General Description and Recognition of 37 Chinese Speech Sounds ㄅ, ㄆ, ㄇ, ㄈ, ㄉ, ㄊ, ㄋ, ㄌ, ㄍ, ㄎ, ㄏ, ㄐ, ㄑ, ㄒ, ㄓ, ㄔ, ㄕ, ㄖ, ㄗ, ㄘ, ㄙ, ㄚ, ㄛ, ㄜ, ㄝ, ㄞ, ㄟ, ㄠ, ㄡ, ㄢ, ㄣ, ㄤ, ㄥ, ㄦ, ㄧ, ㄨ, ㄩ

Ping-yung(or Tom) Cheng

Department of Electrical Engineering,

The University of Adelaide,

Adelaide,

South Australia 5001

December 20, 1979

Submitted to the University of Adelaide in partial fulfilment of the requirements of the degree of Master of Engineering Science

THE UNIVERSITY OF ADELAIDE

This thesis embodies the results of supervised project work which made up two-thirds of the work for the degree.

CONTENTS

Note: Digits in parentheses ( )* in this thesis indicate the numbers of references.

ie no declaration of quotation

ACKNOWLEDGEMENT

I appreciate the amendment and suggestion of my thesis by Prof. Bogner and Dr. Cole. Also Mr. David Fensom and Dr. John Rogers have given me much help and instruction, the author wishes to thank them.

With the consent of Mr. Lin some materials of this thesis in part I were quoted from his papers. (7)* The great assistance offered by Mr. Lin deserves of my deep gratitude.

should be indicated clearly

STATEMENT OF ORIGINALITY

This thesis contains no material which has been accepted for the award to me of any degree or diploma in any University and, to the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis.

Ping-yung(or Tom) Cheng

SUMMARY

This thesis; a "General Description and Recognition of 37 Chinese Speech Sounds", is divided into two parts. A general description of Chinese speech is given first and a method of recognition second.

In the first part, the author will introduce some basic descriptions of the 37 basic Chinese speech sounds , including their characteristics and classifications, spectrum analyses, and will describe spectrum variations of vowels as they are affected by consonants and tones.

A digital computer, a CDC Cyber 173, was used to recognize Chinese speech sounds using linear prediction coefficients,or area functions of Wakita's method, matching and parameter classification.

In the second part, the author also applies BaHli's method and the method of curve fitting, those might be considered as references in the methods of identification.

About 24 of the 37 Chinese speech sounds proved recognizable using linear prediction coefficients, or area functions of Wakita's method, matching and parameter classification.

Using BaHli's method(or the method of curve fitting) we were unable to discriminate between individual speech time-waveforms as BaHli's method(or the method of curve fitting) is only suitable for the identification of smoothed curves.

Both BaHli's method and the method of curve fitting are first applied by the author to the recognition of speech sounds.

PART I : General Description of 37 Chinese Speech Sounds

ABSTRACT

The frequencies and levels of first three formants of all 37 Chinese speech sounds, their time varying wave-forms, and their spectrograms were measured using a D.G.C. NOVA 2 Digital Computer, H.P. 7210A X-Y Digital Plotter, BARCO CTVM 2/51 or H.P. 1310A Screen Display and some interface circuits. Using Fensom's program(Reference 9 and Appendix 6), the frequencies and levels of these first three formants were displayed on the H.P. 1310A. For the sake of brevity, only the results for five vowel sounds and the first two formant frequencies of eleven other vowels were listed.

Useful theories in the classification of speech sounds have been obtained by previous workers using formant frequencies F1 and F2(i.e. Reference 2), and the author used this approach for the Chinese speech sounds. Maximum likelihood regions(Refer (2)*) were computed and used to classify these phonetic sounds.

Spectrum variations of vowels affected by consonants and tones are mentioned in Chapter I.4. The tones of those phonetic sounds can be observed(i.e. Table 4) as pitch(i.e. fundamental frequency) is varied.

I.1 Introduction

Chinese speech sounds do not correspond to English phonemes, but are often approximately related to some compounds of English phonemes.

37 Chinese Speech Sounds and their approximate equivalents in English(6)* (12)* are listed as following:

37 Chinese speech sounds	approximate phonetic equivalents in English
ㄅ	bə
ㄆ	po
ㄇ	mo
ㄈ	fo
ㄉ	də
ㄊ	tə
ㄋ	nə
ㄌ	lə
ㄍ	gə
ㄎ	kə
ㄏ	hə
ㄐ	ji
ㄑ	tʃI
ㄒ	c
ㄓ	tʒʒ'
ㄔ	'tʒu
ㄕ	'sʒ'
ㄖ	'zʒ'
ㄗ	tʒ

37 Chinese speech sounds	approximate phonetic equivalents in English
ㄘ	t_3u
ㄙ	s3''
ㄚ	a
ㄛ	3''
ㄜ	`3'
ㄝ	`ɣe
ㄞ	ai
ㄟ	'eiə'
ㄠ	ɔ:
ㄡ	ɔuə
ㄢ	a:n
ㄣ	`n
ㄤ	a:ng
ㄥ	ong
ㄦ	'ru
ㄧ	i:
ㄨ	wu:
ㄩ	yu:

Speech sound of any Chinese word sound is pronounced as only one syllable. Pronunciations of any Chinese words can be represented by one or more of the 37 Chinese speech sounds, and their four tones.(explanations are seen in Chapter I.4) An analysis of all 37 speech sounds was performed with the D.G.C. NOVA 2 Digital Computer, H.P. 7210A X-Y Digital Plotter, BARCO CTVM 2/51 or H.P. 1310A Screen Display, and their interfaces. Those analyses include the following:

(1) Spectrograms are indicated with time in the horizontal direction, and frequency in the vertical direction. The intensities of time-frequency bounded areas are displayed by the relative tightness or slackness of the lines connecting the peaks of curves. Formant frequencies are represented by those relatively tight lines.(see Fig.1) Refer to pp178,179 in (6)* or p82 of (24)* for a further explanation.

(2) The spectral distributions of intensity or energy within specific short samples of speech waves. Defined by characteristic intensity(db amplitude) vs. frequency (KHz) curves.(eg. Fig.2) (1)* (6)*

(3) An inspection of spectrum variations of vowels affected by consonants and tones.(i.e. Chapter I.4)

(4) Pitch(i.e. fundamental frequency) admits of a rank ordering on a scale ranging from low to high.(Refer to Appendix 5) (1)* (6)* For example; from Fig.3 we can calculate its pitch as following:

$(4.15\text{cm/block})/(1.45\text{cm/pulse})\ 10(10)^3\text{words/sec}\ 256\ \text{words/block} = 111.8$ pps (i.e. pulses/sec)

where the presentation of H.P. 1310A Screen Display or the H.P. 7210A X-Y Digital Plotter are divided into 16 blocks; each block contains 256 addresses; the length of every block and the span of each pulse are equal to 4.15 cm and 1.45 cm respectively. Sampling frequency used for the time waveforms of 37 Chinese speech sounds is equal to 10 KHz.

I.2 The characteristics and classifications of 37 Chinese speech sounds

The 37 Chinese speech sounds are listed as following : ㄅ,ㄆ,ㄇ,ㄈ,ㄉ,ㄊ,ㄋ,ㄌ,ㄍ,ㄎ,ㄏ,ㄐ,ㄑ,ㄒ,ㄓ,ㄔ,ㄕ,ㄖ,ㄗ,ㄘ ㄙ,ㄚ,ㄛ,ㄜ,ㄝ,ㄞ,ㄟ,ㄠ,ㄡ,ㄢ,ㄣ,ㄤ,ㄥ,ㄦ,ㄧ,ㄨ,ㄩ

Every sound has its corresponding spectrogram, spec -tral distribution of energy, and a time waveform. Figures 1,2,and 3, Appendixes 3,4,and 5 indicate each kind of diagram respectively. They show that different speech sounds have different frequency spectrums-i.e. different formants and intensities. The meanings of these three kinds of graphs are explained on the p11.

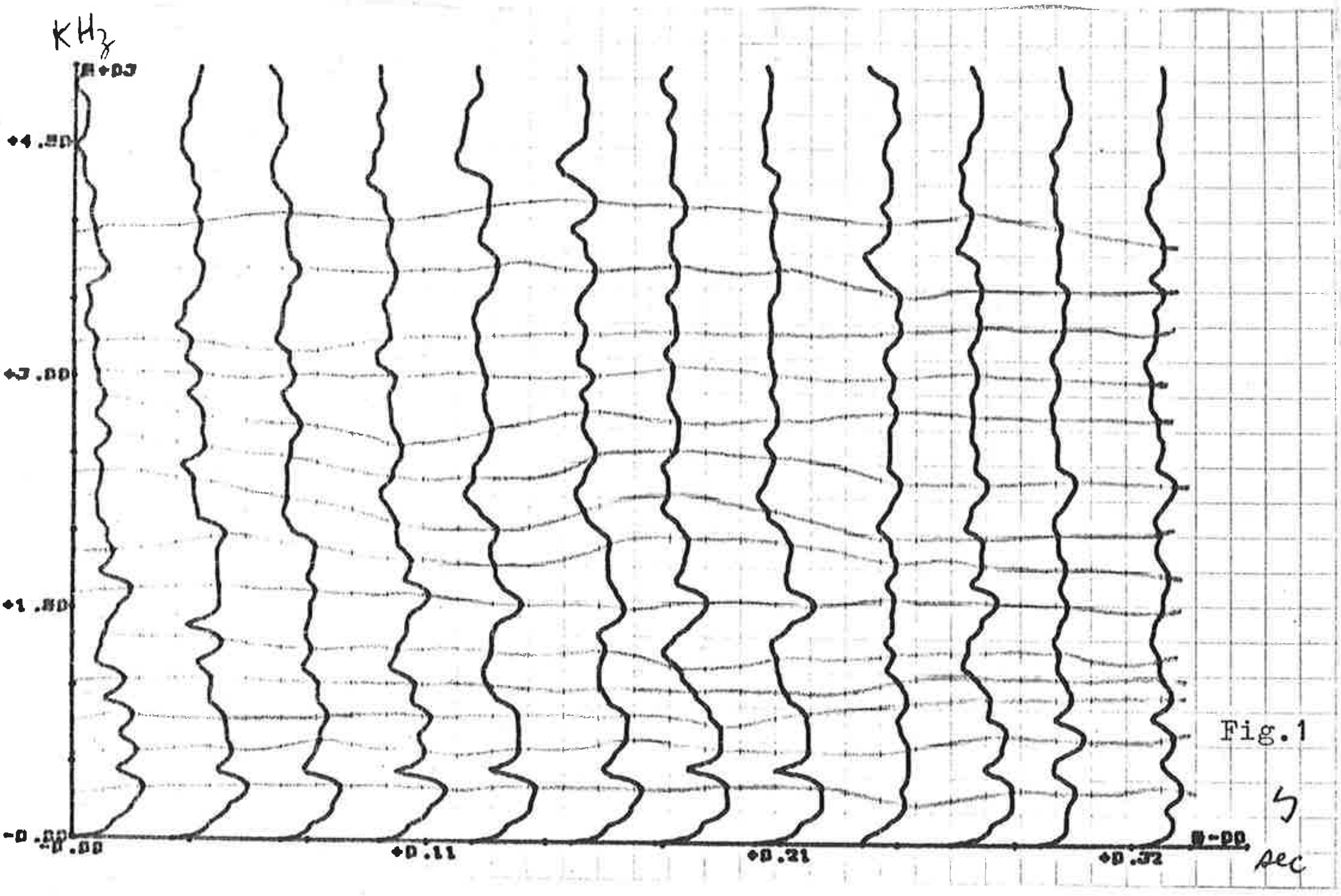


Fig.1

ㄅ

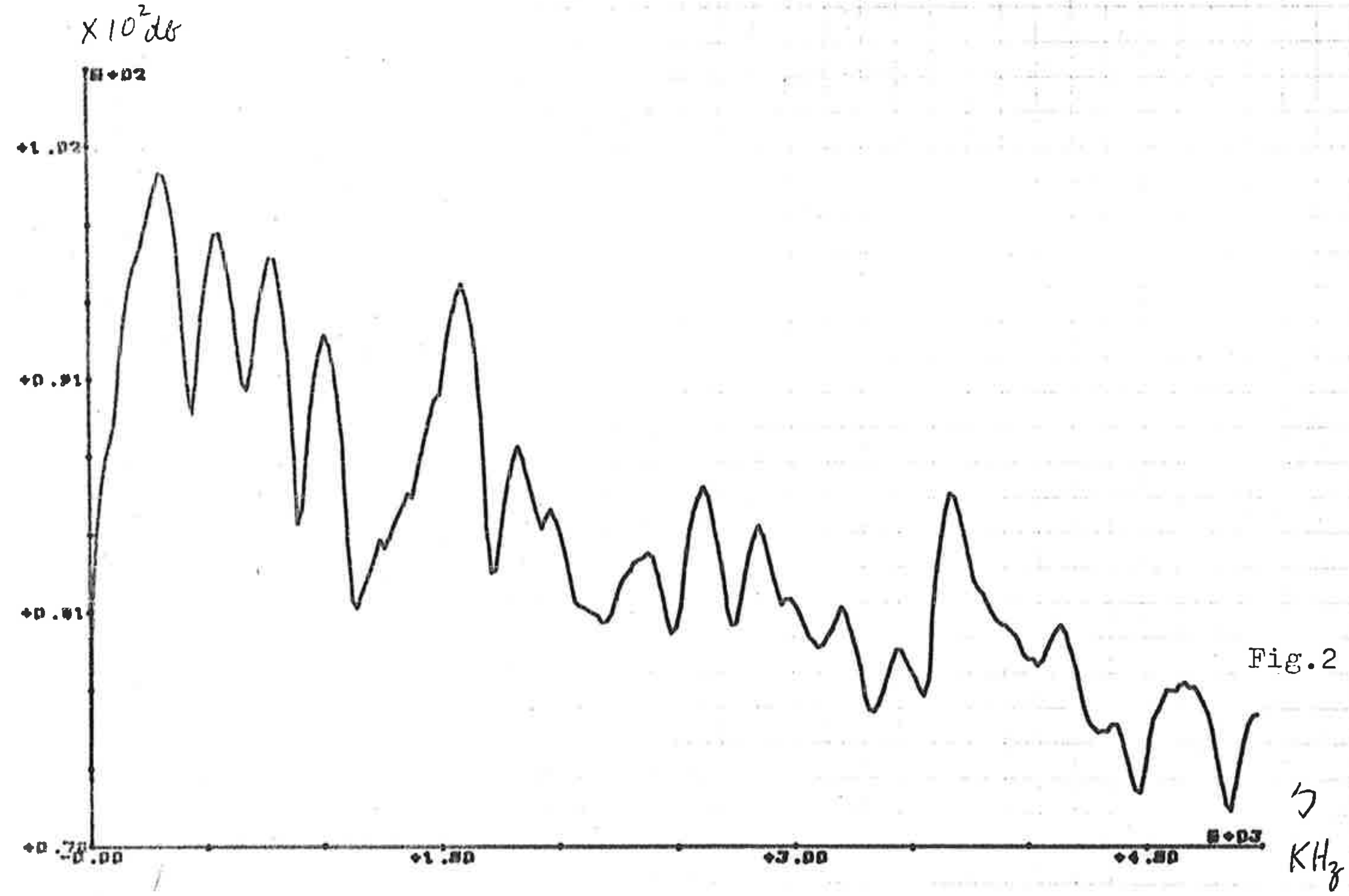

X 10² db
E+02
+1.02
+0.91
+0.81
+0.70
-0.00
+1.50
+3.00
+4.50
E+03
KHz

Fig.2

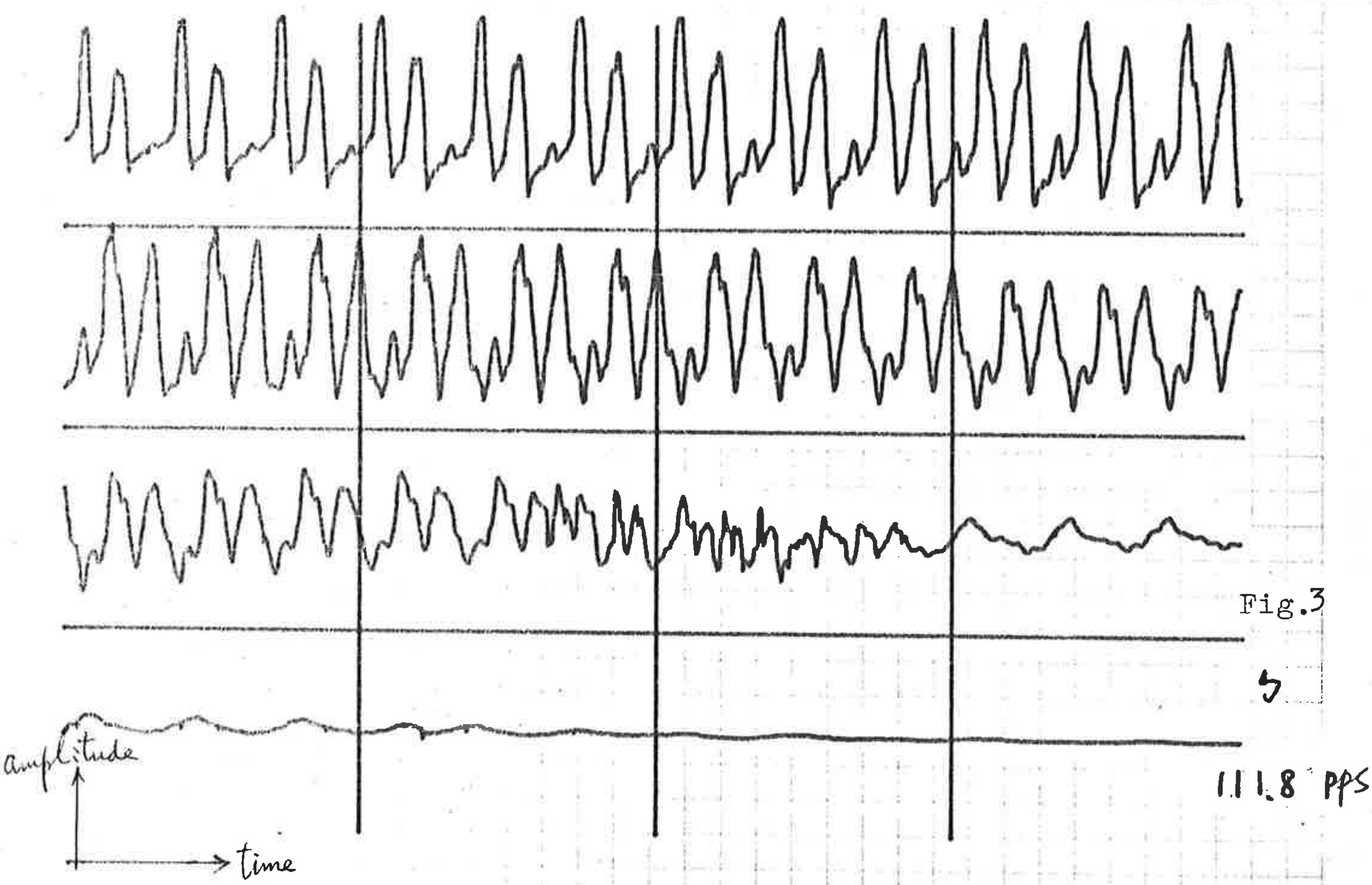

amplitude
time
111.8 pps

Fig.3

Referencing the papers of Nilo Lindgren(2)*, every speech sound is demonstrated by a set of distinctive features implying classifications of sound source, manner and place. In general there are ten kinds of distinctive features found in speech sounds. They are shown in Table 1.

TABLE 1 (2)*

Distinctive features	Characteristics
1. Vocalic/Nonvocalic	Presence vs. absence of a shar-ply defined formant structure
2. Consonant/ Nonconsonant	Low vs. high total energy
3. Compact/Diffuse	High vs. low first formant
4. Tense/Lax	Longer duration & greater depa-rture from a neutral posi-tion vs. shorter duration & less departure from a neutral position
5. Grave/Acute	Low second formant vs. high second formant
6. Flat/Plain	Flat phonemes in contradistinc-tion to the corresponding plain ones are characterized by a downward shift or weaken-ing of some of their upper-frequency components
7. Nasal/Oral	Spreading the available energy over wider(vs. narrower) freq-uency regions by reduction in the intensity of certain formants & introduction of additional formants
8. Discontinuous/ Continuant	Silence followed &/or preceded by spread of energy over a wi-der frequency region vs. absence of abrupt transition between sound & such silence

9. Strident/ Mellow — Higher intensity noise vs. lower intensity noise

10. Voiced/ Voiceless — Presence vs. absence of periodic low frequency excitation

Using the criteria of Table 1, we can list the distinc -tive features of each Chinese speech sound(Note: the author's sounds were adopted as standard models.) These are shown in Table 2. In Table 2, the symbols "+" and "-" represent former and latter distinctive features respectively. In the description of spectrum characteristics in Chapter I.3 the classification standards of items 3,4,5 & 6 as listed in the Table 2 are discussed. The classification standards of other items use common sense definitions, and their corresponding figures are shown in Appendix 3 and Appendix 4.

TABLE 2 (7)*

	ㄚㄛㄜㄝㄧㄨㄩㄦㄞㄟㄠㄡㄢㄣㄤㄥㄅㄆㄉㄊㄍㄎㄐㄑㄓㄔㄗㄘㄇㄋㄌㄈㄏㄒㄕㄙㄖ
1. Vocalic/ Nonvocalic	++++++++++++++++---------------------
2. Consonantal/ Nonconsonantal	----------------+++++++++++++++++++++
3. Compact/ Diffuse	++++---+++++++++----++-+-+-----------
4. Tense/Lax	+---+++----------+-+-+---------+++++-
5. Grave/Acute	-++++---++++--+-++++-----+--++++-----
6. Flat/Plain	--+++-+---------+++++--+-+--++++-----
7. Nasal/Oral	high intensity of formants ------------+++-----+
8. Continuant/ Interrupted	intermediate ---------------+++++-
9. Strident/ Mellow	no noise -+-+-+-+ no noise

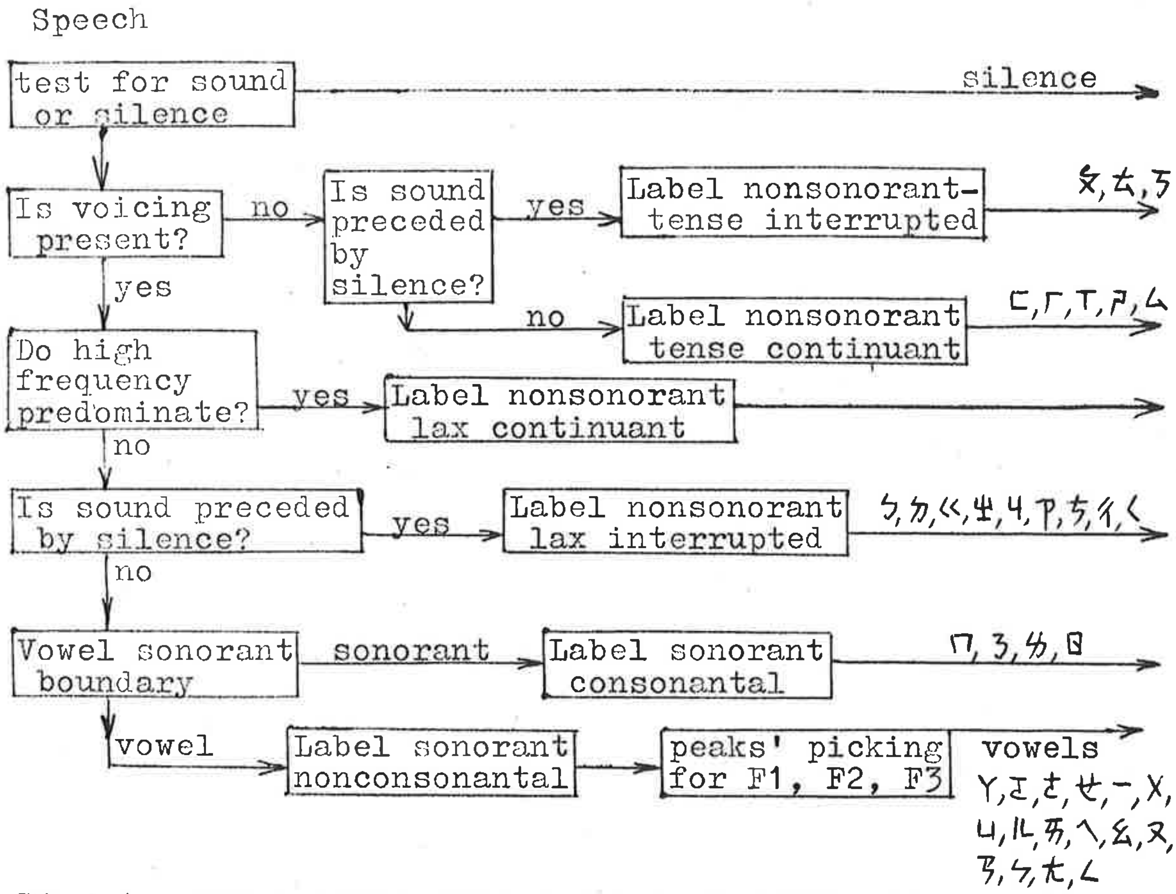


Fig. 4 Independent time segment classification.(2)*

From Table 2 and the definitions within Table 1, we can construct a flow chart as shown in Fig.4. This is the sequence used to classify the characteristics of Chinese speech sounds. Speech sounds are sorted into nonsonorant or sonorant sounds through some property measurements- absence of energy below 350c/s shows lack of voicing; frequency components above 4000 c/s indicate turbulence, etc.(2)* Those results can be shown on the Screen Display of H.P.1310A by means of Fensom's program(i.e. Reference 9 and Appendix 6).

I.3 Spectrum analyses of Chinese Speech Sounds(3)*(4)*(7)*

According to the distributions of formants F1 and F2 for the speech sounds-ㄨ,ㄧ,ㄝ,ㄢ,&ㄚ follow from Fig.6 we know these five sounds have clear ranges. We chose to explain these 5 sounds in detail for that reason. The author was unable to define the boundaries of the other sounds because they aggregate together.

The regions of energy concentration are called formants , so the formants display the frequencies at those ampli-tude peaks shown as in the Appendix 3; F1, F2, and F3 are the 1st, 2nd, and 3rd formants respectively.

1. Experimental equipment

(1) Anechoic room

(2) ATN 2 110 v to 240 v Mains Power Supply

(3) NAGRA 4.2 Tape Recorder

(4) Type 2114 Third-Octave and Octave Spectrometer

(5) Type 2604 Microphone Amplifier

(6) D.G.C. NOVA 2 Digital Computer

(7) H.P. 7210A X-Y Digital Plotter

(8) BARCO CTVM 2/51 or H.P. 1310A Screen Display and accompanied interface circuits

2. Sample Collection

Speech samples were collected from 20 male students of Australia and some Chinese men in Adelaide, South Australia. These sounds were adopted as testing samples. The author's sounds were used as the standard models.

3. Experimental methods

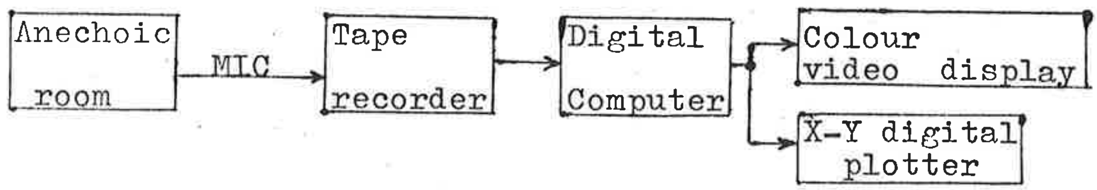


Fig. 5

The samples were recorded in the anechonic room using the equipment shown in Fig. 5. The author first chose the required figures(i.e. time waveforms starting at the first peak of maximum uniform parts in order to adopt the same standard and avoid interference from any other sources of sounds) from the screen image(The screen area was divided into 16 blocks, each block containing 256 addresses. The block numbers corresponded to block addresses on the computer's magnetic disk. The numbers of blocks, and their addresses were software controlled.) and then used the X-Y plotter to plot three kinds of diagrams. From computer experiments, we knew not only that the waveshapes of the maximum uniform parts of time waveforms, but also their corresponding spectrograms and spectral distributions of energy were the same. Therefore, we experienced no problem with segmentations, using this method. The results of these experiments are shown in Figs. 1,2,3 and in the appendixes. I analyzed the speech samples and the results are given in Table 3.(reference Appendixes 3,4) Using the NOVA 2 Computer program-SPGRM(see Appendix 6), the values of formant frequency and level for each sound were shown on the screen. From those values we calculated the corresponding values of standard deviation.

In Table 3, F1, F2, F3, L1, L2, L3 are those mean values of samples and δF1, δF2, δF3, δL1, δL2, δL3 are their average standard deviations respectively.

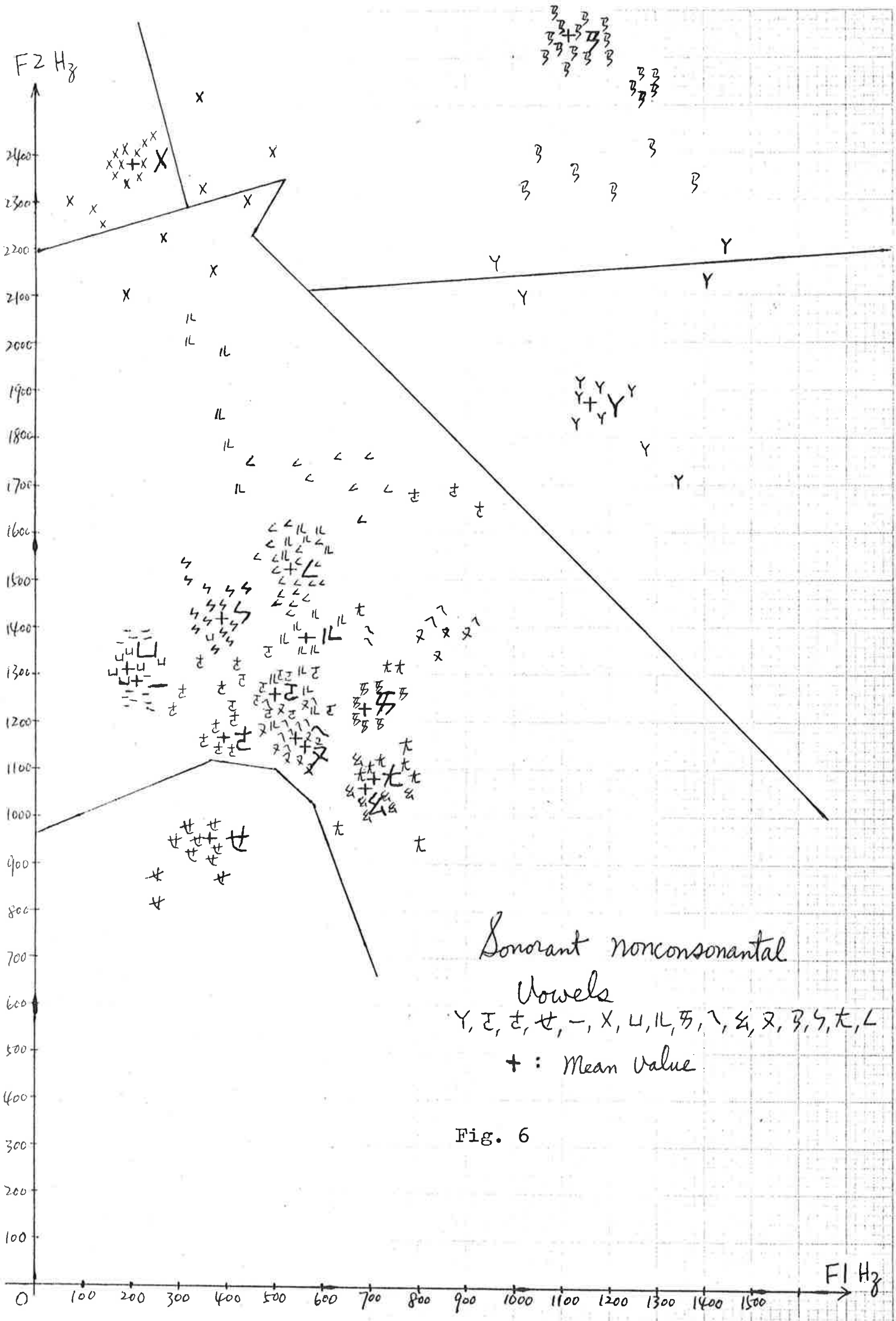

F2 Hz
F1 Hz
2400
2300
2200
2100
2000
1900
1800
1700
1600
1500
1400
1300
1200
1100
1000
900
800
700
600
500
400
300
200
100
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
Sonorant nonconsonantal
Vowels
ㄚ, ㄛ, ㄜ, ㄝ, ㄧ, ㄨ, ㄩ, ㄦ, ㄞ, ㄟ, ㄠ, ㄡ, ㄢ, ㄣ, ㄤ, ㄥ
+ : Mean Value

Fig. 6

TABLE 3

speech sound	average formant frequency & its average standard deviation in Hz						average formant level & its average standard deviation in db					
	F1	δF1	F2	δF2	F3	δF3	L1	δL1	L2	δL2	L3	δL3
ㄨ	203	12	2381	12	3167	333	100	0.3	81	1.5	74	6
ㄧ	214	24	1286	179	2333	47	102	2.6	80	3.2	89	0.4
ㄝ	369	12	952	0.1	1321	36	103	2.1	99	1.6	97	0.3
ㄢ	1107	155	2655	191	3202	190	107	0.5	100	2.7	98	0.1
ㄚ	1155	35	1881	24	2571	24	106	0.9	97	2.1	99	1

The experimental results(i.e. Appendix 4) indicate that the 3rd formant(i.e. F3) has little influence in the for -mation of voiced sound. Because the frequencies of F3 are very high in each speech sound, it is hard to perceive them. The major factors are the positions of F1 and F2 and their relative differences. Hence we can discriminate the vowels ㄨ,ㄧ,ㄝ,ㄢ,&ㄚ as shown in Fig.6.

We assumed that the coordinates F1 vs. F2 of every speech sound are Gaussian distributions. We used maximum likelihood methods to find the equal-probability points between adjacent speech sounds. Those equal-probability curves were used as decision thresholds. We obtained the decision regions between speech sounds, as plotted in Fig.6. In Fig.6 the presented number of each kind of vowels are not all equal to 20 because some vowels of each kind have the same first and second formants.

As an example, the procedure for finding the decision threshold between speech sounds ㄧ and ㄝ is approximated as following. Originally the decision thresholds should be multidimensional quadratic "surfaces"; not straight lines, but for convenience, I assumed that all errors of

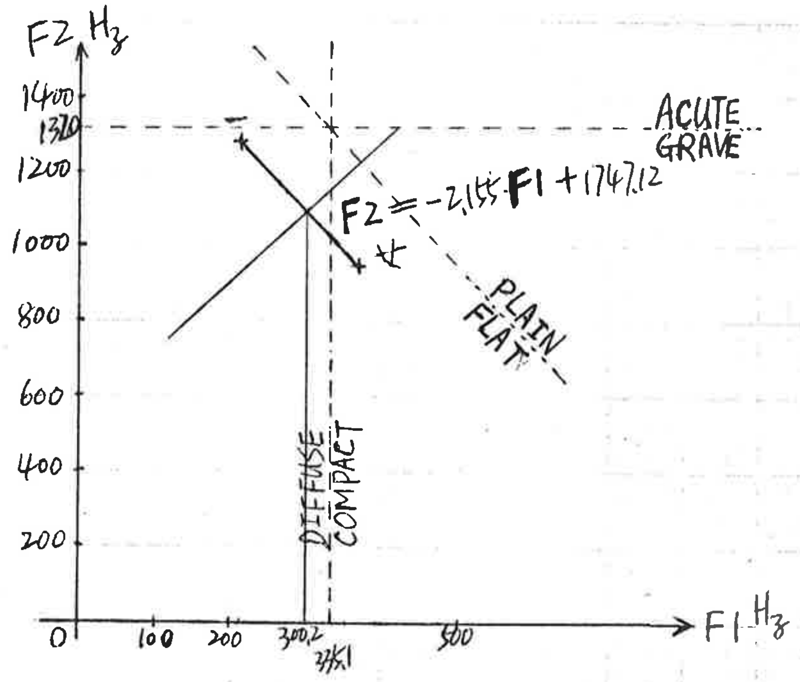


Fig.7

the approximation, especially the numerical values near the decision thresholds, could be neglected.

(1) By connecting the two mean value points of ー and せ the line equation $F2 = -2.155F1 + 1747.12$ (A) as indicated in Fig.7 is obtained.

(2) Assuming the sounds ー and せ at F1 and F2 are indepen -dent, the Joint Gaussian distribution of the sounds is obtained as shown:

$$P_{-}(F1,F2) \qquad \text{(B)}$$
$$= \frac{1}{2\pi\sigma 1\ \sigma 2} \mathrm{Exp}\left\{-(F1-U1)^2/2(\sigma 1)^2-(F2-U2)^2/2(\sigma 2)^2\right\}$$

where $\sigma 1=24$, $\sigma 2=179$, $U1=214$, $U2=1286$ as indicated in Table 3.

Similarly we can produce the Joint Gaussian distribution of speech sound せ

$$P_{せ}(F1,F2) = \frac{1}{2\pi\sigma 1'\sigma 2'} * \qquad \text{(C)}$$
$$\mathrm{Exp}\left\{-(F1-U1')^2/2(\sigma 1')^2-(F2-U2')^2/2(\sigma 2')^2\right.$$

where $\sigma 1'=12$, $\sigma 2'=0.1$, $U1'=369$, $U2'=952$ as indicated in Table 3.

(3) Equation (A) is substituted into equations (B) and (C). Since $P_{\text{ㄧ}} = P_{\text{ㄝ}}$ at the equal-probability point, we obtain F1 = 300.2 Hz

(4) At F1=300.2 Hz we depict the vertical line intersecting with line (A). From the intersection point, a straight line perpendicular to line (A), is the desired decision threshold as indicated in Fig.7.

Similarly we can obtain other decision thresholds between adjacent sounds as shown in Fig.6. F1 vs. F2 figures of the other speech sounds are shown in Appendix 1.

In Chapter I.2 I mentioned the distinctive features of speech sounds such as Acute/Grave, Compact/Diffuse and Flat/Plain. Our experimental results indicated that if the value F2 of a speech sound is larger than 1320 Hz, it is acute; otherwise it is grave. If the value of F1 is greater than 335.1 Hz, the speech sound is compact, other-wise it is diffuse. If the value F1+F2 is greater than 1655.1 Hz(i.e. 1320+335.1 Hz), the speech sound is plain, otherwise it is flat. The classification standards (F1=335.1 Hz, F2=1320 Hz) are not fixed values as shown in Fig.7. They can be changed to discriminate all speech sounds. The methods used in this paragraph are mentioned in the papers of Nilo Lindgren. (2)* (7)*

From the frequency vs. intensity(db) figures shown in Appendix 3, we know that a low intensity F2 will be found in the spectrum of most nasal consonants.(1)* The nasal consonants are ㄇ, ㄋ, ㄣ and ㄥ.

The time-amplitude figures shown in Appendix 5 and Fig.3 show that all speech sounds possess periodic, or rather, quasi-periodic structures as none of the 37 Chinese speech sounds include any fricatives, and accordingly display harmonic spectra. This fine structure originates from the opening and closing movements of the vocal cords, which periodically modulate the volume of the exhaled air during phonation, at a rate of F_0 c/s, which is the voice funda -mental frequency. In narrow-band spectrograms, F_0 is the harmonic spacing and in broad-band spectrograms $1/F_0$ is the time interval between successive striations, each reflecting a single voice cycle. The time variation of F_0 is the physical basis of intonation. (1)* (6)*

Each position of the articulatory organs has its specific F-pattern. Formant bandwidths are slightly greater than vocal tract resonance bandwidths, due to additional losses through the glottis slit. A decrease of the lip-opening area or increase of the length of the lip passage causes a lowering of the frequencies of all formants. Only in non-nasalized, non-lateral sounds produced from a source located at a vibrating or a narrow glottis, can the F-pattern up to F3 be seen clearly.(1)*,(6)*

The frequencies of the three lowest formants, F1, F2, F3, are the main determinants of the phonetic quality of a vowel.

I.4 Spectrum variations of vowels affected by consonants and tones (7)*

The continuous variations in the dimensions of the vocal cavities with time, determine uniquely the variations of the vocal tract resonance frequencies, i.e. the F-pattern. There is, therefore, a continuity of F-pattern within any length of utterance and across any sound segment boundary. The transitional cues, whereby a consonant vowel may in part be identified by its influence on an adjacent vowel , may thus be described in terms of F-pattern variations.

The main articulatory variables are (1)*,(6)*

1. the location,
2. the degree of constriction of the main narrowing between the tongue and the opposite wall of the vocal cavities, and
3. the degree of constriction and lengthening of the lip passage.

(6)*,(7)*

Figures 8(a) and (b) are quoted from the papers of Mr. Lin.

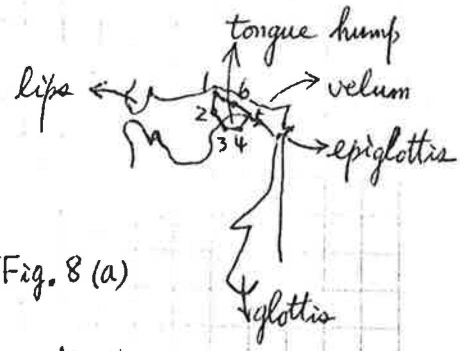


Fig. 8 (a)

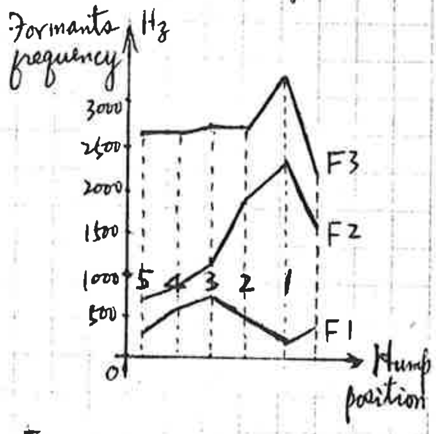


Fig. 8 (b)

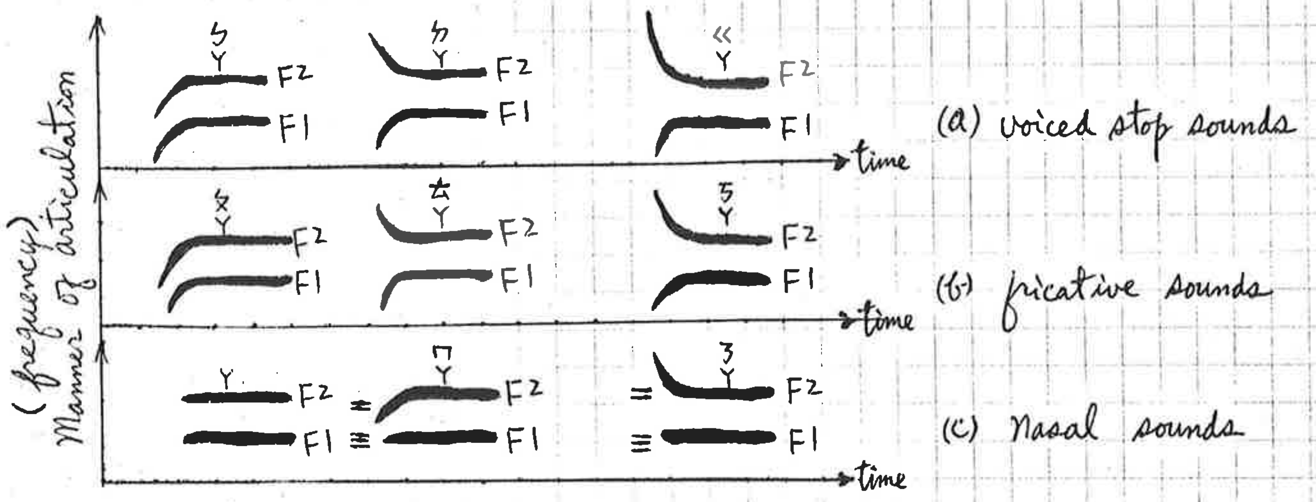


Fig. 9 Vowel ㄚ affected by consonants ㄅ,ㄆ,ㄉ,ㄊ,ㄇ,ㄍ,ㄎ and ㄋ.

Those figures are quoted from the papers of Mr. Lin. (7)*

Segment Pattern Features (1)*,(6)*

Articulation	Speech wave
1. Tongue fronted	F2-F1 large
(a) Prepalatal position	F2 high, F3 maximally high
(b) Midpalatal position	F2 maximally high and close to F3
2. Tongue retracted	F2-F1 small. F1 comparatively high
3. Mouth-opening(including tongue section & lips) narrow	F1 low
4. Retroflex modification	
(a) Alveolar articulation	F4 low and close to F3
(b) Palatal articulation	F3 low and close to F2
5. Bilabial or labio-dental closure	F2: 500～1500 c/s
6. Interdental articulation	F2: 1400～1800 c/s
7. Dental or prealveolar articulation	F2: 1400～1800 c/s, F3 high
8. (a) Palatal retroflex articulation	F3 low
(b) Palatal articulation with tip of tongue down	F2 and F3 high

9. Velar & pharyngeal articulation	F2 medium or low. The F-pattern except F1 is clearly visible.
10. Glottal source	The entire F-pattern including F1 is visible.

The articulation of vowel sounds is generally affected by the position of the tongue hump along the vocal tract (this is often, but not always, the place of greatest constriction) and the degree of the constriction.

(c) Hump positions moved from point 1 thru point 5 as shown in Fig.8(a), cause the formant variations of voiced sound indicated as in Fig.8(b). Using the variations shown in Fig.8, we can evaluate some vowel formants that have been affected by consonants. For example; the hump position of consonant ㄍ corresponds to point 1 of Fig.8(a); This indicates that the value of F2 is raised and the value of F1 is lowered as shown in Fig.8(b); the hump position of vowel ㄚ corresponds to point 3 of Fig.8(a), and the value of F2 is seen to be near that of F1-i.e. as the hump posi -tions varied from point 1 to point 3, the spread of F2 and F1 was decreased. Consider the voice ㄍㄚ, its formant variations shown in Fig.9(a) correspond to the formant variation from point 1 till point 3 as indicated by Fig. 8(b). Hence to analyze the variation of the vowel formant caused by the consonant, we first investigate the variations of hump position and then determine the corresponding formant variations. Fig.9(a), (b) and (c) are represented as the effects of voiced stop sounds, fricative sounds and nasal sounds respectively. This paragraph is quoted from the papers of Mr. Lin.(7)*

Our experiments showed that the tones don't greatly affect formants, but do have some major effects on pitch. Using the fundamental frequency(i.e. pitch) of the first tone as the reference point; changing the vowel pronunci -ation to second tone lowers its fundamental frequency (i.e. pitch). Then changing the vowel sound to third tone , again lowers the fundamental frequency. However at the fourth tone condition, the fundamental frequency(i.e. pitch) will be lifted. These discussions are listed in Table 4. As Mr. Lin's colleague, my sounds were adopted in his experiments. Those experimental results in this paragraph and Table 4, were quoted from his papers (7)*.

B

The values in Table 4 are the average results of measurements of 20 male persons, and were adopted from the experiments of Mr. Lin.

TABLE 4 (7)*

Tones	Effects of fundamental frequencies(i.e. pitch) caused by tones F Hz	Standard deviations of fundamental frequency F Hz
first tone	0 (reference point)	0
second tone	20 (lowered)	8
third tone	31 (lowered more)	13
fourth tone	10 (lifted)	9

The process of implicit investigation was used only for vowels because their sounds are independent of the following features: nasal/oral, continuant/interrupted and strident/mellow. These characteristics of vowels differ from those of consonants, and attracted our interest. All data listed in this chapter was derived by Mr. Lin using a spectrum analyzer(KAY 7030A) and data recorder(GR 1525).

I.5 Conclusions

(1) The distinctive features of the 37 Chinese speech sounds can be used to recognize or classify each one uniquely.

(2) Using Fig.8, we can predict the formant variations of vowels as affected by each possible consonant.

(3) We can distinguish the tones of Chinese speech sounds by measuring variations in fundamental frequency(i.e. pitch) as shown in Table 4.

PART II : Recognition of 37 Chinese Speech Sounds

ABSTRACT

The author used linear prediction techniques to find L.P.C.'s(i.e. linear prediction coefficients) of the time waveforms of speech, and Wakita's method to find their area functions. Parameter classification, illustrated as following, was then used to recognize the 37 possible Chinese speech sounds.

We developed from a set of samples, a decision rule with which we could classify a point in the parameter space of an unknown sound. The process of deriving the decision rule is called parameter classification. The two major areas of parameter recognition are (1) feature selection , and (2) parameter classification procedures.

Using a Digital Computer(CDC Cyber 173) we applied these procedures to recognize two Chinese speech sounds simultaneously. The simultaneous classification of more than two sounds requires considerably more complex software, and is left to those with more experience of dynamic programming.

24 Chinese speech sounds of the possible 37 proved recognizable using linear prediction coefficients(or area function of Wakita's method) and parameter classification.

Both BaHli's method and curve fitting routines were applied to the problem of speech recognition. Unfortunately those two methods were not successful.

II.1 Introduction (17)*,(18)*,(19)*,(20)*,(21)*,(22)*

There are already several commercially available isolated word recognition systems today. A few research systems have been developed for restricted connected speech recognition and speech understanding.(i.e. Hearsay-I, Dragon, Lincoln, and IBM). (17)*,(19)*,(20)*

In connected speech, it is difficult to determine where one word ends and another begins, and the characteristic acoustic patterns of words exhibit much greater variability depending on the context. Isolated word recognition systems do not have these problems since words are separated by pauses. (17)*,(18)*,(19)*,(20)*,(22)*

The other feature that affects the complexity of a system is the vocabulary size. As the size or the confusability of a vocabulary increases, simple brute-force methods of representation and matching become too expensive. (17)*

In restricted speech understanding systems, it is not important to recognize each and every phoneme and/or word correctly, as long as the message is understood. The need to understand the sources of knowledge used to form the message, the representation of the task, conversational context, understanding, and response generation, all add to the difficulty and overall complexity of speech understanding systems. The sources of knowledge include the characteristics of speech sounds(phonetics), variability in pronunciations(phonology), the stress and intonation patterns of speech(prosodics), the sound patterns of words(lexicon), the grammatical structure of language(syntax), the meaning of words and sentences (semantics), and the context of the conversation(pragmatics). (17)*

The (restricted) dictation machine problem requires larger vocabularies(1000 to 10000 words). It is assumed that the user would be willing to spell any word that is unknown to the system. The task requires an English-like syntax, but can assume a cooperative speaker speaking clearly in a quiet room. (17)*

The unrestricted speech understanding problem requires unlimited vocabulary connected speech recognition, but permits the use of all the available task-specific information. The most difficult of all recognition tasks is the unrestricted connected speech recognition problem which requires unlimited vocabulary, but does not assume the availability of any task-specific information. (17)*

In general, for a given system and task, performance depends on the size and speed of the computer and on the accuracy of the algorithm used. Accuracy is often task dependent. Accuracy versus response time tradeoff is also possible,i.e. it is often possible to tune a system and adjust thresholds so as to improve the response time while reducing accuracy and vice versa. (17)*,(21)*,(22)*

Given a known vocabulary(of about 30 to 200 words) and a known speaker, the systems of Itakura, Martin, and White can recognize a word spoken in isolation with accuracies around 99 percent. The general paradigm involves comparing the parameter or feature representation of the incoming utterance with the prototype reference patterns of each of the words in the vocabulary. (17)* ,(18)*,(19)*,(20)*

The 37 Chinese speech sounds and any other Chinese words have their corresponding linear prediction coefficients(or area functions). The pronunciation of every one of those is a single phonetic segment so we do not have the problem of no segmentation. The author suggests that the methods-parameter extraction of time waveforms and their corresponding parameter classifications, could be applied to the recognition of Chinese speech sounds.

The time waveform of a speech sound can be modeled as a linear combination of its past values, and present and past values of a hypothetical input to a system whose output is the given time waveform of speech sound. The parameters of a suitable all-pole model can be obtained by a least squares analysis in the time domain. The time waveform of speech sound is assumed to be stationary. From those linear prediction coefficients(or, either area functions or other kinds of coefficients using BaHli's method or the method of curve fitting) of the time waveforms, we could use the method of parameter classification shown following to identify two speech sounds each time.(5)*, (24)*,(25)*, Appendixes 8 & 9 The principles of linear prediction coefficients, area functions and other kinds of coefficients using BaHli's method or the method of curve fitting are given in Chapter II.2.

A sophisticated approach for speech waveform recognition is based on a set of selected measurements extracted from the time waveforms of speech sounds. These selected measurements, called "features", have less variation and distortion. The criterion of feature selection or ordering is often based on either the importance of the features in characterizing the time waveforms or the contribution of the features to the performance of recognition(i.e. the accuracy of recognition). The problem of classification(or making a decision on the class assignment to the time waveforms) is making the measurements of the selected features. A simplified block diagram of a speech recognition system is shown in Fig.10.

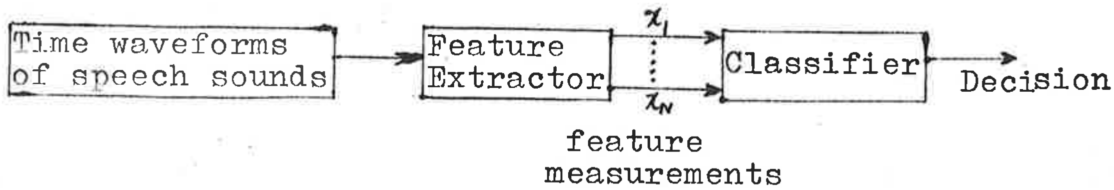


Fig.10 Recognition system of speech time-waveforms.

II.2 Recognition process of linear prediction coefficients ,area functions, or other kinds of coefficients with respect to their corresponding time waveforms of 37 Chinese speech sounds

The linear prediction coefficients, the vocal tract area functions(also the reflection coefficients) or other kinds of coefficients using BaHli's method or curve fitting of any isolated sound time-waveforms could be considered as the required features in the recognition system.(18)* ,(5)*,(24)*,(25)* Let us explain briefly as following:

A. Linear Prediction (5)*

During the production of speech-sound, the vocal tract is excited by a series of nearly periodic pulses generated by the vocal cords. If both the excitation and the shape of vocal tract remain fixed, the resulting speech-signal can be considered to be stationary. In order to get the model of speech-sound we approximate the continuously-varying shape of vocal tract by a discretely-varying shape of vocal tract, i.e. a vocal tract whose shape changes at discrete time intervals. This discrete time interval is called the sampling interval T. Hence the sampling frequency is then $f_s=1/T$. Every continuous-time speech s(t) could be sampled to obtain its discrete-time speech s(nT), where n is an integer variable.

Linear prediction was introduced as a spectral modeling technique in which 37 Chinese speech sounds were modeled by the all-pole spectra. This method allows for the modeling of selected portions of a speech sound, for arbitrary spectral shaping in the frequency domain, and for the modeling of continuous as well as discrete spectra.

The method of linear prediction was introduced to speech processing basically as a time-domain analysis. Referring to (5)* we know that linear prediction is a correlation type of analysis which can be approached either from the time or frequency domain. Here I restrict my attention to the spectral modeling properties of the autocorrelation method of linear prediction.

Given a speech spectrum P(w), we desire to model this spectrum by an all-pole spectrum $\hat{P}(w)$ indicated by (assuming a sampling frequency of 1 Hz)

$$\hat{P}(w) = G^2 / \left| 1 + \sum_{k=1}^{p} a_k e^{-jkw} \right|^2 \qquad (1)$$

where the predictor coefficients a_k, $1 \leq k \leq p$, and the gain factor G are the parameters of all-pole model. The parameters a_k are obtained as a result of the minimization of the spectral error E, which is defined as

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{P(w)}{\hat{P}(w)} dw \qquad (2)$$

where $w = 2\pi f$ and f is the frequency.

By setting $\partial E / \partial a_i = 0$, $1 \leq i \leq p$, there results

$$\sum_{k=1}^{p} a_k R_{|i-k|} = -R_i \quad , \quad 1 \leq i \leq p \qquad (3)$$

where $$R_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} P(w) \mathrm{Cos}(kw) dw \qquad (4)$$

is the autocorrelation function corresponding to the continuous spectrum P(w). By equating the total energy in the speech sound and model spectra, the gain G is obtained

$$G^2 = R_0 + \sum_{k=1}^{p} a_k R_k \qquad (5)$$

Equations (3),(4), and (5) completely specify the all-pole model spectrum $\hat{P}(w)$.

In practice, the speech sound s'_n is defined for all time such that it is identically zero outside a portion of the speech sound s_n. s'_n is N samples long and N is any integer variable. Hence we get

$$s'_n = \begin{cases} s_n w_n , & 0 \leqslant n \leqslant N-1 \\ 0 , & \text{otherwise} \end{cases} \qquad (6)$$

where w_n is the window function.
Hence the autocorrelation function (4) is given by

$$R(i) = \sum_{n=0}^{N-1-i} s'_n \, s'_{n+i} \quad , \; i \geqslant 0 \qquad (7)$$

The predictor coefficients a_k, $1 \leqslant k \leqslant p$, can be computed using autocorrelation(i.e. stationary) techniques.(see Eq.(3)) According to (5)*, Durbin's recursive procedure is used to solve the normal autocorrelation Eq.(3).

$$E_0 = R(0) \qquad (8a)$$

$$k_i = -\left\{R(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)\right\} / E_{i-1} \qquad (8b)$$

$$a_i^{(i)} = k_i$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)} \quad , \; 1 \leqslant j \leqslant i-1 \qquad (8c)$$

$$E_i = (1-k_i^2) E_{i-1} \qquad (8d)$$

Equations (8a),(8b),(8c), and (8d) are solved recursively for $i=1,2,\ldots\ldots,p$ The final solution is given by

$$a_j = a_j^{(p)} \quad , \; 1 \leqslant j \leqslant p \qquad (8e)$$

Those parameters k_i, $1 \leqslant i \leqslant p$, are known as the reflection coefficients or the partial correlation coefficients.

The normalized autocorrelation coefficients r(i) are defined as

$$r(i) = R(i)/R(0)\ , \quad |r(i)| \leq 1 \qquad (9)$$

B. Estimation of the Vocal Tract Area (5)*,(24)*,(25)*

(1) In 1972, Wakita showed that an acoustic tube model consisting of M cylindrical sections of equal length can be used to represent the inverse filter $I(z)=1+\sum_{k=1}^{p} a_k z^{-k}$ obtained from linear prediction of the speech sounds.

(2) If the speech is properly preemphasized and the boundary conditions of the acoustic tube model are properly chosen, then quite accurate vocal tract shapes can be directly estimated by means of the autocorrelation method of linear prediction.

(3) A set of reflection coefficients in the acoustic tube model can be derived using inverse filter processing of speech. The discrete area function of this kind of acoustic tube can easily be obtained from its corresponding reflection coefficients.

In order to derive the acoustic tube model of the vocal tract we make the following assumptions.

(1) A set of M cylindrical sections can be interconnected as the physical model of vocal tract. Each individual section has uniform area A_m and equal length l.

(2) The speech sound transmitted through each section of cylinders can be considered as a plane wave because its wave-length is much larger than the transverse dimension of individual cylindrical section.

(3) Internal losses due to wall vibration, viscosity, and heat conduction are negligible in the rigid sections of cylinders.

(4) Elementary wave equations of acoustics are valid in the derivation of acoustic tube model.

(5) This model is linear and the effects of the glottis and the nasal tract can be neglected.

(6) Proper boundary conditions exist which make the acoustic tube model equivalent to the linear prediction model $1/I(z)$, $z=\exp(jwT)$.

A series of 8 uniform cylindrical sections was adopted as the schematic model of the vocal tract. Each section had a constant area A_m, whose value was estimated as the average area of the mth non-uniform section of the vocal tract.

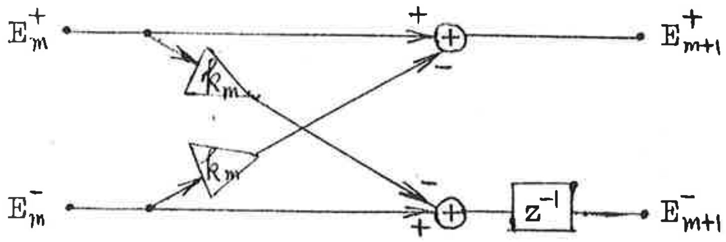


Fig.11 Digital filter of linear prediction model.

Also according to (25)*, a linear prediction digital filter as Fig.11 can be constructed in a recursive manner. The z-transforms of the forward- and the backward- going speech signals, $E_m^+(z)$ and $E_m^-(z)$ ($z=\exp(jwT)$; T is a sampling period), at the input of the m-th stage of the filter(see Fig.11) can be expressed by a recursive equation,

$$\begin{bmatrix} E_{m+1}^+(z) \\ E_{m+1}^-(z) \end{bmatrix} = \begin{bmatrix} 1 & -k_m \\ -k_m z^{-1} & z^{-1} \end{bmatrix} \begin{bmatrix} E_m^+(z) \\ E_m^-(z) \end{bmatrix} \qquad (10)$$

where k_m is the reflection coefficient.

If pressure $p_m(z)$ and volume velocity $u_m(z)$ are defined in terms of a linear combination of $E_m^+(z)$ and $E_m^-(z)$, i.e.,

$$\begin{bmatrix} p_m(z) \\ u_m(z) \end{bmatrix} = G_m(z) \begin{bmatrix} E_m^+(z) \\ E_m^-(z) \end{bmatrix} \qquad (11)$$

where $G_m(z)$ is a 2X2 operational matrix, then

$$\begin{bmatrix} p_{m+1}(z) \\ u_{m+1}(z) \end{bmatrix} = H_m(z) \begin{bmatrix} p_m(z) \\ u_m(z) \end{bmatrix} \qquad (12)$$

where, by the use of (10) and (11),

$$H_m(z) = G_m(z) \begin{bmatrix} 1 & -k_m \\ -k_m z^{-1} & z^{-1} \end{bmatrix} G_m^{-1}(z) \qquad (13)$$

$$\equiv \begin{bmatrix} h_{11}^{(m)}(z) & h_{12}^{(m)}(z) \\ h_{21}^{(m)}(z) & h_{22}^{(m)}(z) \end{bmatrix} \qquad (14)$$

For simplicity, the system is assumed to be reciprocal and the matrix $H_m(z)$ is assumed to be symmetric. Then, for

reciprocity $\quad \det H_m(z) = 1 \qquad (15)$

and for symmetry

$$h_{22}^{(m)}(z) = h_{11}^{(m)}(z) \qquad (16)$$

Under these assumptions, there may be various choices of matrix $G_m(z)$ to satisfy (13). Here, $G_m(z)$ is chosen to be of the form

$$G_m(z) = C_m \begin{bmatrix} z_m & \beta z_m \\ 1 & -\beta \end{bmatrix} \qquad (17)$$

By substituting (17) into (13), taking the conditions of (15) and (16) into consideration, z_{m+1}/z_m is obtained as

$$\frac{z_{m+1}}{z_m} = 1 + \frac{\beta(1+z^{-1})}{1+\beta^2 z^{-1}} \qquad (18)$$

In this case, β is to be chosen so as to make z_{m+1}/z_m independent of z^{-1}. This results in $\beta = \pm 1$ or $\beta = \pm z$. Here, $\beta = 1$ is chosen(other choices of β give equivalent results). Then

$$z_{m+1}/z_m = (1+k_m)/(1-k_m) \qquad (19)$$

and

$$C_{m+1}/C_m = z^{1/2}/(1+k_m) \qquad (20)$$

In (19) $z_m = \rho c/A_m$, ρ defines the air density and c is the speed of sound in the air. Hence we can write (19) as

$$k_m = (A_m - A_{m+1})/(A_m + A_{m+1}) \qquad (21)$$

or

$$A_{m+1}/A_m = (1-k_m)/(1+k_m) \qquad (22)$$

where m = M, M-1,.......1; the constraint on sampling frequency is $f_s = Mc/(2L)$; the number of sections is M, and the length of the acoustic tube, L = Ml. Note that if the two sections A_m and A_{m+1} have identical areas, there is no reflection ($k_m = 0$).

From (24)* and (25)*, we can state that the linear prediction digital filter is equivalent to the non-uniform acoustic tube model of the vocal tract. The boundary condition at the lip end is realized by connecting a tube of infinite area or by short-circuiting the vocal tract tube. At the glottis end, a tube of infinite length with cross-sectional area A_{m+1} is connected, i.e., the vocal tract tube is terminated with the characteristic impedance of $\rho c/A_{m+1}$.

C. BaHli's method (23)* (27)*

One period of the time waveform, starting at the first peak of maximum uniform parts, is divided equally into 30 points (the more points, the more precise the result) on the time axis t. On the screen image the time values of these 30 points and their corresponding amplitude values h(t) are measured. Referencing Fig.13 (shown later) the time function h(t) could be obtained by an inverse Laplace transform of Eq.(28). The first 8 constant coefficients of h(t) can be considered as 8 features applied in the method of parameters classification, or we can use them as follows.

Referring to section A of this chapter the author supposes that each of the 37 Chinese speech time waveforms may be approximated by its corresponding time function with poles and zeros. Comparing this time function of tested speech sample h*(t) with that of the reference time waveform h(t) (see Fig.12) the computer might recognize every sound. The values h(t) or h*(t) were not obtained by the author for each of the possible sounds (reference Appendix 7) because BaHli's method is only suitable for smoothed curves whose functions are linear. Also the transfer functions of speech sounds are only roughly estimated by their poles and zeros which were computed without regard for minimization of total-squared error (reference section A of this chapter). Therefore BaHli's method was not suitable in this instance.

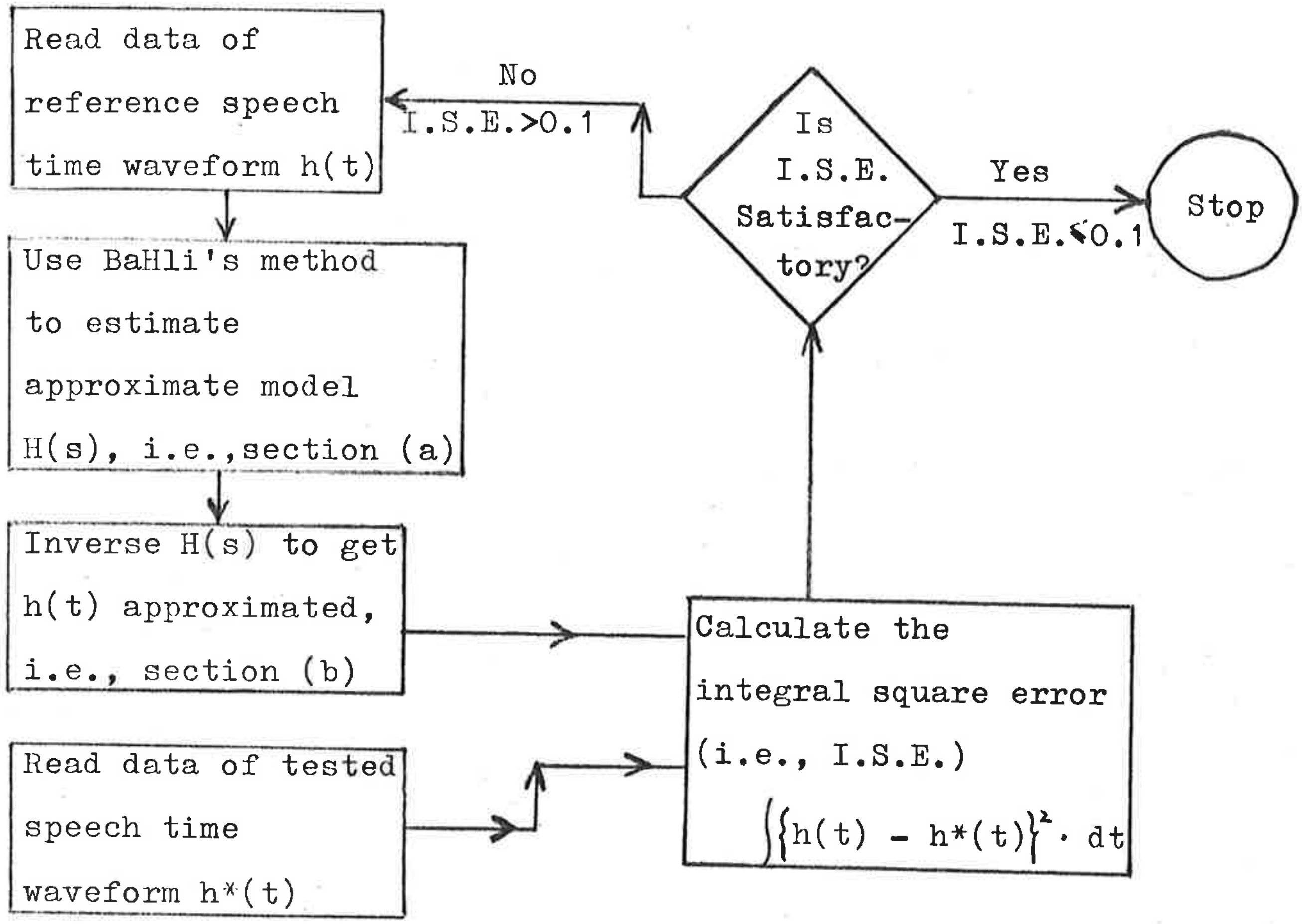


Fig.12 Flow diagram for the identification of speech time-waveform.

BaHli's method for speech recognition is as follows:

(a) Principles of BaHli's method

Assume that the time waveforms of speech are linear, stationary and continuous.

In a stationary condition, the speech time function h(t) is assumed to be zero for $t \geqslant \tau_k$. The Laplace transform of h(t) in discrete form can be expressed as

$$H(s) = \sum_{n=1}^{k} a_n e^{-s\tau_n} \qquad (23)$$

Using the power series expansion for Eq.(23)

$$H(s) = \sum_{n=1}^{k} a_n - \left(\frac{1}{1!}\sum_{n=1}^{k} a_n \tau_n\right)s + \left(\frac{1}{2!}\sum_{n=1}^{k} a_n \tau_n^2\right)s^2 - \ldots\ldots. \qquad (24)$$

The transfer function can also be written as

$$H(s) = b_1 + b_2 s + b_3 s^2 + b_4 s^3 + \ldots\ldots\ldots. \qquad (25)$$

By comparing equations (24) and (25), we obtain

$$b_1 = \sum_{n=1}^{k} a_n \quad , \quad b_2 = -\frac{1}{1!}\sum_{n=1}^{k} a_n \tau_n \quad , \quad \ldots\ldots\ldots.$$

$$b_i = \frac{(-1)^{i-1}}{(i-1)!}\sum_{n=1}^{k} a_n \tau_n^{i-1} \qquad (26)$$

In general the b's should be defined as the continuous form

$$b_i = \frac{(-1)^{i-1}}{(i-1)!}\int_0^{\tau_k} h(t)\, t^{i-1} dt \qquad (27)$$

Assume that BaHli's model of speech estimation H(s) can not respond instantaneously to an input. It's impulse response must then have a finite derivative near the time origin. Therefore the transfer function of BaHli's model will be of the following form.

$$H(s) = P(s)/Q(s) = \frac{p_1+p_2 s+p_3 s^2+\ldots\ldots+p_n s^{n-1}}{1+q_1 s+q_2 s^2+\ldots\ldots+q_m s^m} \qquad (28)$$

where $m \geqslant n$.

Equating (25) and (28) we obtain the following set of simultaneous, linear equations:

$$\begin{aligned}
p_1 &= b_1 \\
p_2 &= b_2+b_1q_1 \\
p_3 &= b_3+b_2q_1+b_1q_2 \\
&\ldots\ldots\ldots\ldots\ldots\ldots \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
p_i &= b_i+b_{i-1}q_1+b_{i-2}q_2+\ldots\ldots+b_1q_{i-1} \\
&\ldots\ldots\ldots\ldots\ldots\ldots
\end{aligned} \qquad (29)$$

The known quantities of Eq.(29) are b_1, b_2, b_3,......,b_i,.... calculated by Eq.(26) and the (m+n) unknowns are p_1, p_2,... ......,p_n, q_1, q_2, ..., q_m. The final value theorem

$$\lim_{s\to 0} sH(s) = \lim_{t\to\infty} h(t) \qquad (30)$$

implies that the higher powers of s can be neglected under steady state conditions ($t\to\infty$) so that the steady state error is negligible. From the initial value theorem

$$\lim_{t\to 0} h(t) = \lim_{s\to\infty} sH(s) \qquad (31)$$

i.e., the transfer function behaves like $p_n s^n/(q_m s^m)$ in the frequency domain when the region near the origin of its counterpart in the time domain is considered.

Fig.13 summarize the procedures of BaHli's method.

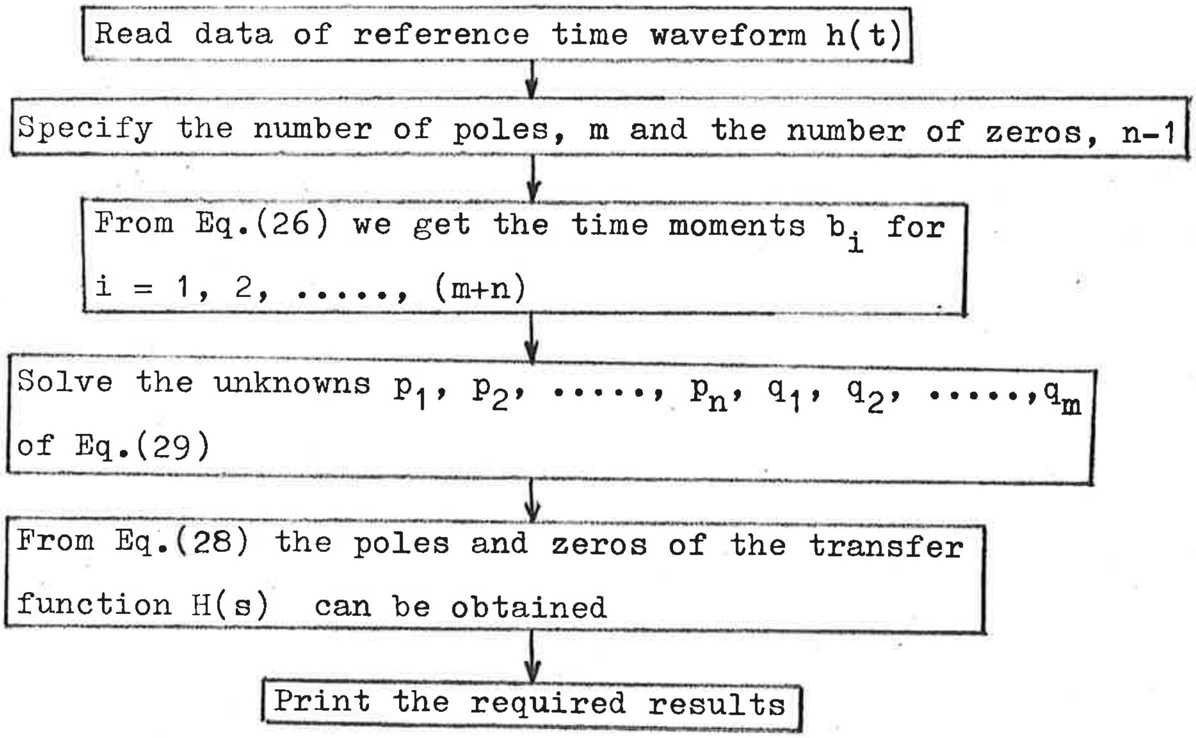


Fig.13 Flow chart of BaHli's method.

(b) Numerical solution of Inverse Laplace transforms

In order to transform the transfer function H(s) of frequency domain into its counterpart h(t) in the time domain we use the following methods.

Consider the transfer function of BaHli's model

$$H(s) = \frac{c_1 s^{n-1} + c_2 s^{n-2} + \ldots\ldots + c_n}{s^n + d_1 s^{n-1} + \ldots\ldots + d_n} \qquad (32)$$

Where $c_1, c_2, \ldots\ldots, c_n, d_1, d_2, \ldots\ldots, d_n$ are constants.

In Eq.(28) if the degree of denominator polynomial, m, is the same as that of the numerator polynomial, n, then the form of Eq.(28) can be turned into the form of Eq.(32). In matrix form Eq.(32) becomes

$$(s^n+d_1 s^{n-1}+......+d_n)\ H(s) = \left(s^{n-1}\ s^{n-2}.....s\ \ 1\right) \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \qquad (33)$$

For $h^{(n)}(t)$ its Laplace transform is

$$L\ h^{(n)}(t) = s^n H(s)-s^{n-1}h(0)-s^{n-2}h^{(1)}(0)-....-h^{(n-1)}(0) \qquad (34)$$

If the Laplace transform of the nth order differential equation

$$h^{(n)}(t)+d_1 h^{(n-1)}(t)+.......+d_{n-1}h^{(1)}(t)+d_n h(t) = 0 \qquad (35)$$

is taken, we get

$$(s^n+d_1 s^{n-1}+....+d_n)\ H(s)$$

$$= s^{n-1}h(0)+s^{n-2}h^{(1)}(0)+.....+h^{(n-1)}(0)$$

$$+d_1 \left\{ s^{n-2}h(0)+......+h^{(n-2)}(0) \right\}$$

$$+.........................$$

$$+d_{n-1}h(0) \qquad (36)$$

Eq.(36) is written as

$$(s^n+d_1 s^{n-1}+\ldots\ldots+d_n)\ H(s)$$

$$= \left[s^{n-1}\ s^{n-2}\ldots\ldots\ s\ 1\right] \begin{bmatrix} 1 & 0 & \ldots\ldots\ldots & 0 \\ d_1 & 1 & \ldots\ldots\ldots & 0 \\ d_2 & d_1 & \ldots\ldots\ldots & 0 \\ \cdot & & \ldots\ldots\ldots & \\ \cdot & & \ldots\ldots\ldots & \\ \cdot & & \ldots\ldots\ldots & \\ d_{n-1} & d_{n-2} & \ldots d_1 & 1 \end{bmatrix} \cdot \begin{bmatrix} h(0) \\ h^{(1)}(0) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ h^{(n-1)}(0) \end{bmatrix} \tag{37}$$

The initial condition vector of the speech time function can be obtained by comparing the right hand sides of (37) and (33), i.e.,

$$\begin{bmatrix} h(0) \\ h^{(1)}(0) \\ \cdot \\ \cdot \\ \cdot \\ h^{(n-1)}(0) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \ldots\ldots\ldots & 0 \\ d_1 & 1 & \ldots\ldots\ldots & 0 \\ \cdot & \cdot & \ldots\ldots\ldots & \\ \cdot & \cdot & \ldots\ldots\ldots & \\ \cdot & \cdot & \ldots\ldots\ldots & \\ d_{n-1} & d_{n-2} & \ldots d_1 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ \cdot \\ c_n \end{bmatrix} \tag{38}$$

i.e., $\left[H(0)\right] = \left[D'\right]^{-1} \left[C\right]$ (39)

Where the matrices in Eq.(39) are defined in Eq.(38).

If we write Eq.(35) in the following matrix form

$$\begin{bmatrix} h^{(1)}(t) \\ h^{(2)}(t) \\ \cdot \\ \cdot \\ \cdot \\ h^{(n)}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots\ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots\ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \ldots\ldots & \\ \cdot & \cdot & \cdot & \cdot & \ldots\ldots & \\ 0 & 0 & \cdot & \ldots\ldots & 0 & 1 \\ -a_n & -a_{n-1} & & \ldots\ldots & & -a_1 \end{bmatrix} \cdot \begin{bmatrix} h(t) \\ h^{(1)}(t) \\ \cdot \\ \cdot \\ \cdot \\ h^{(n-1)}(t) \end{bmatrix} \tag{40}$$

i.e., $$[\dot{H}] = [D]\;[H] \qquad (41)$$

Where the matrices $[D]$ and $[H]$ of (41) are defined as in (40). Hence the solution of the nth order differential equation of Eq.(35) will be reduced to the n first order differential equations as Eq.(41), with initial conditions given by Eq.(38). This kind of differential equation can be solved by the Runge-Kutta method (i.e., subroutine RUNGS). Fig.14 explains the solution procedure for the inverse Laplace transform.

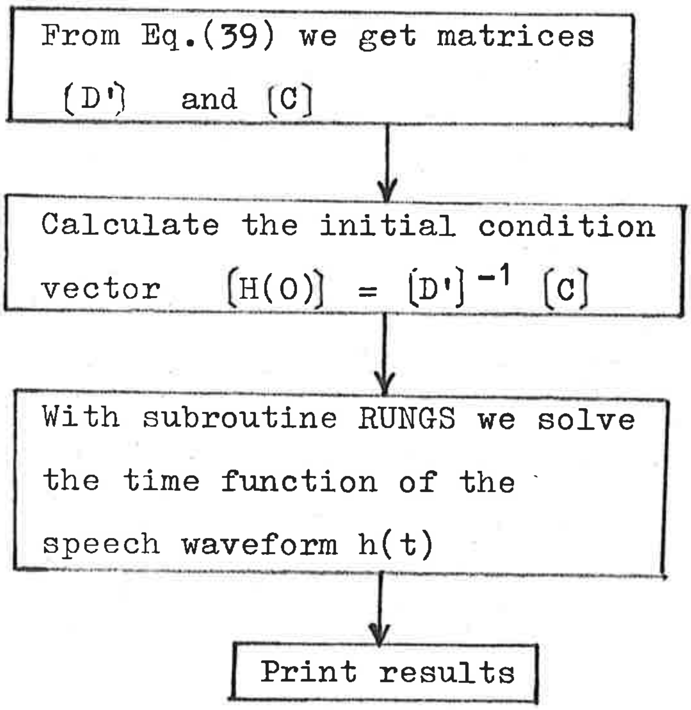


Fig.14 Flow diagram determining h(t) from H(s)

All equations of BaHli's method in this section C were quoted from (23)* and (27)*.

D. Curve fitting (26)*

The fourth method applied to the recognition of speech sounds is curve fitting (26)*. From the time waveform starting at the first peak of maximum uniform parts, we adopt one period of this time waveform, which is divided equally into 30 points (the more the better) on the horizontal axis (i.e., time axis) X. On the screen image we obtain the time values of these 30 points and their corresponding amplitude values (i.e., horizontal time value X vs. corresponding vertical amplitude value Y).

The author attempted to model the function using the first 8 terms of the Fourier series expansion of the waveform from the best fit of the experimental values of (x_1,y_1), (x_2,y_2), and so forth. Referencing Appendix 11 we obtained 8 terms (i.e., $a_0/2$, a_1, a_2, a_3, b_1, b_2, b_3 and b_4) of this Fourier series i.e., $y = f(x) = a_0/2 + \sum_{n=1}^{\infty} a_n \mathrm{Cos}(n\pi x/1) + \sum_{n=1}^{\infty} b_n \mathrm{Sin}(n\pi x/1)$. Note that the period is 21 which corresponds to 2π, and that $f(x+21) = f(x)$. Those 8 coefficients might be considered as the features applied in the method of parameter classification. Unfortunately I failed to obtain these 8 coefficients of each speech sound because the method of curve fitting is only adequate to the solution of smooth curves. Nonlinear functions of speech sounds cannot be approximated by any linear equations.

Traditional Fourier analysis requires a relatively long speech segment to provide adequate spectral resolution. Rapidly changing speech cannot be accurately followed. Also having insufficient information about the spectrum between pitch harmonics meant the frequency-domain techniques didn't perform satisfactorily for highly-pitched voices.

The principles of curve fitting are as following: Curve fitting uses an approximating function which graphs as a smooth curve having the general shape suggested by the data values, but generally not passing exactly through all of the data points.

A function y=f(x), consisting of m functionally linear undetermined constants, C_1, C_2, ..., C_m, for a least-squares fit to n known data values, requires the simultaneous solution of m linear equations. The better procedure can be generalized in the form of a matrix equation to facilitate the generation of m linear equations which can be solved by Cholesky's method, explained in the Appendix 12.

The function y=f(x) can be written in the general form

$$y_i = \sum_{k=1}^{m} C_k f_k(x_i) \qquad (42)$$

in which known values for x_i and y_i correspond to experimental data.

Considering the residuals $r_1, r_2, \ldots, r_n$, for a least-squares fit, it follows from Eq. (42) that

$$\begin{aligned} \sum_{k=1}^{m} C_k f_k(x_1) - y_1 &= r_1 \\ &\cdots\cdots \\ \sum_{k=1}^{m} C_k f_k(x_n) - y_n &= r_n \end{aligned} \qquad (43)$$

where r_1, r_2, and so on, are the "residuals" which would all be zero in the case of a perfect fit. However, the resi -duals will generally not be zero and a best fit is obtained by making them as small as possible. This is accomplished by finding values for $C_1, C_2, \ldots\ldots, C_m$ such that the sum of the squares of the residuals is a minimum. That is,

$$\sum_{i=1}^{n} (r_i)^2 = \min. \qquad (44)$$

Considering r_i as a function of the constants C_1, C_2, and so on, it follows from Eq.(44) that

$$\frac{\partial}{\partial C_k} \sum_i (r_i)^2 = 0$$

or $$\sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial C_k} = 0 \qquad k = 1,2,\ldots\ldots,m \qquad (45)$$

which may be written in the matrix form

$$\begin{bmatrix} r_1 & r_2 & \ldots\ldots r_n \end{bmatrix} \begin{bmatrix} \partial r_1/\partial C_1 & \partial r_1/\partial C_2 & \ldots\ldots \partial r_1/\partial C_m \\ \partial r_2/\partial C_1 & \partial r_2/\partial C_2 & \ldots\ldots \partial r_2/\partial C_m \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \partial r_n/\partial C_1 & \partial r_n/\partial C_2 & \ldots\ldots \partial r_n/\partial C_m \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & \ldots\ldots & 0 \end{bmatrix} \qquad (46)$$

From Eq.(43), we see that $\partial r_i/\partial C_k = f_k(x_i)$ (47)

Thus Eq.(46) may be written in the form

$$\begin{bmatrix} r_1 & r_2 \ldots\ldots r_n \end{bmatrix} \underset{\sim}{F} = \begin{bmatrix} 0 & 0 & \ldots\ldots 0 \end{bmatrix} \qquad (48)$$

where $$\underset{\sim}{F} = \begin{bmatrix} f_1(x_1) & f_2(x_1) \ldots\ldots\ldots f_m(x_1) \\ f_1(x_2) & f_2(x_2) \ldots\ldots\ldots f_m(x_2) \\ \cdots & \cdots \\ \cdots & \cdots \\ f_1(x_n) & f_2(x_n) \ldots\ldots\ldots f_m(x_n) \end{bmatrix} \qquad (49)$$

We now denote $\left[r_1 \ r_2 \ \ldots\ldots r_n\right] = \underset{\sim}{R}$ and take the transpose of Eq.(48) to obtain

$$\left[\underset{\sim}{R}\ \underset{\sim}{F}\right]^T = \left[0\ 0 \ldots\ldots 0\right]^T \qquad (50)$$

Since $\left[\underset{\sim}{R}\ \underset{\sim}{F}\right]^T = \underset{\sim}{F}^T\ \underset{\sim}{R}^T$ and $\left[0\ 0 \ldots\ldots 0\right]^T$ is a column matrix ,Eq.(50) takes the form

$$\underset{\sim}{F}^T \begin{Bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_n \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{Bmatrix} \qquad (51)$$

where

$$\underset{\sim}{F}^T = \begin{bmatrix} f_1(x_1) & f_1(x_2) & \ldots\ldots & f_1(x_n) \\ f_2(x_1) & f_2(x_2) & \ldots\ldots & f_2(x_n) \\ \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ f_m(x_1) & f_m(x_2) & \ldots\ldots & f_m(x_n) \end{bmatrix} \qquad (52)$$

and from Eq.(43)

$$\begin{Bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_n \end{Bmatrix} = \begin{Bmatrix} \sum_{k=1}^{m} C_k f_k(x_1) \\ \sum_{k=1}^{m} C_k f_k(x_2) \\ \cdot \\ \cdot \\ \cdot \\ \sum_{k=1}^{m} C_k f_k(x_n) \end{Bmatrix} - \begin{Bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{Bmatrix} \qquad (53)$$

From Eq.(53), it is apparent that Eq.(51) may be written in the form

$$\underset{\sim}{F}^T \begin{Bmatrix} \sum_{k=1}^{m} C_k f_k(x_1) \\ \sum_{k=1}^{m} C_k f_k(x_2) \\ \cdot \\ \cdot \\ \cdot \\ \sum_{k=1}^{m} C_k f_k(x_n) \end{Bmatrix} = \underset{\sim}{F}^T \begin{Bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{Bmatrix} \qquad (54)$$

According to Eq.(49), we note that

$$\begin{Bmatrix} \sum_{k=1}^{m} C_k f_k(x_1) \\ \sum_{k=1}^{m} C_k f_k(x_2) \\ \vdots \\ \sum_{k=1}^{m} C_k f_k(x_n) \end{Bmatrix} = \underset{\sim}{F} \begin{Bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{Bmatrix} \qquad (55)$$

Thus Eq. (54) may be written in the form

$$\underset{\sim}{F}^T \underset{\sim}{F} \begin{Bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{Bmatrix} = \underset{\sim}{F}^T \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{Bmatrix}$$

or more simply as $\underset{\sim}{F}^T \underset{\sim}{F} \{C\} = \underset{\sim}{F}^T \{y\}$ (56)

The use of Eq.(56) facilitates the formulation of a system of m linear algebraic equations for the least-squares curve-fitting criterion since the $\underset{\sim}{F}$ matrix is readily generated from the selected f(x) function and the known data values for x.

All theories of the curve fitting routine were quoted from (26)*.

II. 3 Basic theories about recognition process of linear prediction coefficients, area functions, or other kinds of coefficients with respect to their corresponding time waveforms of 37 Chinese speech sounds (13)*, (2)*, (3)*, (4)*.

this is a heading

From equations (8), (9) and (1) we can find the linear prediction coefficients of any sound. Also from (8), (9) and (22) we can obtain the vocal tract area functions or the reflection coefficients of sounds. Because every sound has its corresponding time-waveform, those parameters can be used as the features to distinguish between two isolated speech sounds (18)*. Using linear prediction coefficients and parameter classification (an example of speech recognition) is explained in Chapter II. 4. Principles of the process of parameter recognition are described as follows.

Shown in Fig.15 are all the labeled or reference samples of parameters collected and the method of determining the best decision rules to classify unlabeled samples.

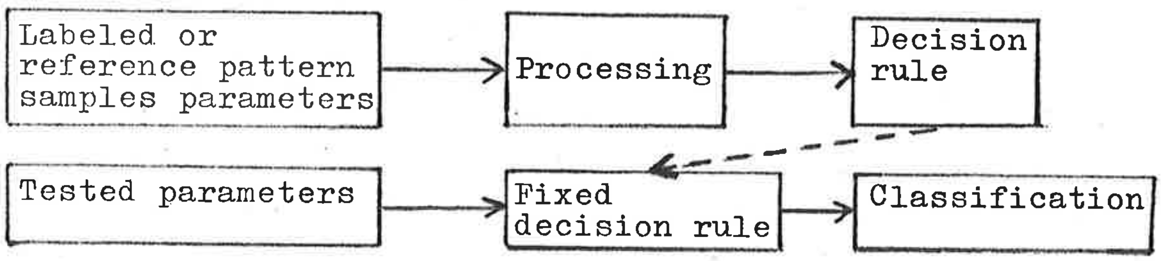


Fig.15 Learning before recognition as a classification strategy.

In order to derive the decision rule for parameter recognition (Fig.15), the procedure shown in Fig.16 is used. The speech model from which given parameters arise is characterized, completely, only by its physical embodiment. This physical embodiment is represented numerically by some set of measurements. A sample of the speech waveform is characterized by some specific values of all the measurements, corresponding to points in the measurement space. The parameter space may be identical with the measurement space, or several stages of intermediate processing may be necessary.

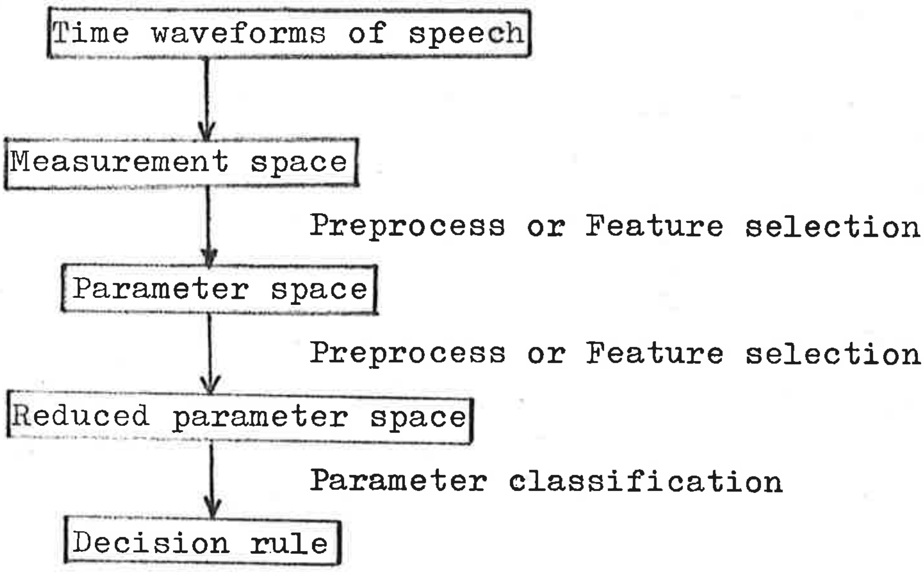


Fig.16 Flow diagram for derivation of the decision rule.

Feature selection or preprocessing is the process by which a sample in the measurement space is described by a finite and usually smaller set of numbers called features, say x_1, x_2, ..., x_n, which become components of the parameter space. As in Fig.17 each sample becomes a point in the parameter space, once the features are defined.

The decision rule is derived from the set of labeled samples using the parameter classification algorithm.

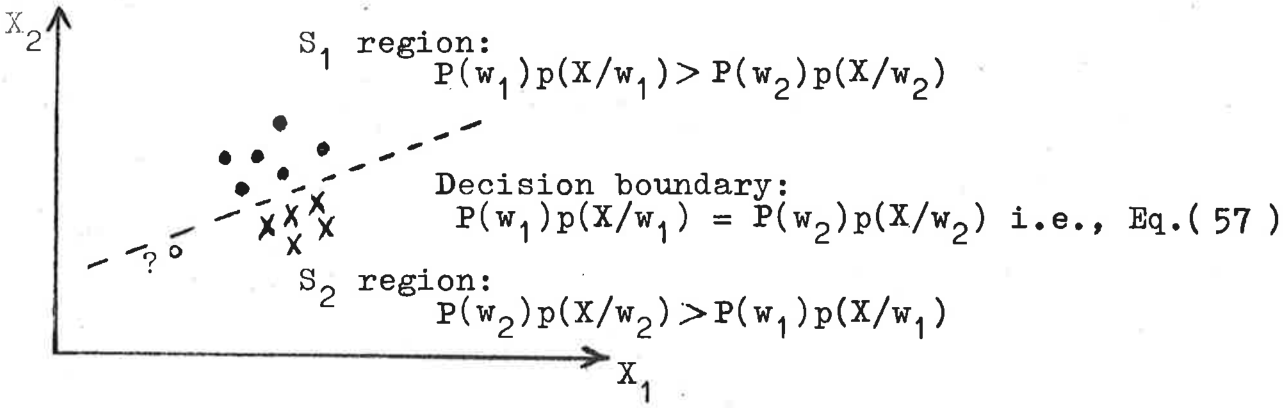


Fig. 17 Samples in the two-dimensional parameter space:
• samples of class S_1; X samples of class S_2;
o unknown parameter.

In Fig. 17 the labeled samples belong to two parameter classes; S_1 and S_2. To classify an unknown point, such as the one indicated, into one of the two classes, Fig. 17 is used. This diagram relates the assumed correspondence between two samples in the parameter space, to the similarity of the parameters which represent them.

Instead of matching the speech time waveforms with their reference templates, recognition is based on a set of selected measurements extracted from the tested waveforms. These selected measurements, called "features", are invariant with, or less sensitive to the commonly encountered variations and distortions, and also contain less redundancies.

For each parameter class w_j, $j = 1, \ldots\ldots, m$ we assume that the multivariate (N-dimensional) probability density (or distribution) function of the feature vector X is equal to $p(X/w_j)$, and the probability of occurrence of w_j is $P(w_j)$. Having the known informations $p(X/w_j)$ and $P(w_j)$, j = 1, ..., m, the classifier is used to minimize the probability of misrecognition. Parameter classification can now be formulated as a statistical decision problem (testing of m statistical hypotheses) by defining a decision function d(x), where $d(x) = d_i$ means that the hypothesis $H_i : X \sim w_i$ is accepted.

If $P(w_i)p(X/w_i) \geqslant P(w_j)p(X/w_j)$ for all j = 1, ...,m then by means of the Bayes decision rule we get

$$d = d_i, \quad \text{i.e.} \quad X \sim w_i \qquad (57)$$

The likelihood ratio λ between classes w_i and w_j is defined as

$$\lambda = p(X/w_i)/p(X/w_j) \qquad (58)$$

so (57) becomes

$$d = d_i \quad \text{if} \quad \lambda \geqslant P(w_j)/P(w_i) \quad \text{for all } j = 1,....,m \qquad (59)$$

With a mean vector M_i, and a covariance matrix K_i, the multivariate Gaussian density function can be represented as following.

$$p(X/w_i) = \left\{(2\pi)^{N/2}|K_i|^{1/2}\right\}^{-1} \operatorname{Exp}\left\{-(1/2)(X-M_i)^T K_i^{-1}(X-M_i)\right\} \qquad (60)$$

where i = 1, ..., m.

Following Wald's sequential analysis, it has been shown that a classifier using the sequential probability ratio test (SPRT) has an optimal property for the case of two parameter classes (14)* (15)* (16)*. After the nth feature is measured, the classifier computes the sequential likelihood ratio

$$\lambda_n = p_n(X/w_1)/p_n(X/w_2) \qquad (61)$$

where the multivariate n-dimensional $p_n(X/w_i)$, i = 1, 2, is the conditional probability density function of X for parameter class w_i. The coefficient λ_n computed by (61) is then compared with the upper stopping boundary A and the lower stopping boundary B. If

$$\lambda_n \geqslant A, \text{ then } \quad X \sim w_1 \qquad (62)$$

and if $\lambda_n \leqslant B$, then $X \sim w_2$ (63)

An additional feature measurement will be taken when $B < \lambda_n < A$, and the process is proceeding to the (n+1)th stage.

Suppose that e_{ij} is the probability of deciding $X \sim w_i$ when actually $X \sim w_j$ is true, i,j = 1,2. At the nth stage of the recognition process, we find that $\lambda_n = A$ (64) leading to a decision of class w_1. From (61) and (64), $p(X/w_1) = Ap(X/w_2)$ which is equivalent to

$$\int p(X/w_1)dX = A \int p(X/w_2)dX \qquad (65)$$

where both integrations are over the region for accepting class w_1. By the definitions of e_{12} and e_{21}, (65) is reduced to

$$(1 - e_{21}) = Ae_{12}$$

Similarly; if $\lambda_n = B$, we get $e_{21} = B(1 - e_{12})$

i.e. $A = (1 - e_{21})/e_{12}$ and $B = e_{21}/(1 - e_{12})$

(66)

Equations (62) and (63) are used to represent the decision boundaries which divide the feature space into three regions: the region associated with w_1; the region of w_2; and the null region. In a sequential classification process, the decision boundaries are varied with the number of feature measurements n. For example, suppose that x_1, x_2, ... are independent feature measurements with $p(x_j/w_i)$, j = 1, 2, ..., i = 1, 2, a univariate Gaussian density function with mean m_i and variance σ^2. For the nth stage of this process,

$$\log \lambda_n = \sum_{i=1}^{n} \log p(x_i/w_1)/p(x_i/w_2)$$

$$= \frac{(m_1-m_2)}{\sigma^2} \sum_{i=1}^{n} \{x_i-(1/2)(m_1+m_2)\} \qquad (67)$$

the procedure of classification is listed as following:

if $\sum_{i=1}^{n} x_i \geqslant \frac{\sigma^2}{m_1-m_2} \log A + (n/2)(m_1+m_2)$, then the decision is that $x \sim w_1$;

if $\sum_{i=1}^{n} x_i \leqslant \frac{\sigma^2}{m_1-m_2} \log B + (n/2)(m_1+m_2)$, then the decision is that $x \sim w_2$;

and if $\frac{\sigma^2}{m_1-m_2} \log B + (n/2)(m_1+m_2) < \sum_{i=1}^{n} x_i < \frac{\sigma^2}{m_1-m_2} \log A + (n/2)(m_1+m_2)$,

then x_{n+1} will be observed.

To improve the accuracy of classification the SPRT is used to reduce the error (misrecognition) probability by varying stopping boundaries. Another advantage of varying stopping boundaries is that, by starting with relatively large values of stopping boundaries and gradually decreasing them, the average number of feature measurements will not be excessively large compared with the case in which small values of stopping boundaries are used through the whole process. Consequently, the probability of misrecognition and the average number of feature measurements can be to some degree, simultaneously controlled by properly adjusting the stopping boundaries.

The selection of features is closely related to the performance of classification. The ordering of features for successive measurements is also very important in the recognition system of sequential parameters. Feature ordering is used to provide a feature which is the most informative among all possible choices of features for the next measurement so the recognition process can be terminated earlier.

Let $L = \log\lambda = \log p(X/w_i) - \log p(X/w_j)$ (68)

Substituting (60) into (68) we have

$$L = \log\lambda = (1/2)\log\frac{|K_j|}{|K_i|} + (1/2)(X-M_j)^T K_j^{-1}(X-M_j) - (1/2)(X-M_i)^T K_i^{-1}(X-M_i) \qquad (69)$$

When $|K_i| = |K_j| = K$, from (69) we obtain

$$L = X^T K^{-1}(M_i-M_j) - (1/2)(M_i+M_j)^T K^{-1}(M_i-M_j) \qquad (70)$$

and

$$E\left[L/w_i\right] = (1/2)(M_i-M_j)^T K^{-1}(M_i-M_j) \qquad (71)$$

Define the divergence between w_i and w_j as

$$J(w_i, w_j) = E\left[L/w_i\right] - E\left[L/w_j\right] \qquad (72)$$

Then, from (71) and (72)

$$J(w_i, w_j) = (M_i-M_j)^T K^{-1}(M_i-M_j) \qquad (73)$$

In (73) if the covariance matrix K is equal to the identity matrix I, then $J(w_i, w_j)$ represents the squared distance between M_i and M_j.

Using a fixed-sample size or nonsequential Bayes' decision rule to the classifier, if $P(w_i) = P(w_j) = 1/2$, from (68) we obtain

$$X \sim w_i \quad \text{if} \quad \lambda > 1, \text{ or } L > 0$$

and

$$X \sim w_j \quad \text{if} \quad \lambda < 1, \text{ or } L < 0$$

The probability of misrecognition is

$$e = (1/2)\, P\{L > 0/w_j\} + (1/2)\, P\{L < 0/w_i\} \qquad (74)$$

From (70), (71) and (73) we know that $p(L/w_i)$ is a Gaussian density function with mean $(1/2)J$ and variance J. Also $p(L/w_j)$ is a Gaussian density function with mean $(-1/2)J$ and variance J. So from (74),

$$e = (1/2)\int_0^{\infty} (2\pi J)^{-1/2}\ \mathrm{Exp}\{-(1/2)(\theta + \tfrac{1}{2}J)/J\}\cdot d\theta$$
$$+ (1/2)\int_{-\infty}^{0} (2\pi J)^{-1/2}\ \mathrm{Exp}\{-1/2(\theta - \tfrac{1}{2}J)/J\}\cdot d\theta$$

Let $y = (\theta \pm \frac{1}{2}J)/\sqrt{J}$ then $e = \int_{\sqrt{J}/2}^{\infty} (2\pi)^{-1/2}\ \mathrm{Exp}(-\frac{1}{2}y^2)\cdot dy$ (75)

From (75), e is a monotonically decreasing function of $J(w_i, w_j)$. Therefore, features selected or ordered according to the magnitude of $J(w_i, w_j)$ will imply their corresponding discriminatory power between parameter classes w_i and w_j with Gaussian distributed feature measurement vector X.

Hence the information of divergence relating to parameter classes characterized by the features has been proposed as the criteria for feature "goodness".

II.4 Example

On the screen image we can define the required half-block of time waveforms starting at the first peak of maximum uniform parts. This half block contains 128 addresses (reference Chapter I.3 of this thesis). Equally divide this half-block into 128 sample values at the rate of 10 KHz.

(A) With 128 sample values the digital computer program - MBREC (Appendix 8) was used to compute the autocorrelation functions, then solve the normal equations to obtain the linear prediction coefficients by means of Durbin's procedure.

(B) With 128 sample values we use the digital computer program - QBREC (Appendix 9) we first calculate the autocorrelation functions, then the reflection coefficients utilizing Durbin's algorithm. From those reflection coefficients we get the area functions.

The sequential probability ratio test was used for classifying linear prediction coefficients (or area functions) with respect to the sound time-waveforms. Referring Fig.19 the mean vectors and the covariance matrices for the features of linear prediction coefficients (or area functions) were estimated from 20 test samples. Standard feature values of each kind of 37 Chinese speech sounds were established by 20 samples recorded from 20 Chinese males. Unknown feature values of any kinds of 37 Chinese speech sounds were obtained from 20 samples of Australian male students and Chinese males in Adelaide, South Australia.

Recognition of speech sounds ㄅ and ㄇ. The eight features of linear prediction coefficients(or area functions) were treated as though they were statistically independent for convenience, and were measured sequentially(refer the results of Appendix 8 or 9). Referencing Fig. 22 and formula (66) the stopping boundaries were preset symmetrically (log B = -log A) at successive stages as

±2.1, ±2.2, ±2.3, ±2.4, ±2.5, ±2.6, ±2.7, ±2.8

The classification process was truncated at the eigth stage. Referencing Figs 20 , 21 and formula (73) the features of linear prediction coefficients (or area functions), if ordered, were arranged according to the descending order of their corresponding divergences. For speech sounds ㄅ and ㄇ using the methods of linear prediction coefficients and parameter classification (reference Fig. 22), the order is x_3, x_7, x_2, x_1, x_5, x_8, x_6, x_4. Their recognition results are shown in the following table.

	Input			
	Features unordered		Features ordered	
Output	ㄅ	ㄇ	ㄅ	ㄇ
ㄅ	20	0	20	0
ㄇ	16	4	14	6
Recognition percentage	100%	20%	100%	30%
Total number of recognition to accomplish those above recognition percentage	46		49	
Average number of recognition *	1.15		1.225	

* i.e. total number of recognition times to accomplish a specific recognition percentage divided by total number of test samples

Note that, for the same set of stopping boundaries (reference formula (66)) the classification process for the case of ordered features terminates earlier (The time needed for each recognition decreases.) than that for the unordered features (i.e., following the natural order x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) but doesn't have to be more precise.

The flow charts for the process are listed on the following pages. The computer programs for this example are shown in the Appendix 2.

For the speech sounds ㄅ and ㄇ the author chose 15 standard (or reference) feature values of each sound. They are sorted as classes A and B respectively. For each computer run, no more than 5 unknown (or tested) feature values of any kind of the 37 Chinese speech sounds can be set into each class (i.e. A or B). In this example I put 5 unknown feature values of any speech sounds into each class (i.e. ㄅ or ㄇ), the digital computer could differentiate them into classes A (i.e. ㄅ) and B (i.e. ㄇ) or other (unrecognized). Any unknown feature values of unrecognized classes can be used as the tested feature values in the next computer run until all the sounds have been classified. Of course if only one unknown feature value of any sound was tested among 39 standard feature values of speech sounds ㄅ and ㄇ , its recognition result would be the best.

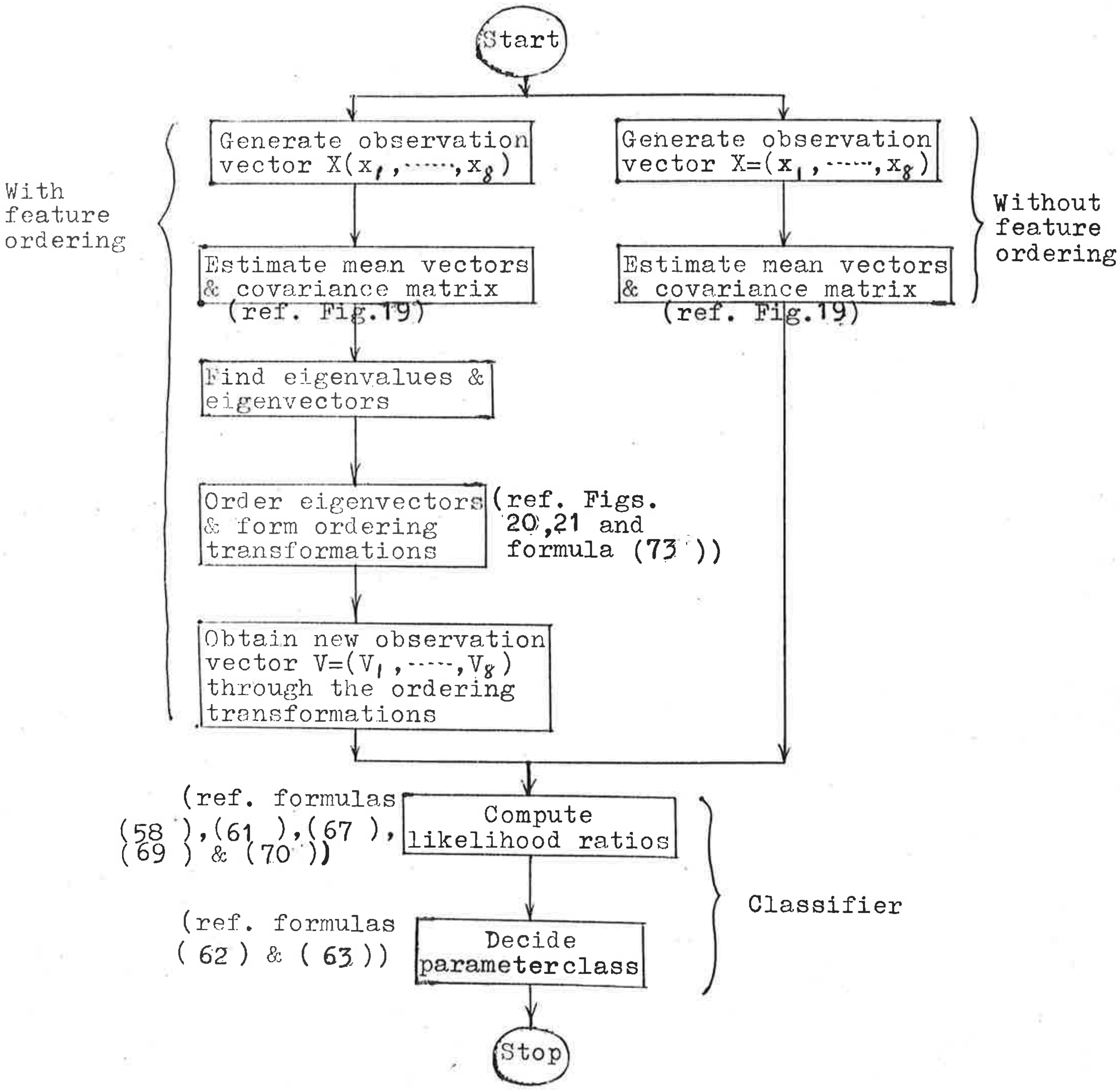


Fig. (18) A simplified flow diagram of Computer-simulated recognition system.

MEANCO

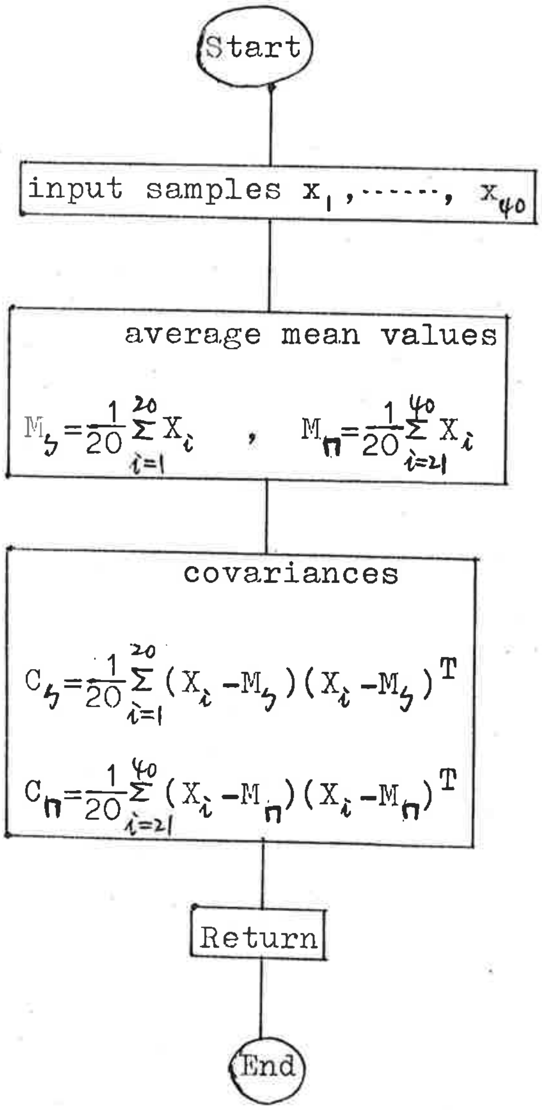


Fig. (19) Flow diagram of Mean values and Covariances.

DIVER(DIV)

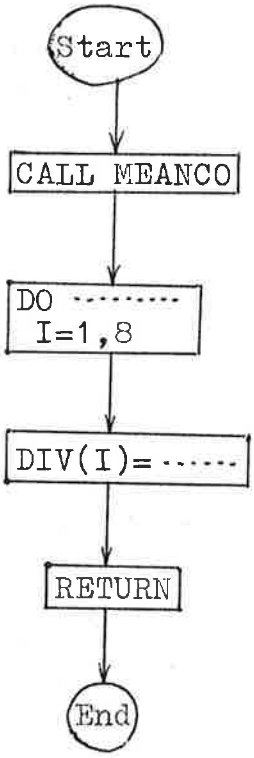


Fig. (20) Flow diagram of Divergence.

$$DIV(w_\xi, w_\eta/X)=\frac{1}{2}tr(CMAX_\xi-CMAX_\eta)(CMAX_\eta^{-1}-CMAX_\xi^{-1})$$
$$+\frac{1}{2}tr(CMAX_\xi^{-1}+CMAX_\eta^{-1})(AM_\xi-AM_\eta)(AM_\xi-AM_\eta)^T$$

Reference the formula (73)

ORDER(OD)

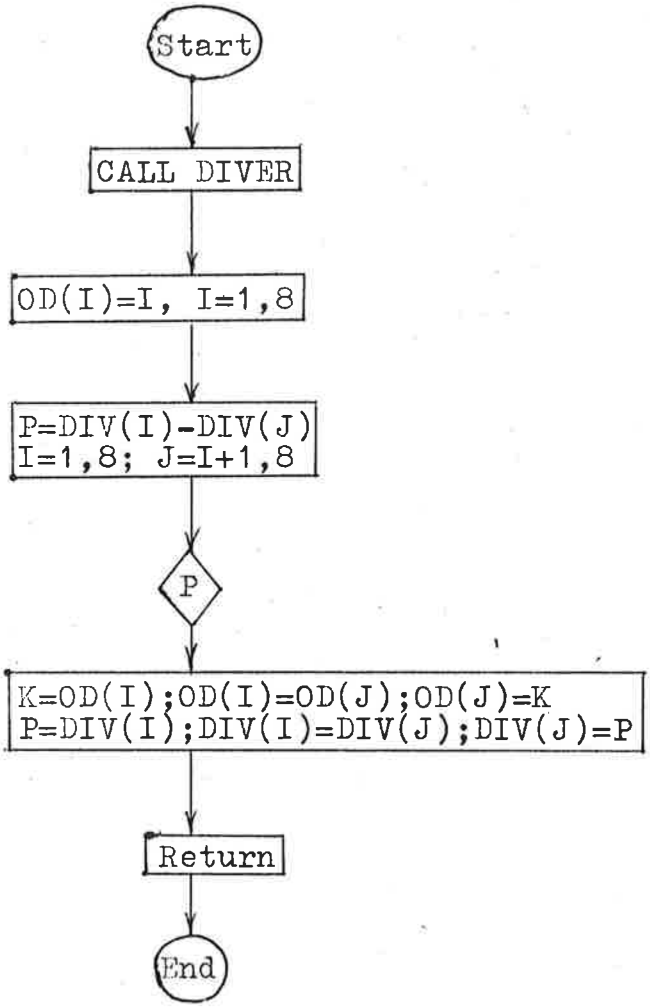


Fig. (21) Flow diagram of ORDER.

AM(j,I)	I th element of mean vector of class j
CM(j,I,I)	I th column, I th row element of Covariance matrix of class j
NNI	input number of X
NK	sum number of stages
B	stopping boundary

λ_i sequential likelihood probability ratio

L1 number of features

$RA(I)=\log\lambda_i$, $SUM=\log\lambda_1+\log\lambda_2+\cdots+\log\lambda_8$ (Ref. formulas (67),(68),(69))

Assume that $e=e_{21}=e_{12}=0.1$ or 10% (i.e. % error of required probability)

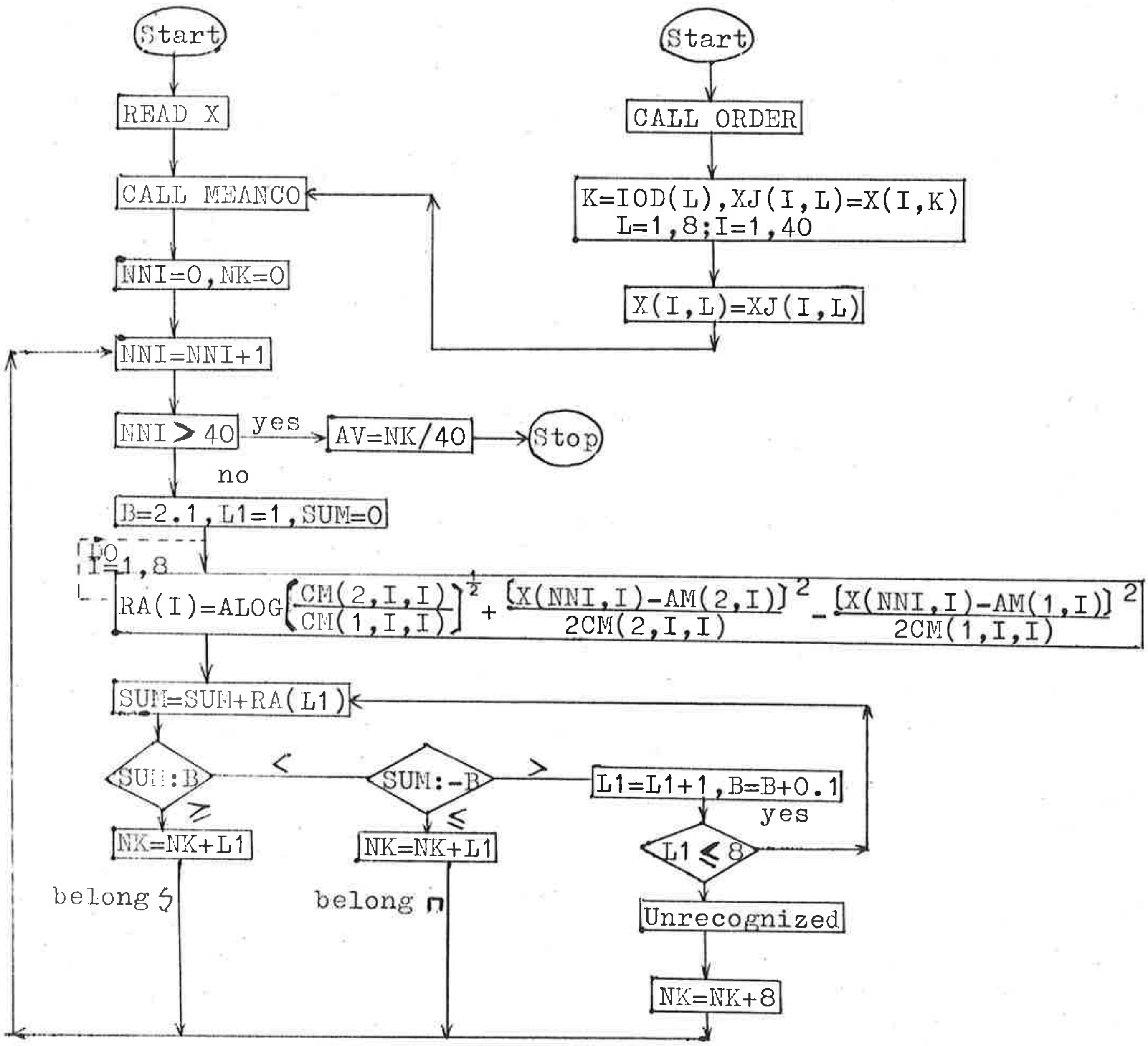


Fig. (22) Flow chart of main computer program.

II.5 Conclusions

Referencing the papers of John Makhoul (5)* and Fumitada Itakura(18)*, linear prediction coefficients are derived for two sounds, are used to describe those sounds.

The author also uses Wakita's method (25)* finding the area functions of speech waveforms, and then utilizing parameter classification, identifies two Chinese speech sounds each time. Although the reflection coefficients of Wakita's method are quite different from predictor coefficients of linear prediction model, we use parameter classification which still can recognize two Chinese speech sounds each time around the same identification percentage.

About 24 Chinese speech sounds could be recognized among the possible total of 37, a success rating of about 65%. Those groups of sounds which couldn't be recognized might be classified using their spectrograms, spectral distributions of energy, and F1 vs. F2 figures. I regret that I didn't complete the research due to a lack of time.

We can define a good decision boundary quite readily in two dimensions. Valid representation in higher dimensions will require some advanced skills and complicated computer programmes (13)*, (14)*, (15)*.

Linear prediction coefficients (or area functions) of the time-waveforms of each kind of sound collected from people, do vary slightly (Ref. (18)* and data collected by the author) and therefore the method of parameter classification is suitable for the recognition of simple speech sounds.

Originally the author assumed that the representations of the time waveforms of any Chinese speech sounds might be approximated by their corresponding functions of time which could be obtained by means of BaHli's method or the method of curve fitting. To identify speech sounds from each other I adopt those coefficients of time functions as the features of the recognition process mentioned in part II. Unfortunately my aim failed because I couldn't get certain coefficients of any Chinese speech sounds.(i.e. The digital computer failed to print the results required.)

PART III: CONCLUSIONS

In China all Chinese words are pronounced by the spelling of Roman alphabets but in Taiwan the pronunciations of any Chinese words are still composed with 37 Chinese speech sounds and their four tones. This thesis is the first work in the research of all the 37 possible Chinese speech sounds; previously only Mr. J.H. Lin has done any basic research on Chinese vowels. This was done in 1974 (7)*. From this thesis we can obtain some preliminary knowledge about Chinese speech sounds. Pronunciations of Chinese are different from those of English. The same spelling sound with different tones may represent different words.

I suggest the methods of linear prediction (5)* or area function (25)* and parameter recognition which are utilized to distinguish the sounds of Chinese words. Because any one of Chinese words has only one kind of sound, and any Chinese sentences can be separated into words which are mutually independent. Pronunciations of Chinese sentences are those of grammatical words spelled one by one.

In a word, using the methods of linear prediction (5)* or area function (25)* and parameter recognition we can discriminate the sounds of Chinese words or sentences easily. But the sounds of English words or sentences are hard to be identified by means of the same method. Since the sound characteristics of English are quite different from those of Chinese. Sounds of every English words are composite of several phonemes those are difficult to decide the boundaries of segmentation. But every Chinese word is pronounced only one corresponding syllable which isn't worried at the problem of segmentation.

I haven't the knowledge of dynamic programming (18)*, so the author couldn't use the digital computer to classify more speech sounds simultaneously. Only two kinds of speech sounds are identified each time.

From the waveshapes of time waveform we could classify 37 Chinese speech sounds except the following groups: ㄐ, ㄒ & ㄓ; ㄛ, ㄟ & ㄡ; ㄎ & ㄏ; ㄑ & ㄕ; ㄖ, ㄗ, ㄘ & ㄙ; ㄜ & ㄝ; ㄠ & ㄤ; ㄧ & ㄩ or ㄣ & ㄦ by observations of their similarities in time waveform. The digital computer also failed in identifying those groups of sounds because the 128 sample values of corresponding time waveforms are duplicate from each other among the data collected by the author. Those results are appended in the end of each table i.e. Appendix 10. For them the percentages of recognitions with each method (i.e. linear prediction coefficients or vocal tract areas) are equal to or less than 50%.

In order to test other pairs of sounds the digital computer needs runs of 276 times for recognitions of each method (i.e. linear prediction coefficients or vocal tract areas).

The author lists all results of 37 Chinese speech sounds distinguished by digital computer CDC cyber 173 as shown in Appendix 10. Here I only narrate briefly as following:

A. Recognitions with linear prediction coefficients

Except the following pairs of sounds, the percentages of recognitions for other pairs of sounds within the computer runs of 276 times are equal to 100% (i.e. They are identified completely).

Pairs of sounds	Percentage of recognition	
ㄅ, ㄇ	ㄅ 100%	ㄇ 30%
ㄈ, ㄜ	ㄈ 100%	ㄜ 95%
ㄇ, ㄋ	ㄇ 95%	ㄋ 100%
ㄇ, ㄌ	ㄇ 95%	ㄌ 100%
ㄇ, ㄗ	ㄇ 15%	ㄗ 100%
ㄇ, ㄖ	ㄇ 95%	ㄖ 100%
ㄇ, ㄜ	ㄇ 25%	ㄜ 100%
ㄇ, ㄟ	ㄇ 95%	ㄟ 100%
ㄇ, ㄠ	ㄇ 95%	ㄠ 100%
ㄇ, ㄣ	ㄇ 95%	ㄣ 100%
ㄇ, ㄧ	ㄇ 15%	ㄧ 100%
ㄊ, ㄞ	ㄊ 100%	ㄞ 95%
ㄋ, ㄥ	ㄋ 100%	ㄥ 95%
ㄞ, ㄠ	ㄞ 95%	ㄠ 100%
ㄞ, ㄣ	ㄞ 95%	ㄣ 100%
ㄞ, ㄨ	ㄞ 95%	ㄨ 100%
ㄣ, ㄧ	ㄣ 95%	ㄧ 100%

I adopt the better recognition percentages of each sound pair, doesn't matter the methods of parameter classification with or without ordered features. Except for the example of II.4 the results of parameter classification without ordered features are better or equal to those of parameter classification with ordered features in most cases. (Refer to II.4 and Appendix 2)

B. Recognitions with vocal tract areas

Except for the following pairs of sounds, the percentages of recognitions for other pairs of sounds within the computer runs of 276 times are equal to 100% (i.e. They are identified completely.).

Pairs of sounds	Percentage of recognition	
ㄇ, ㄚ	ㄇ 90%	ㄚ 100%
ㄋ, ㄠ	ㄋ 100%	ㄠ 95%
ㄋ, ㄥ	ㄋ 100%	ㄥ 70%
ㄖ, ㄠ	ㄖ 100%	ㄠ 95%
ㄖ, ㄥ	ㄖ 100%	ㄥ 95%
ㄛ, ㄢ	ㄛ 95%	ㄢ 100%
ㄥ, ㄧ	ㄥ 100%	ㄧ 95%
ㄥ, ㄨ	ㄥ 95%	ㄨ 100%
ㄧ, ㄨ	ㄧ 95%	ㄨ 100%

I adopted the better recognition percentages of each sound pair, doesn't matter the methods of parameter classification with or without ordered features.(Refer to II.4 and Appendix 2)

In all cases the results of parameter classification without ordered features are better or equal to those of parameter classification with ordered features.

From those results of recognition listed as above we know the method of vocal tract area could be a little bit better than that of linear prediction coefficients.

Those methods utilized in this thesis are more simple than those of Fumitada Itakura (18)* since the latter can recognize many sounds simultaneously with the aid of complicated dynamic programming, but the principles of identification are almost the same.

Methods listed in this thesis are only suitable for recognitions of single sounds (or phonemes) from each other because it is not concerned with segmentation and matching of different phonemes within a single word.

Using BaHli's method (or the method of curve fitting) we can't discriminate speech time-waveforms one by one because BaHli's method (or the method of curve fitting) is only suitable for the smoothed curves. Also the speech time waveforms have to be modeled dynamically as a linear combination of its past values, present and past values of a hypothetical input to a system. BaHli's method or the method of curve fitting is based on the assumption that any speech time waveform could be modeled statically as some linear equations which have certain coefficients. But this kind of assumption isn't consistent with the actual condition so we can't use BaHli's method or the method of curve fitting to recognize speech sounds.

PART IV

REFERENCES

1. Gunnar Fant: Speech sounds and features
The MIT Press, U.S.A. 1973
2. Nilo Lindgren: Machine recognition of human language
Part I- Automatic Speech Recognition p115-p136
IEEE Spectrum Vol.2,No.1 March 1965
Part II- Theoretical models of speech perception
and language p45-p59
IEEE Spectrum Vol.2,No.1 April 1965
3. L.C.W. Pols, H.R.C. Tromp, and R. Plomp:
Frequency analysis of Dutch vowels from 50 male speakers p1093-p1101 The Journal of the Acoustical Society of America Vol.53,No.4 1973
4. W. Klein, R. Plomp and L.C.W. Pols:
Vowel spectra, Vowel spaces, & Vowel identification p999-p1008 The Journal of the Acoustical Society of America Vol.48,No.4(part 2) 1970
5. John Makhoul: Linear prediction: A Tutorial Review
p561-p578 Proc. of the IEEE Vol.63,No.4 April 1975
6. J. L. Flanagan: Speech Analysis Synthesis and Perception Springer-Verlag,
2nd expanded edition, 1972
7. J.H. Lin: Frequency Analysis of Chinese Vowels from 20 male speakers
Telecommunication Labs., Ministry of Communication, Taiwan, R. O. C. July 1974

8. John Rogers: Subroutines- GRAPH, UNSHF, DPREAD, DIFF, WAKITA & REPLAY, May 1976 Dept. of Electrical Engineering, The Univ. of Adelaide, South Australia

9. David Fensom: Program generating a spectrograph of part of the speech waveform held on disk, May 1976 Dept. of Electrical Engineering , The Univ. of Adelaide, South Australia

10. R. E. Bogner: Speech Spectrometry-The measurement of speech spectra Invited article for Encyclopaedic Dictionary of Physics, Pergamon. Suppl. Vol.5, p298-p304 (J. Thewlis, Ed.)

11. Alan V. Oppenheim: Speech spectrograms using the fast Fourier transform IEEE Spectrum August 1970

12. Liang Shih-chiu: Far East English-Chinese Dictionary The Far East Book Co., Ltd. Taipei, Taiwan 1975

13. Meisel: Computer-oriented approaches to Pattern Recognition, Academic Press, 1972

14. K. S. Fu: Sequential Methods in Pattern Recognition and Machine Learning, Academic Press, 1968

15. G. M. Mendel and K. S. Fu: Adaptive Learning and Pattern Recognition System, Academic Press, 1970

16. A. Wald: Sequential Analysis, Wiley, 1947

17. D. Raj Reddy: Speech Recognition by Machine: A Review, Proc. of the IEEE, Vol. 64, No. 4, April 1976

18. F. Itakura: Minimum prediction residual principle applied to speech recognition, IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-23, p67-p72, Feb. 1975

19. G. L. Clapper: Automatic word recognition, IEEE Spectrum, August 1971

20. John N. Shearme & Peter F. Leach: Some Experiments with a Simple Word Recognition System, IEEE Trans. on Audio and Electroacoustics, Vol. AU-16, No.2, June 1968

21. David J. Broad: Basic Directions in Automatic Speech Recognition, Int. J. Man-machine Studies, 1972, 4, p105-p118

22. George M. White and Richard B. Neely: Speech Recognition Experiments with Linear prediction, Bandpass filtering, and Dynamic Programming, IEEE Trans. on Acou. Speech and Signal Processing, Vol. ASSP-24, No. 2, April 1976

23. Alex Yew-Choy Quan: On-line system identification, Master thesis, Dept. of Electrical Engineering, The Univ. of Adelaide, South Australia

24. J. D. Markel and A. H. Gray, Jr.: Linear Prediction of Speech, Springer-Verlag, 1976

25. Hisashi Wakita: Direct Estimation of the Vocal Tract Shape by Inverse Filtering of Acoustic Speech waveforms, IEEE Trans. on Audio and Electroacoustics, Vol. AU-21, No. 5, Oct. 1973

26. M. L. James, G. M. Smith and J. C. Wolford: Applied Numerical Methods for Digital Computation with FORTRAN and CSMP, Dun-Donnelley Publisher, 1976

27. F. BaHli: Network Synthesis by Impulse Response for specified input and output in the Time Domain, M.I.T. Research Laboratory of Electronics, Technical Report No.261, July 1953

PART V : Appendixes

APPENDIX 1

F1 vs. F2 figures 4 pages

1. Nonsonorant, consonantal oral, tense interrupted
   ㄆ ㄊ ㄎ

2. Nonsonorant, consonantal oral, tense continuant
   ㄈ ㄏ ㄒ ㄕ ㄙ

3. Nonsonorant, consonantal oral, lax interrupted
   ㄅ ㄉ ㄍ ㄓ ㄐ ㄗ ㄘ ㄔ ㄑ

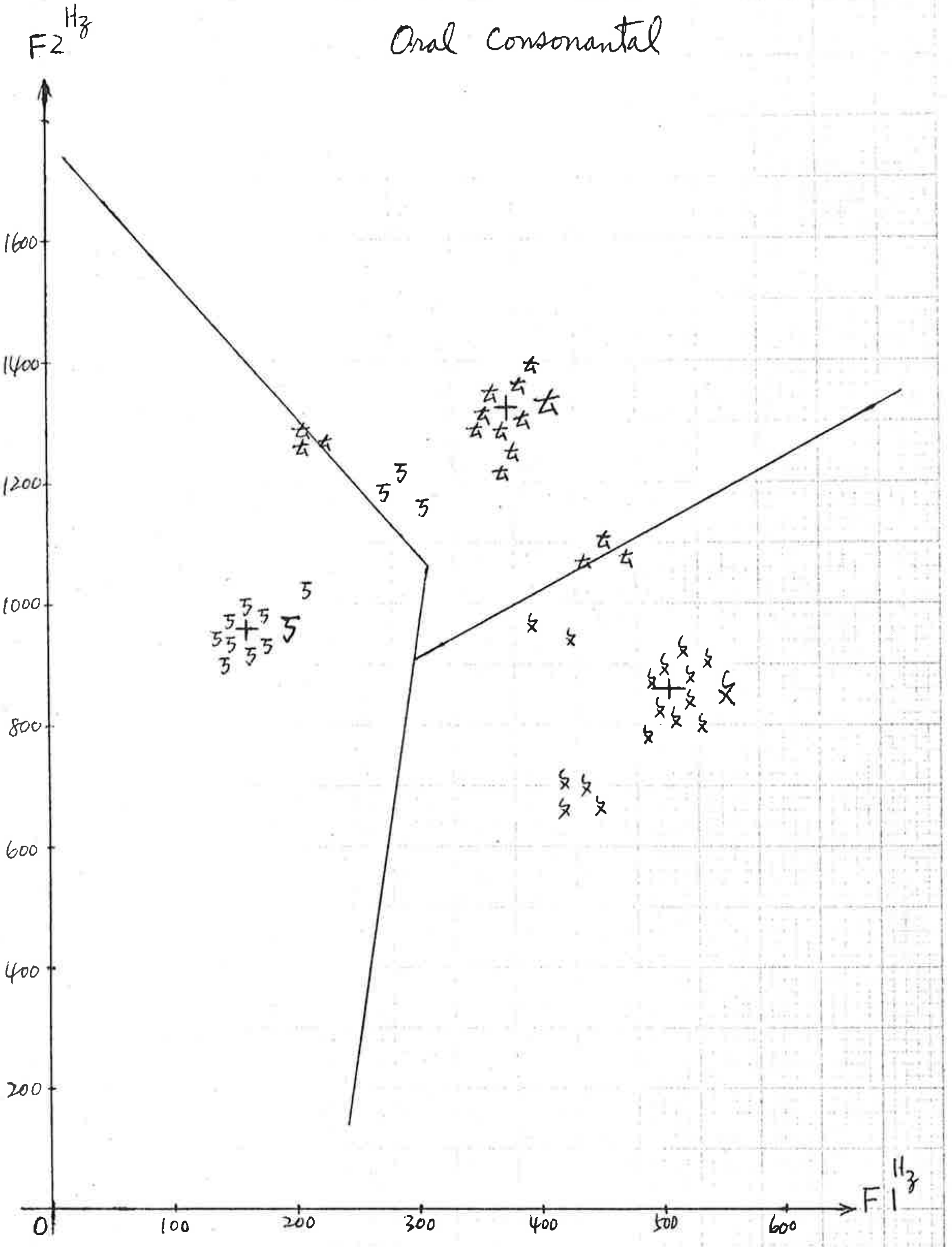
4. Sonorant, consonantal nasal, lax interrupted
   ㄇ ㄋ ㄌ ㄖ

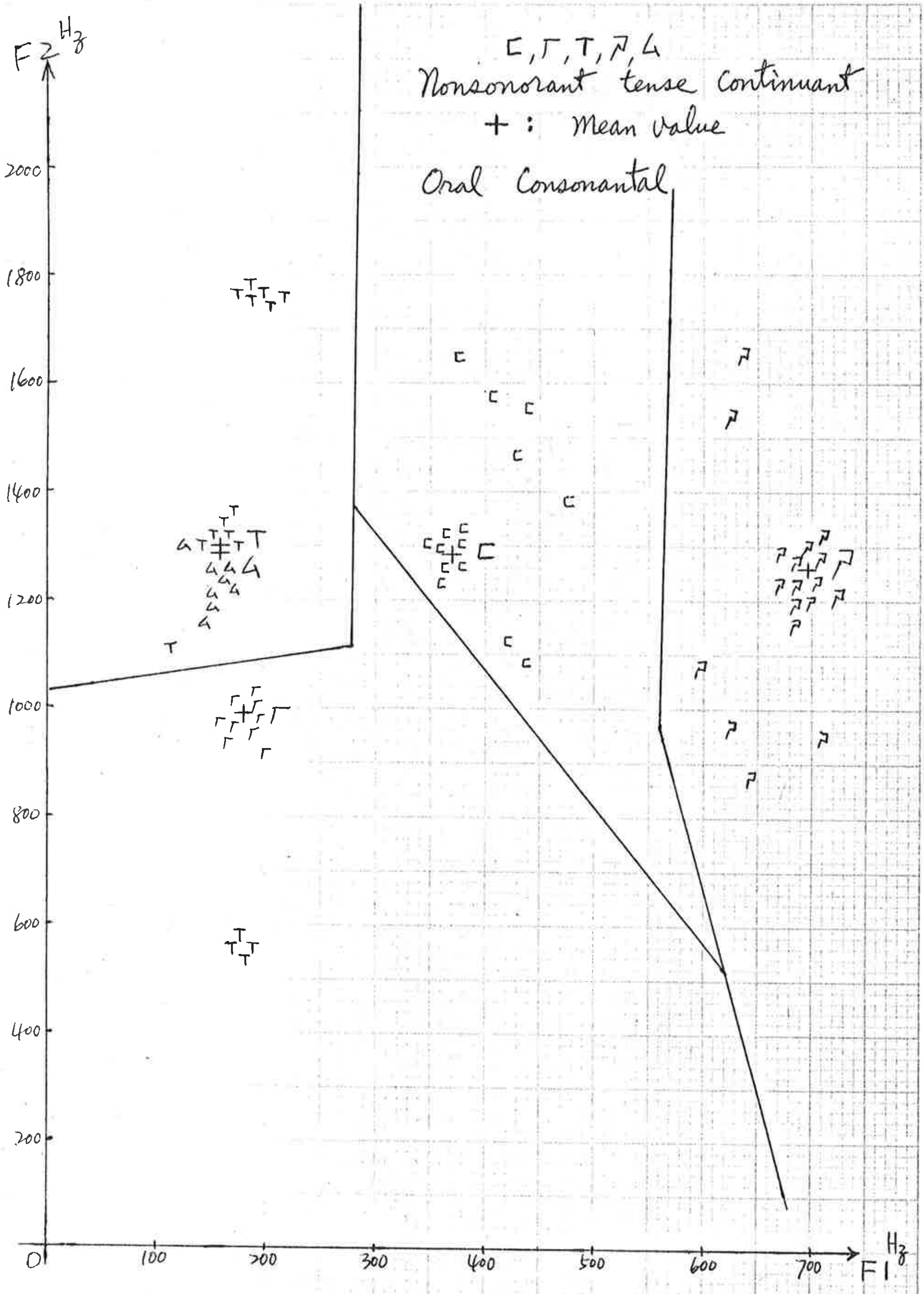
ㄆ, ㄊ, ㄎ

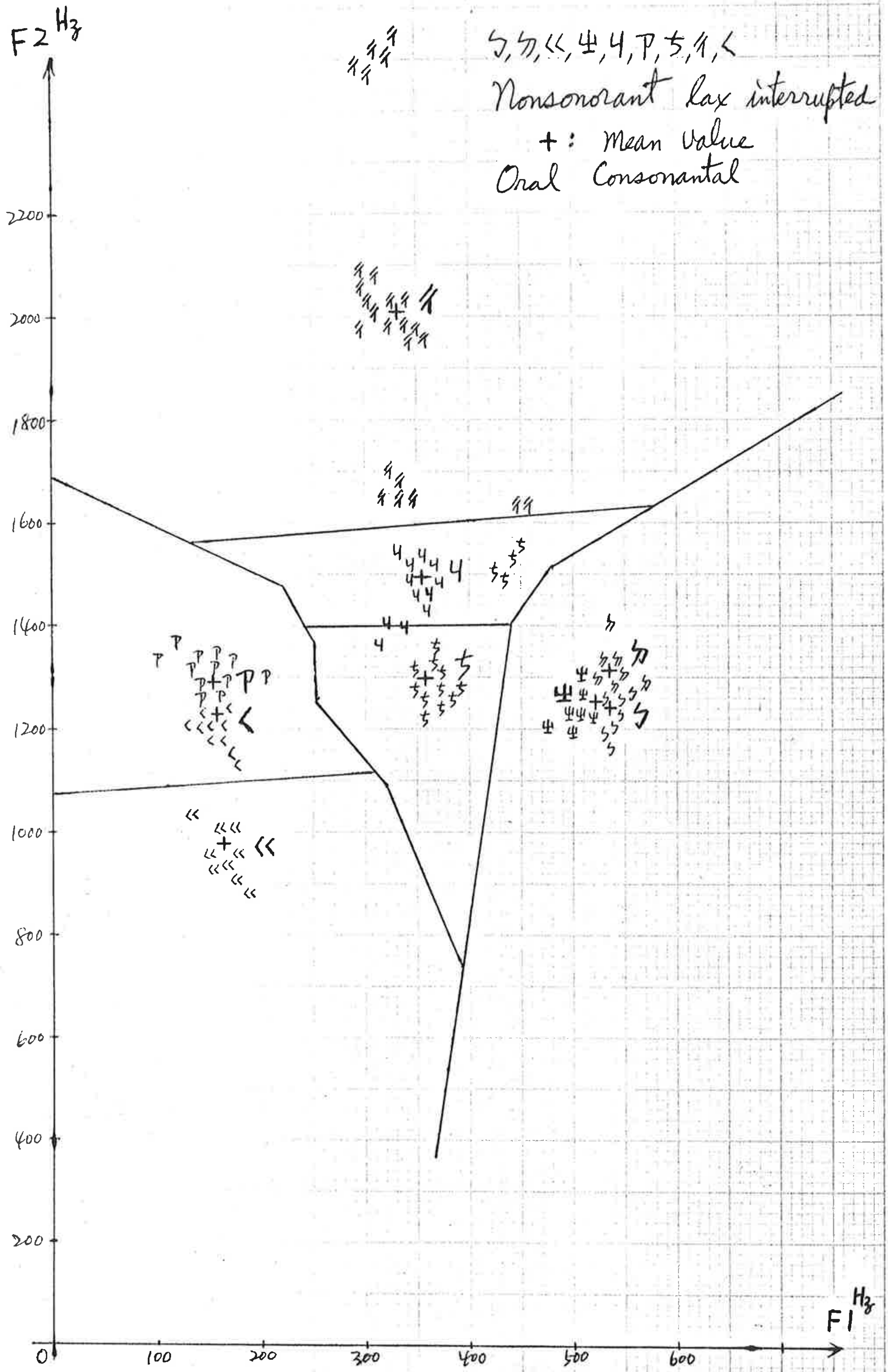
Nonsonorant tense interrupted

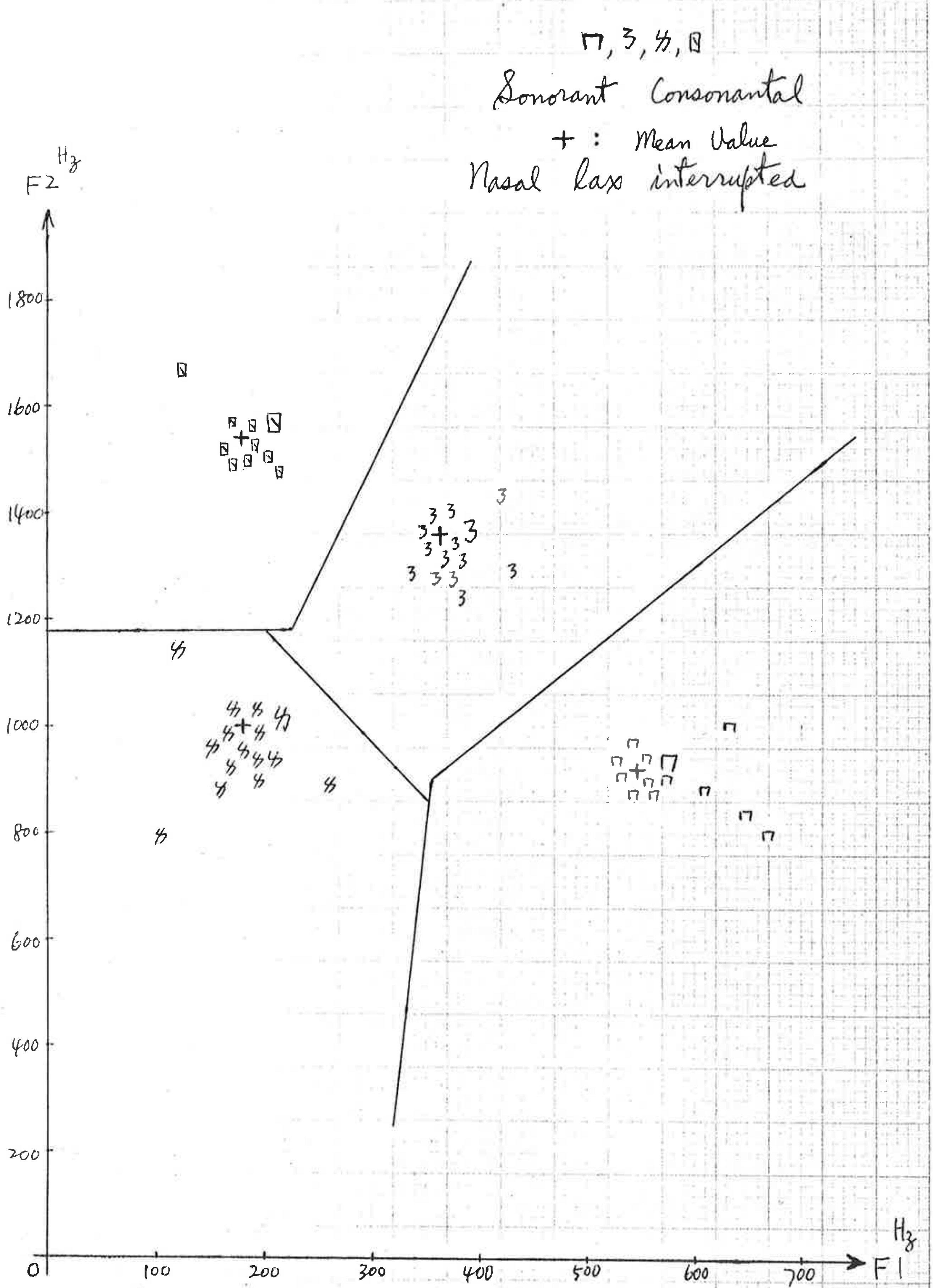
+ : Mean Value

Oral Consonantal




F2 Hz
ㄷ, ㄱ, ㅜ, ㄱ, ㄴ
Nonsonorant tense Continuant
+ : Mean value
Oral Consonantal
2000
1800
1600
1400
1200
1000
800
600
400
200
0
100
200
300
400
500
600
700
F1 Hz


F2 Hz
ㄅ,ㄉ,ㄍ,ㄓ,ㄐ,ㄗ,ㄘ,ㄔ,ㄑ
Nonsonorant lax interrupted
+ : Mean Value
Oral Consonantal
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
100
200
300
400
500
600
F1 Hz



ㅁ, ㄴ, ㅇ, ㅂ
Sonorant Consonantal
+ : Mean Value
Nasal lax interrupted
F2 Hz
1800
1600
1400
1200
1000
800
600
400
200
0
100
200
300
400
500
600
700
F1 Hz

APPENDIX 2

CDC Cyber 173 Computer Program- ABREC

11 pages

```
      PROGRAM ABREC (OUTPUT,INPUT,TAPE1=INPUT,TAPE3=OUTPUT)
      DIMENSION RA(8),AM(2,8),CM(2,8,8),X(40,8),IOD(8),XJ(40,8),IS(2)
      COMMON X,XJ
      IS=1
      DO 100 I=1,40
100   READ 101, (X(I,J),J=1,8)
101   FORMAT(8F10.2)
2222  PRINT 111, ((X(I,J),J=1,8),I=1,40)
111   FORMAT(8F10.2)
      CALL MEANCO(AM,CM,40,2)
155   NNI=0
      NK=0
250   NNI=NNI+1
      IF(NNI.GT.40) GO TO 2200
      B=2.1$ L1=1 $ SUM=0.0
      DO 555 I=1,8
      S=SQRT(CM(2,I,1)/CM(1,I,I))
      SS=(X(NNI,I)-AM(2,I))**2
      SR=(X(NNI,I)-AM(1,I))**2
555   RA(I)=ALOG(S)+SS/(2*CM(2,I,I))-SR/(2*CM(1,I,I))
120   SUM=SUM+RA(L1)
      IF(SUM.LT.B) GO TO 189
      NK=NK+L1
      PRINT 108,NNI,(X(NNI,L),L=1,8)
108   FORMAT(1X,3HTHE,I5,2HTH,26H INPUT VECTOR BELONGS TO A/8F10.2)
      GO TO 250
189   IF(SUM.GT.-B) GO TO 199
      NK=NK+L1
      PRINT 109, NNI,(X(NNI,L),L=1,8)
109   FORMAT(1X,3HTHE,I5,2HTH,26H INPUT VECTOR BELONGS TO B/8F10.2)
      GO TO 250
199   L1=L1+1
      B=B+0.1
      IF(L1.GT.8) GO TO 121
      GO TO 120
121   PRINT 179,NNI,(X(NNI,I),I=1,8)
179   FORMAT(1X,3HTHE,I5,2HTH,35H INPUT VECTOR NOT BELONGS TO A OR B/
     1 8F10.3)
      NK=NK+9
      GO TO 250
2200  IF(IS.EQ.2) GO TO  1000
220   PRINT 201,NK
201   FORMAT(1X,59H THE TOTAL NUMBER OF RECOGNITION IF FEATURES ARE UNOR
     1DERED=,I3)
      AV=NK/40.0
      PRINT 202, AV
202   FORMAT(1X,61H THE AVERAGE NUMBER OF RECOGNITION IF FEATURES AREUN
     1ORDERED=,F8.3)
      CALL ORDER(IOD)
      IS=2
      DO 400 I=1,40
      DO 401 L=1,8
      K=IOD(L)
401   XJ(I,L)=X(I,K)
400   CONTINUE
      DO 501 I=1,40
      DO 501 L=1,8
501   X(I,L)=XJ(I,L)
      GO TO 2222
1000  PRINT 1001,NK
1001  FORMAT(1X,57H THE TOTAL NUMBER OF RECOGNITION IF FEATURES ARE ORDE
     1RED=,I3)
      AV=NK/40.0
      PRINT 1002,AV
1002  FORMAT(1X,59H THE AVERAGE NUMBER OF RECOGNITION IF FEATURES ARE OR
     1DERED=,F8.3)
      STOP
      END
```

```
      SUBROUTINE ORDER(OD)
C THIS PROGRAM FIND THE ORDER OF FEATURES
      INTEGER OD
      DIMENSION DIV(8),OD(8)
      CALL DIVER(DIV)
      DO 21 I=1,8
21    OD(I)=I
      DO 23 I=1,8
      M=I+1
      DO 22 J=M,8
      P=DIV(I)-DIV(J)
      IF(P.GE.0) GO TO 22
      K=OD(I) $ OD(I)=OD(J) $ OD(J)=K
      P=DIV(I) $ DIV(I)=DIV(J) $ DIV(J)=P
22    CONTINUE
23    CONTINUE
      DO 25 I=1,8
25    PRINT 24, I, OD(I)
24    FORMAT(1X,3HTHE,I3,23HTH ORDER OF FEATURES IS,I3)
      RETURN
      END
```

```
      SUBROUTINE DIVER(DIV)
C THIS IS THE PROGRAM COMPUTE THE DIVERGENCE OF FEATURES
      DIMENSION AM(2,8), CM(2,8,8), DIV(8)
      CALL MEANCO(AM,CM,40,2)
      DO 11 I=1,8
      DIV(I)=0.5*(CM(1,I,I)-CM(2,I,I))*(1/CM(2,I,I)-1/CM(1,I,I))+0.5*(1/
     1CM(1,I,I)+1/CM(2,I,I))*(AM(1,I)-AM(2,I))**2
11    CONTINUE
      DO 13 I=1,8
      PRINT 12, I, DIV(I)
12    FORMAT(1X,21HDIVERGENCE OF FEATURE,I2,2H =,F9.5/)
13    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MEANCO(AM,CMAX,N1,N2)
C THIS SUBROUTINE COMPUTE THE MEAN AND COVARIANCE OF THE DISTRIBUTION
      DIMENSION X(40,8),CMAX(2,8,8),AM(2,8)
      COMMON X,XJ
      N=1 $ NF=N1/2 $ N2=1
9     DO 4 J=1,8
      AM(N,J)=0.0
      DO 3 K=N2,NF
3     AM(N,J)=AM(N,J)+X(K,J)
4     AM(N,J)=AM(N,J)/20
      DO 6 I=1,8
      DO 6 J=1,8
      CMAX(N,I,J)=0.0
      DO 5 K= N2,NF
5     CMAX(N,I,J)=CMAX(N,I,J)+(X(K,J)-AM(N,J))*(X(K,I)-AM(N,I))
6      CMAX(N,I,J)=CMAX(N,I,J)/20
      IF(N2.EQ.2) GO TO 101
7     PRINT 8,N,(AM(N,I),I=1,8),((CMAX(N,I,J),I=1,8),J=1,8)
8     FORMAT(1H,10X,7HFOR THE,I2,7HTH WORD//1X,15HTHE MEAN VECTOR/8(2X,F
     110.4)//1X,21HTHE COVARIANCE MATRIX/1X,8(2X,F12.4))
101   N=N+1 $ NF=N1 $ N2=N1/2+1
      IF(N.EQ.2) GO TO 9
10    RETURN
      END
```

-.064	-.042	-.039	-.120	-.143	-.037	-.024	-.045
-.062	-.040	-.038	-.121	-.143	-.039	-.025	-.044
-.065	-.042	-.037	-.120	-.142	-.037	-.024	-.045
-.064	-.042	-.039	-.120	-.142	-.037	-.023	-.044
-.064	-.042	-.039	-.121	-.144	-.037	-.022	-.045
-.063	-.041	-.040	-.121	-.144	-.040	-.023	-.045
-.062	-.040	-.040	-.119	-.140	-.040	-.025	-.046
-.060	-.043	-.039	-.119	-.140	-.038	-.025	-.046
-.061	-.043	-.038	-.118	-.143	-.036	-.024	-.043
-.060	-.040	-.037	-.122	-.143	-.035	-.024	-.043
-.061	-.044	-.038	-.118	-.143	-.036	-.024	-.045
-.063	-.039	-.037	-.122	-.144	-.036	-.024	-.045
-.063	-.043	-.039	-.120	-.145	-.035	-.023	-.042
-.060	-.042	-.039	-.120	-.145	-.035	-.022	-.042
-.065	-.042	-.038	-.120	-.141	-.037	-.022	-.047
-.066	-.041	-.038	-.120	-.141	-.038	-.025	-.047
-.059	-.040	-.037	-.118	-.141	-.038	-.025	-.048
-.061	-.040	-.039	-.118	-.142	-.036	-.020	-.041
-.062	-.040	-.040	-.040	-.119	-.140	-.040	-.025
-.060	-.042	-.038	-.121	-.142	-.037	-.026	-.045
-.058	-.038	-.042	-.078	-.125	-.015	-.011	-.028
-.053	-.038	-.041	-.080	-.126	-.021	-.011	-.028
-.056	-.039	-.042	-.079	-.127	-.014	-.013	-.026
-.058	-.037	-.044	-.076	-.123	-.013	-.015	-.025
-.058	-.037	-.043	-.075	-.120	-.012	-.017	-.027
-.058	-.038	-.042	-.078	-.119	-.016	-.010	-.026
-.058	-.040	-.042	-.078	-.123	-.015	-.008	-.023
-.057	-.035	-.045	-.078	-.121	-.018	-.009	-.031
-.056	-.036	-.039	-.079	-.122	-.011	-.011	-.032
-.057	-.036	-.039	-.077	-.124	-.015	-.011	-.029
-.055	-.041	-.040	-.076	-.124	-.015	-.011	-.028
-.060	-.040	-.041	-.075	-.123	-.013	-.011	-.028
-.061	-.039	-.042	-.078	-.125	-.019	-.015	-.028
-.060	-.038	-.042	-.078	-.126	-.020	-.013	-.028
-.059	-.038	-.038	-.077	-.126	-.015	-.014	-.027
-.056	-.038	-.039	-.078	-.125	-.015	-.015	-.025
6.300	5.021	5.800	5.000	4.025	8.300	17.500	-3.470
-.054	-.038	-.041	-.078	-.125	-.015	-.017	-.024
-.015	-.017	-.160	-.009	-.005	-.173	-.030	-.021
-.055	-.039	-.043	-.080	-.125	-.014	-.016	-.023

FOR THE 1TH WORD

THE MEAN VECTOR

-.0622	-.0414	-.0384	-.1159	-.1414	-.0422	-.0245	-.0436

THE COVARIANCE MATRIX

.00000379	.00000015	.00000014	.00000168	.00000031	-.00000080	-.00000067	.00000104
.00000015	.00000174	.00000012	.00000504	.00000186	-.00000773	-.00000120	.00000119
.00000014	.00000012	.00000095	-.00000636	-.00000176	.00000846	.00000088	-.00000184
.00000168	.00000504	-.00000636	.00030469	.00008998	-.00039073	-.00006175	.00007441
.00000031	.00000186	-.00000176	.00008998	.00002833	-.00011607	-.00001898	.00002042
-.00000080	-.00000773	.00000846	-.00039073	-.00011607	.00050546	.00008055	-.00009453
-.00000067	-.00000120	.00000088	-.00006175	-.00001898	.00008055	.00001455	-.00001393
.00000104	.00000119	-.00000184	.00007441	.00002042	-.00009453	-.00001393	.00002133

FOR THE 2TH WORD

THE MEAN VECTOR

.2628	.2160	.2448	.1797	.0896	.3926	.8621	-.1989

THE COVARIANCE MATRIX

1.91839256	1.52683419	1.76492960	1.53178943	1.25071556	2.51225666	5.28661407	-1.03938667
1.52683419	1.21520745	1.40478949	1.21912213	.99538443	1.99961613	4.20766381	-.82726079
1.76492960	1.40478949	1.62492209	1.40899576	1.14998429	2.31287919	4.86471293	-.95645831
1.53178943	1.21912213	1.40899576	1.22316063	.99882159	2.00562934	4.22102229	-.82988010
1.25071556	.99538443	1.14998429	.99882159	.81581865	1.63697595	3.44609445	-.67751698
2.51225666	1.99961613	2.31287919	2.00562934	1.63697595	3.29211855	6.92451574	-1.36143653
5.28661407	4.20766381	4.86471293	4.22102229	3.44609445	6.92451574	14.56947929	-2.86448437
-1.03938667	-.82726079	-.95645831	-.82988010	-.67751698	-1.36143653	-2.86448437	.56318693

THE 1TH INPUT VECTOR BELONGS TO A

-.064	-.042	-.039	-.120	-.143	-.037	-.024	-.045

THE 2TH INPUT VECTOR BELONGS TO A

-.062	-.040	-.038	-.121	-.143	-.039	-.025	-.044

THE 3TH INPUT VECTOR BELONGS TO A

-.065	-.042	-.037	-.120	-.142	-.037	-.024	-.045

THE 4TH INPUT VECTOR BELONGS TO A

-.064	-.042	-.039	-.120	-.142	-.037	-.023	-.044

THE 5TH INPUT VECTOR BELONGS TO A

-.064	-.042	-.039	-.121	-.144	-.037	-.022	-.045

THE 6TH INPUT VECTOR BELONGS TO A

-.063	-.041	-.040	-.121	-.144	-.040	-.023	-.045

THE 7TH INPUT VECTOR BELONGS TO A

-.062	-.040	-.040	-.119	-.140	-.040	-.025	-.046

THE 8TH INPUT VECTOR BELONGS TO A

-.060	-.043	-.039	-.119	-.140	-.038	-.025	-.046

```
THE    9TH INPUT VECTOR BELONGS TO A
   -.061     -.043     -.038     -.118     -.143     -.036     -.024     -.043
THE   10TH INPUT VECTOR BELONGS TO A
   -.060     -.040     -.037     -.122     -.143     -.035     -.024     -.043
THE   11TH INPUT VECTOR BELONGS TO A
   -.061     -.044     -.038     -.118     -.143     -.036     -.024     -.045
THE   12TH INPUT VECTOR BELONGS TO A
   -.063     -.039     -.037     -.122     -.144     -.036     -.024     -.045
THE   13TH INPUT VECTOR BELONGS TO A
   -.063     -.043     -.039     -.120     -.145     -.035     -.023     -.042
THE   14TH INPUT VECTOR BELONGS TO A
   -.060     -.042     -.039     -.120     -.145     -.035     -.022     -.042
THE   15TH INPUT VECTOR BELONGS TO A
   -.065     -.042     -.038     -.120     -.141     -.037     -.022     -.047
THE   16TH INPUT VECTOR BELONGS TO A
   -.066     -.041     -.038     -.120     -.141     -.038     -.025     -.047
THE   17TH INPUT VECTOR BELONGS TO A
   -.059     -.040     -.037     -.118     -.141     -.038     -.025     -.048
THE   18TH INPUT VECTOR BELONGS TO A
   -.061     -.040     -.039     -.118     -.142     -.036     -.020     -.041
THE   19TH INPUT VECTOR BELONGS TO A
   -.062     -.040     -.040     -.040     -.119     -.140     -.040     -.025
THE   20TH INPUT VECTOR BELONGS TO A
   -.060     -.042     -.038     -.121     -.142     -.037     -.026     -.045
THE   21TH INPUT VECTOR BELONGS TO A
   -.058     -.038     -.042     -.078     -.125     -.015     -.011     -.028
THE   22TH INPUT VECTOR BELONGS TO B
   -.053     -.038     -.041     -.080     -.126     -.021     -.011     -.028
THE   23TH INPUT VECTOR BELONGS TO A
   -.056     -.039     -.042     -.079     -.127     -.014     -.013     -.026
THE   24TH INPUT VECTOR BELONGS TO A
   -.058     -.037     -.044     -.076     -.123     -.013     -.015     -.025
THE   25TH INPUT VECTOR BELONGS TO A
   -.058     -.037     -.043     -.075     -.120     -.012     -.017     -.027
THE   26TH INPUT VECTOR BELONGS TO A
   -.058     -.038     -.042     -.078     -.119     -.016     -.010     -.026
THE   27TH INPUT VECTOR BELONGS TO A
   -.058     -.040     -.042     -.078     -.123     -.015     -.008     -.023
THE   28TH INPUT VECTOR BELONGS TO A
   -.057     -.035     -.045     -.078     -.121     -.018     -.009     -.031
THE   29TH INPUT VECTOR BELONGS TO A
   -.056     -.036     -.039     -.079     -.122     -.011     -.011     -.032
THE   30TH INPUT VECTOR BELONGS TO A
   -.057     -.036     -.039     -.077     -.124     -.015     -.011     -.029
```

THE 31TH INPUT VECTOR BELONGS TO A

-.055 -.041 -.040 -.076 -.124 -.015 -.011 -.028

THE 32TH INPUT VECTOR BELONGS TO A

-.060 -.040 -.041 -.075 -.123 -.013 -.011 -.028

THE 33TH INPUT VECTOR BELONGS TO A

-.061 -.039 -.042 -.078 -.125 -.019 -.015 -.028

THE 34TH INPUT VECTOR BELONGS TO A

-.060 -.038 -.042 -.078 -.126 -.020 -.013 -.028

THE 35TH INPUT VECTOR BELONGS TO A

-.059 -.038 -.038 -.077 -.126 -.015 -.014 -.027

THE 36TH INPUT VECTOR BELONGS TO A

-.056 -.038 -.039 -.078 -.125 -.015 -.015 -.025

THE 37TH INPUT VECTOR BELONGS TO B

6.300 5.021 5.800 5.000 4.025 8.300 17.500 -3.470

THE 38TH INPUT VECTOR BELONGS TO B

-.054 -.038 -.041 -.078 -.125 -.015 -.017 -.024

THE 39TH INPUT VECTOR BELONGS TO B

-.015 -.017 -.160 -.009 -.005 -.173 -.030 -.021

THE 40TH INPUT VECTOR BELONGS TO A

-.055 -.039 -.043 -.080 -.125 -.014 -.016 -.023

THE TOTAL NUMBER OF RECOGNITION IF FEATURES ARE UNORDERED= 46

THE AVERAGE NUMBER OF RECOGNITION IF FEATURES AREUN ORDERED= 1.150

FOR THE 1TH WORD

THE MEAN VECTOR

-.0622 -.0414 -.0384 -.1159 -.1414 -.0422 -.0245 -.0436

THE COVARIANCE MATRIX

.00000379	.00000015	.00000014	.00000168	.00000031	-.00000080	-.00000067	.00000104
.00000015	.00000174	.00000012	.00000504	.00000186	-.00000773	-.00000120	.00000119
.00000014	.00000012	.00000095	-.00000636	-.00000176	.00000846	.00000088	-.00000184
.00000168	.00000504	-.00000636	.00030469	.00008998	-.00039073	-.00006175	.00007441
.00000031	.00000186	-.00000176	.00008998	.00002833	-.00011607	-.00001898	.00002042
-.00000080	-.00000773	.00000846	-.00039073	-.00011607	.00050546	.00008055	-.00009453
-.00000067	-.00000120	.00000088	-.00006175	-.00001898	.00008055	.00001455	-.00001393
.00000104	.00000119	-.00000184	.00007441	.00002042	-.00009453	-.00001393	.00002133

FOR THE 2TH WORD

THE MEAN VECTOR

.2628 .2160 .2448 .1797 .0896 .3926 .8621 -.1989

THE COVARIANCE MATRIX

1.91839256	1.52683419	1.76492960	1.53178943	1.25071556	2.51225666	5.28661407	-1.03938667
1.52683419	1.21520745	1.40478949	1.21912213	.99538443	1.99961613	4.20766381	-.82726079

1.76492960	1.40478949	1.62492209	1.40899576	1.14998429	2.31287919	4.86471293	-.95645831
1.53178943	1.21912213	1.40899576	1.22316063	.99882159	2.00562934	4.22102229	-.82988010
1.25071556	.99538443	1.14998429	.99882159	.81581865	1.63697595	3.44609445	-.67751698
2.51225666	1.99961613	2.31287919	2.00562934	1.63697595	3.29211855	6.92451574	-1.36143653
5.28661407	4.20766381	4.86471293	4.22102229	3.44609445	6.92451574	14.56947929	-2.86448437
-1.03938667	-.82726079	-.95645831	-.82988010	-.67751698	-1.36143653	-2.86448437	.56318693

DIVERGENCE OF FEATURE 1 = 267200.35592

DIVERGENCE OF FEATURE 2 = 368227.89794

DIVERGENCE OF FEATURE 3 = 899800.78062

DIVERGENCE OF FEATURE 4 = 2149.59940

DIVERGENCE OF FEATURE 5 = 15339.84034

DIVERGENCE OF FEATURE 6 = 3442.55162

DIVERGENCE OF FEATURE 7 = 527680.77440

DIVERGENCE OF FEATURE 8 = 13767.01970

THE 1TH ORDER OF FEATURES IS 3
THE 2TH ORDER OF FEATURES IS 7
THE 3TH ORDER OF FEATURES IS 2
THE 4TH ORDER OF FEATURES IS 1
THE 5TH ORDER OF FEATURES IS 5
THE 6TH ORDER OF FEATURES IS 8
THE 7TH ORDER OF FEATURES IS 6
THE 8TH ORDER OF FEATURES IS 4

-.039	-.024	-.042	-.064	-.143	-.045	-.037	-.120
-.038	-.025	-.040	-.062	-.143	-.044	-.039	-.121
-.037	-.024	-.042	-.065	-.142	-.045	-.037	-.120
-.039	-.023	-.042	-.064	-.142	-.044	-.037	-.120
-.039	-.022	-.042	-.064	-.144	-.045	-.037	-.121
-.040	-.023	-.041	-.063	-.144	-.045	-.040	-.121
-.040	-.025	-.040	-.062	-.140	-.046	-.040	-.119
-.039	-.025	-.043	-.060	-.140	-.046	-.038	-.119
-.038	-.024	-.043	-.061	-.143	-.043	-.036	-.118
-.037	-.024	-.040	-.060	-.143	-.043	-.035	-.122
-.038	-.024	-.044	-.061	-.143	-.045	-.036	-.118
-.037	-.024	-.039	-.063	-.144	-.045	-.036	-.122
-.039	-.023	-.043	-.063	-.145	-.042	-.035	-.120
-.039	-.022	-.042	-.060	-.145	-.042	-.035	-.120

-.038	-.022	-.042	-.065	-.141	-.047	-.037	-.120
-.038	-.025	-.041	-.066	-.141	-.047	-.038	-.120
-.037	-.025	-.040	-.059	-.141	-.048	-.038	-.118
-.039	-.020	-.040	-.061	-.142	-.041	-.036	-.118
-.040	-.040	-.040	-.062	-.119	-.025	-.140	-.040
-.038	-.026	-.042	-.060	-.142	-.045	-.037	-.121
-.042	-.011	-.038	-.058	-.125	-.028	-.015	-.078
-.041	-.011	-.038	-.053	-.126	-.028	-.021	-.080
-.042	-.013	-.039	-.056	-.127	-.026	-.014	-.079
-.044	-.015	-.037	-.058	-.123	-.025	-.013	-.076
-.043	-.017	-.037	-.058	-.120	-.027	-.012	-.075
-.042	-.010	-.038	-.058	-.119	-.026	-.016	-.078
-.042	-.006	-.040	-.058	-.123	-.023	-.015	-.078
-.045	-.009	-.035	-.057	-.121	-.031	-.018	-.078
-.039	-.011	-.036	-.056	-.122	-.032	-.011	-.079
-.039	-.011	-.036	-.057	-.124	-.029	-.015	-.077
-.040	-.011	-.041	-.055	-.124	-.028	-.015	-.076
-.041	-.011	-.040	-.060	-.123	-.028	-.013	-.075
-.042	-.015	-.039	-.061	-.125	-.028	-.019	-.078
-.042	-.013	-.038	-.060	-.126	-.028	-.020	-.078
-.038	-.014	-.038	-.059	-.126	-.027	-.015	-.077
-.039	-.015	-.038	-.056	-.125	-.025	-.015	-.078
5.800	17.500	5.021	6.300	4.025	-3.470	8.300	5.000
-.041	-.017	-.038	-.054	-.125	-.024	-.015	-.078
-.160	-.030	-.017	-.015	-.005	-.021	-.173	-.009
-.043	-.016	-.039	-.055	-.125	-.023	-.014	-.080

FOR THE 1TH WORD

THE MEAN VECTOR

-.0384	-.0245	-.0414	-.0622	-.1414	-.0436	-.0422	-.1159

THE COVARIANCE MATRIX

.00000095	.00000088	.00000012	.00000014	-.00000176	-.00000184	.00000846	-.00000636
.00000088	.00001455	-.00000120	-.00000067	-.00001898	-.00001393	.00008055	-.00006175
.00000012	-.00000120	.00000174	.00000015	.00000186	.00000119	-.00000773	.00000504
.00000014	-.00000067	.00000015	.00000379	.00000031	.00000104	-.00000080	.00000168
-.00000176	-.00001898	.00000186	.00000031	.00002833	.00002042	-.00011607	.00008998
-.00000184	-.00001393	.00000119	.00000104	.00002042	.00002133	-.00009453	.00007441
.00000846	.00008055	-.00000773	-.00000080	-.00011607	-.00009453	.00050546	-.00039073
-.00000636	-.00006175	.00000504	.00000168	.00008998	.00007441	-.00039073	.00030469

FOR THE 2TH WORD

THE MEAN VECTOR

.2448	.8621	.2160	.2628	.0896	-.1989	.3926	.1797

THE COVARIANCE MATRIX

1.62492209	4.86471293	1.40478949	1.76492960	1.14998429	-.95645831	2.31287919	1.40899576
4.86471293	14.56947929	4.20766381	5.28661407	3.44609445	-2.86448437	6.92451574	4.22102229
1.40478949	4.20766381	1.21520745	1.52683419	.99538443	-.82726079	1.99961613	1.21912213
1.76492960	5.28661407	1.52683419	1.91839256	1.25071556	-1.03938667	2.51225666	1.53178943
1.14998429	3.44609445	.99538443	1.25071556	.81581865	-.67751698	1.63697595	.99882159
-.95645831	-2.86448437	-.82726079	-1.03938667	-.67751698	.56318693	-1.36143653	-.82988010
2.31287919	6.92451574	1.99961613	2.51225666	1.63697595	-1.36143653	3.29211855	2.00562934
1.40899576	4.22102229	1.21912213	1.53178943	.99882159	-.82988010	2.00562934	1.22316063

THE 1TH INPUT VECTOR BELONGS TO A

-.039 -.024 -.042 -.064 -.143 -.045 -.037 -.120

THE 2TH INPUT VECTOR BELONGS TO A

-.038 -.025 -.040 -.062 -.143 -.044 -.039 -.121

THE 3TH INPUT VECTOR BELONGS TO A

-.037 -.024 -.042 -.065 -.142 -.045 -.037 -.120

THE 4TH INPUT VECTOR BELONGS TO A

-.039 -.023 -.042 -.064 -.142 -.044 -.037 -.120

THE 5TH INPUT VECTOR BELONGS TO A

-.039 -.022 -.042 -.064 -.144 -.045 -.037 -.121

THE 6TH INPUT VECTOR BELONGS TO A

-.040 -.023 -.041 -.063 -.144 -.045 -.040 -.121

THE 7TH INPUT VECTOR BELONGS TO A

-.040 -.025 -.040 -.062 -.140 -.046 -.040 -.119

THE 8TH INPUT VECTOR BELONGS TO A

-.039 -.025 -.043 -.060 -.140 -.046 -.038 -.119

THE 9TH INPUT VECTOR BELONGS TO A

-.038 -.024 -.043 -.061 -.143 -.043 -.036 -.118

THE 10TH INPUT VECTOR BELONGS TO A

-.037 -.024 -.040 -.060 -.143 -.043 -.035 -.122

THE 11TH INPUT VECTOR BELONGS TO A

-.038 -.024 -.044 -.061 -.143 -.045 -.036 -.118

THE 12TH INPUT VECTOR BELONGS TO A

-.037 -.024 -.039 -.063 -.144 -.045 -.036 -.122

THE 13TH INPUT VECTOR BELONGS TO A

-.039 -.023 -.043 -.063 -.145 -.042 -.035 -.120

THE 14TH INPUT VECTOR BELONGS TO A

-.039 -.022 -.042 -.060 -.145 -.042 -.035 -.120

THE 15TH INPUT VECTOR BELONGS TO A

-.038 -.022 -.042 -.065 -.141 -.047 -.037 -.120

THE 16TH INPUT VECTOR BELONGS TO A

-.038 -.025 -.041 -.066 -.141 -.047 -.038 -.120

THE 17TH INPUT VECTOR BELONGS TO A

-.037 -.025 -.040 -.059 -.141 -.048 -.038 -.118

THE 18TH INPUT VECTOR BELONGS TO A

-.039 -.020 -.040 -.061 -.142 -.041 -.036 -.118

```
THE    19TH INPUT VECTOR BELONGS TO A
   -.040      -.040      -.040      -.062      -.119      -.025      -.140      -.040
THE    20TH INPUT VECTOR BELONGS TO A
   -.038      -.026      -.042      -.060      -.142      -.045      -.037      -.121
THE    21TH INPUT VECTOR BELONGS TO A
   -.042      -.011      -.038      -.058      -.125      -.028      -.015      -.078
THE    22TH INPUT VECTOR BELONGS TO A
   -.041      -.011      -.038      -.053      -.126      -.028      -.021      -.080
THE    23TH INPUT VECTOR BELONGS TO A
   -.042      -.013      -.039      -.056      -.127      -.026      -.014      -.079
THE    24TH INPUT VECTOR BELONGS TO B
   -.044      -.015      -.037      -.058      -.123      -.025      -.013      -.076
THE    25TH INPUT VECTOR BELONGS TO B
   -.043      -.017      -.037      -.058      -.120      -.027      -.012      -.075
THE    26TH INPUT VECTOR BELONGS TO A
   -.042      -.010      -.038      -.058      -.119      -.026      -.016      -.078
THE    27TH INPUT VECTOR BELONGS TO A
   -.042      -.008      -.040      -.058      -.123      -.023      -.015      -.078
THE    28TH INPUT VECTOR BELONGS TO B
   -.045      -.009      -.035      -.057      -.121      -.031      -.018      -.078
THE    29TH INPUT VECTOR BELONGS TO A
   -.039      -.011      -.036      -.056      -.122      -.032      -.011      -.079
THE    30TH INPUT VECTOR BELONGS TO A
   -.039      -.011      -.036      -.057      -.124      -.029      -.015      -.077
THE    31TH INPUT VECTOR BELONGS TO A
   -.040      -.011      -.041      -.055      -.124      -.028      -.015      -.076
THE    32TH INPUT VECTOR BELONGS TO A
   -.041      -.011      -.040      -.060      -.123      -.028      -.013      -.075
THE    33TH INPUT VECTOR BELONGS TO A
   -.042      -.015      -.039      -.061      -.125      -.028      -.019      -.078
THE    34TH INPUT VECTOR BELONGS TO A
   -.042      -.013      -.038      -.060      -.126      -.028      -.020      -.078
THE    35TH INPUT VECTOR BELONGS TO A
   -.038      -.014      -.038      -.059      -.126      -.027      -.015      -.077
THE    36TH INPUT VECTOR BELONGS TO A
   -.039      -.015      -.038      -.056      -.125      -.025      -.015      -.078
THE    37TH INPUT VECTOR BELONGS TO B
   5.800     17.500      5.021      6.300      4.025     -3.470      8.300      5.000
THE    38TH INPUT VECTOR BELONGS TO A
   -.041      -.017      -.038      -.054      -.125      -.024      -.015      -.078
THE    39TH INPUT VECTOR BELONGS TO B
   -.160      -.030      -.017      -.015      -.005      -.021      -.173      -.009
THE    40TH INPUT VECTOR BELONGS TO B
   -.043      -.016      -.039      -.055      -.125      -.023      -.014      -.080
  THE TOTAL NUMBER OF RECOGNITION IF FEATURES ARE ORDERED= 49
  THE AVERAGE NUMBER OF RECOGNITION IF FEATURES ARE ORDERED=   1.225
```

APPENDIX 3

Spectral distribution of energy- frequency vs. amplitude

18 pages

Sampling frequency	10 KHz
Overflow allowance (i.e. numerical overflow to be permitted)	10
Overlap factor (i.e. proportional overlap number of plots in one segment)	0
Differencing (i.e. to remove nett 20 db/decade droop causing by vocal organ)	1
Fourier transform to be normalized (i.e. Fourier transforms were not normalized)	0

Standard hamming window function:

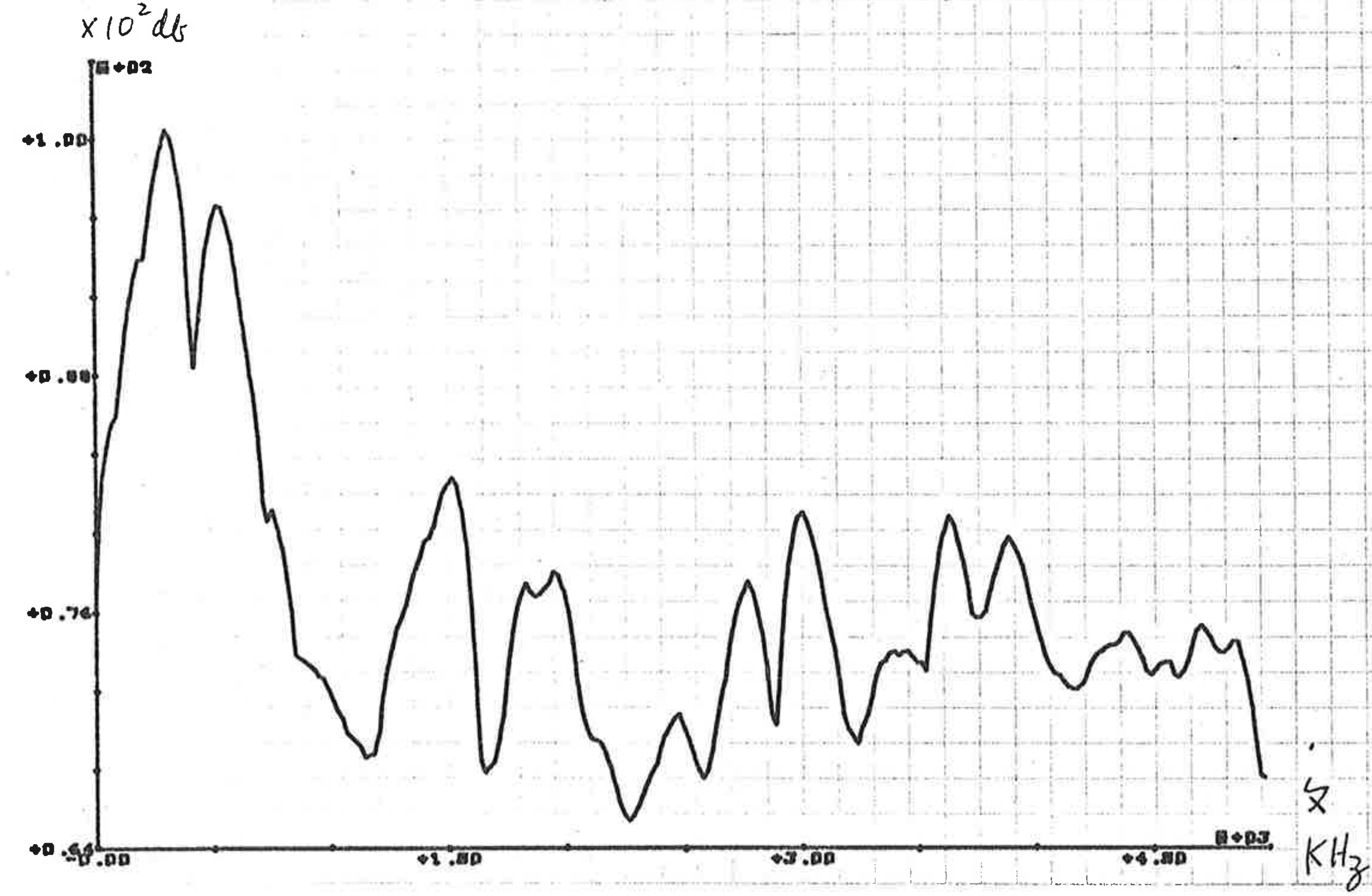
$$0.54-0.46\mathrm{Cos}(2\pi(J-1)/(M-1))$$

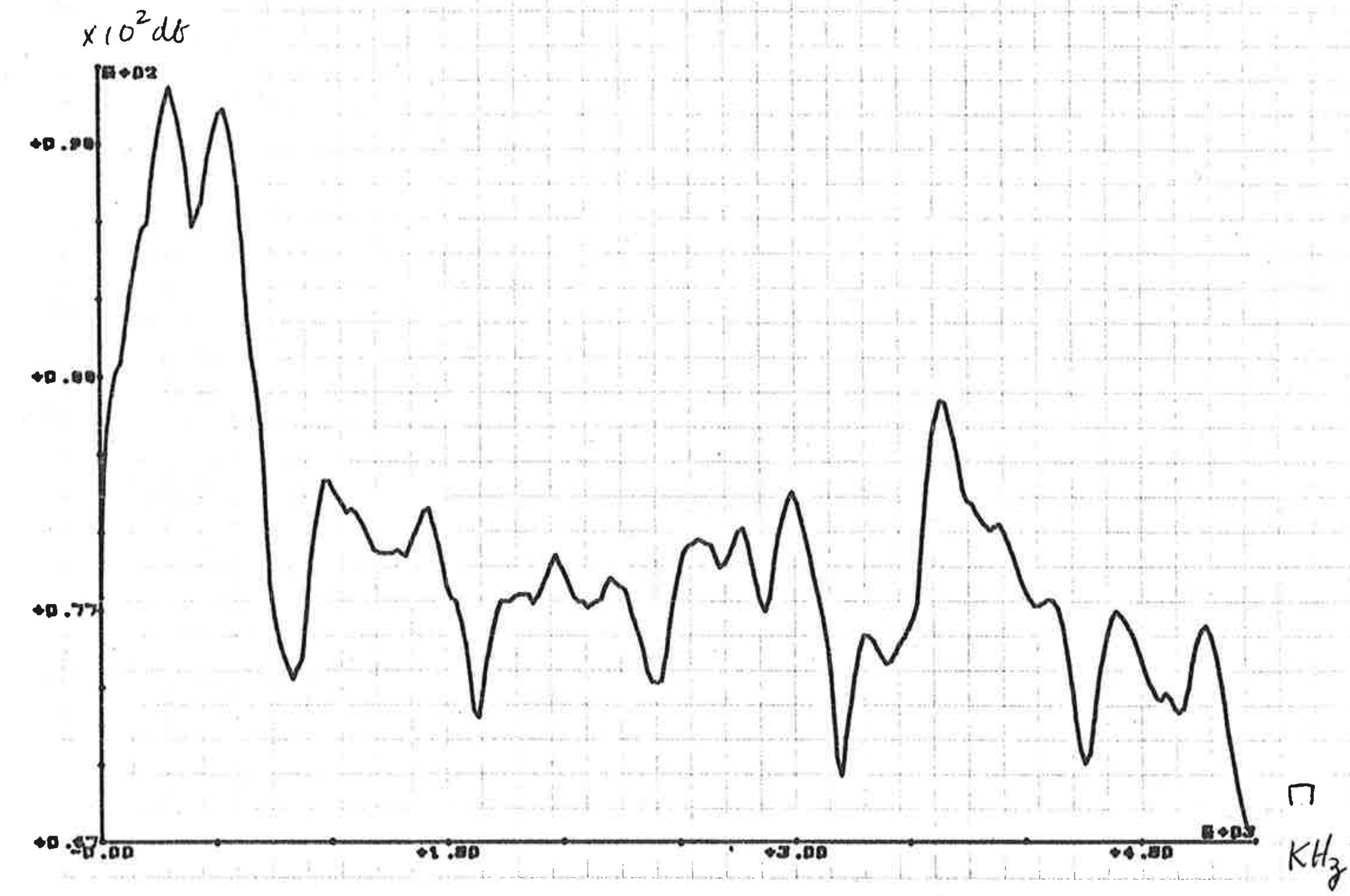
where J: sample index

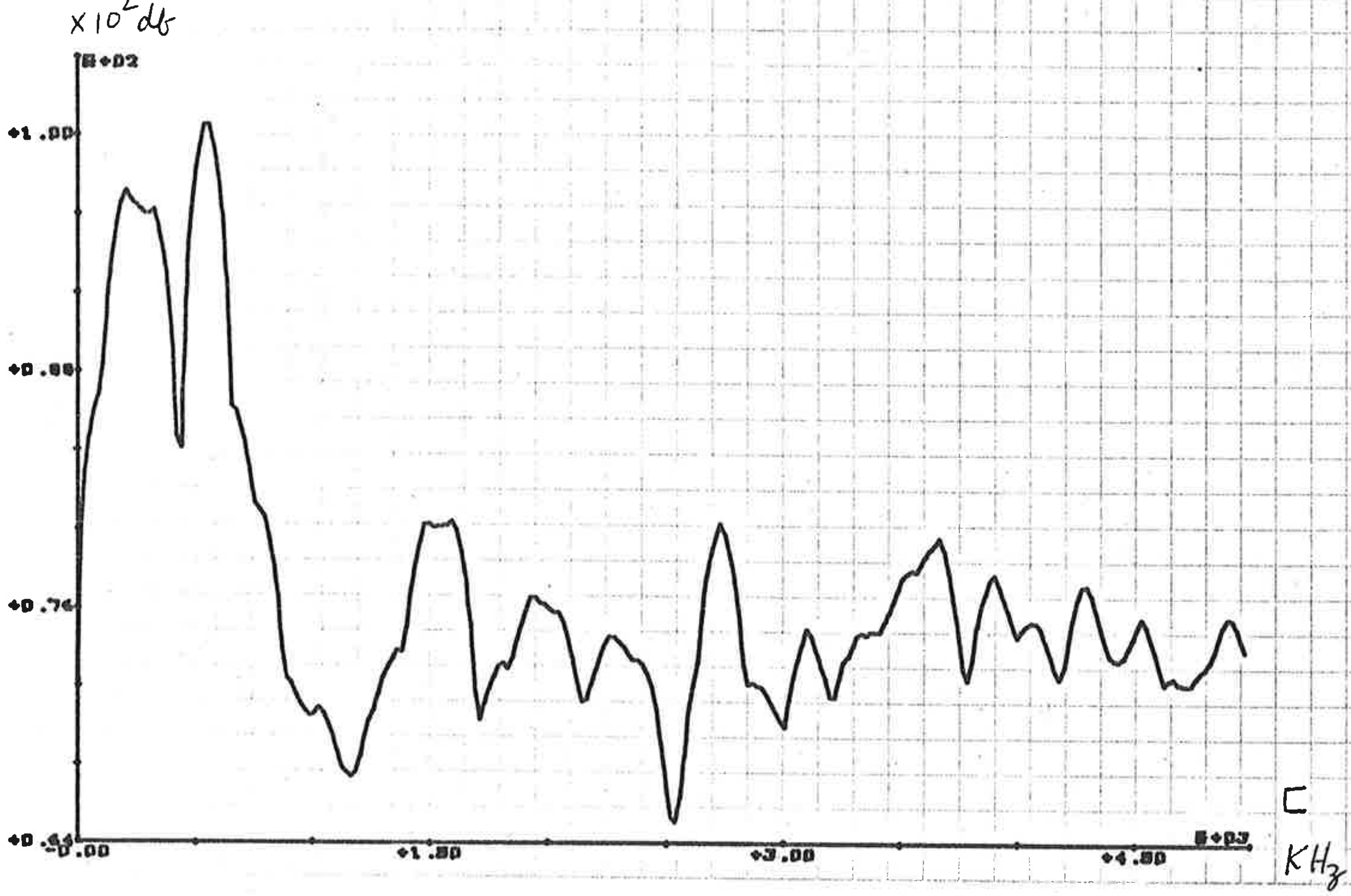
M: window length(512 points)

Successive windows overlap 50% in analysis

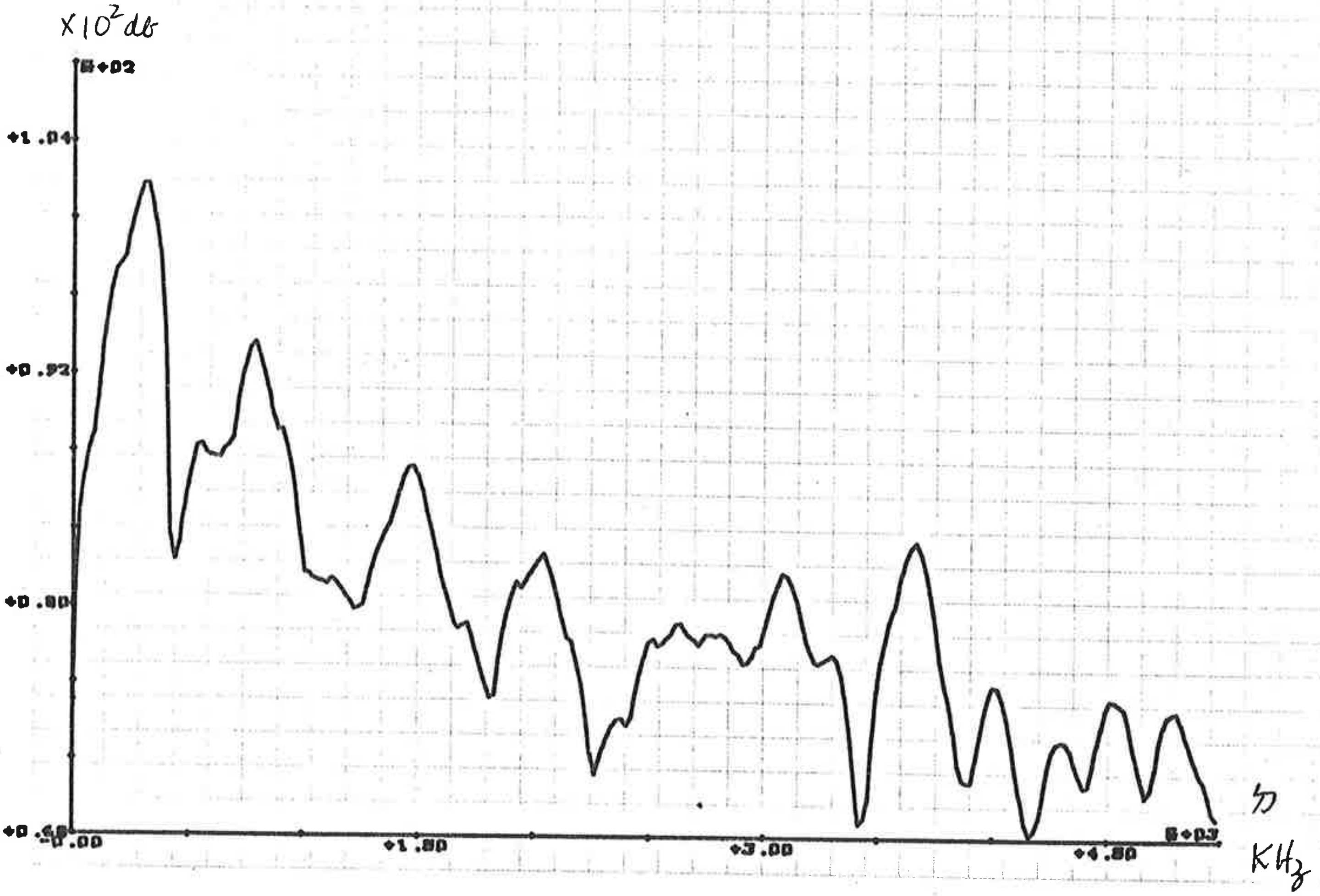
If the display is more than one spectrum, it is necessary to set up axes. Number of sections needed for horizontal axis is the number sum of spectra and overlap factor. It is obvious in the spectrograms.


x10²db
E+02
+1.00
+0.88
+0.76
+0.64
-0.00
+1.50
+3.00
+4.50
E+03
ㄆ
KHz

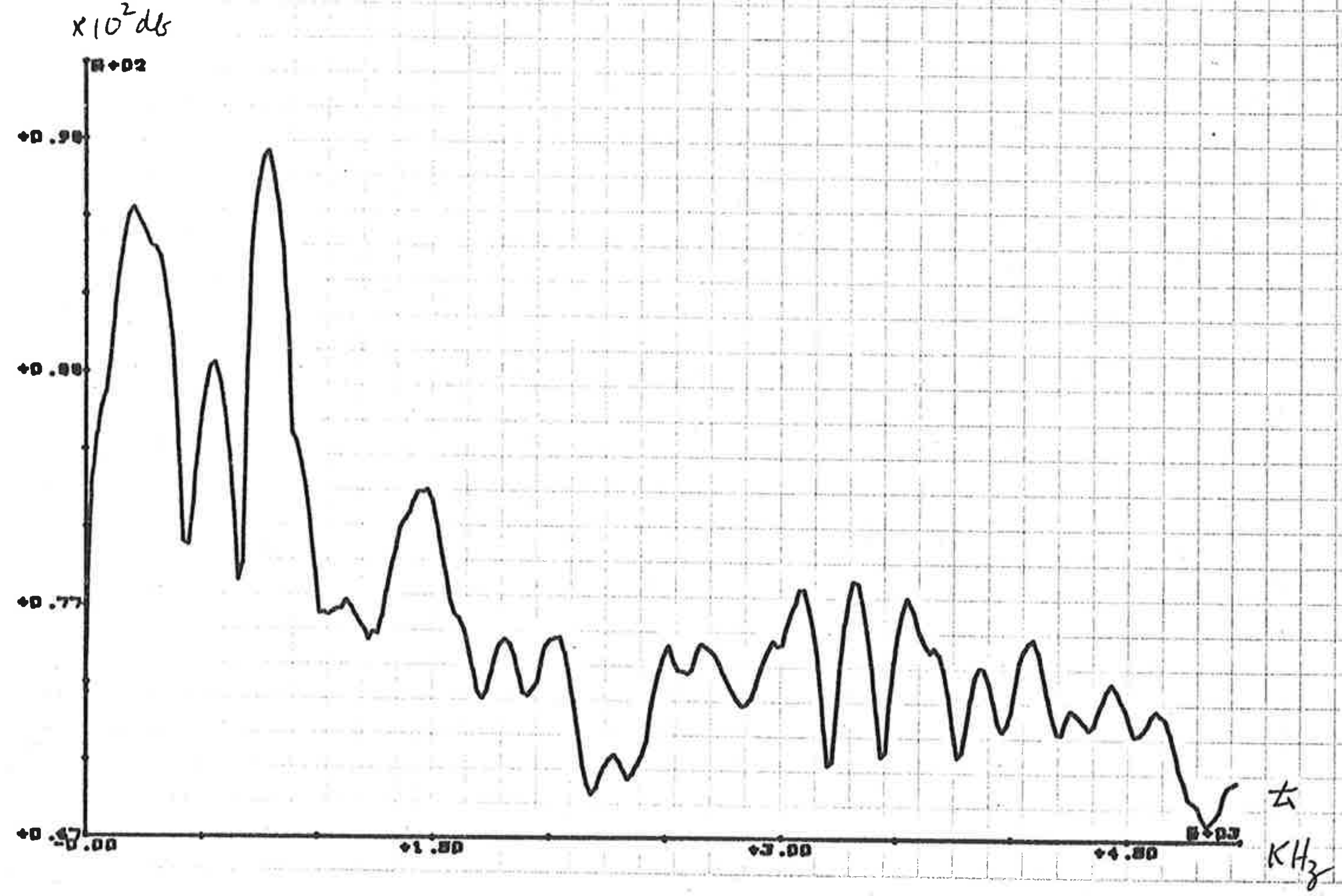

x10²db
E+02
+0.98
+0.88
+0.77
+0.67
-0.00
+1.50
+3.00
+4.50
E+03
ㄇ
KHz



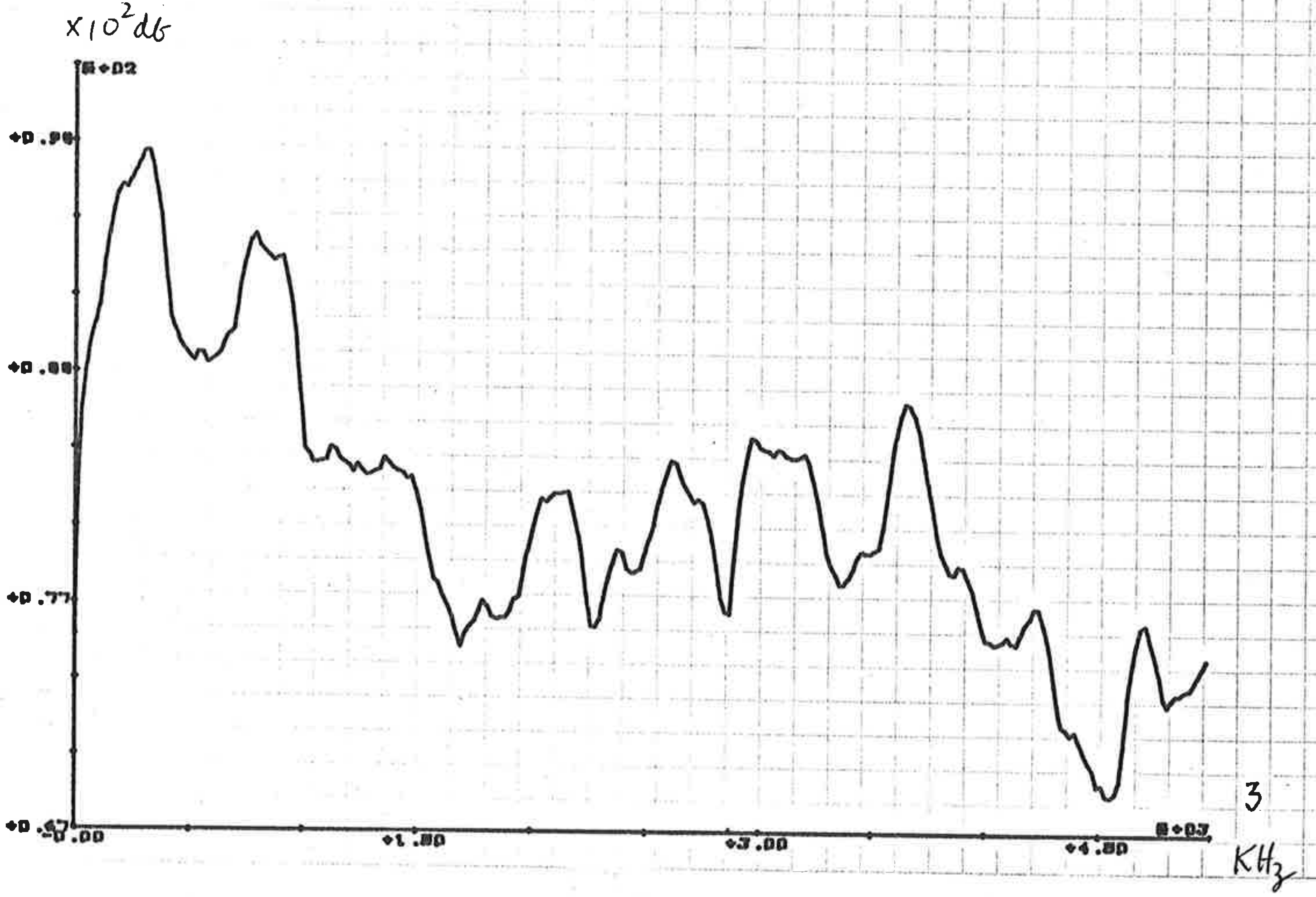
x10² db
E+02
+1.00
+0.88
+0.76
+0.64
-0.00
+1.50
+3.00
+4.50
E+03
KHz
C



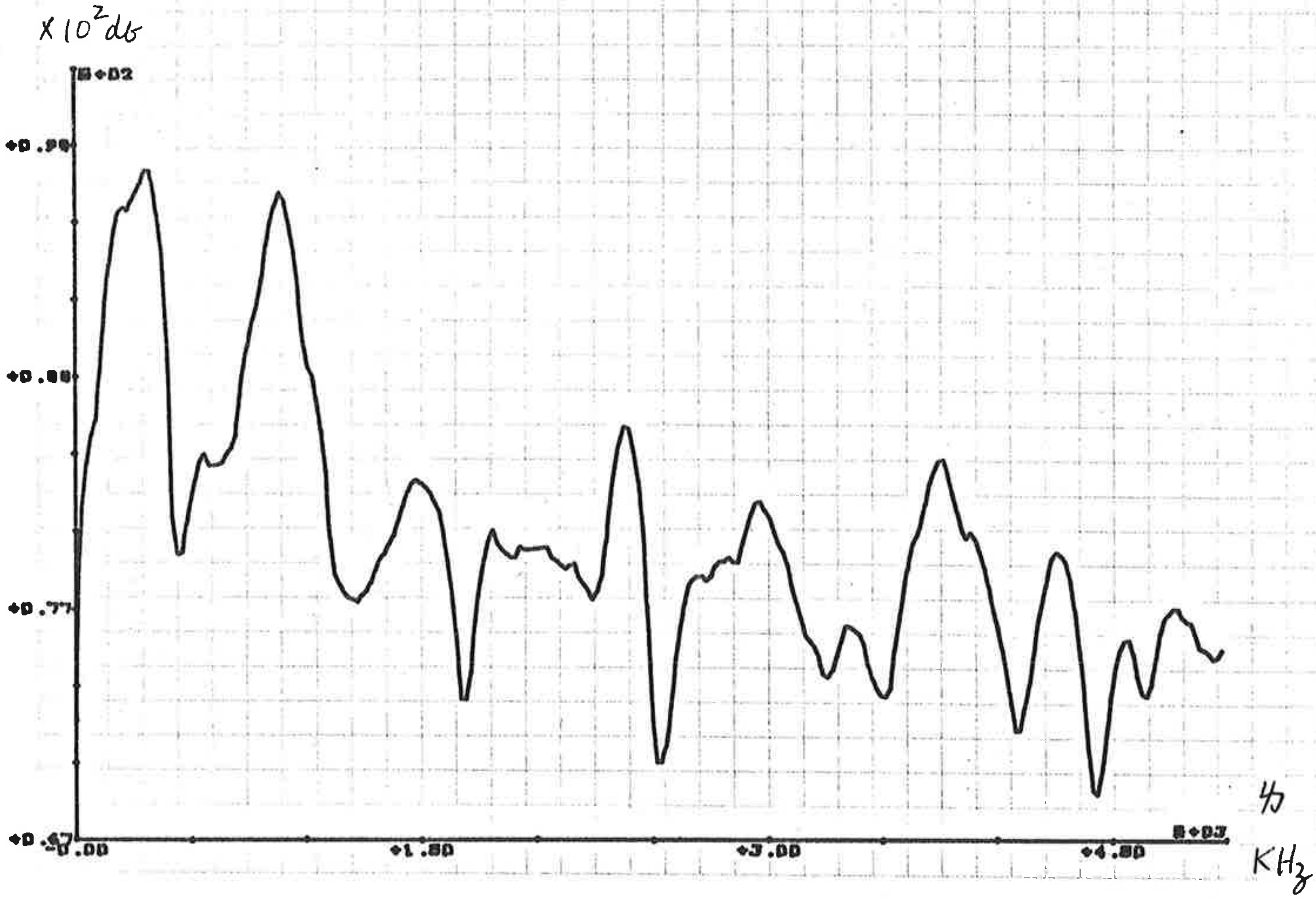
x10² db
E+02
+1.04
+0.92
+0.80
+0.68
-0.00
+1.50
+3.00
+4.50
E+03
KHz

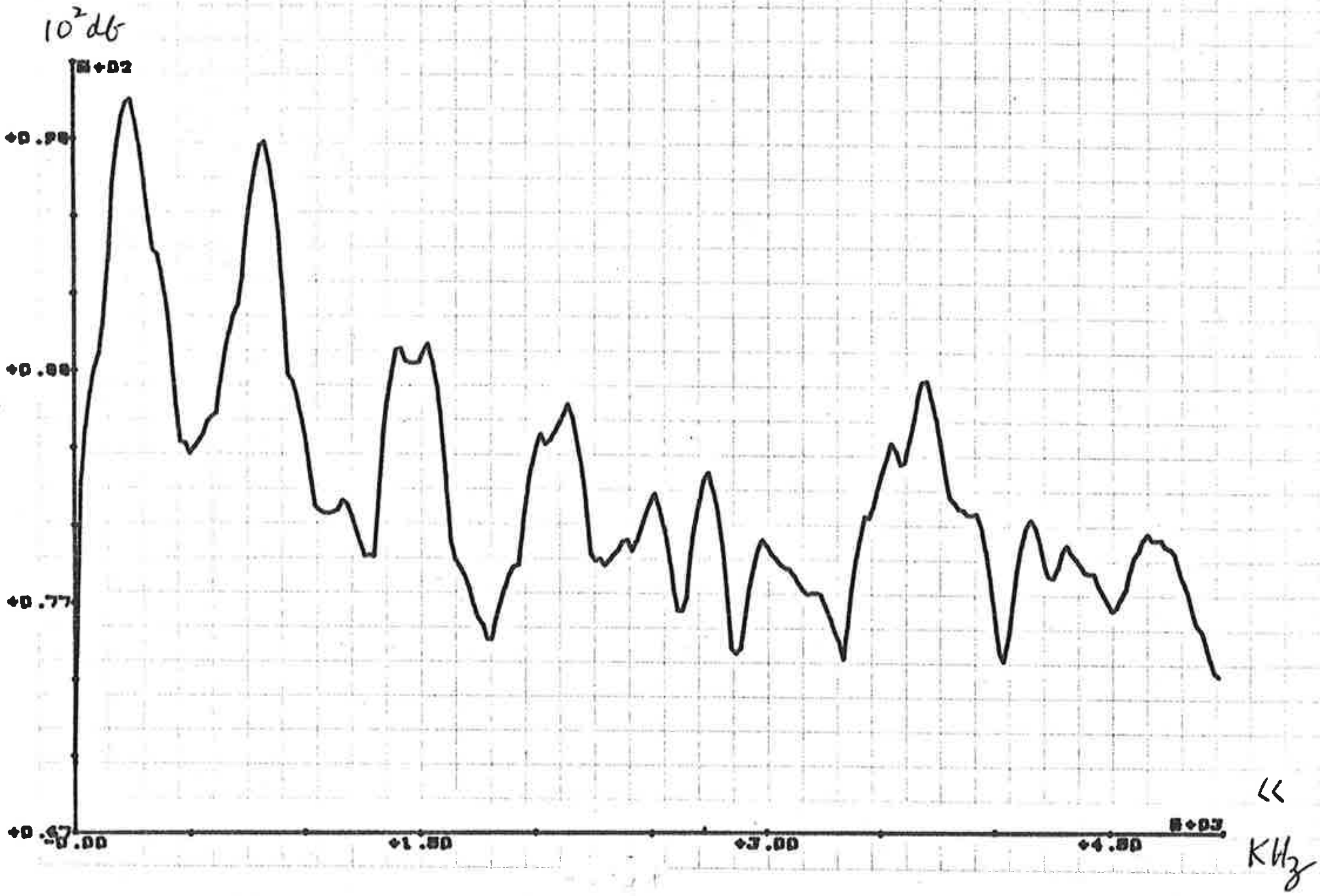


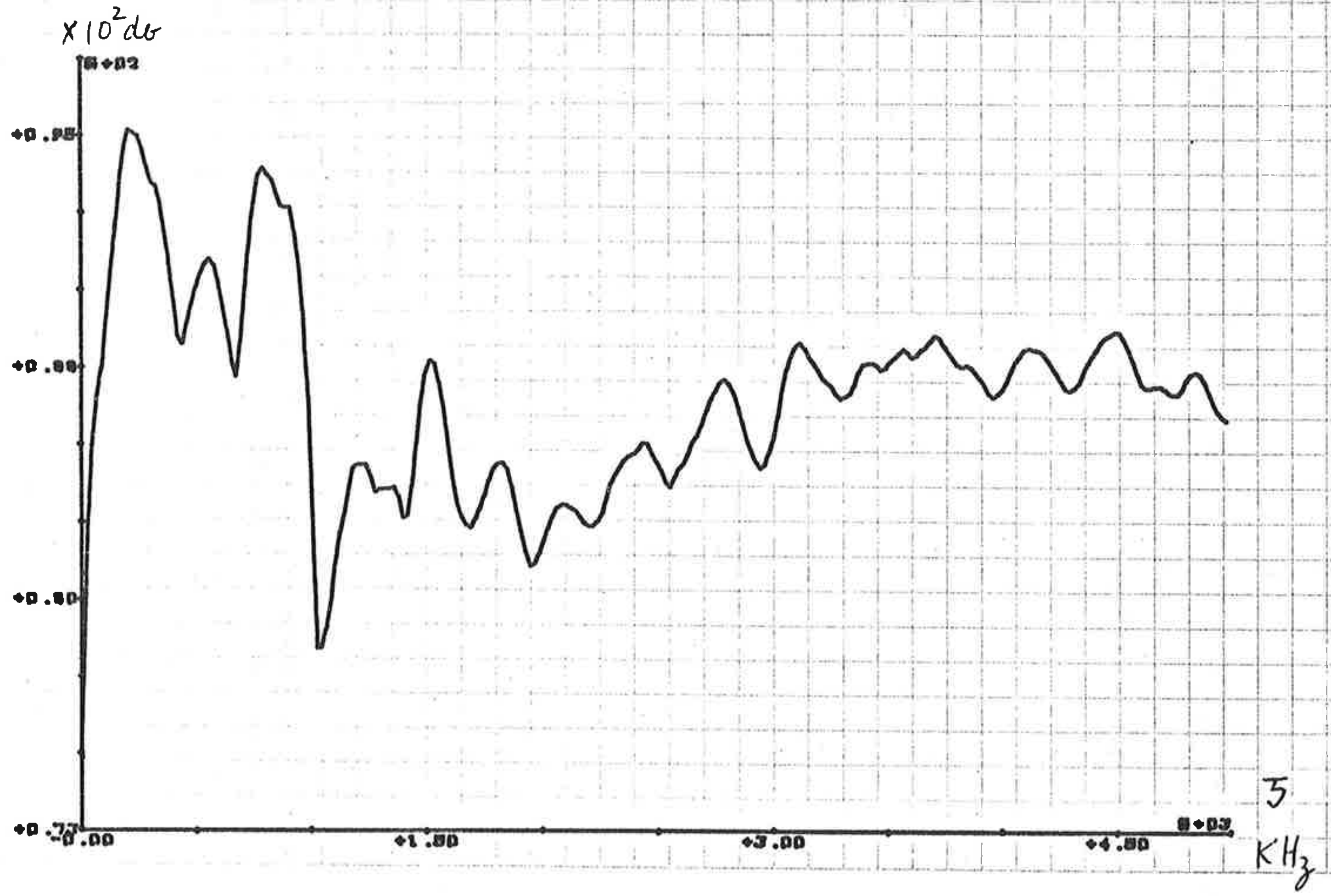
x10² db
E+02
+0.98
+0.88
+0.77
+0.67
+0.00
+1.50
+3.00
+4.50
E+03
去
KHz

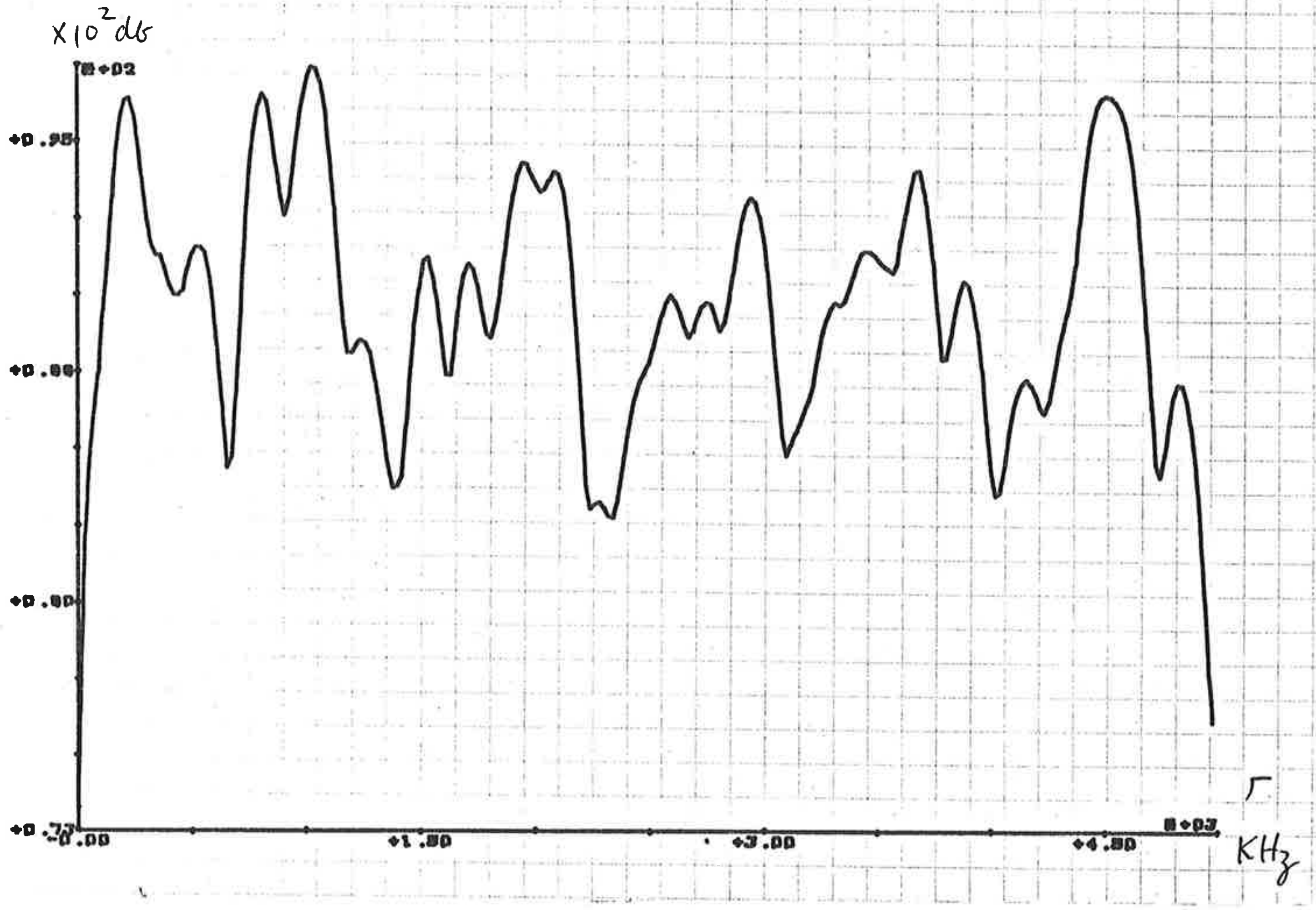


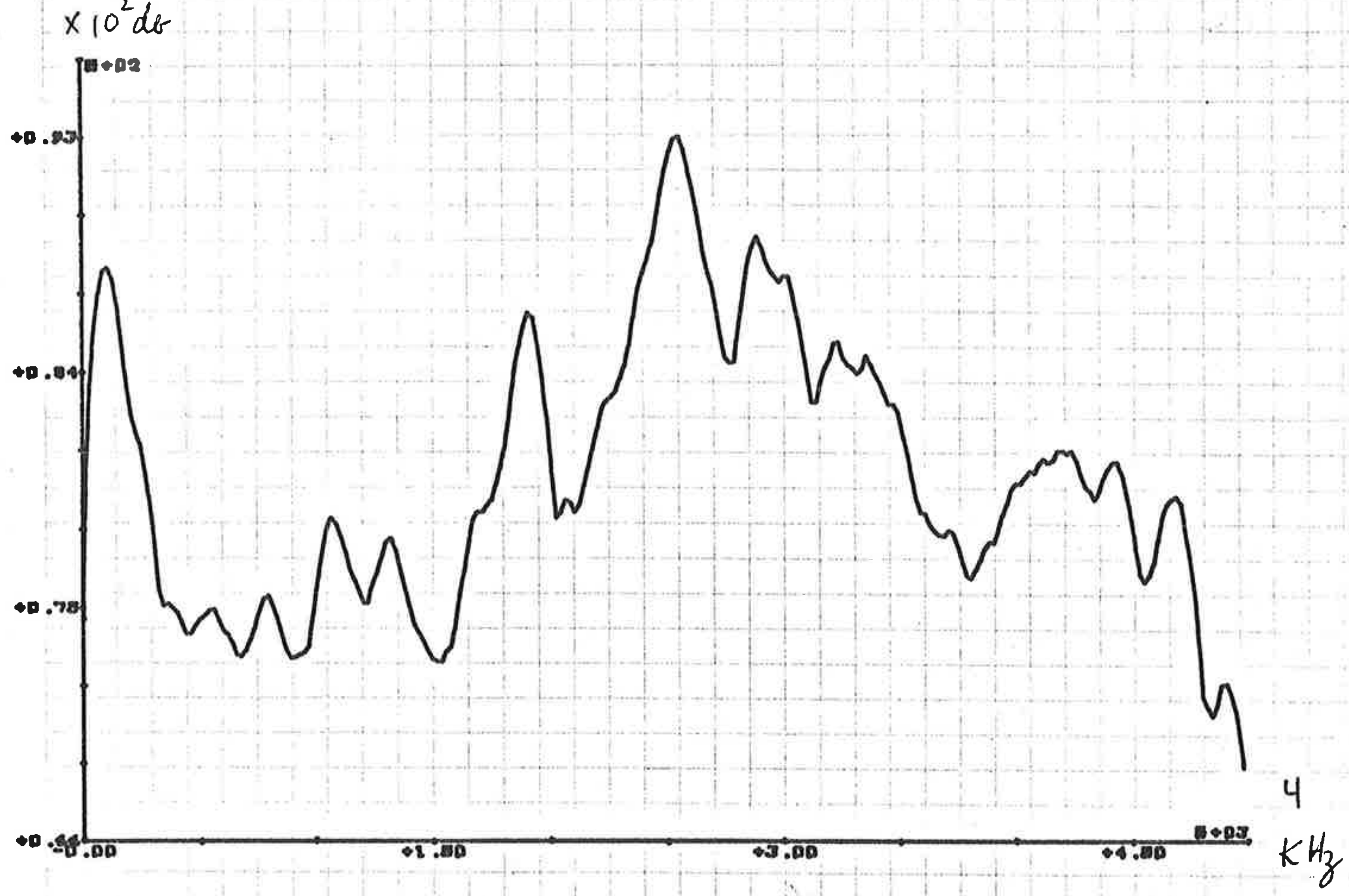
x10² db
E+02
+0.98
+0.88
+0.77
+0.67
+0.00
+1.50
+3.00
+4.50
E+03
3
KHz


X 10² db
E+02
+0.98
+0.88
+0.77
+0.67
+0.00
+1.50
+3.00
+4.50
E+03
4/5
KHz

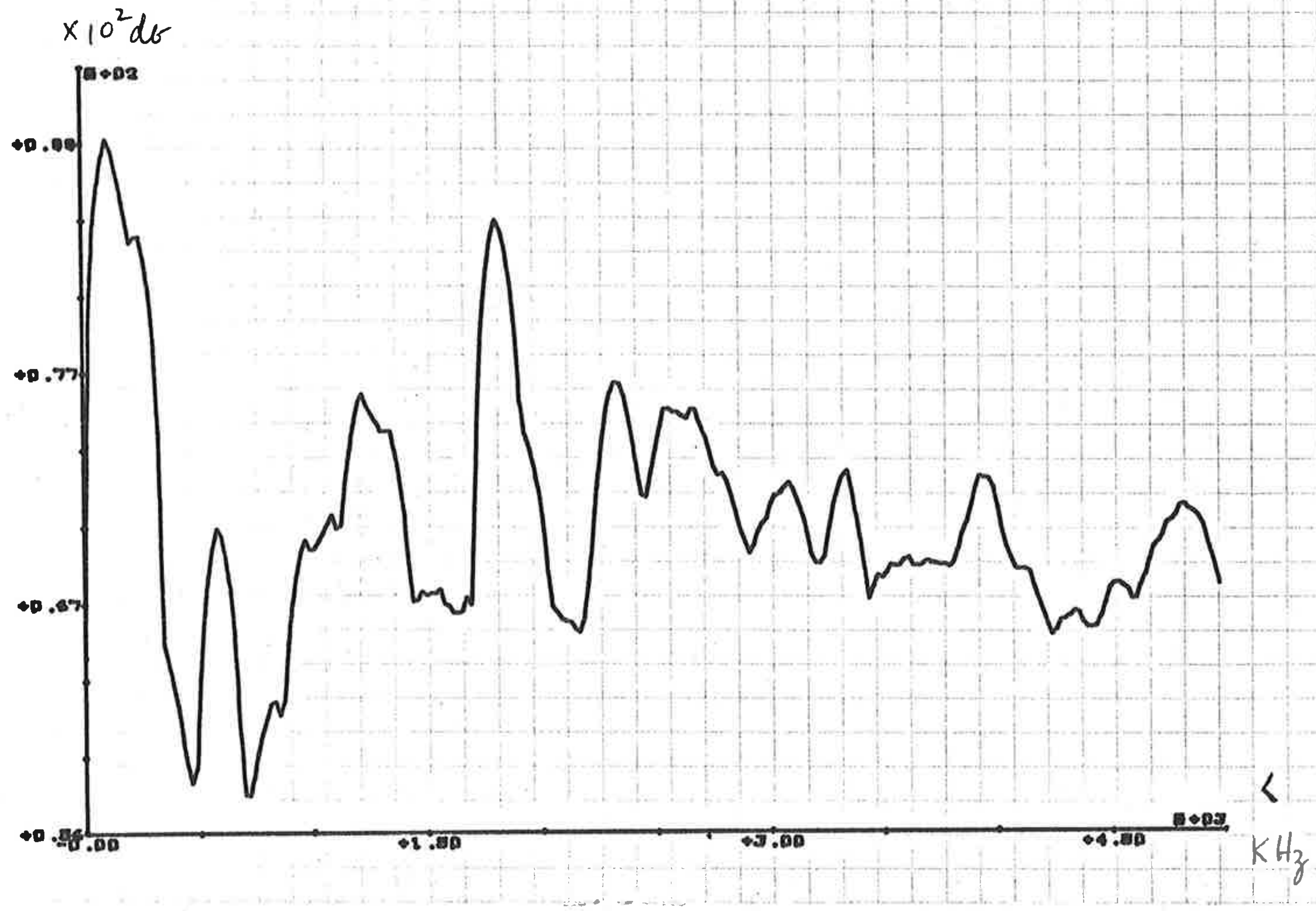

10² db
E+02
+0.98
+0.88
+0.77
+0.67
+0.00
+1.50
+3.00
+4.50
E+03
"
KHz


x10^{2}db
8+02
+0.95
+0.88
+0.80
+0.73
+0.00
+1.50
+3.00
+4.50
8+03
KHz
Б

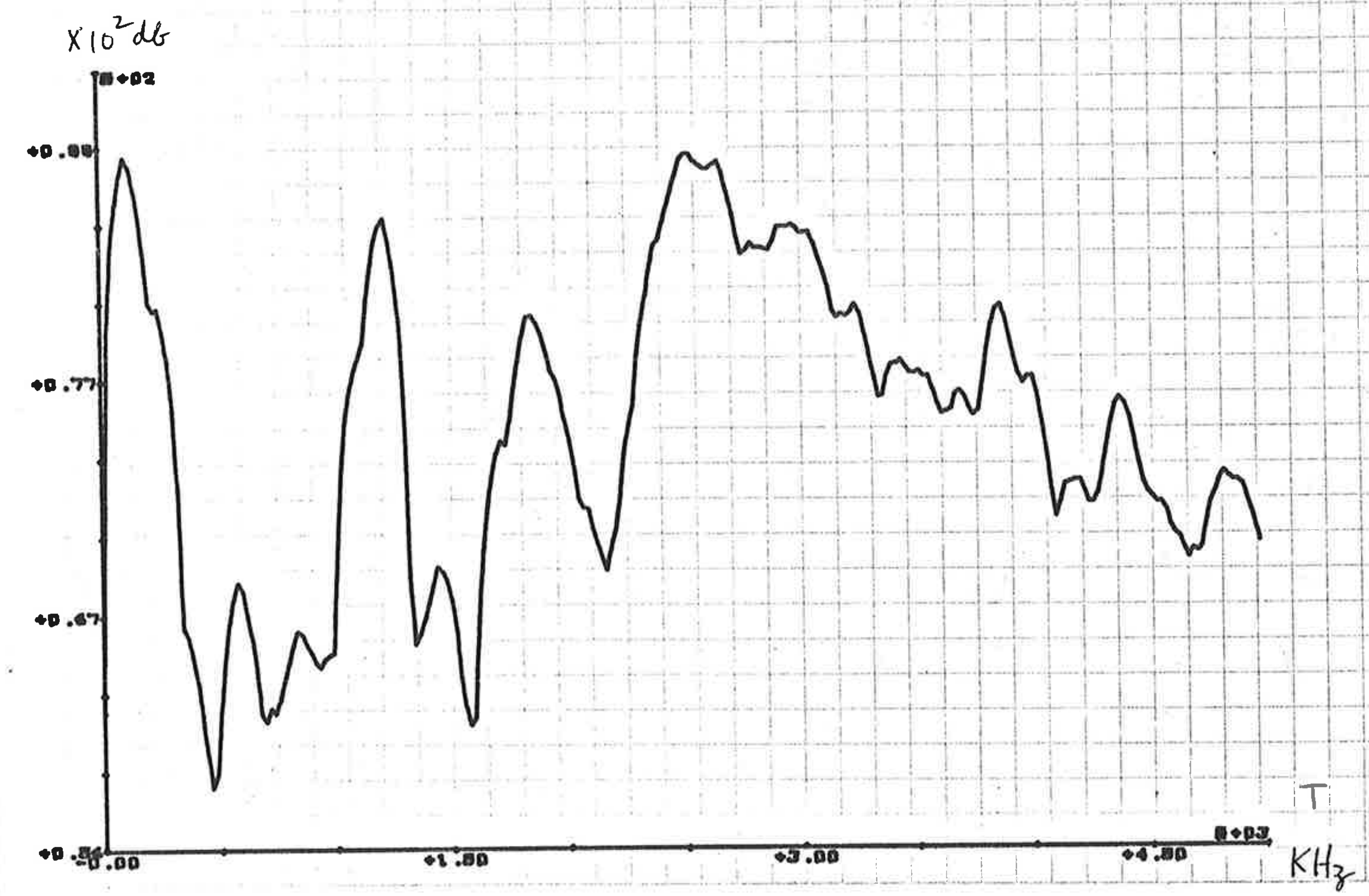

x10^{2}db
8+02
+0.95
+0.88
+0.80
+0.73
+0.00
+1.50
+3.00
+4.50
8+03
KHz
Г

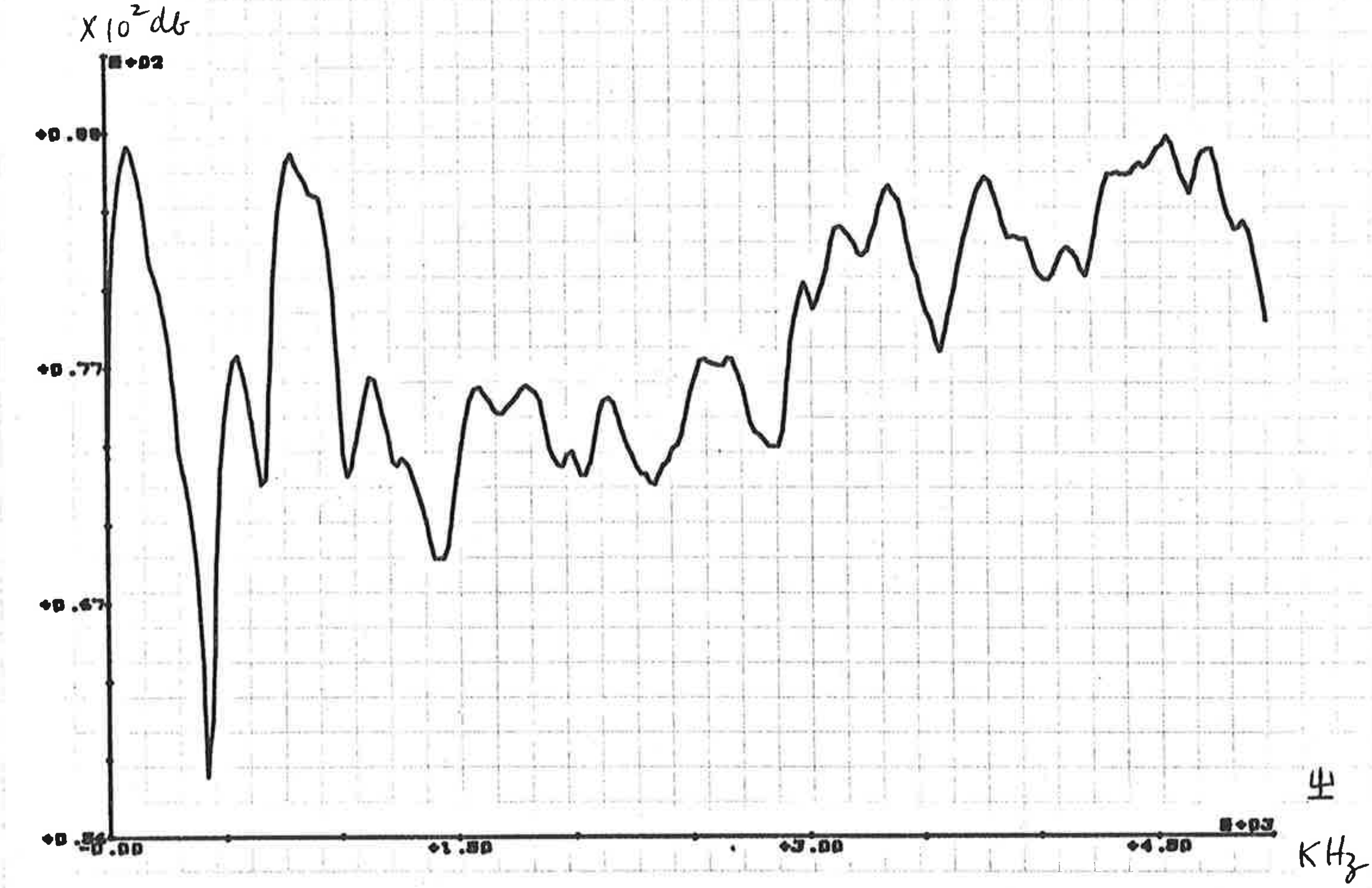


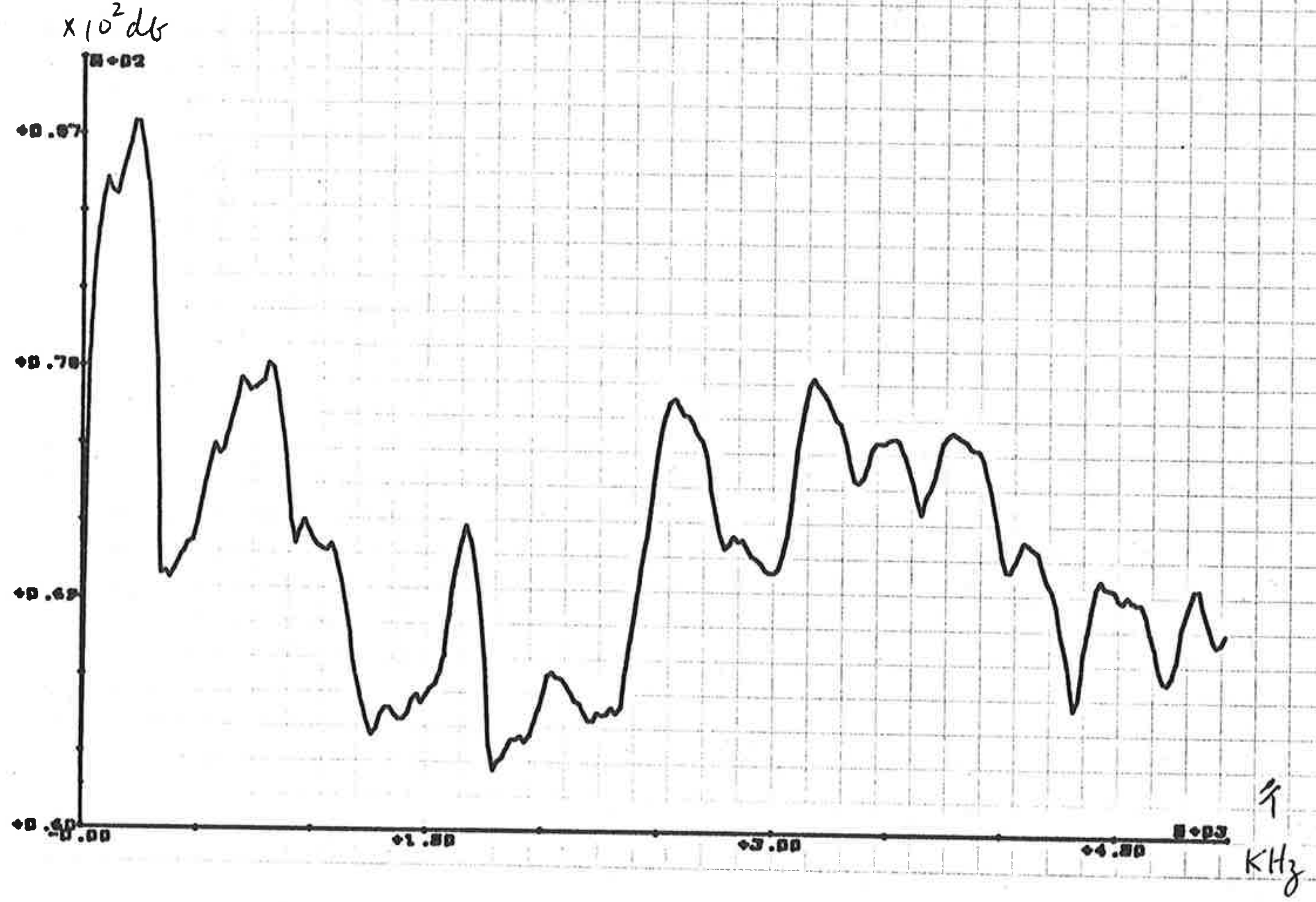
X 10² db
E+02
+0.93
+0.84
+0.75
+0.66
-0.00
+1.50
+3.00
+4.50
E+03
4
KHz



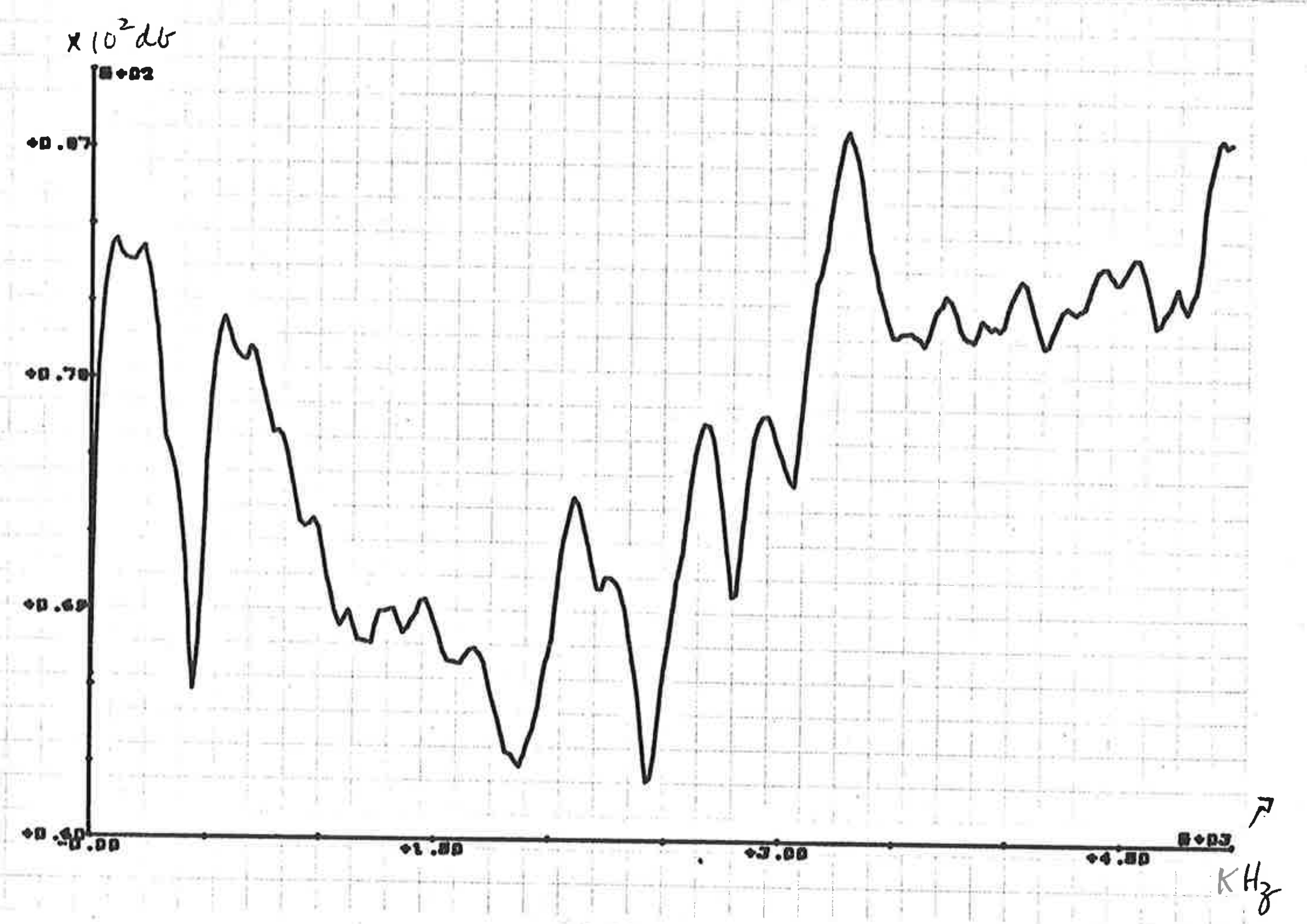
X 10² db
E+02
+0.88
+0.77
+0.67
+0.56
-0.00
+1.50
+3.00
+4.50
E+03
<
KHz


X 10² db
E+02
+0.88
+0.77
+0.67
+0.56
-0.00
+1.50
+3.00
+4.50
E+03
T
KHz

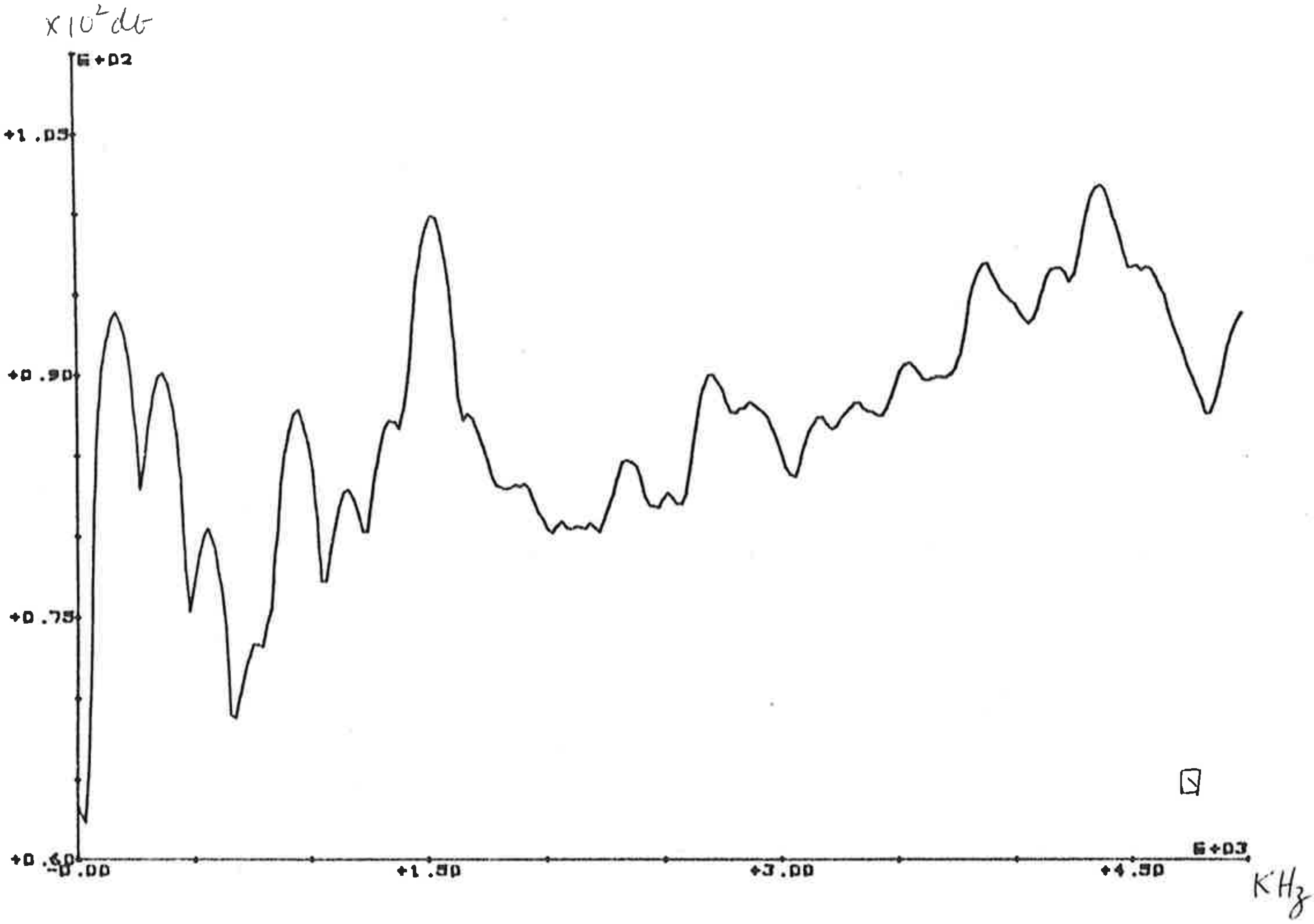

X 10² db
E+02
+0.88
+0.77
+0.67
+0.56
-0.00
+1.50
+3.00
+4.50
E+03
KHz

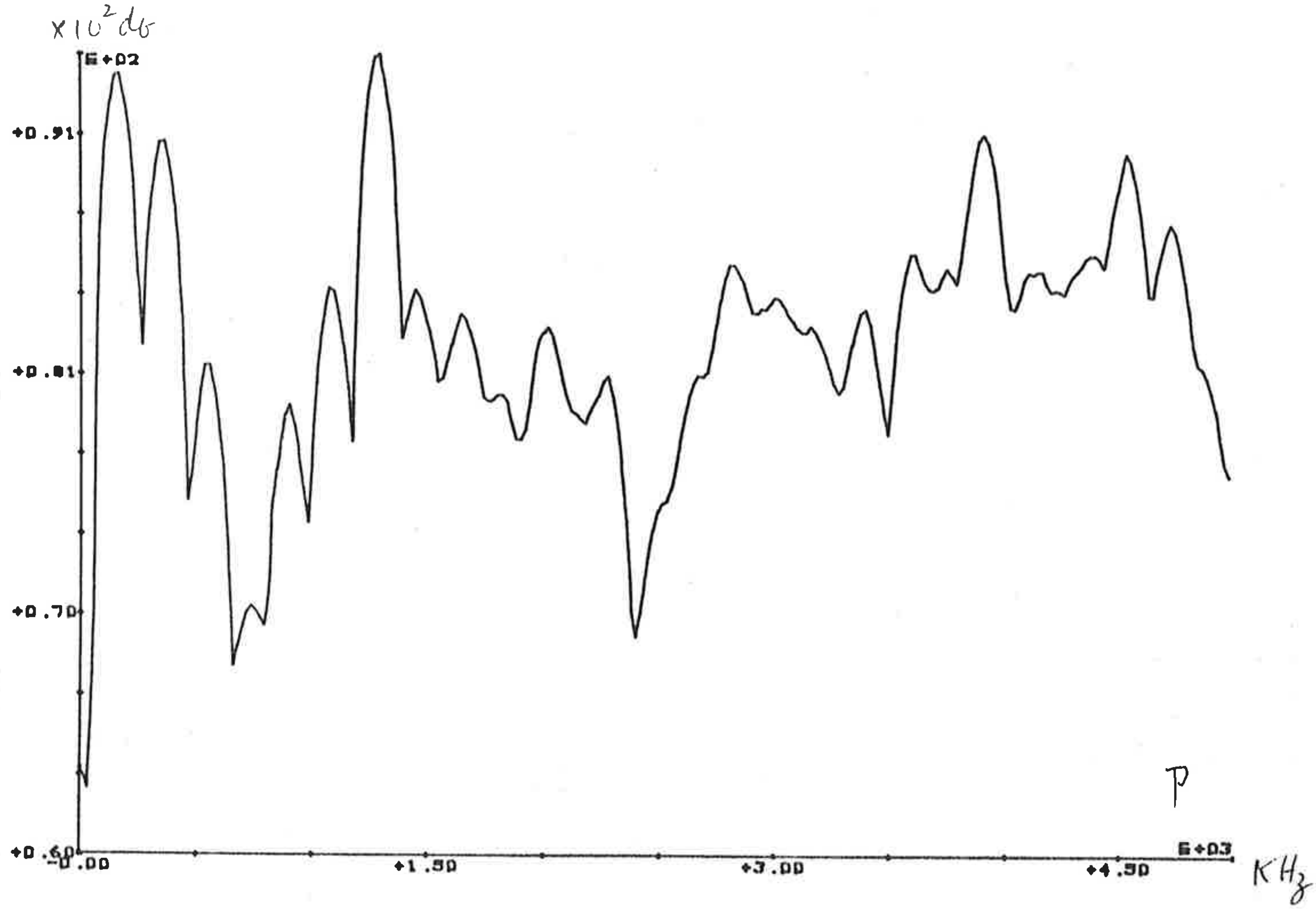


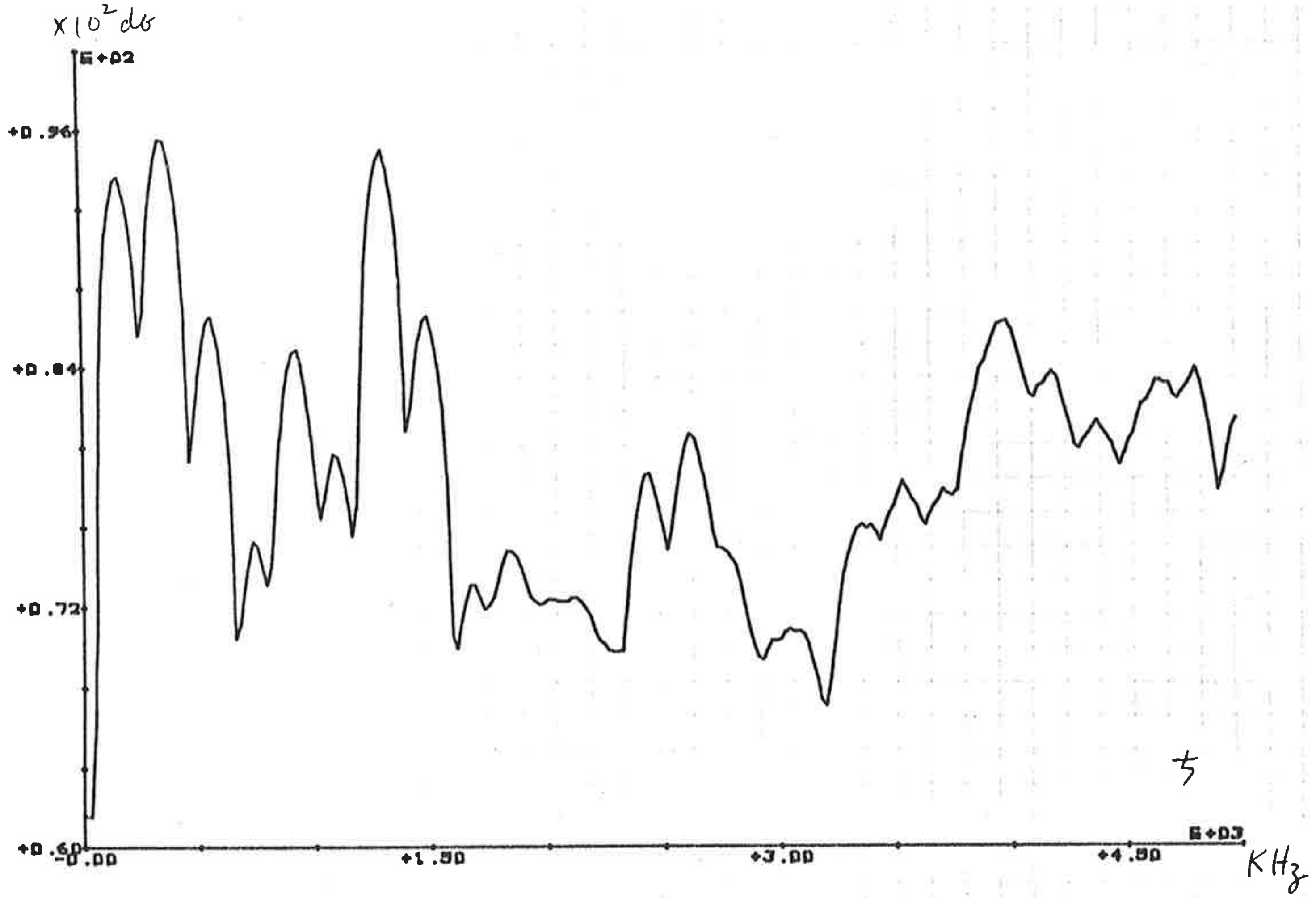
x10²db
E+02
+0.87
+0.78
+0.69
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz

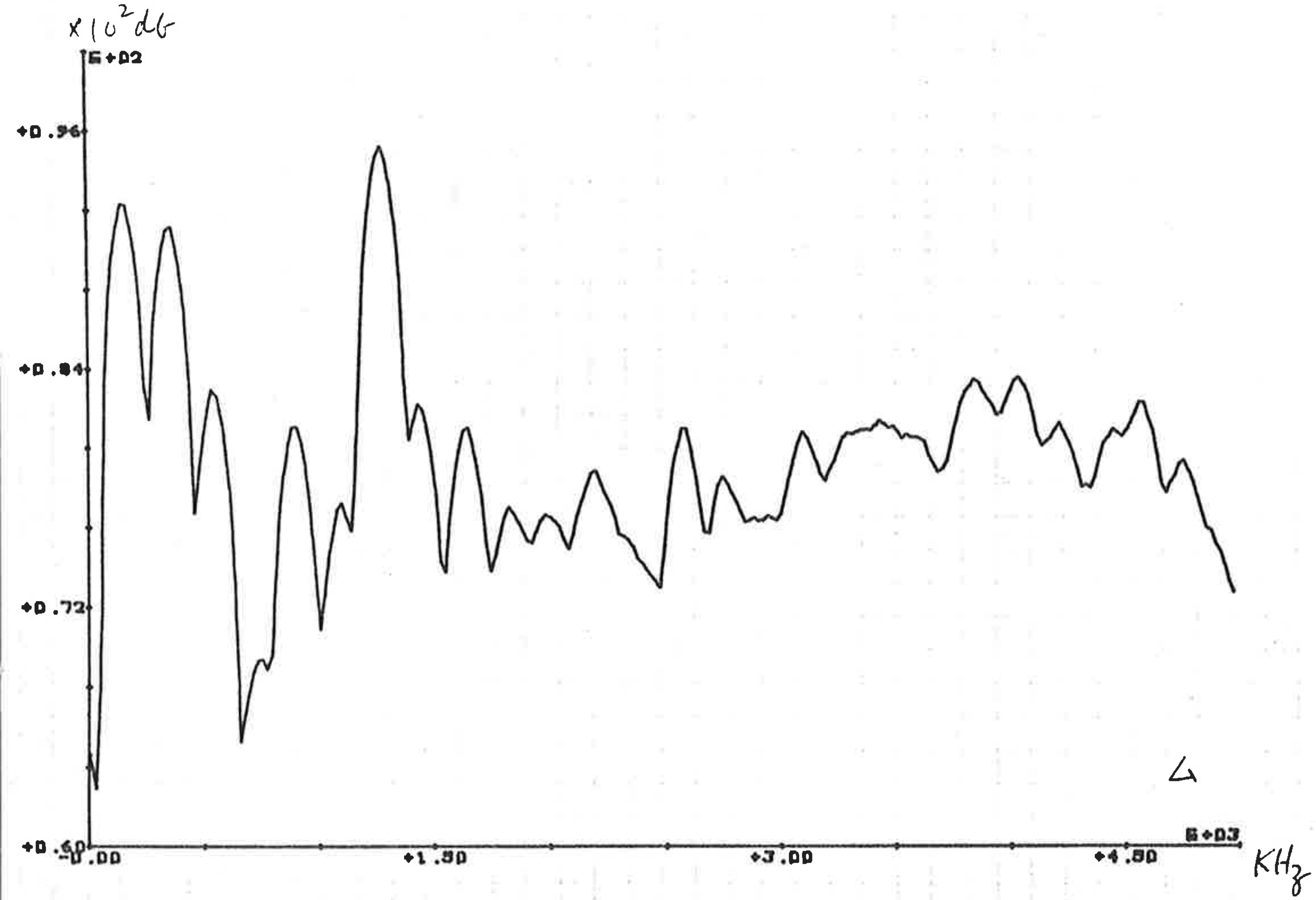


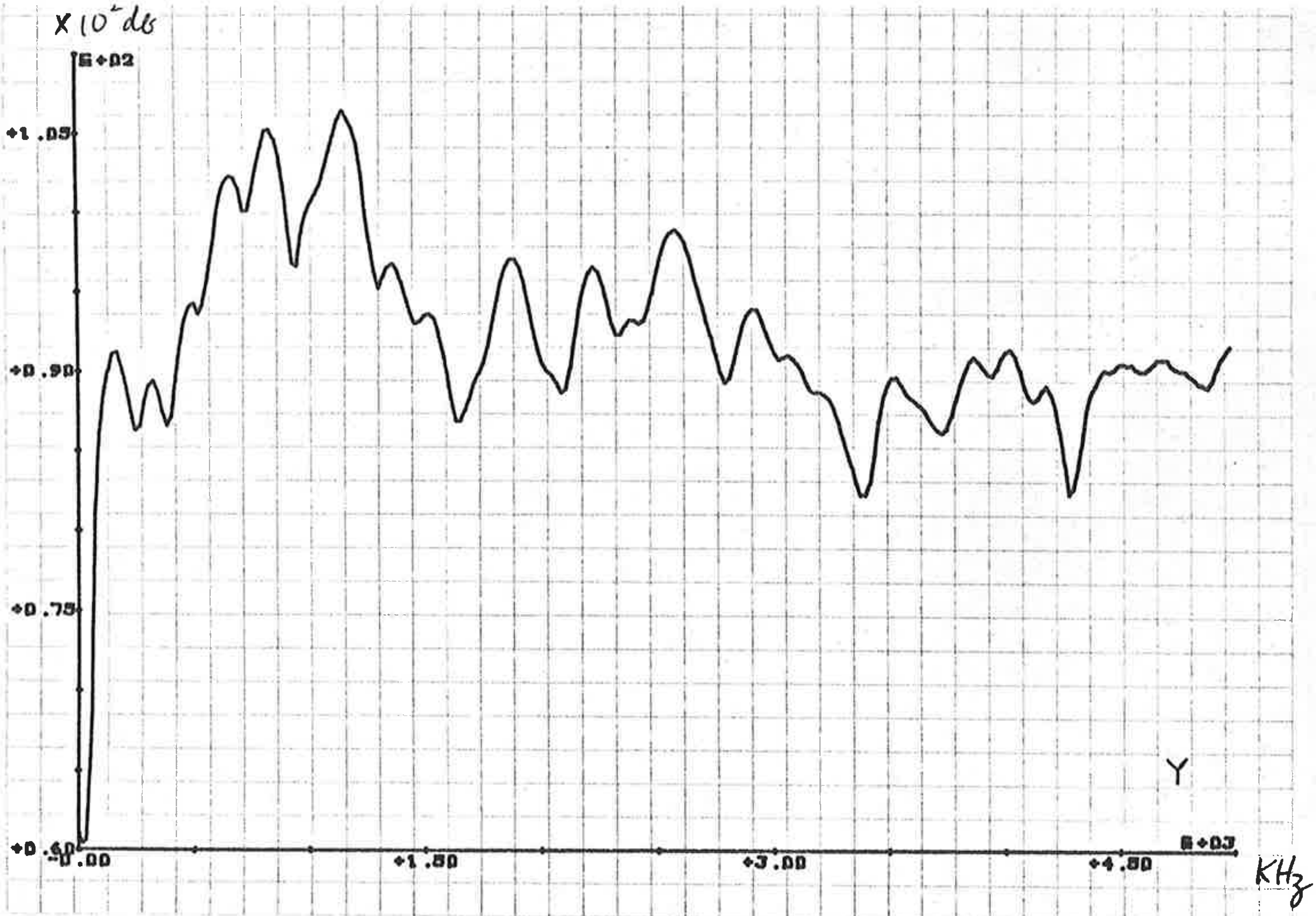
x10²db
E+02
+0.87
+0.78
+0.69
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz

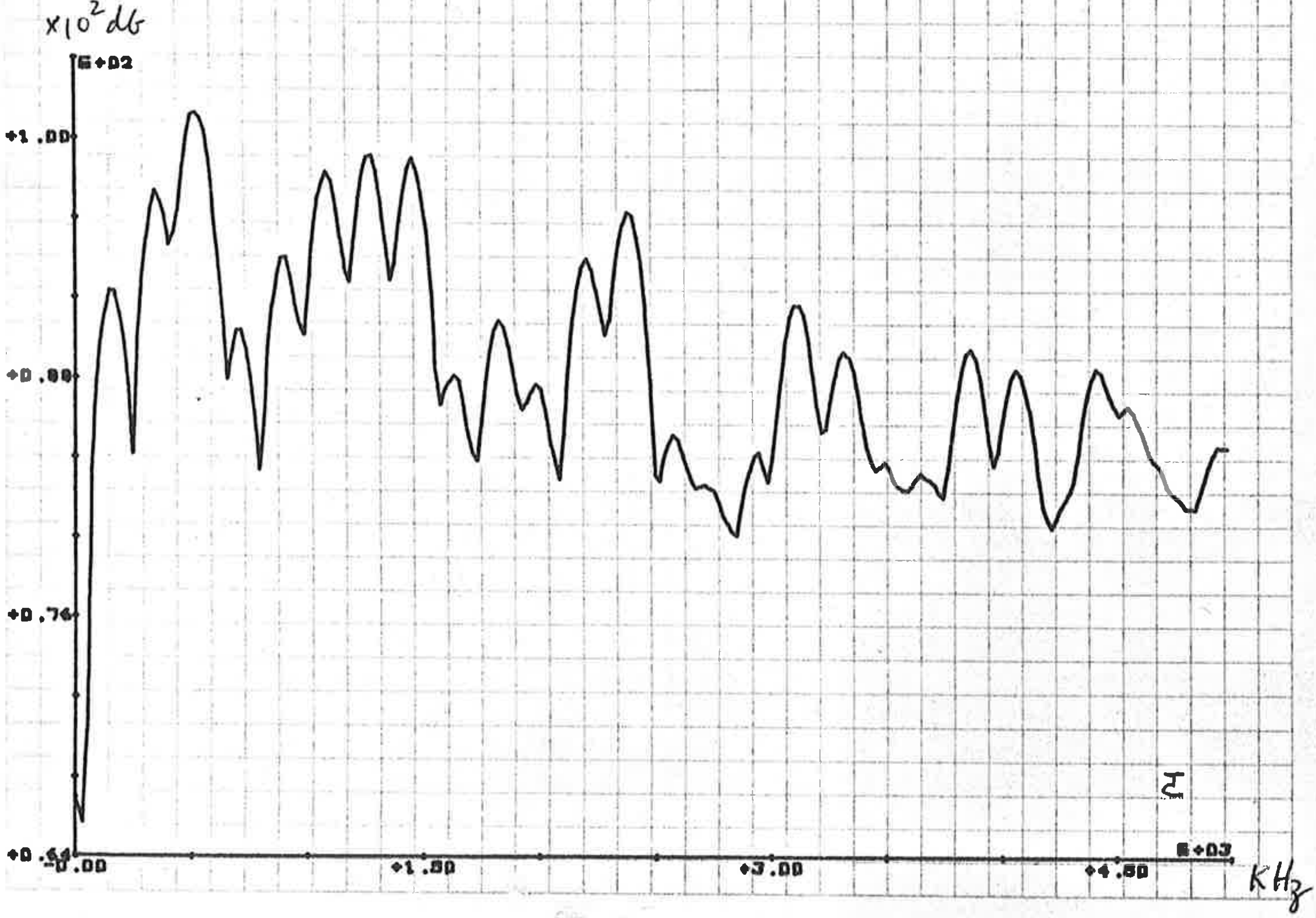

x10² db
E+02
+1.05
+0.90
+0.75
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz
Q

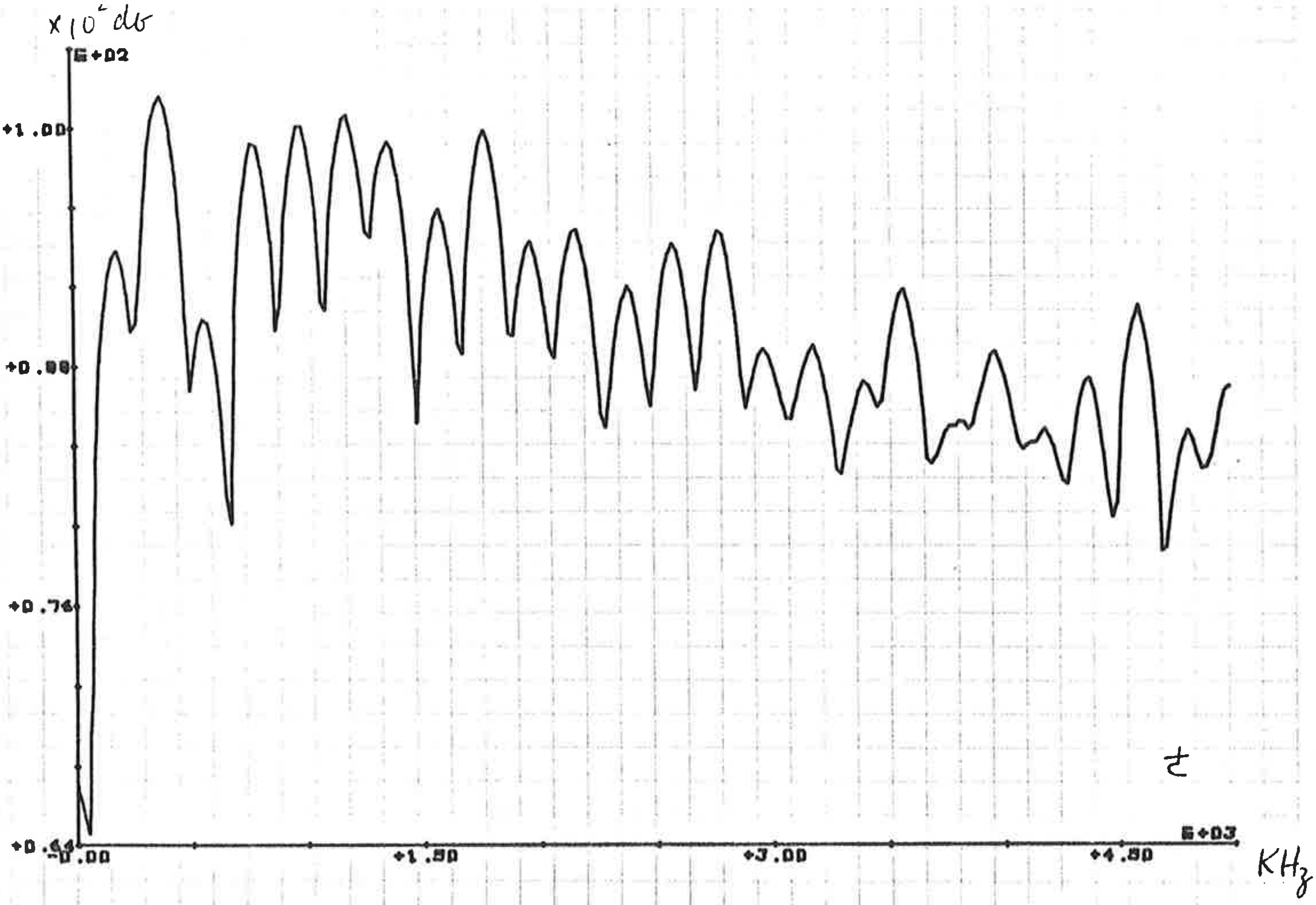

x10² db
E+02
+0.91
+0.81
+0.70
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz
P

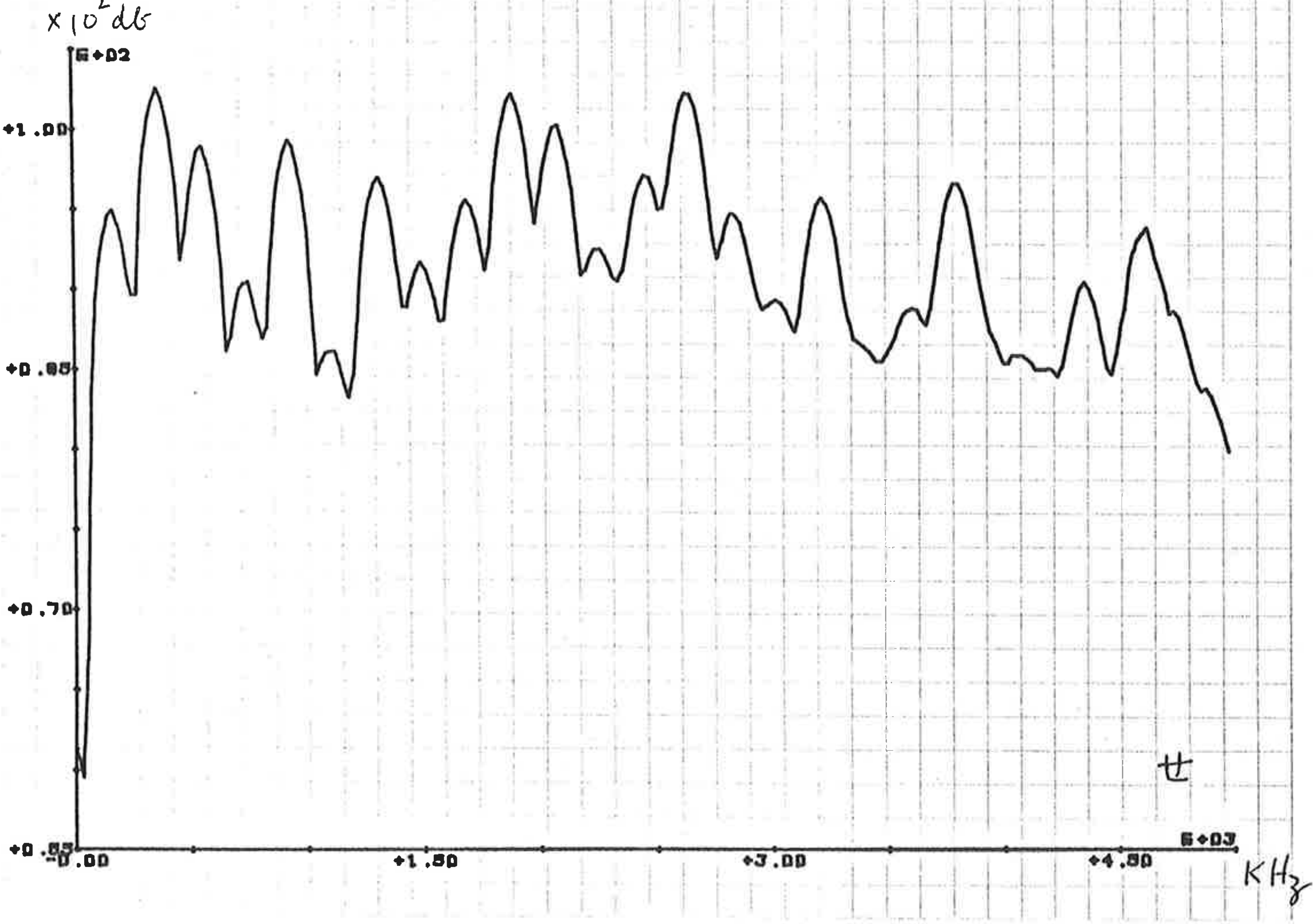

×10² db
E+02
+0.96
+0.84
+0.72
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz

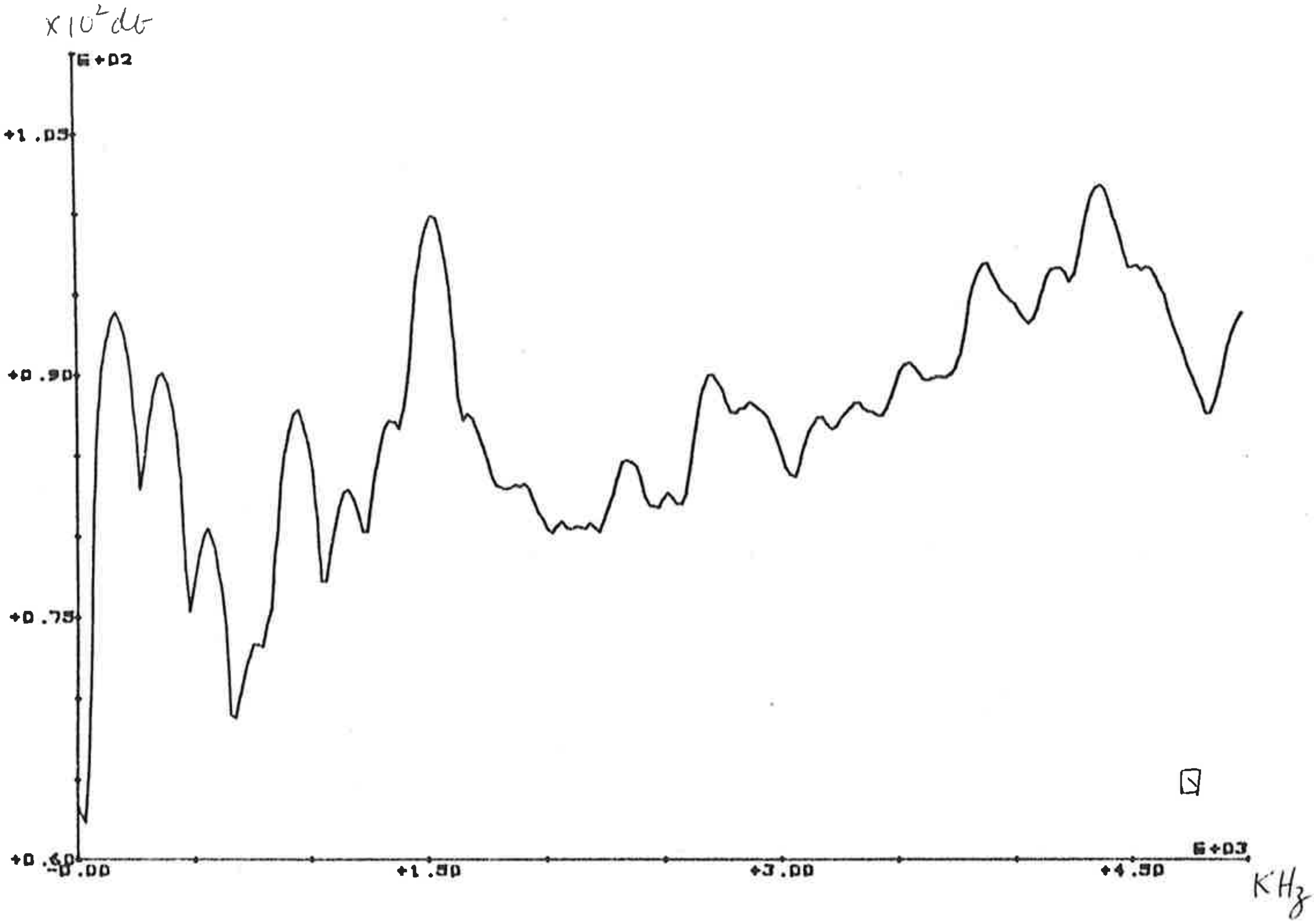

×10² db
E+02
+0.96
+0.84
+0.72
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz

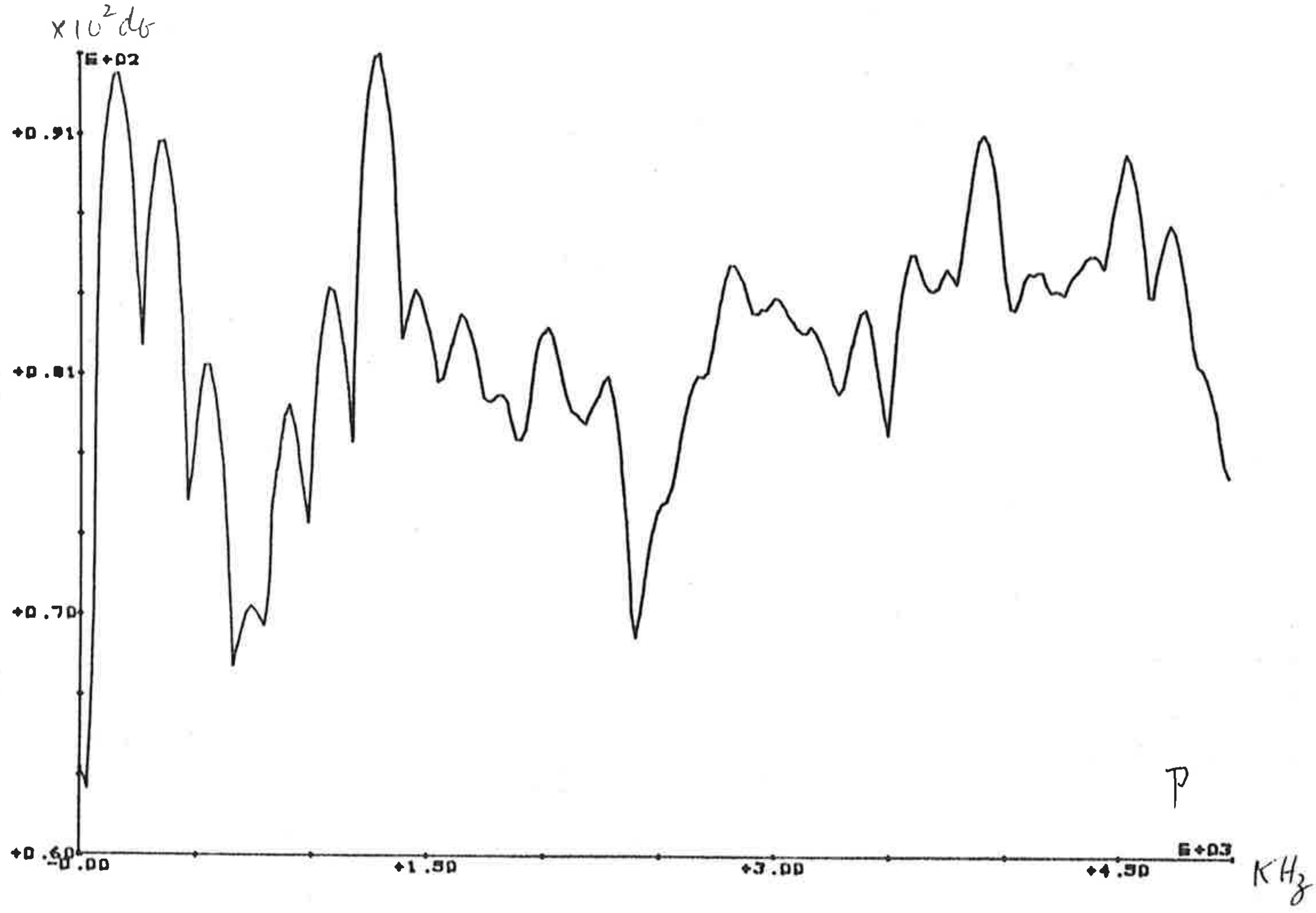

X 10² db
E+02
+1.05
+0.90
+0.75
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz
Y

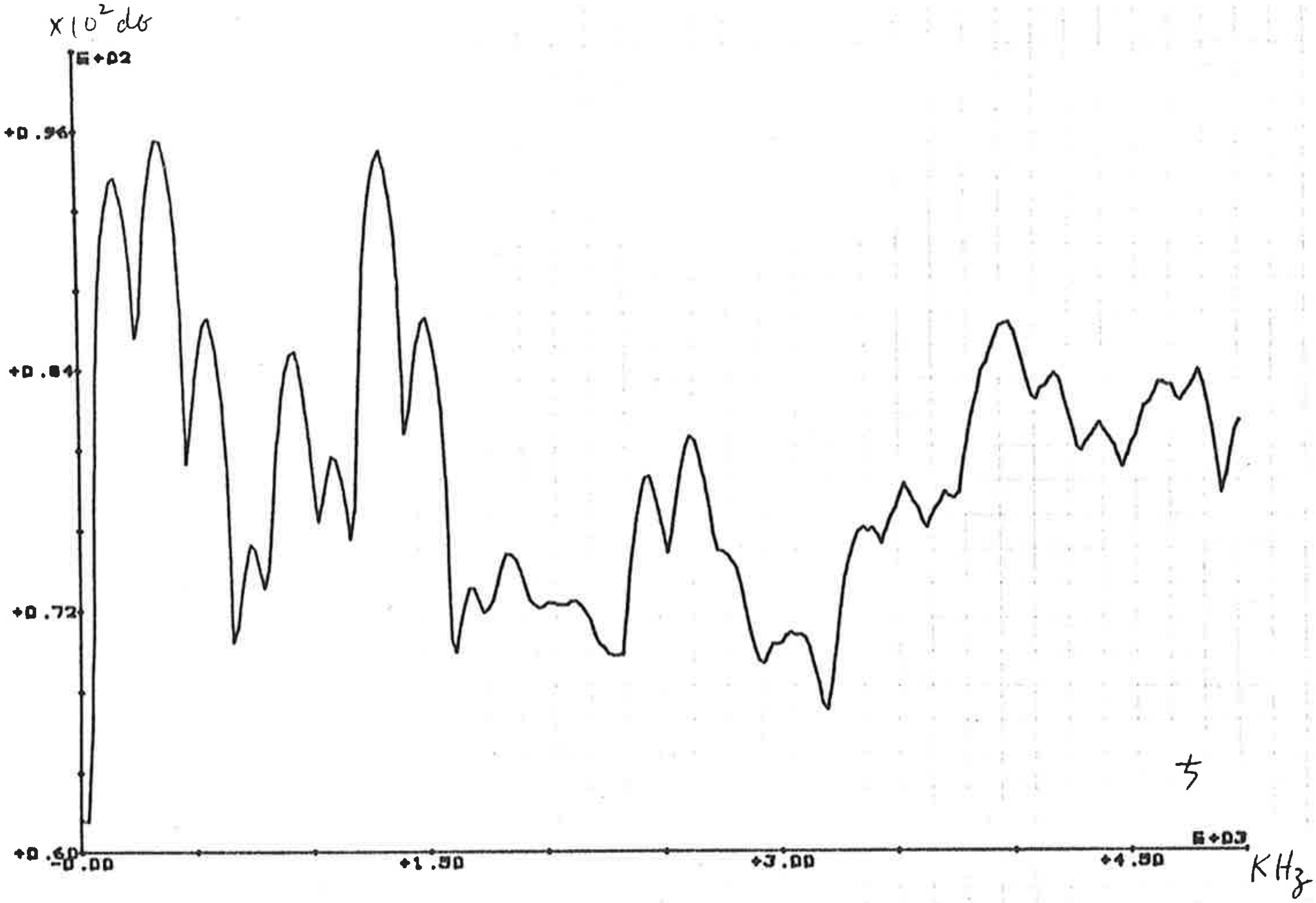

x10² db
E+02
+1.00
+0.88
+0.76
-0.00
+1.50
+3.00
+4.50
E+03
KHz
Z

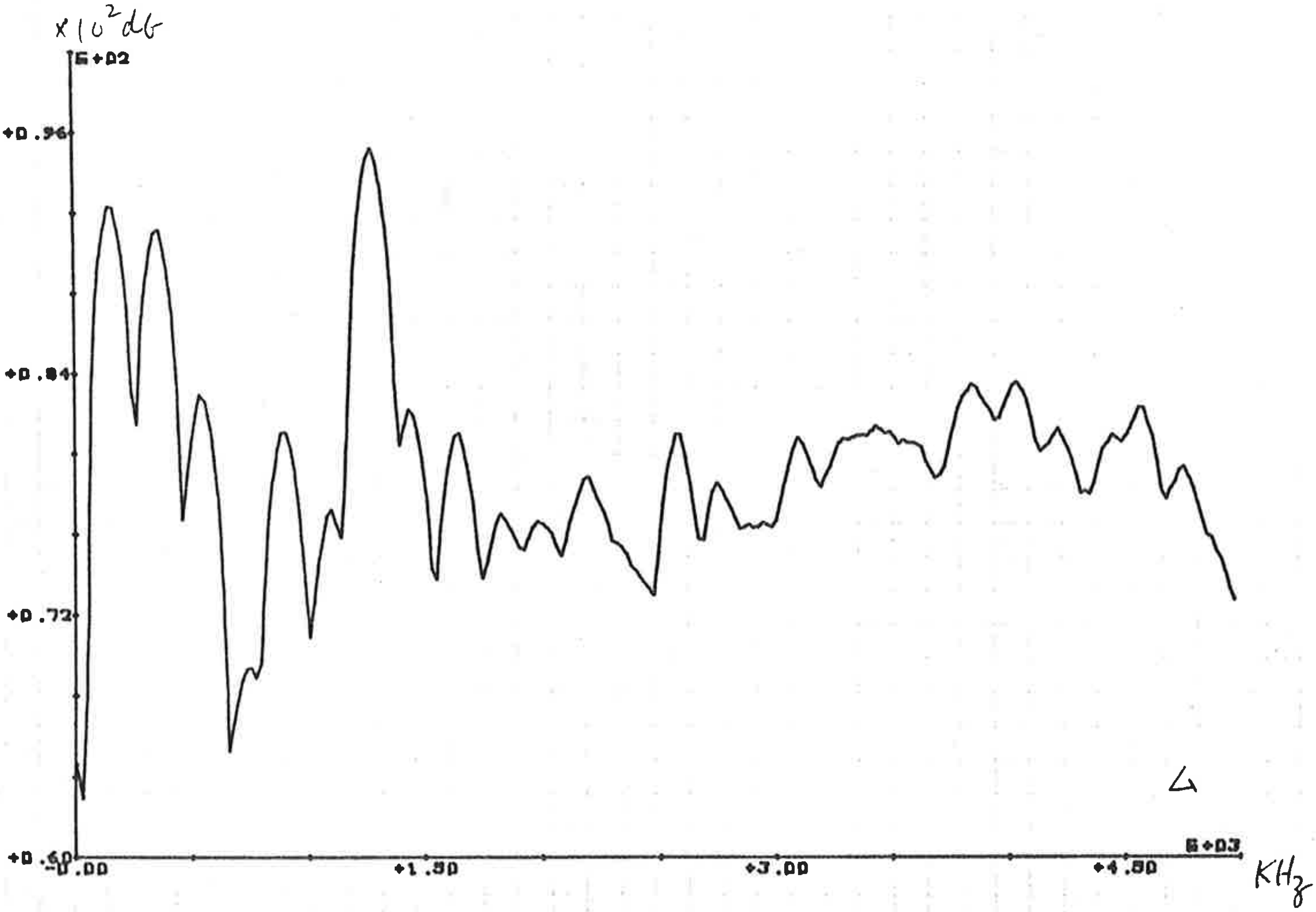

x10² db
E+02
+1.00
+0.88
+0.76
+0.64
-0.00
+1.50
+3.00
+4.50
E+03
KHz
さ

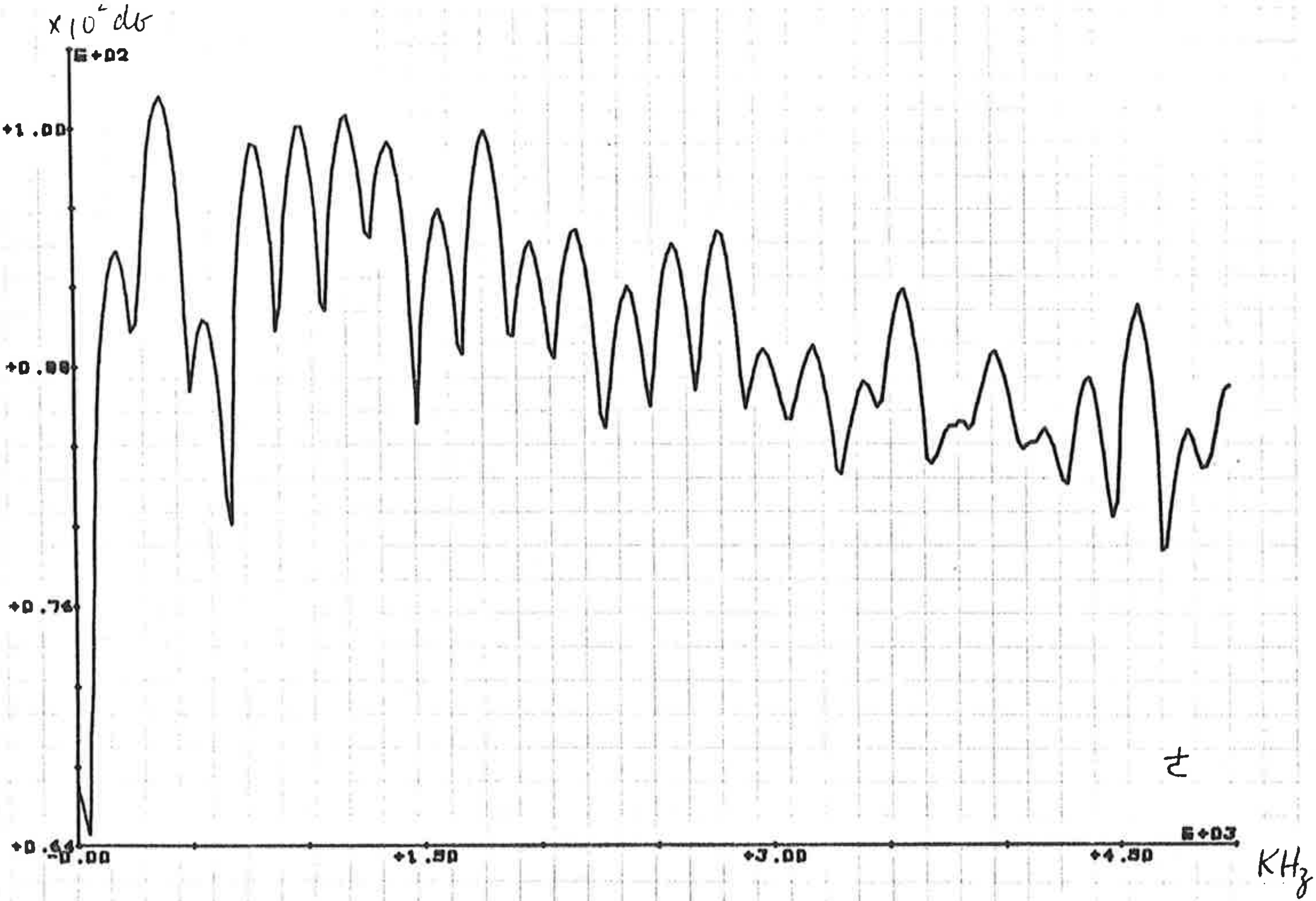

x10² db
E+02
+1.00
+0.85
+0.70
+0.55
-0.00
+1.50
+3.00
+4.50
E+03
KHz
せ

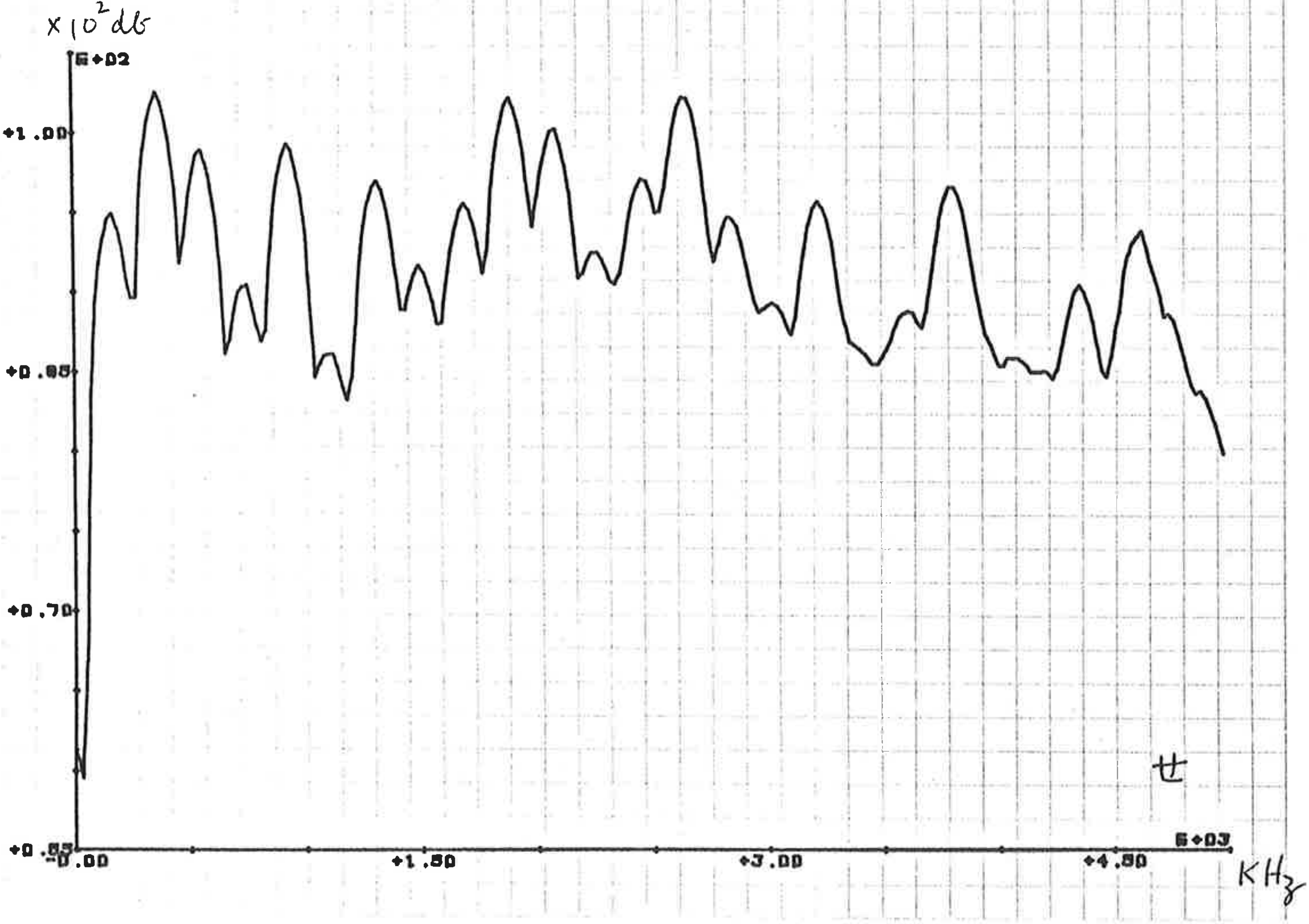

x10² db
E+02
+1.05
+0.90
+0.75
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz
Q

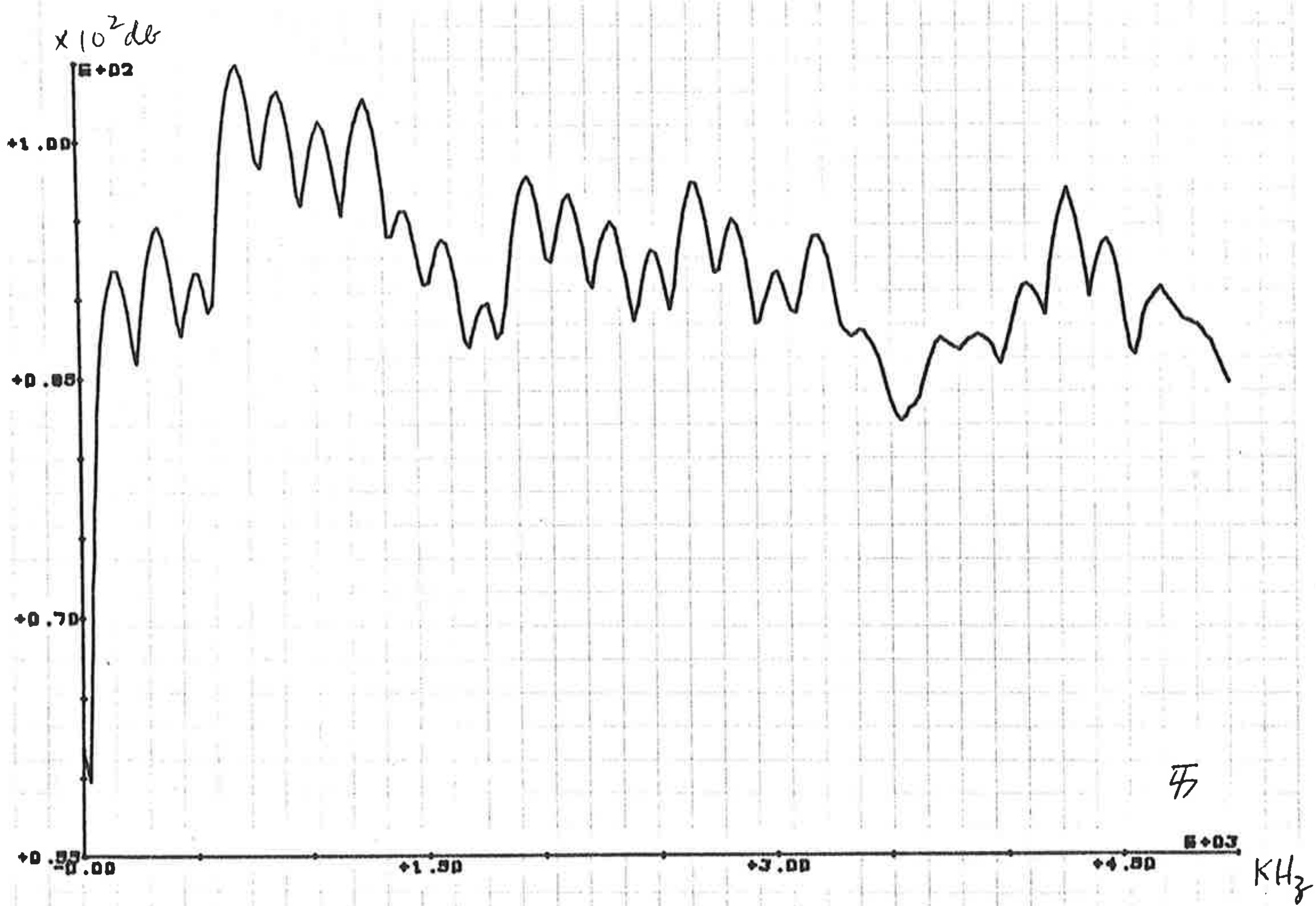

x10² db
E+02
+0.91
+0.81
+0.70
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz
P

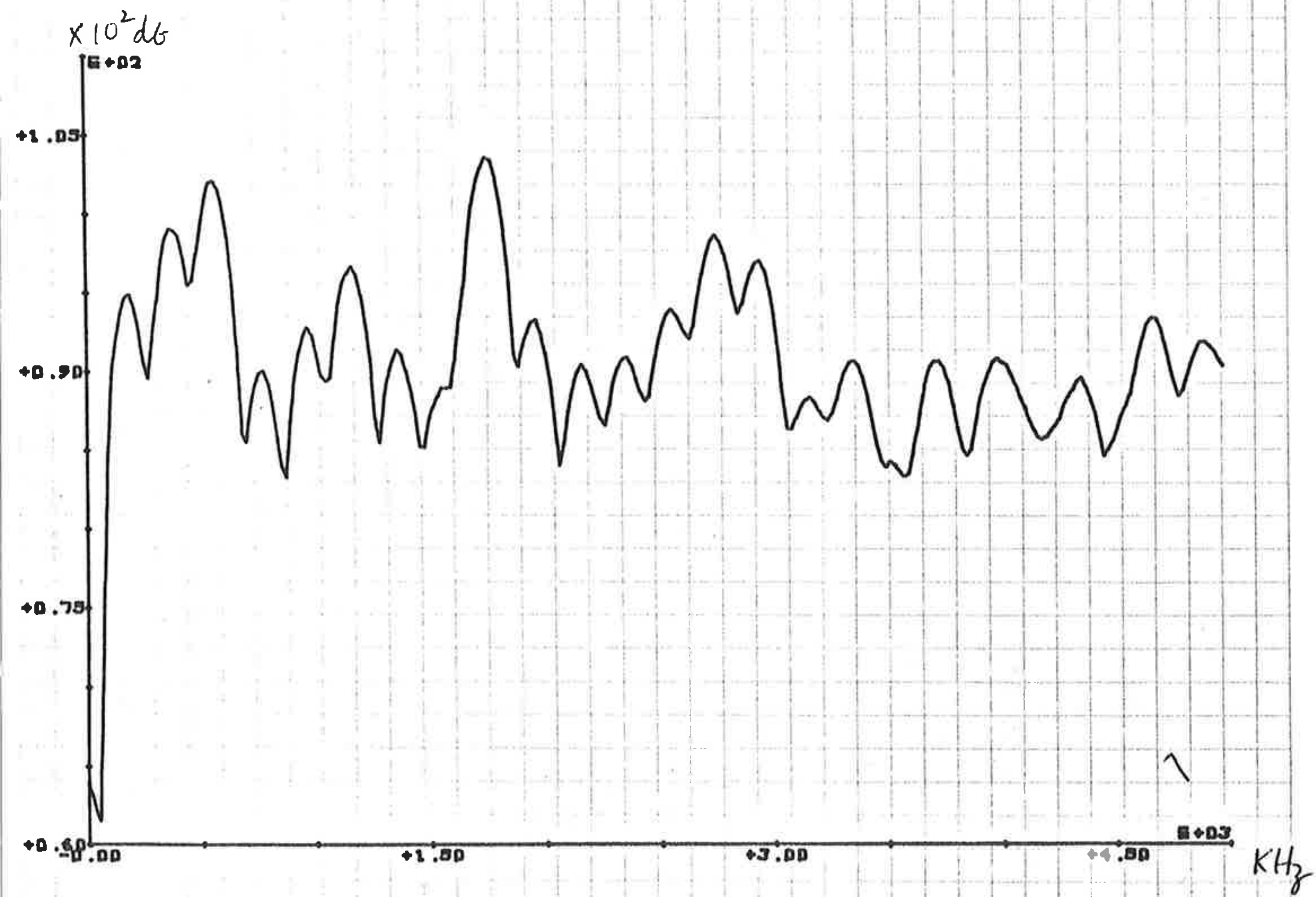

x10² db
E+02
+0.96
+0.84
+0.72
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz
5

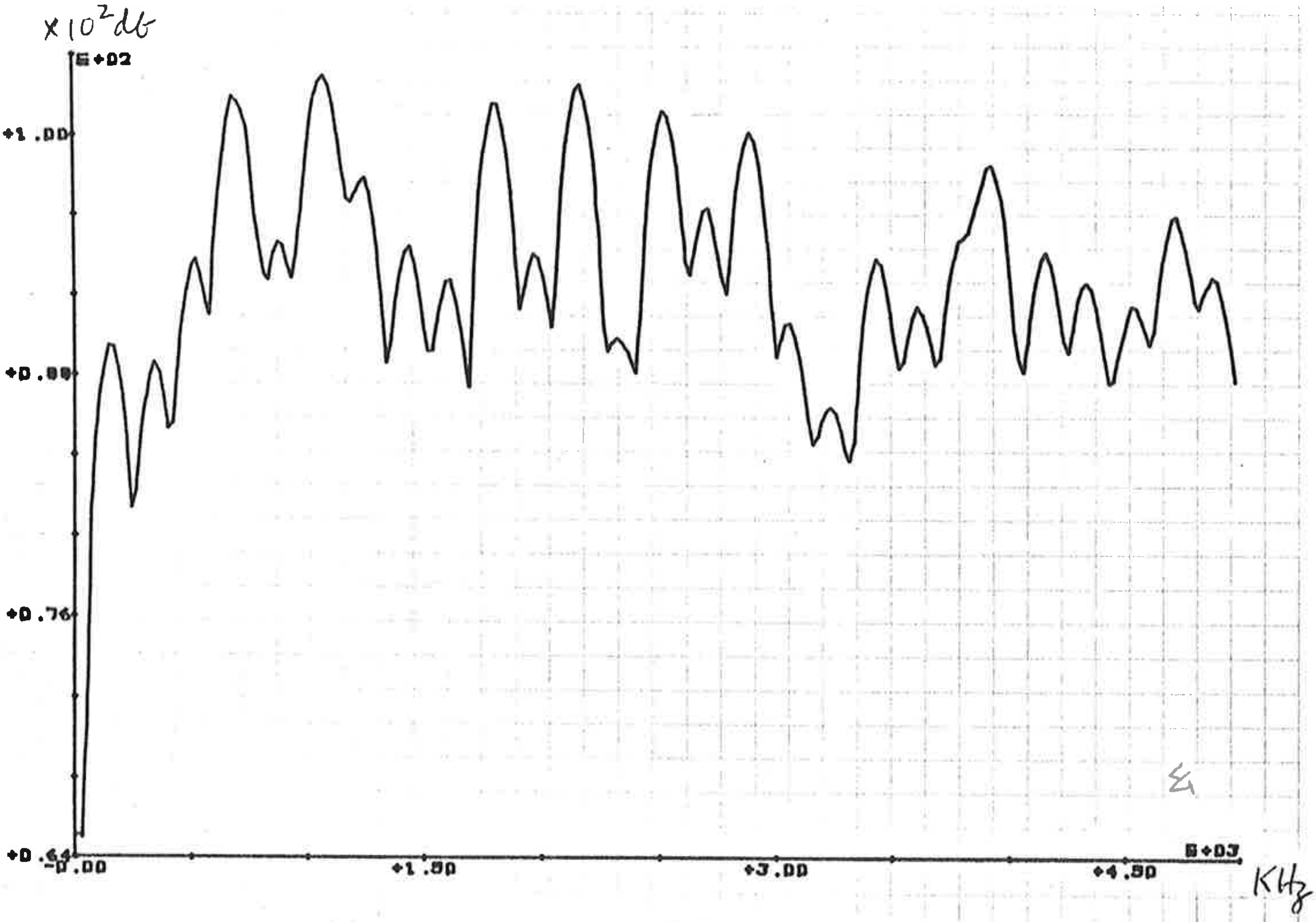

x10² db
E+02
+0.96
+0.84
+0.72
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz
4

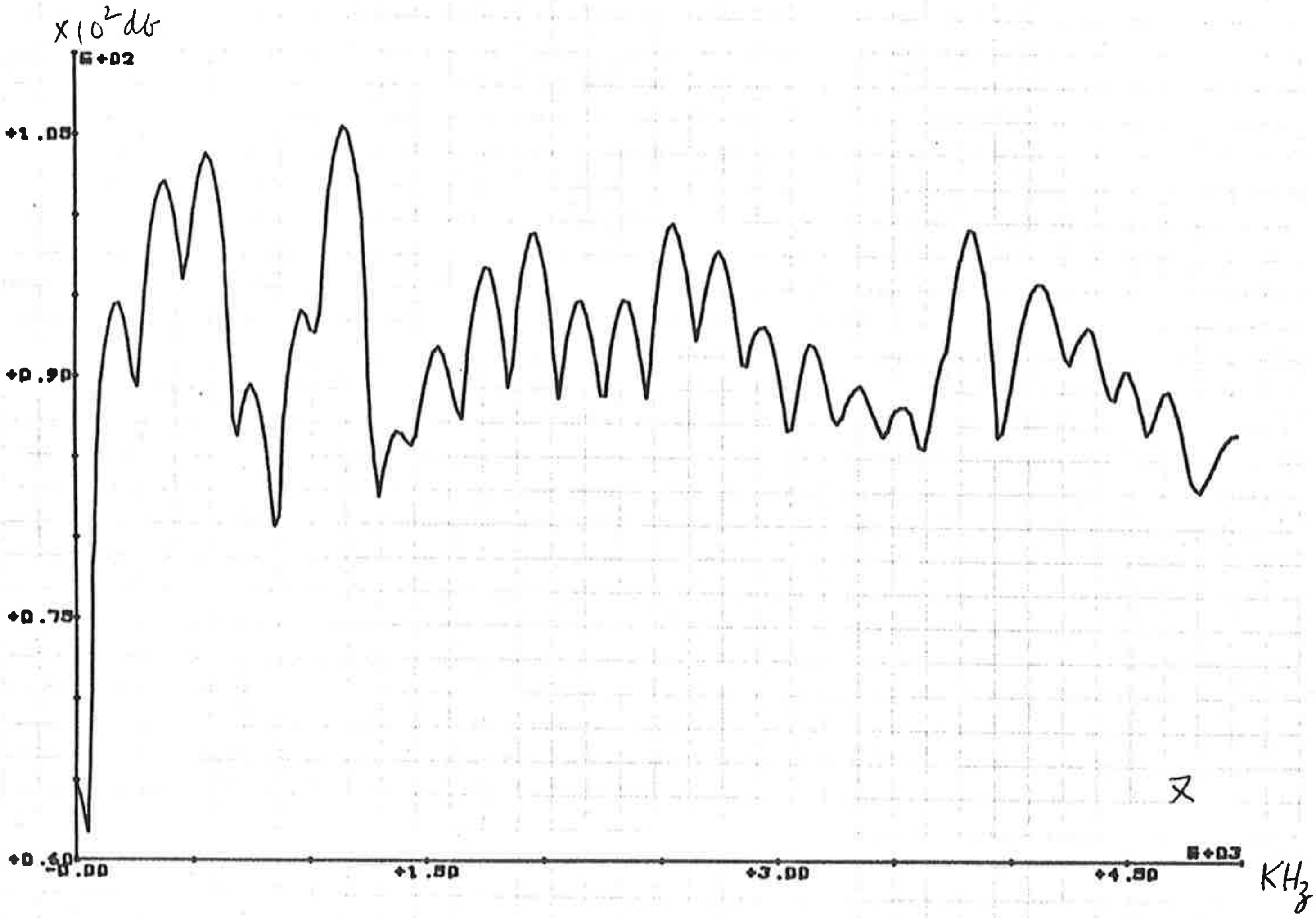

x10² db
E+02
+1.00
+0.88
+0.76
+0.64
-0.00
+1.50
+3.00
+4.50
E+03
KHz
さ

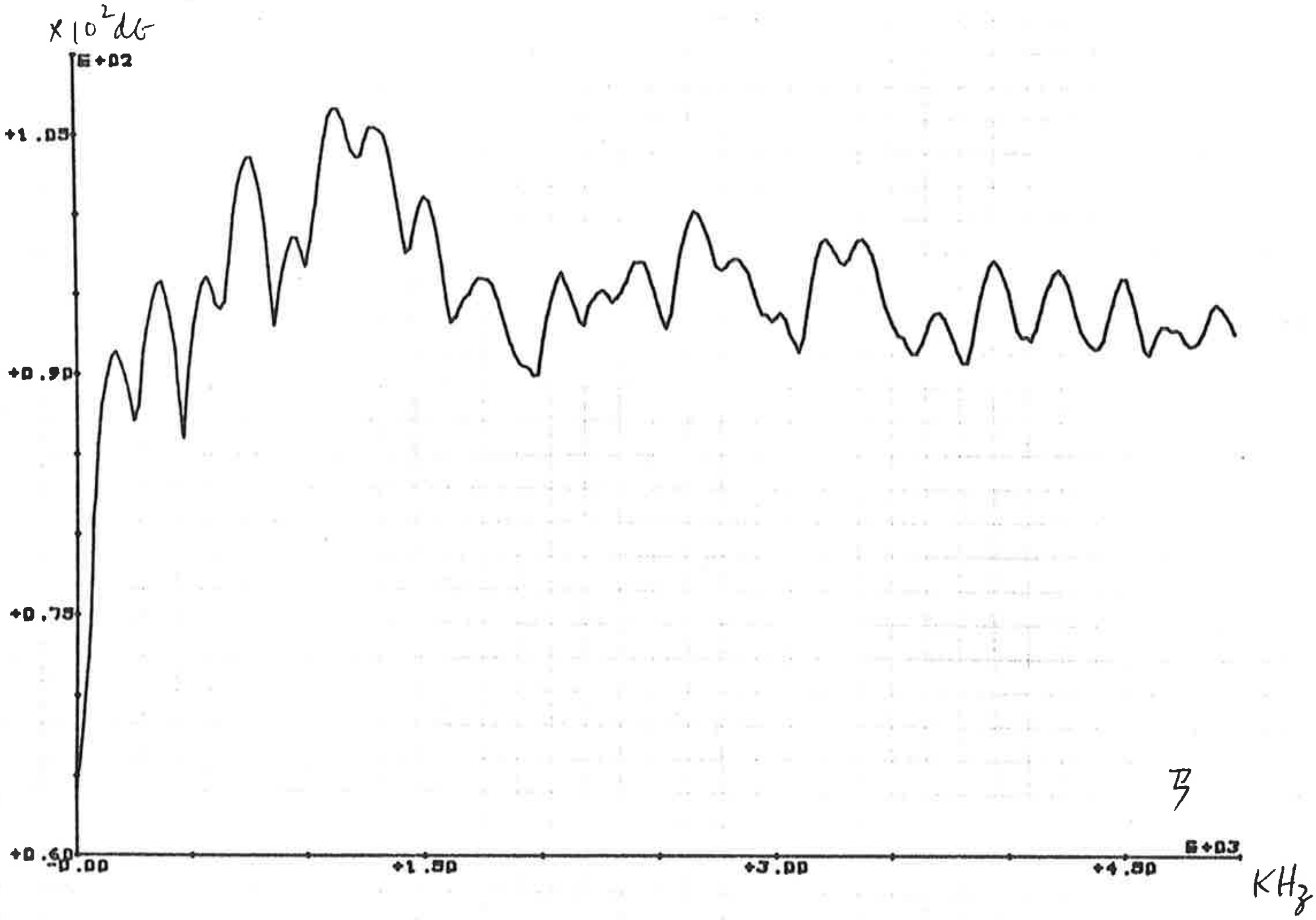

x10² db
E+02
+1.00
+0.85
+0.70
+0.55
-0.00
+1.50
+3.00
+4.50
E+03
KHz
せ

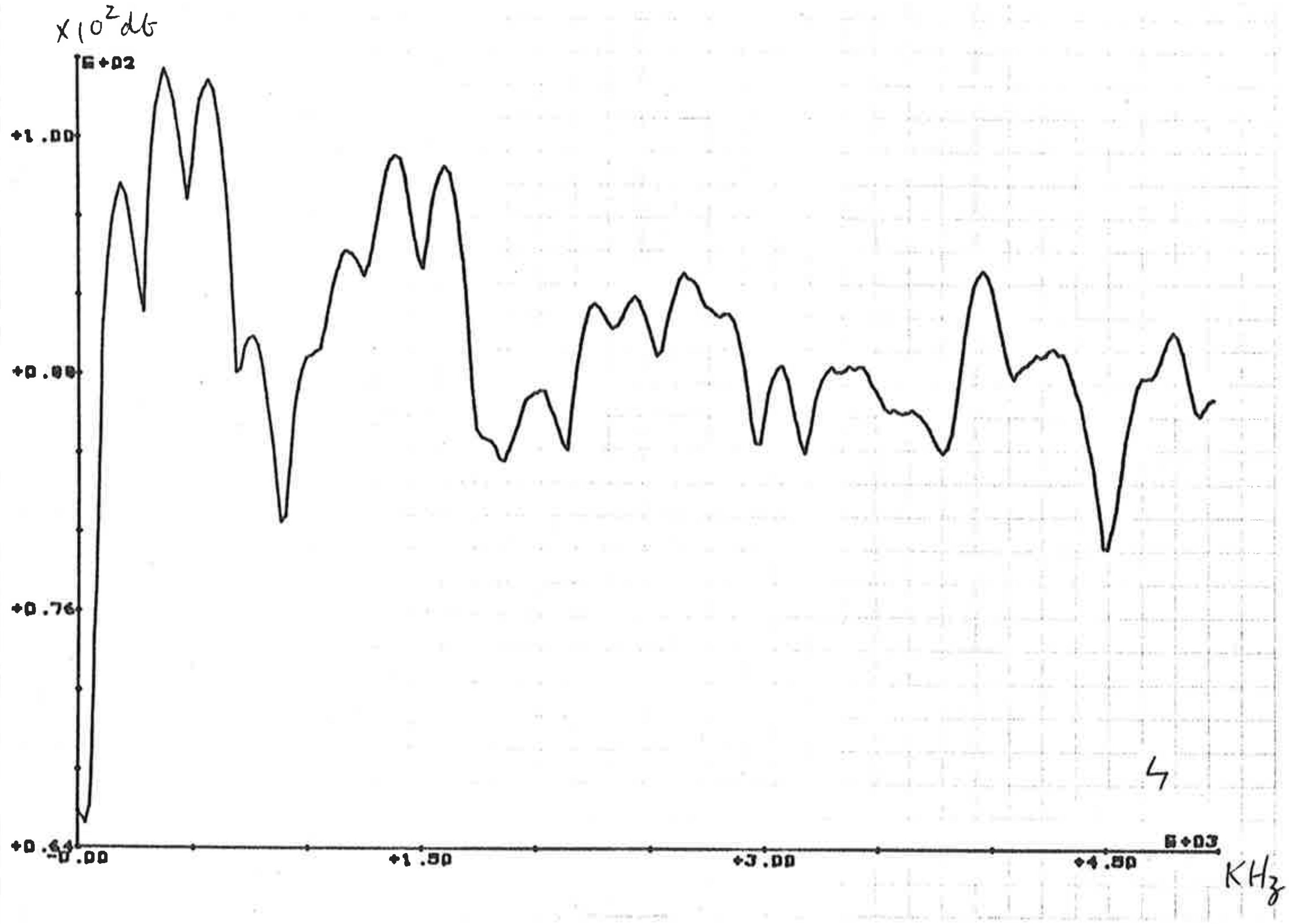

x 10² db
E+02
+1.00
+0.85
+0.70
+0.55
+0.00
+1.50
+3.00
+4.50
E+03
KHz
五

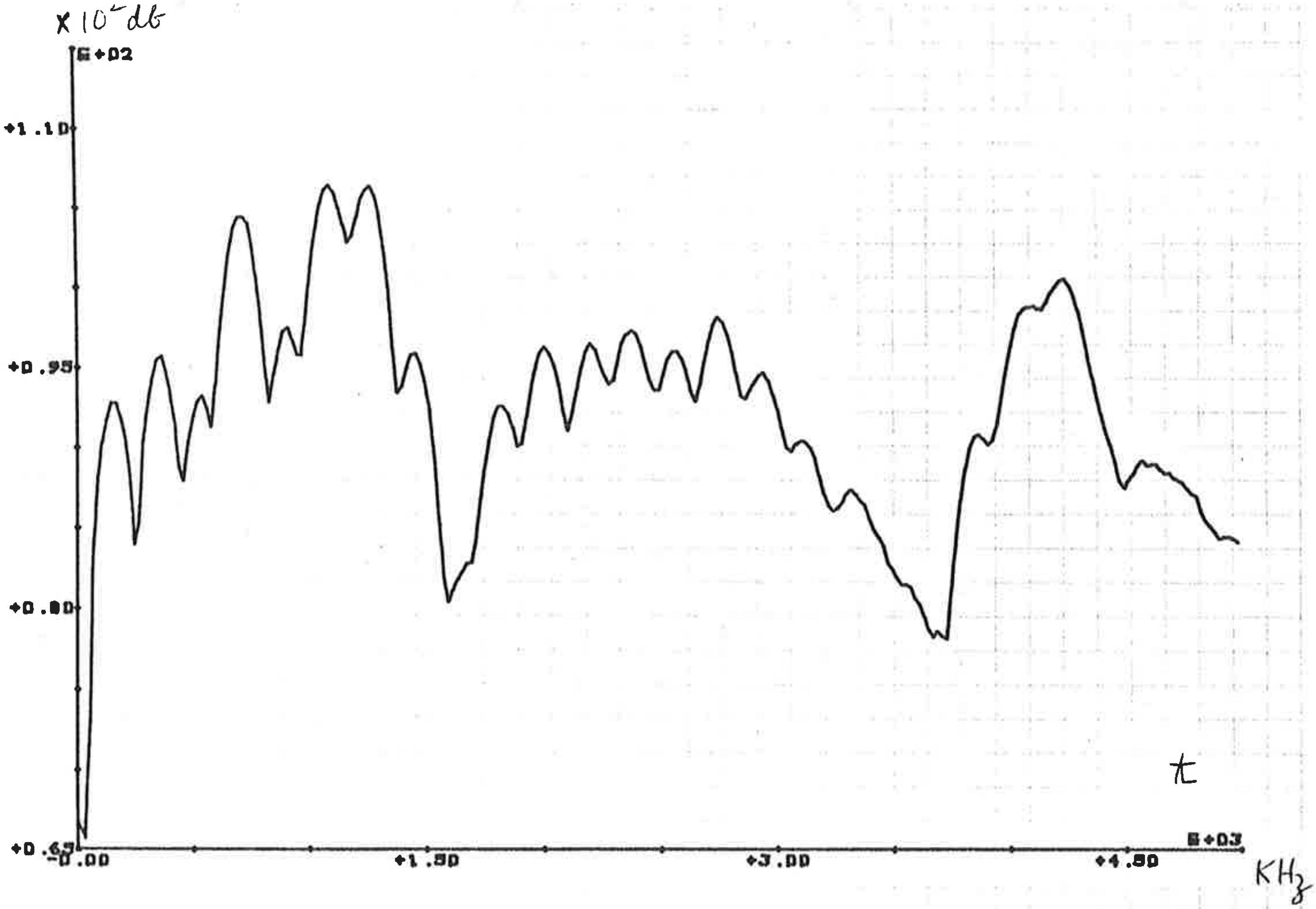

x 10² db
E+02
+1.05
+0.90
+0.75
+0.60
+0.00
+1.50
+3.00
+4.50
E+03
KHz

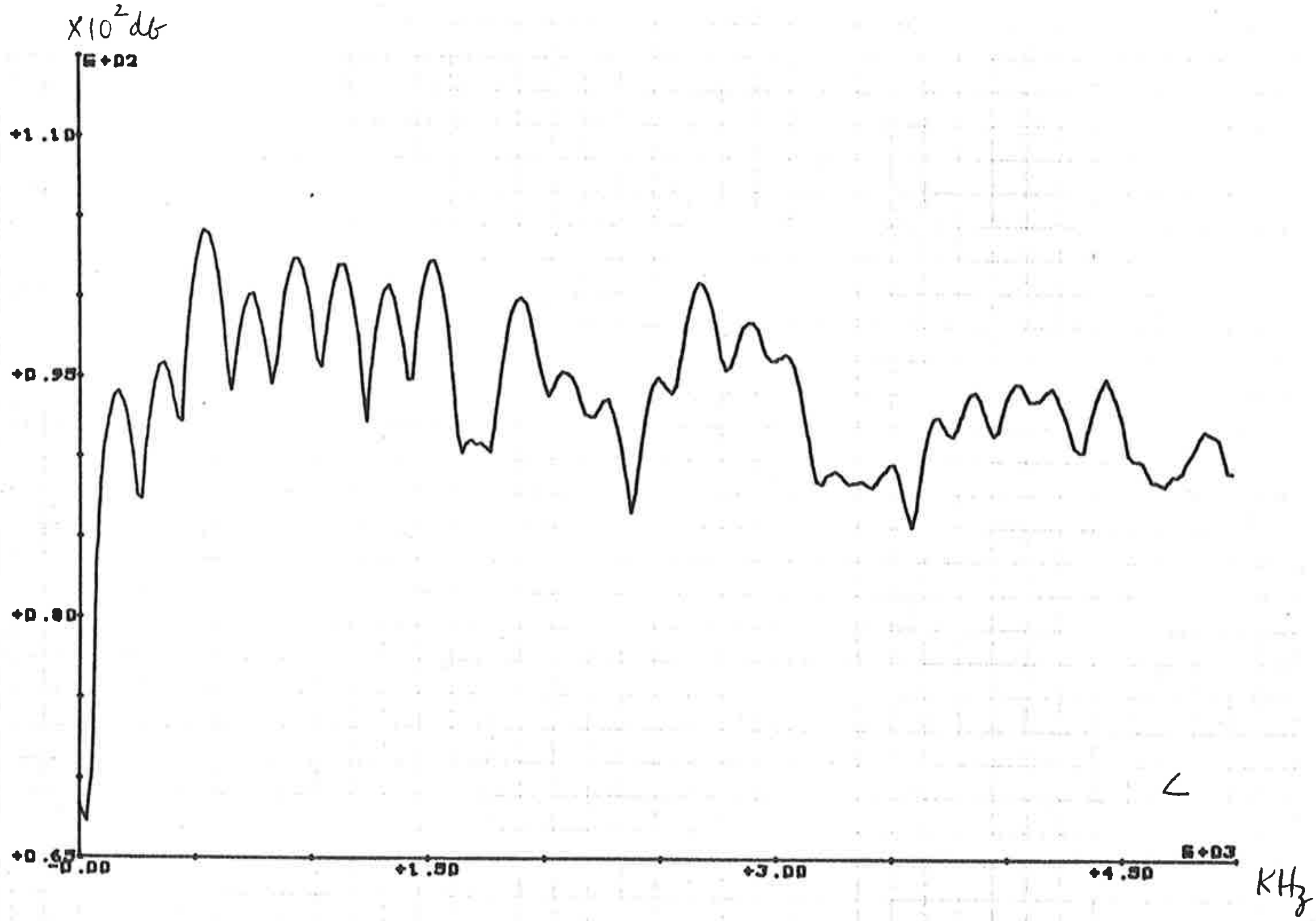

x10²db
E+02
+1.00
+0.88
+0.76
+0.64
+0.00
+1.50
+3.00
+4.50
E+03
KHz

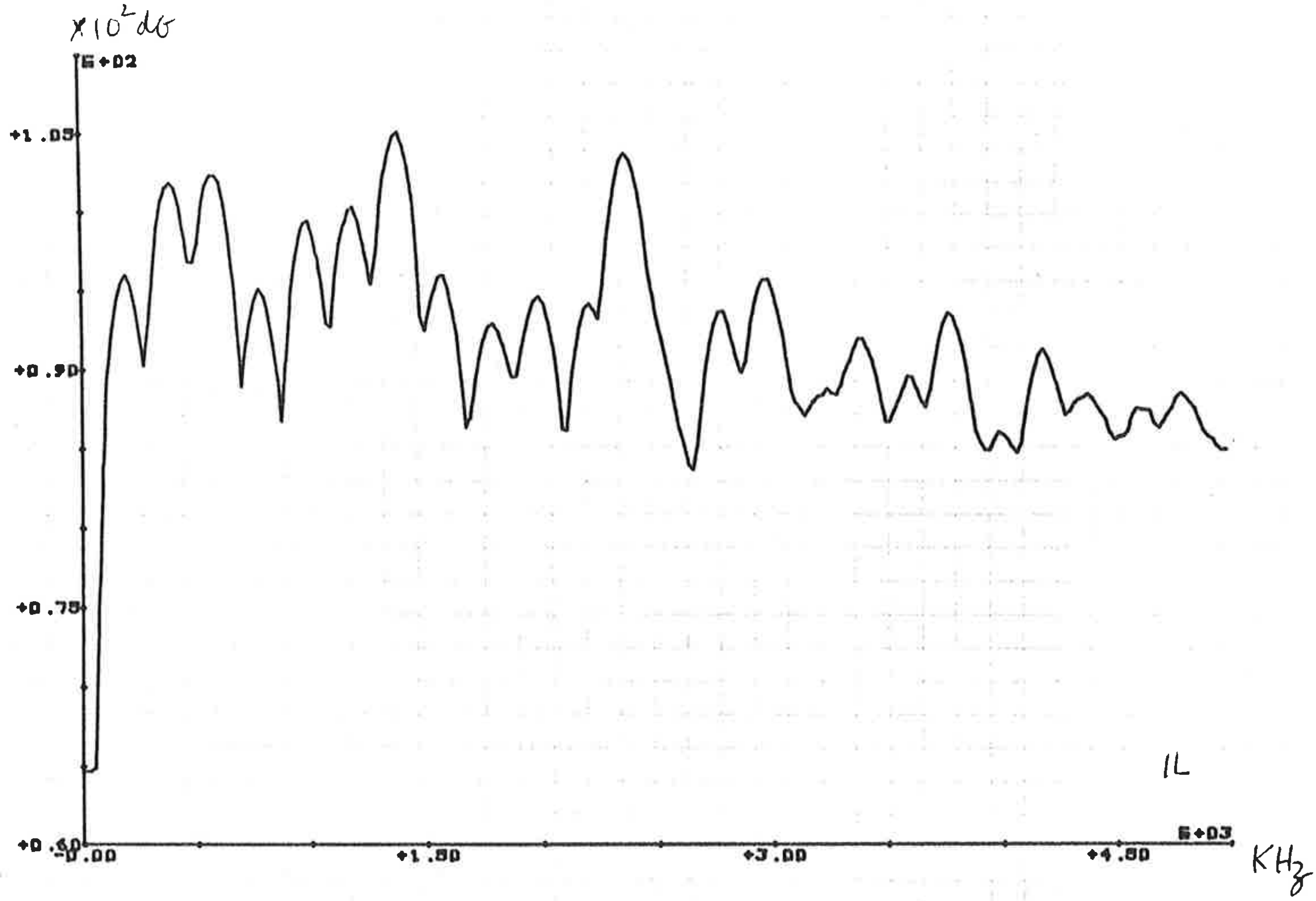

x10²db
E+02
+1.05
+0.90
+0.75
+0.60
+0.00
+1.50
+3.00
+4.50
E+03
KHz

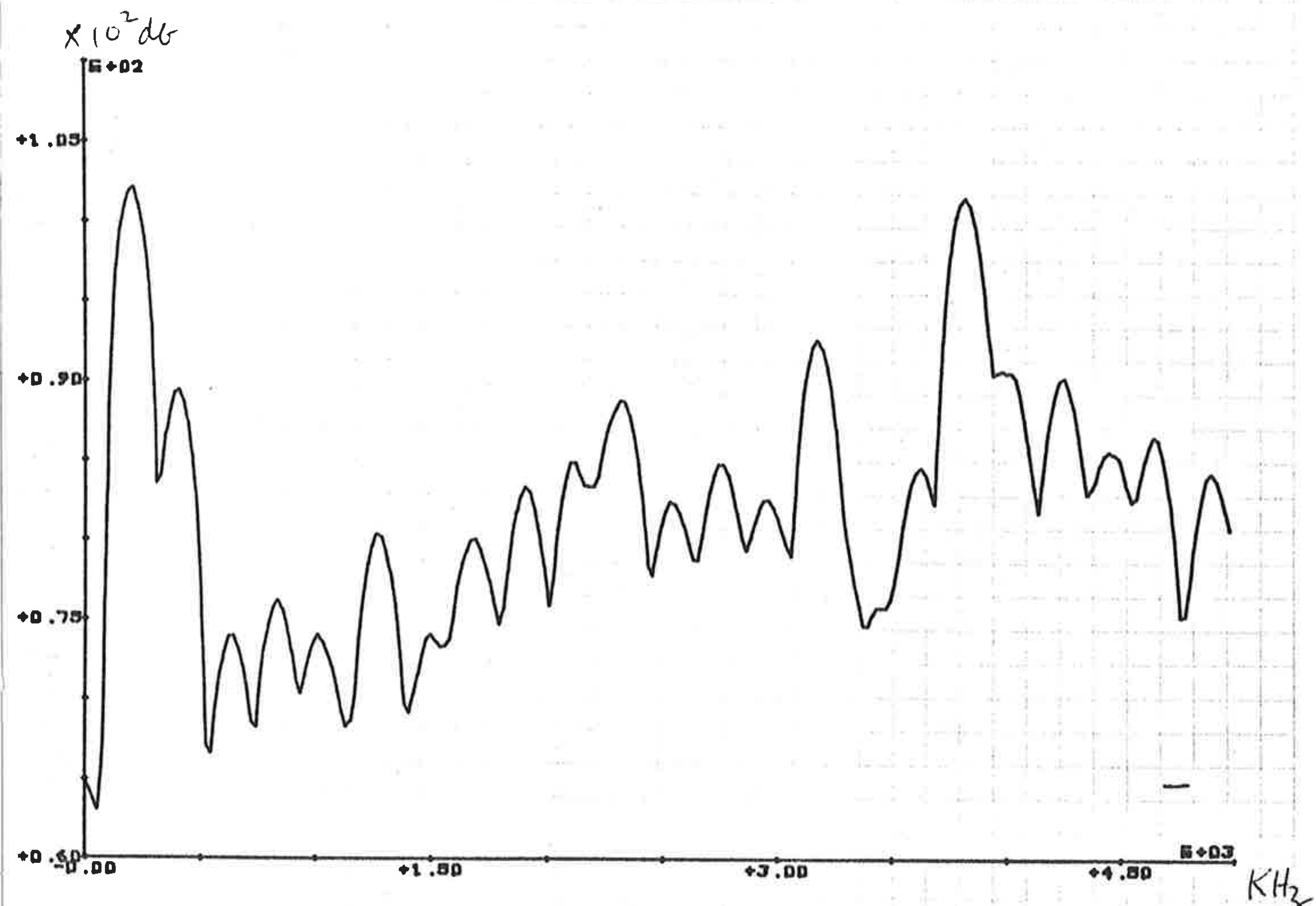

x10² dB
E+02
+1.05
+0.90
+0.75
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz
3

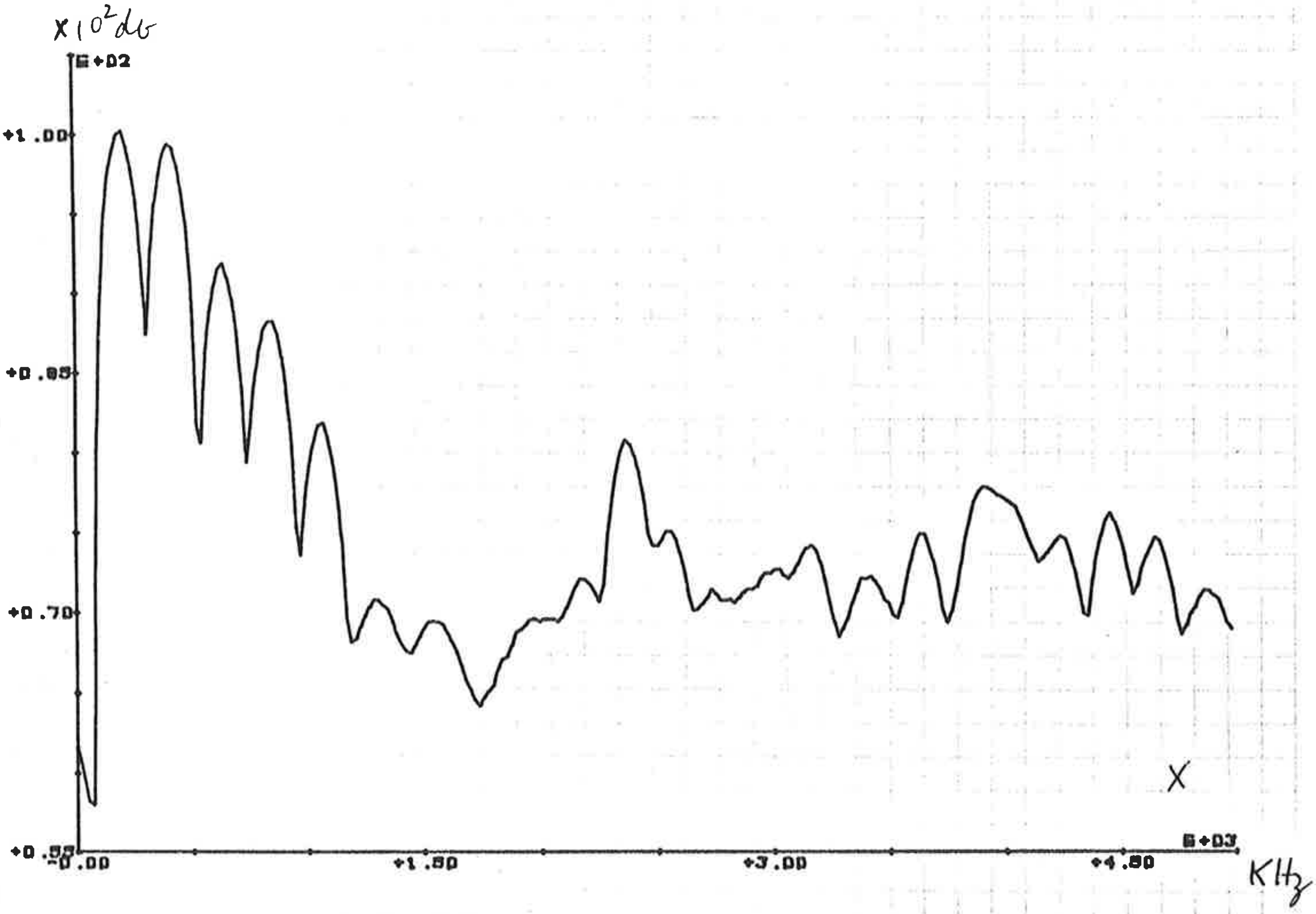

x10² dB
E+02
+1.00
+0.88
+0.76
+0.64
-0.00
+1.50
+3.00
+4.50
E+03
KHz
4

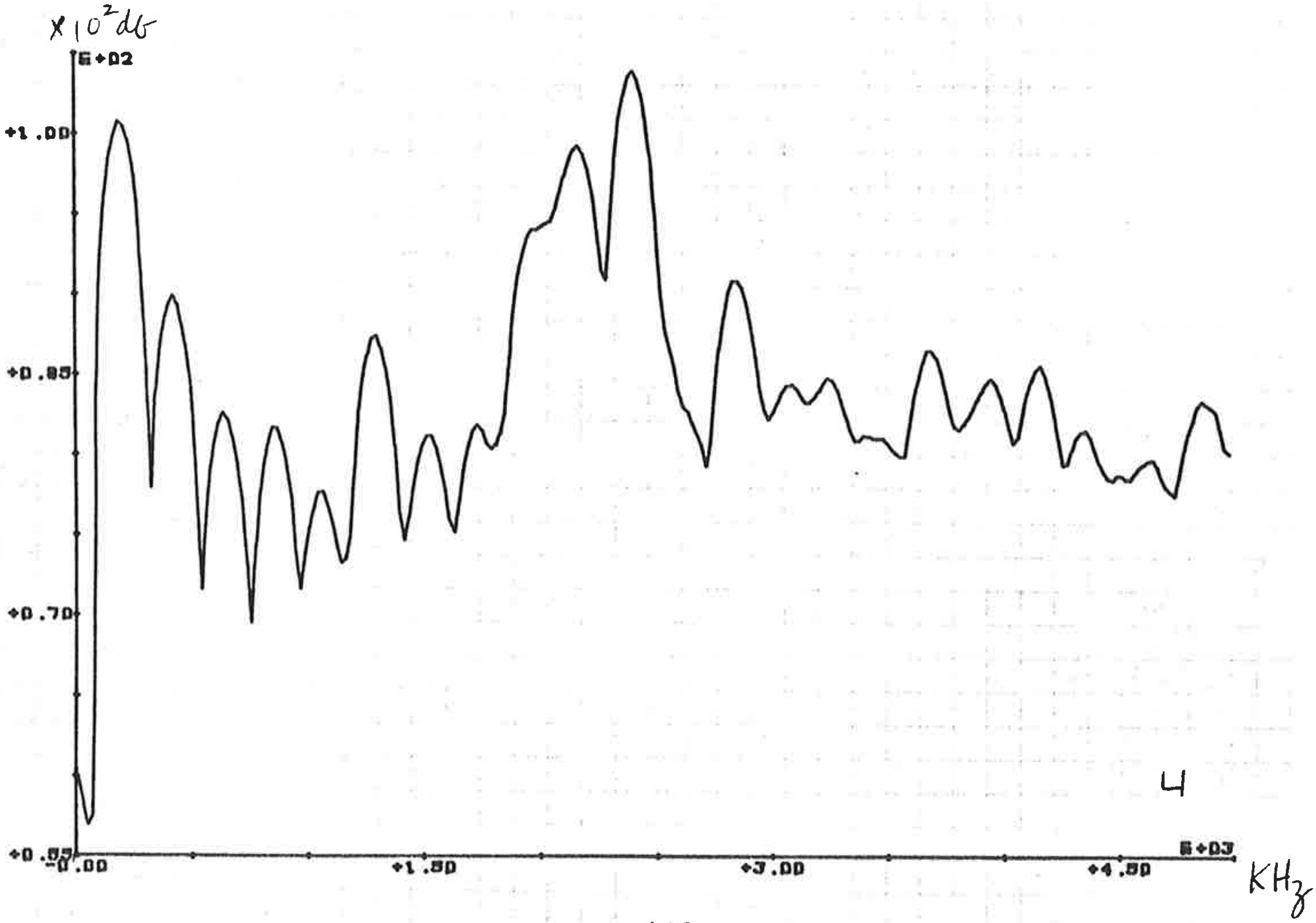

X10² db
E+02
+1.10
+0.95
+0.80
+0.65
+0.00
+1.50
+3.00
+4.50
E+03
KHz


X10² db
E+02
+1.10
+0.95
+0.80
+0.65
+0.00
+1.50
+3.00
+4.50
E+03
KHz


x10² db
E+02
+1.05
+0.90
+0.75
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz
IL


x10² db
E+02
+1.05
+0.90
+0.75
+0.60
-0.00
+1.50
+3.00
+4.50
E+03
KHz


x10^2 db
E+02
+1.00
+0.85
+0.70
+0.55
-0.00
+1.50
+3.00
+4.50
E+03
KHz
X


x10^2 db
E+02
+1.00
+0.85
+0.70
+0.55
-0.00
+1.50
+3.00
+4.50
E+03
KHz
4

APPENDIX 4

Spectrograms- time vs. frequency

18 pages

Sampling frequency	10 KHz
Overlap factor (i.e. proportional overlap number of plots in one segment)	0
Overflow allowance (i.e. numerical overflow to be permitted)	10
Differencing (i.e. to remove nett 20 db/decade droop causing by vocal organ)	1
Fourier transform to be normalized (i.e. Fourier transforms were not normalized)	0

Standard hamming window function:

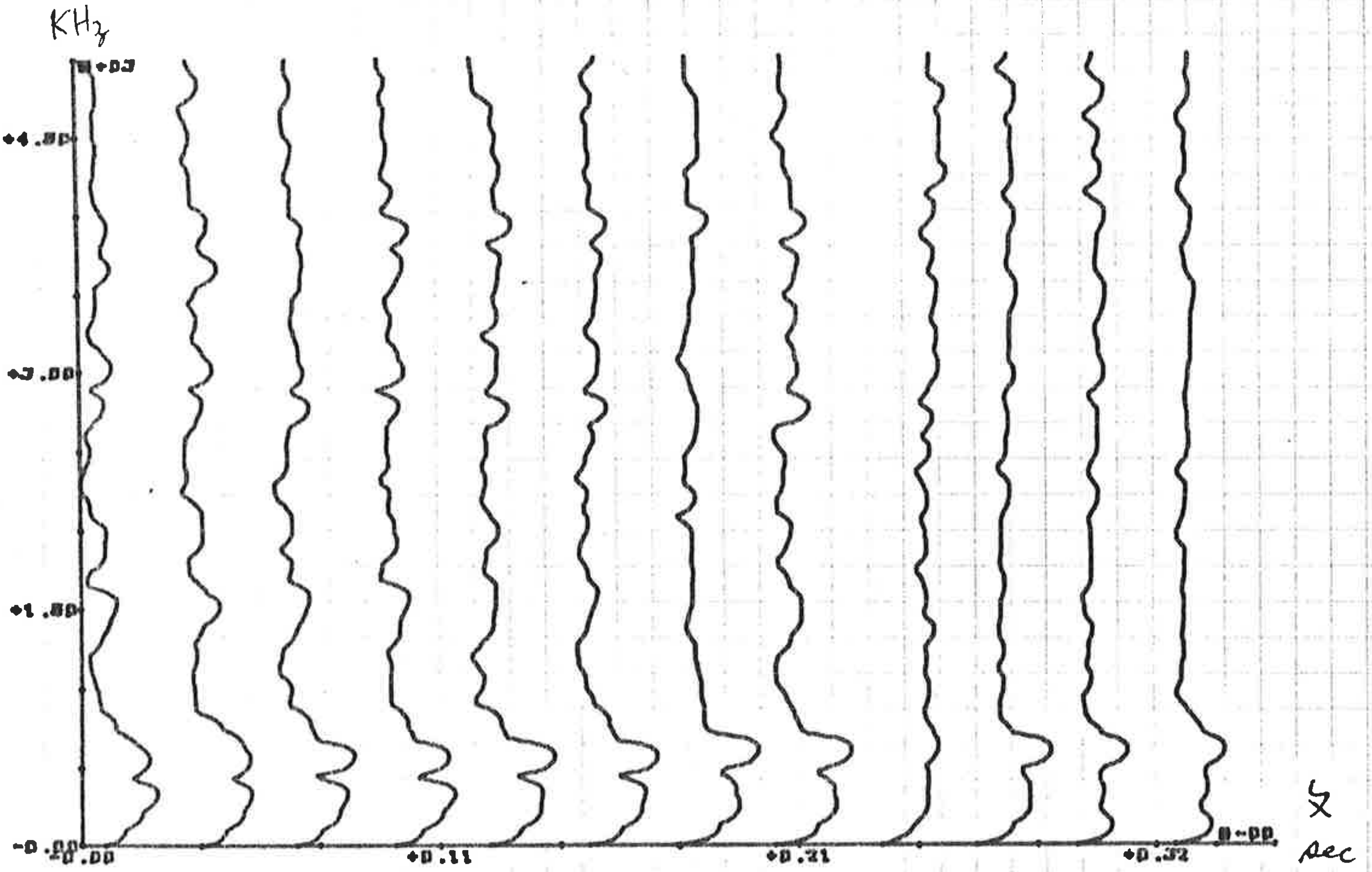
$$0.54-0.46\mathrm{Cos}(2\pi(J-1)/(M-1))$$

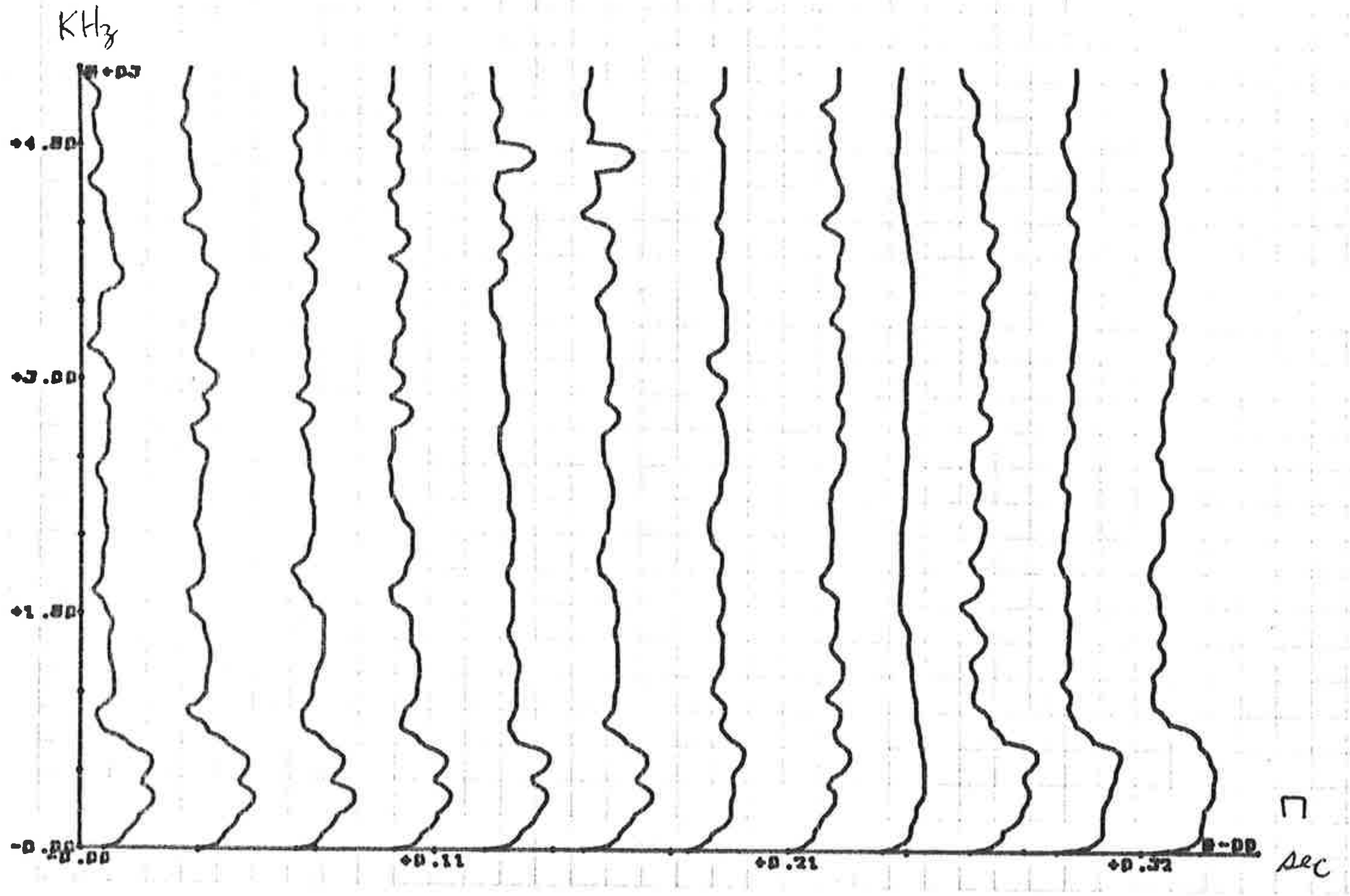
where J: sample index

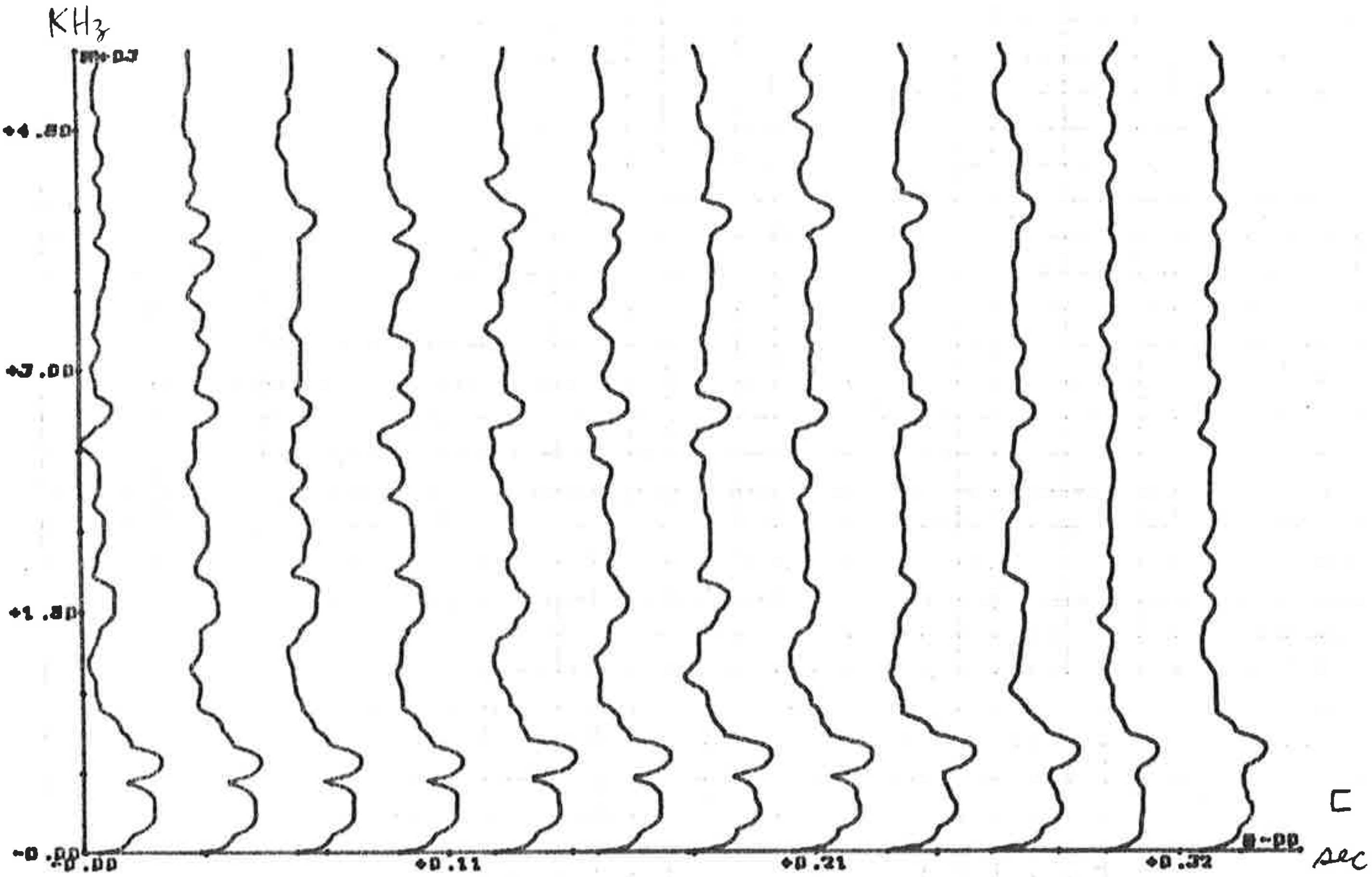
M: window length(512 points)

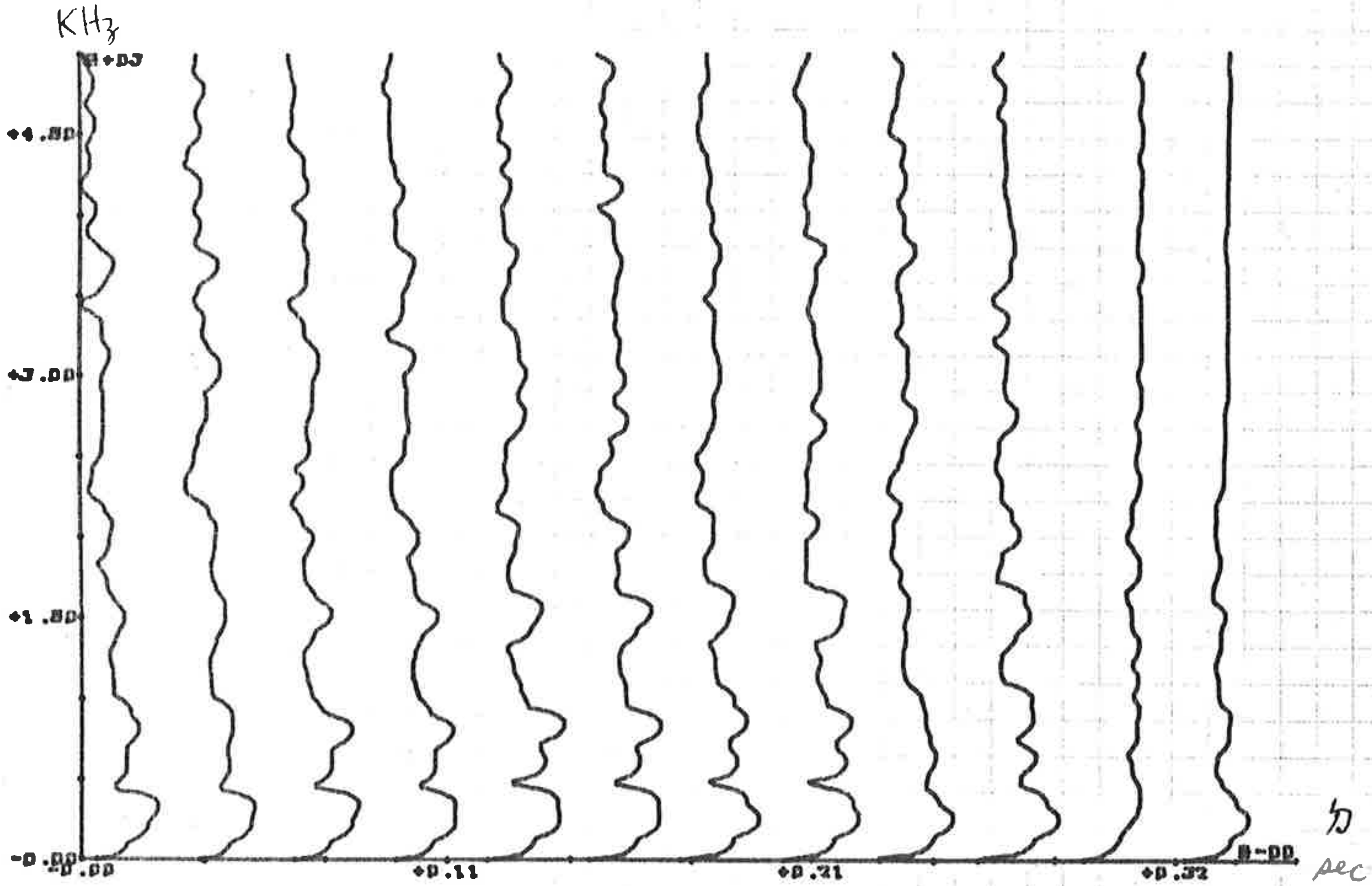
Successive windows overlap 50% in analysis

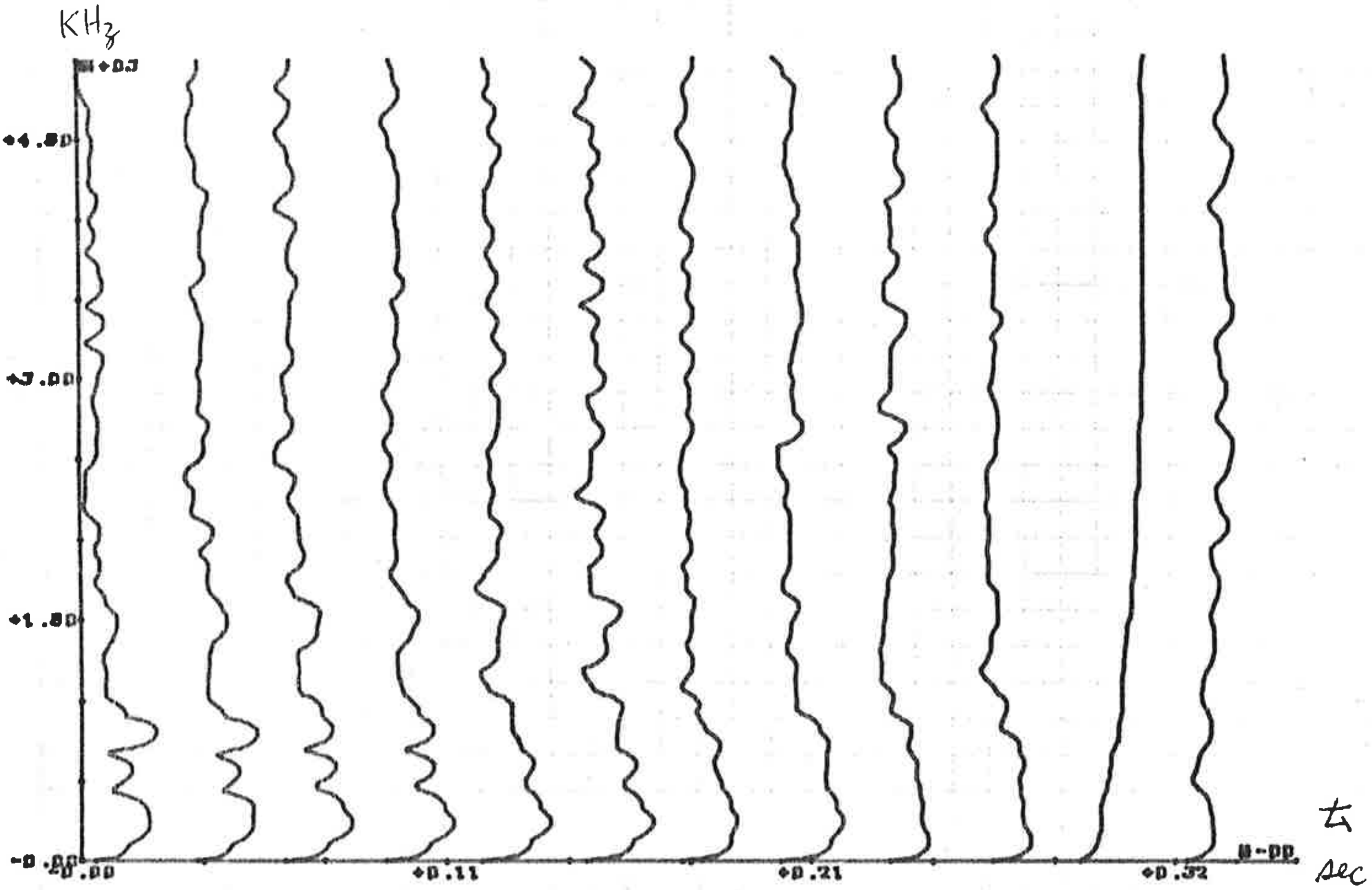
If the display is more than one spectrum, it is necessary to set up axes. Number of sections needed for horizontal axis is the number sum of spectra and overlap factor. It is obvious in the spectrograms.

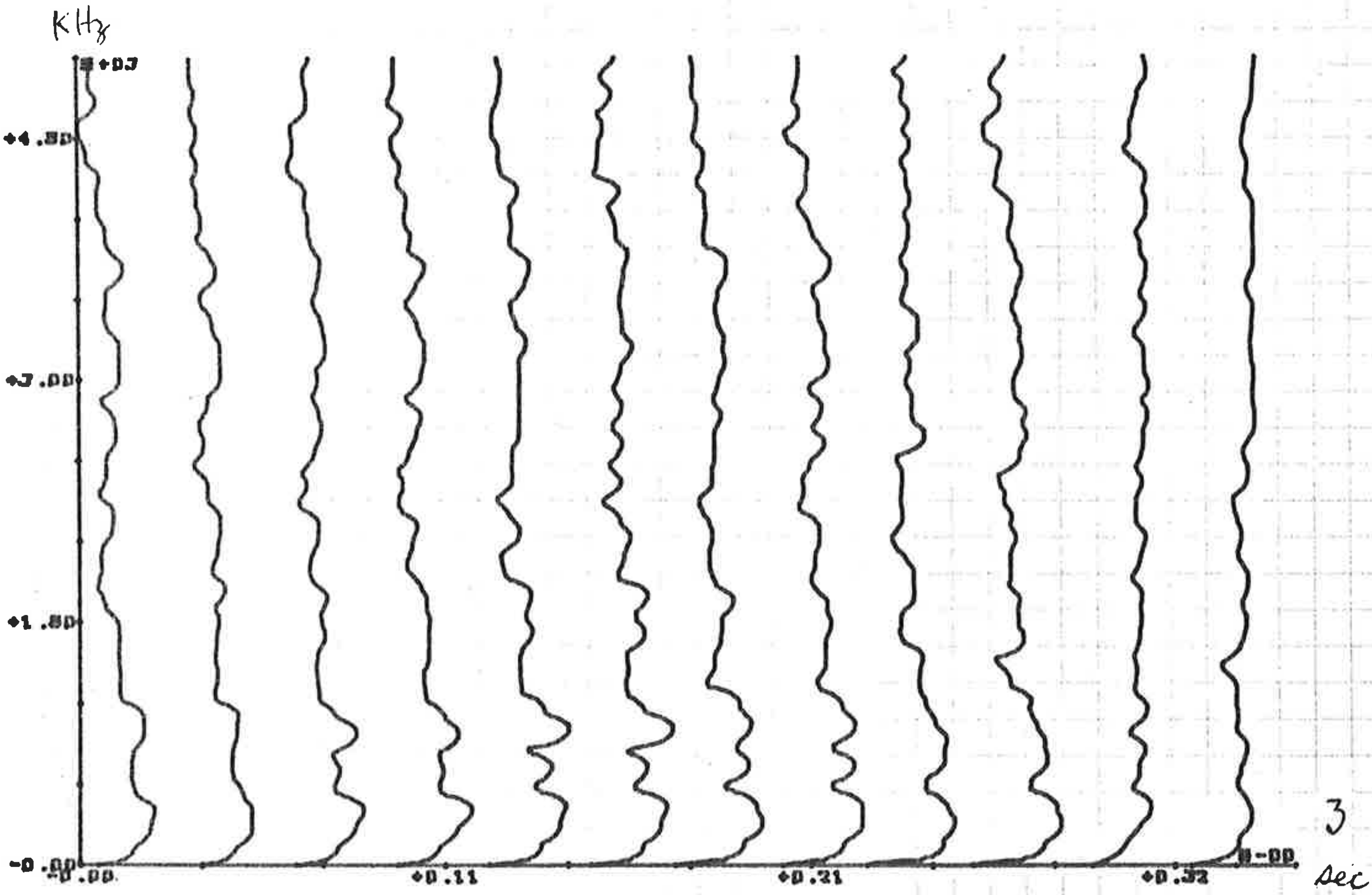

KHz
+4.00
+3.00
+1.00
-0.00
+0.11
+0.21
+0.32
sec

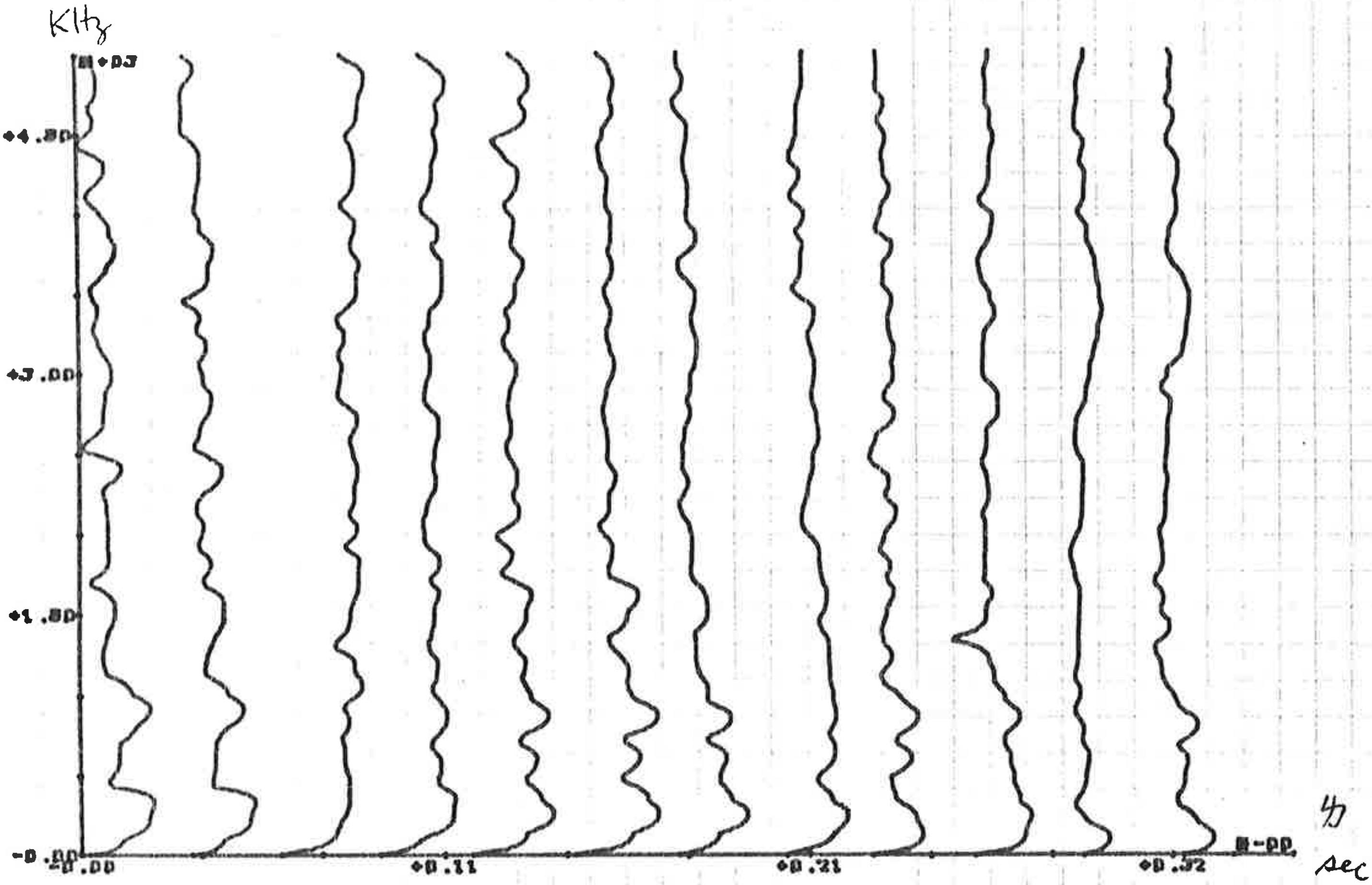

KHz
+4.00
+3.00
+1.00
-0.00
+0.11
+0.21
+0.32
sec

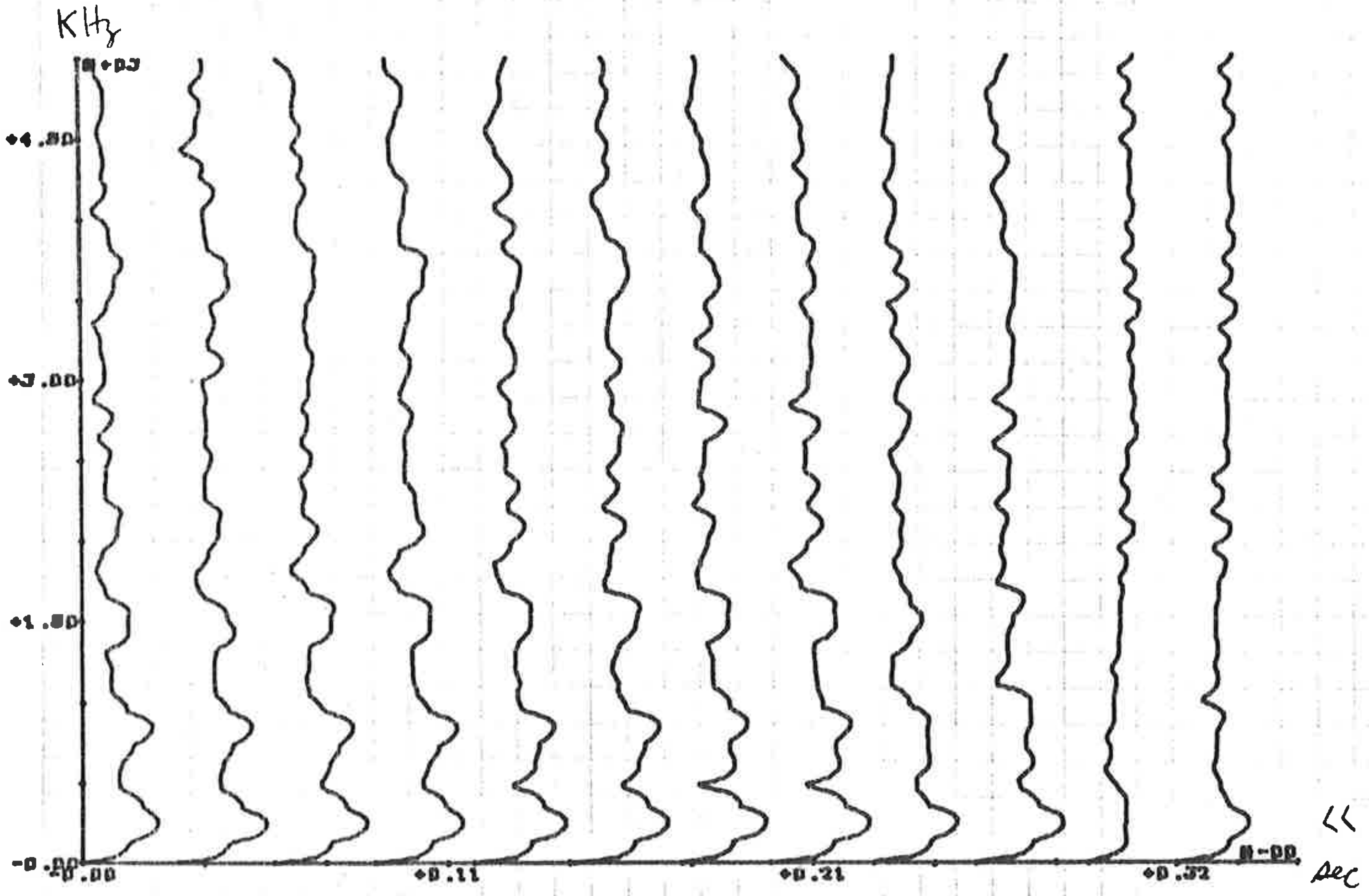

KHz
+4.80
+3.00
+1.80
-0.00
+0.00
+0.11
+0.21
+0.32
C
sec

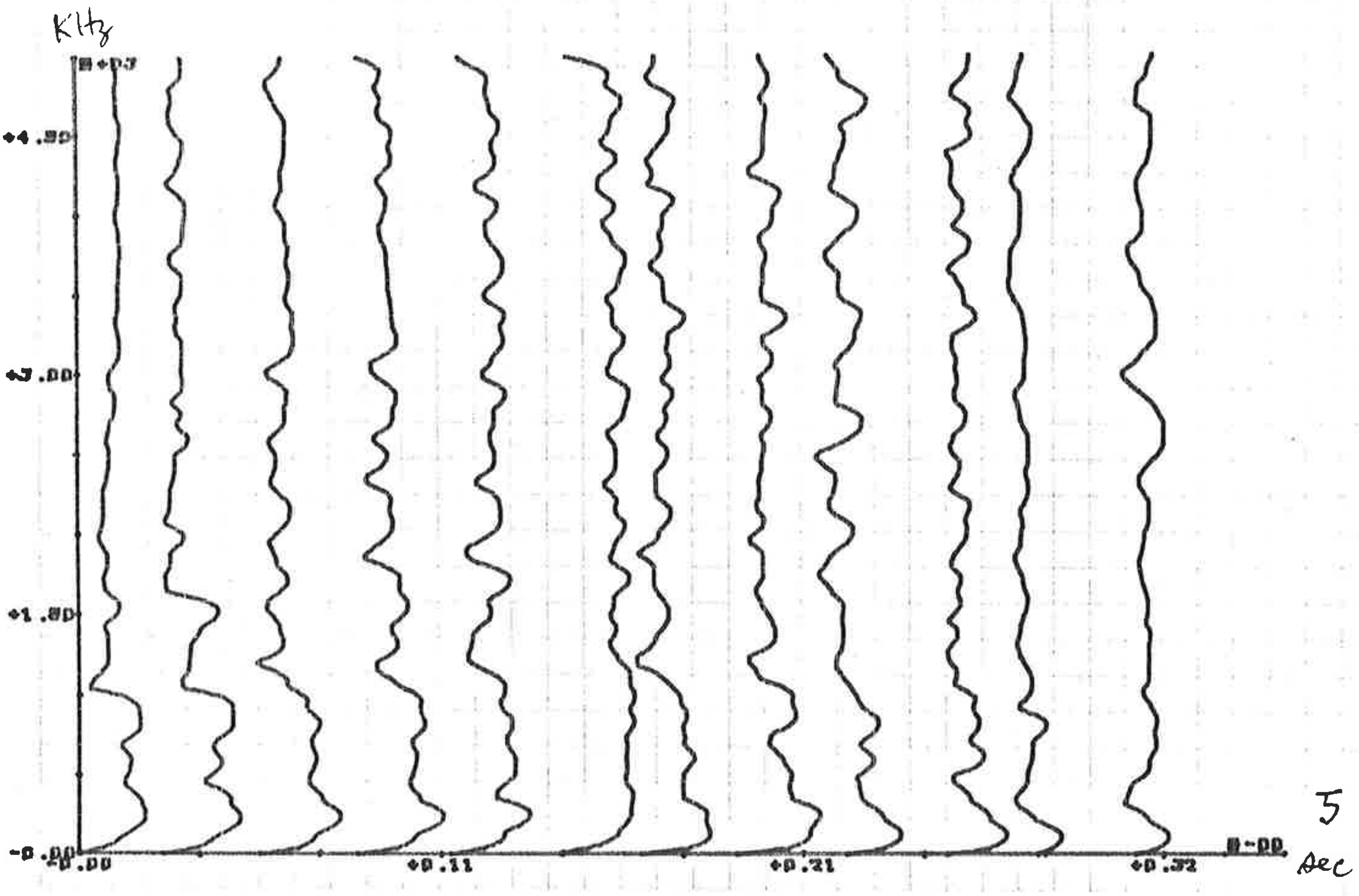

KHz
+4.80
+3.00
+1.80
-0.00
+0.00
+0.11
+0.21
+0.32
sec

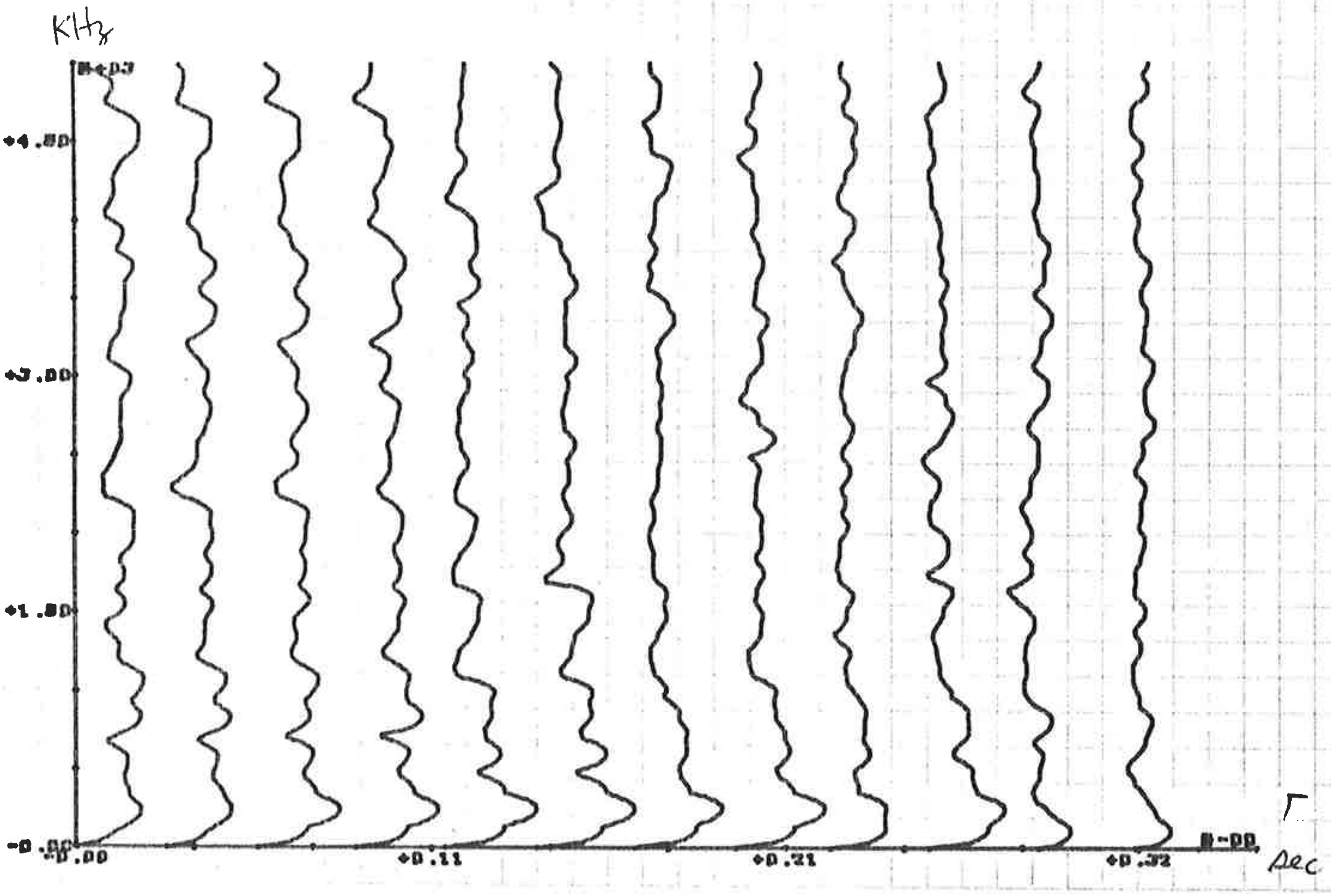

KHz
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
t₁
sec

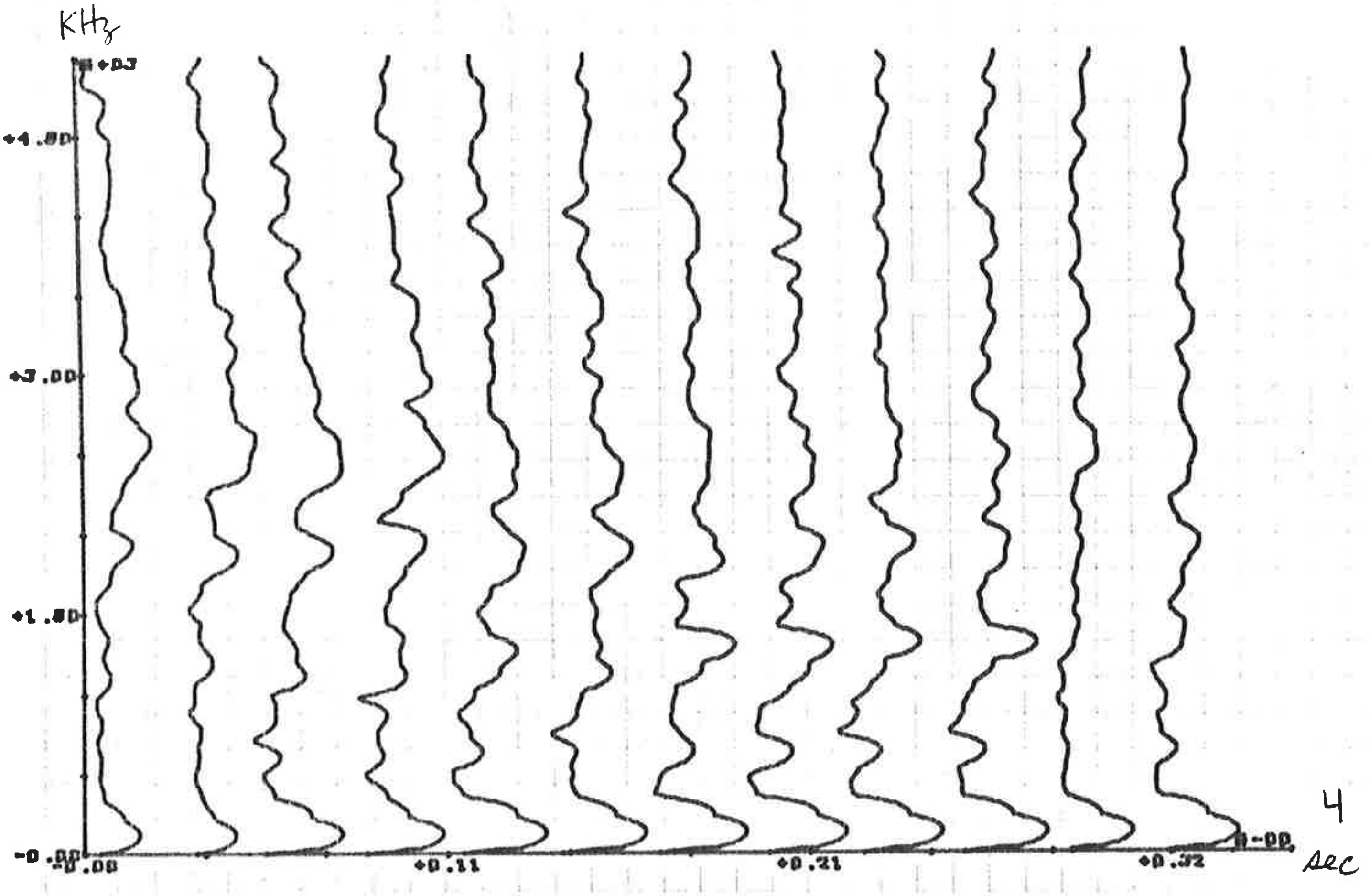

KHz
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
3
sec

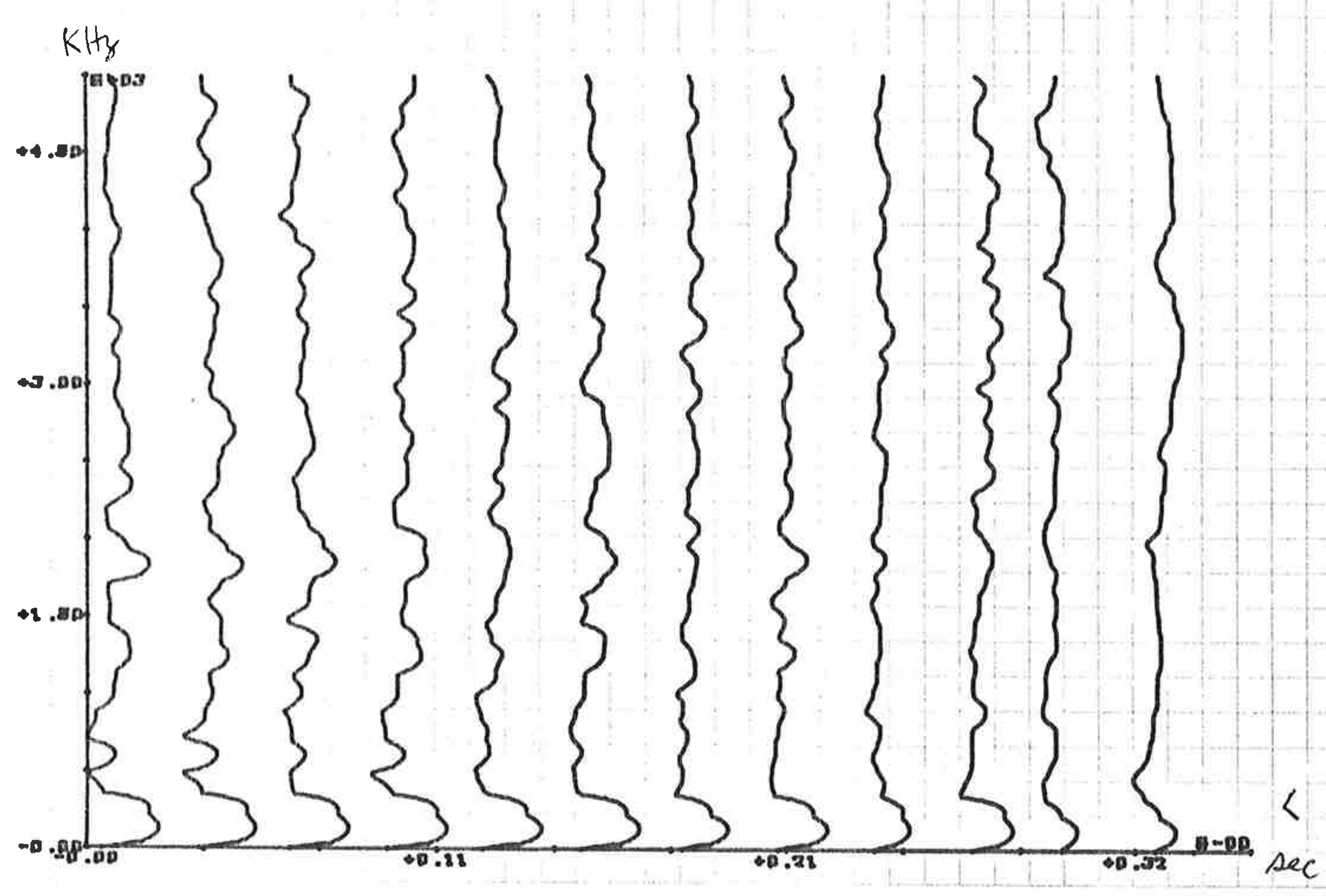

KHz
+4.00
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
sec

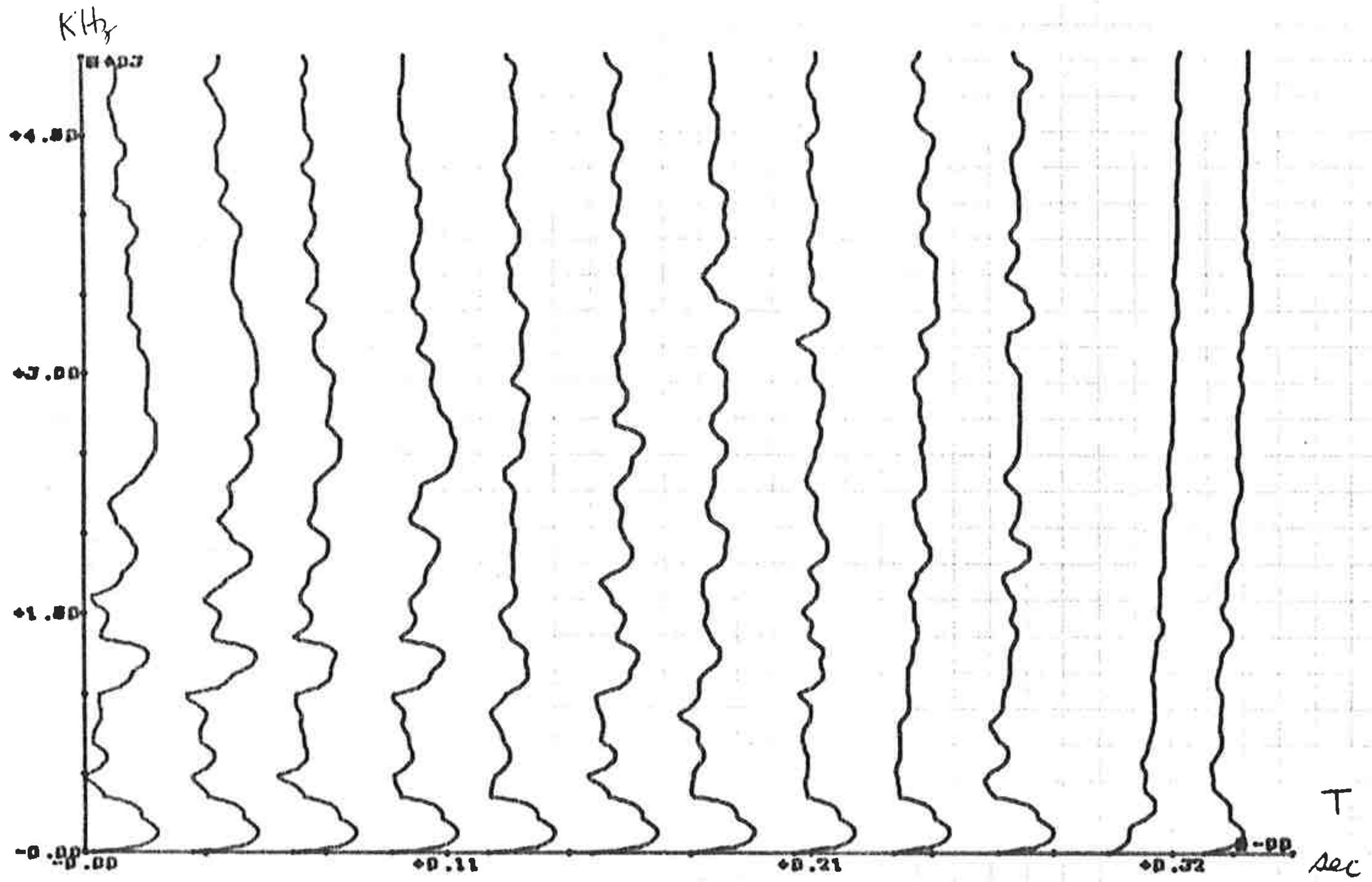

KHz
+4.00
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
sec

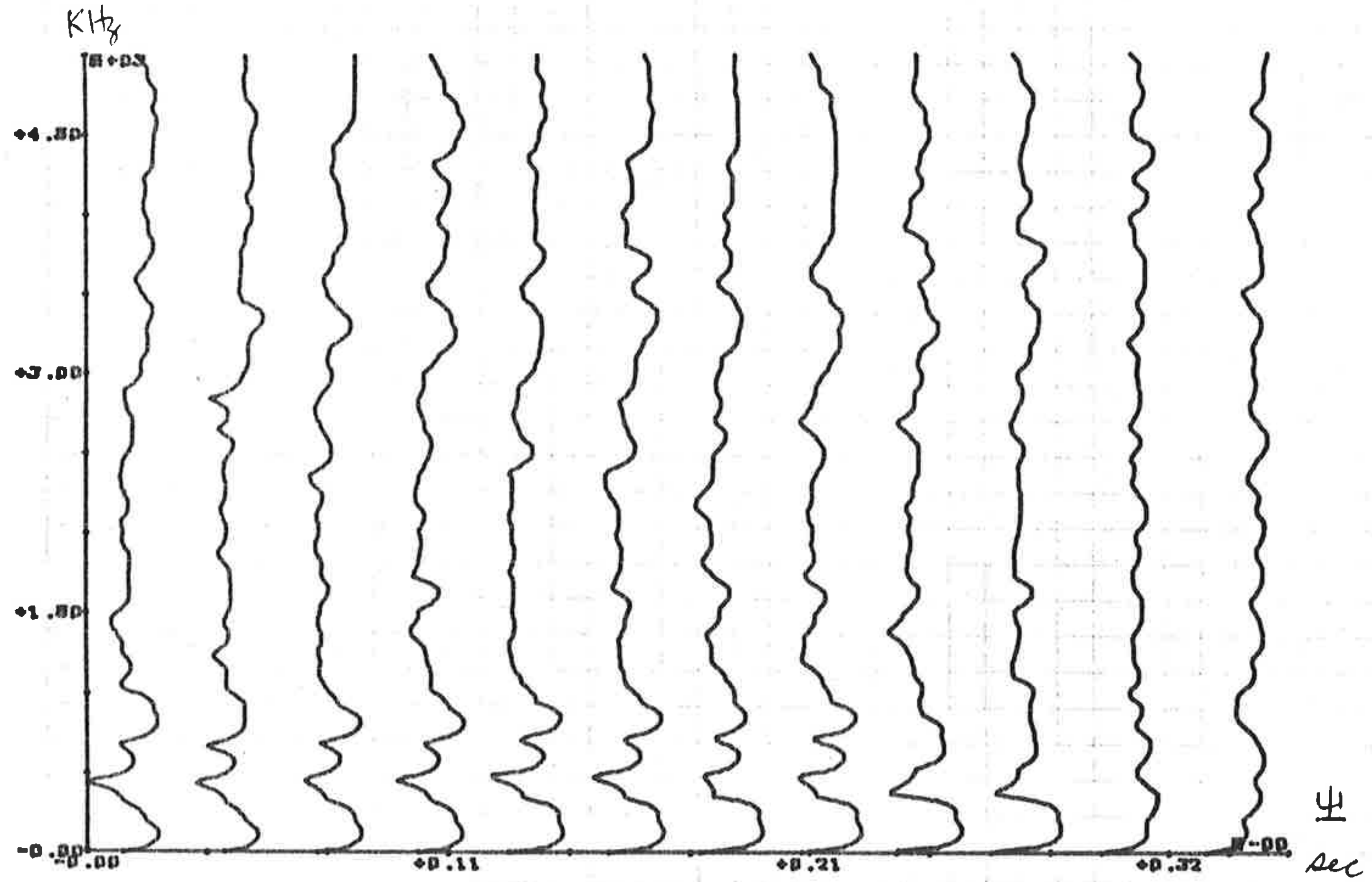

KHz
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
sec
5

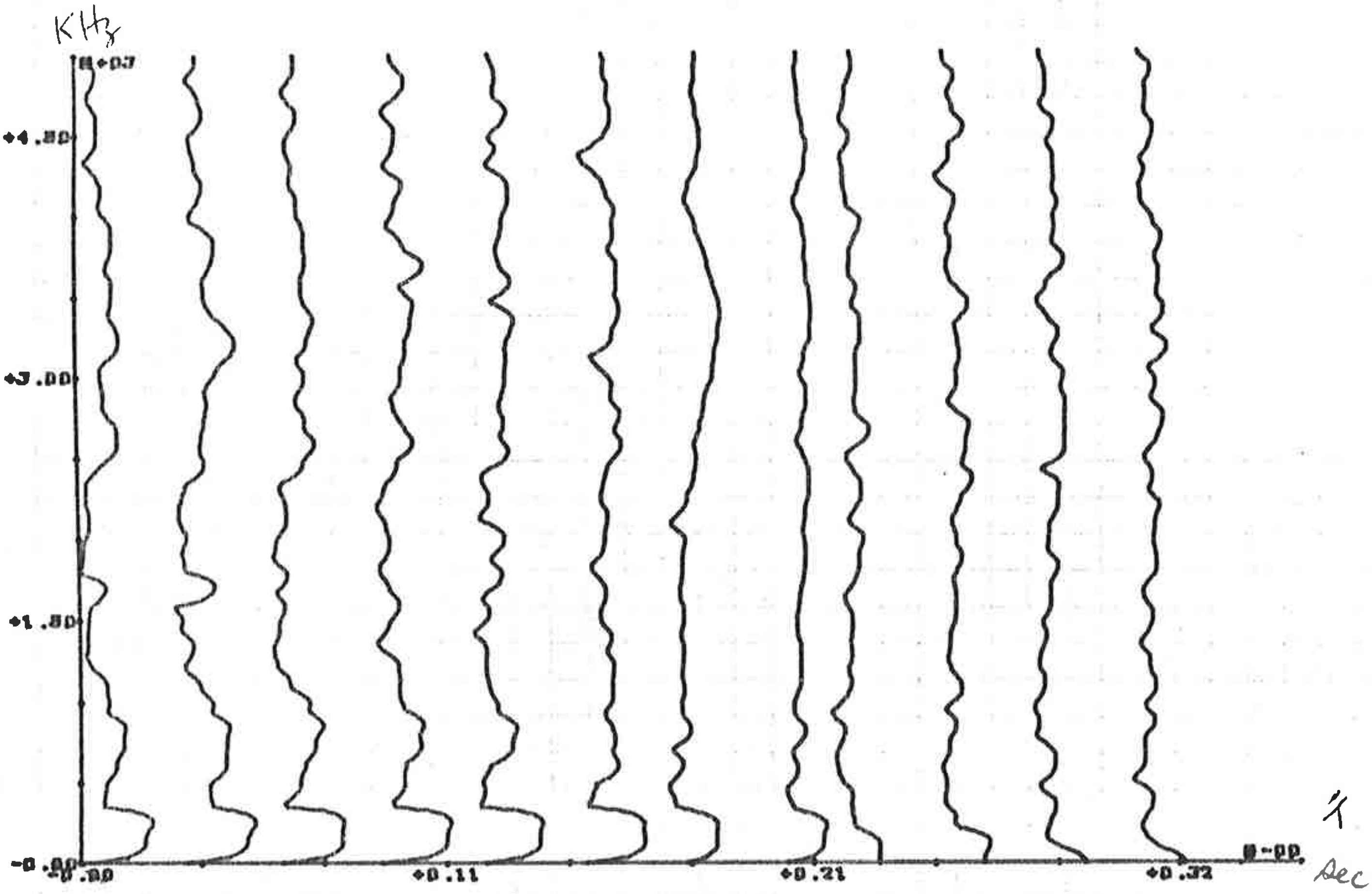

KHz
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
sec
Γ

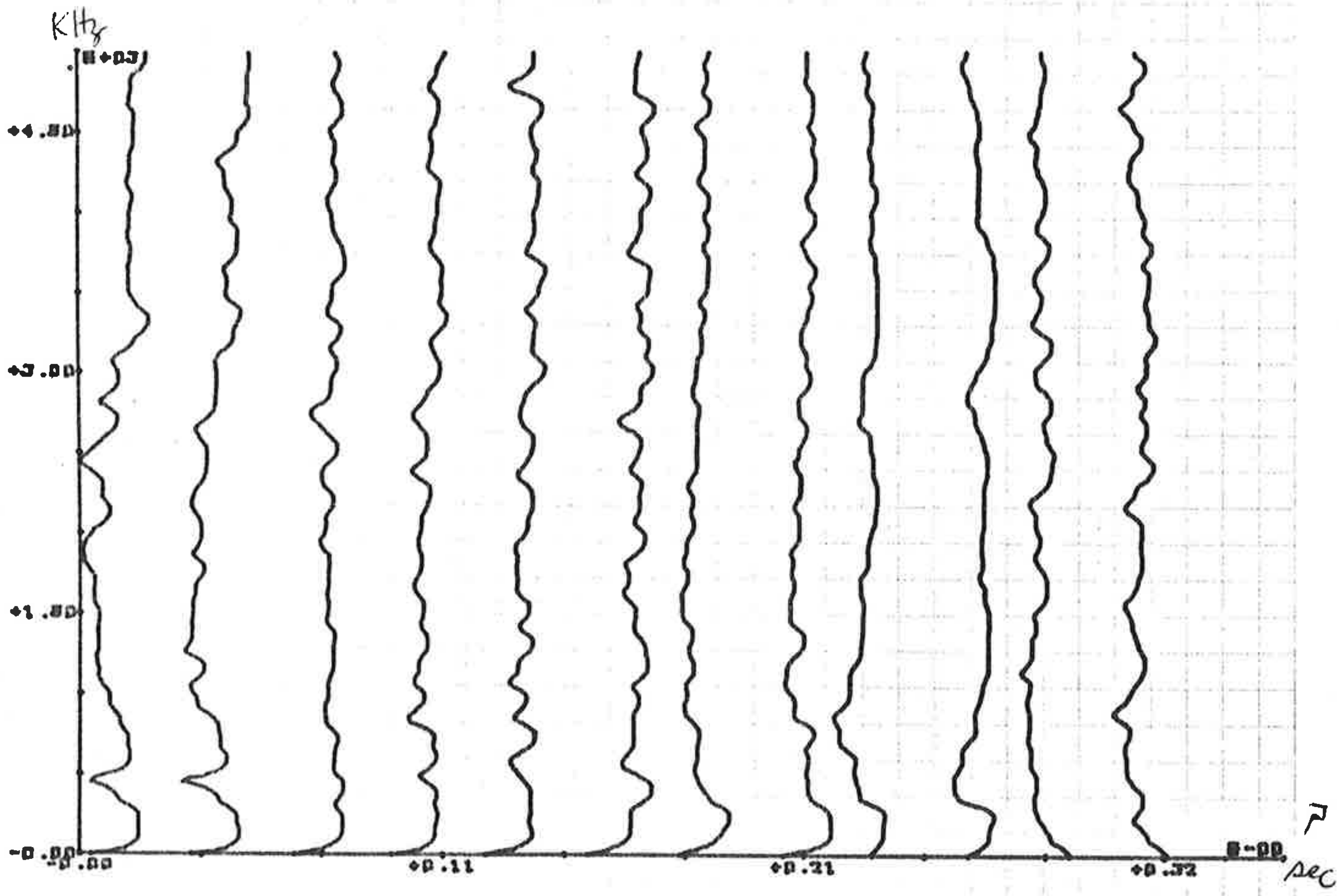

KHz
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
4
sec

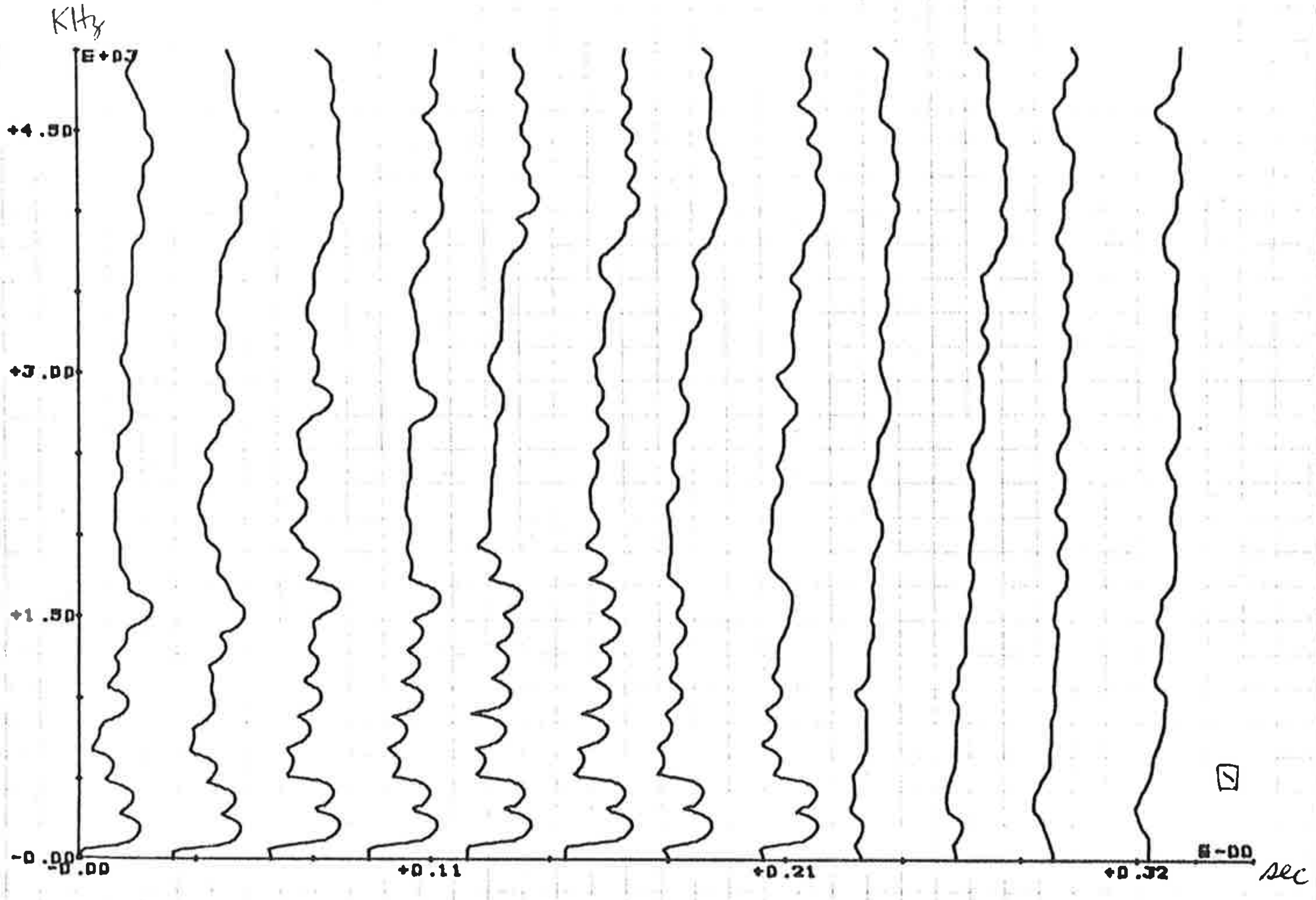

KHz
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
l
sec

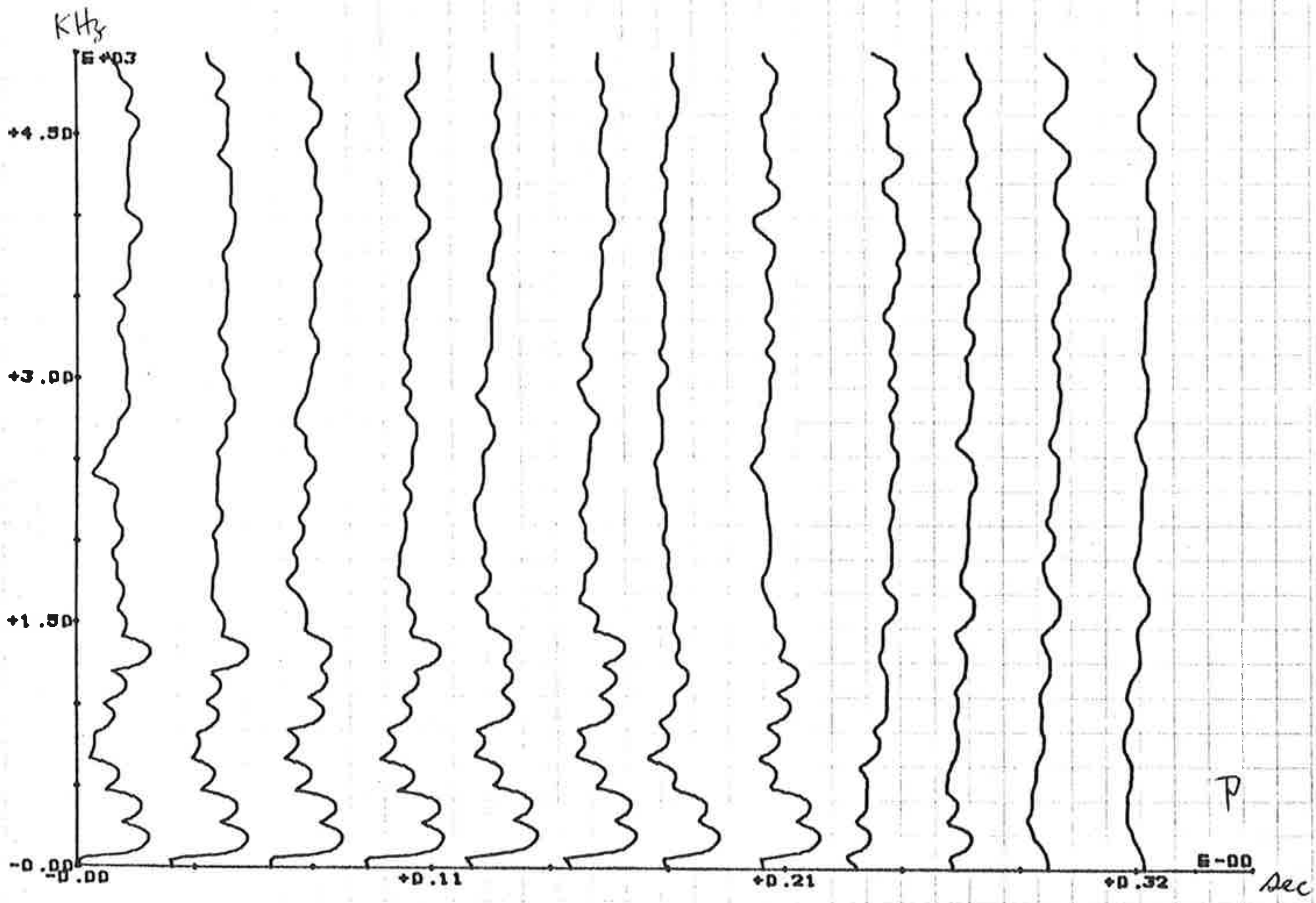

KHz
+4.50
+3.00
+1.50
-0.00
+0.00
+0.11
+0.21
+0.32
T
sec

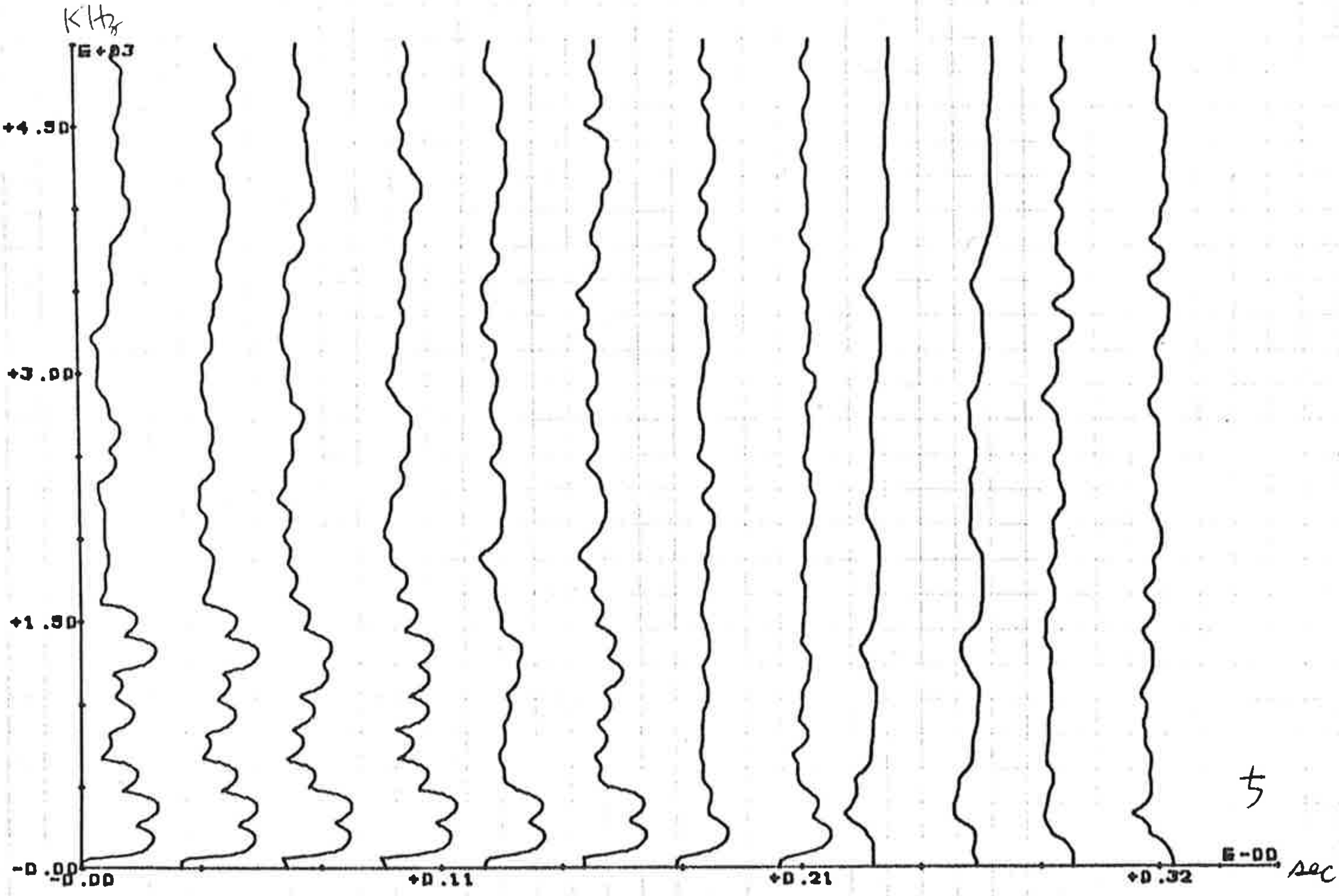

KHz
+4.50
+3.00
+1.50
-0.00
+0.00
+0.11
+0.21
+0.32
sec

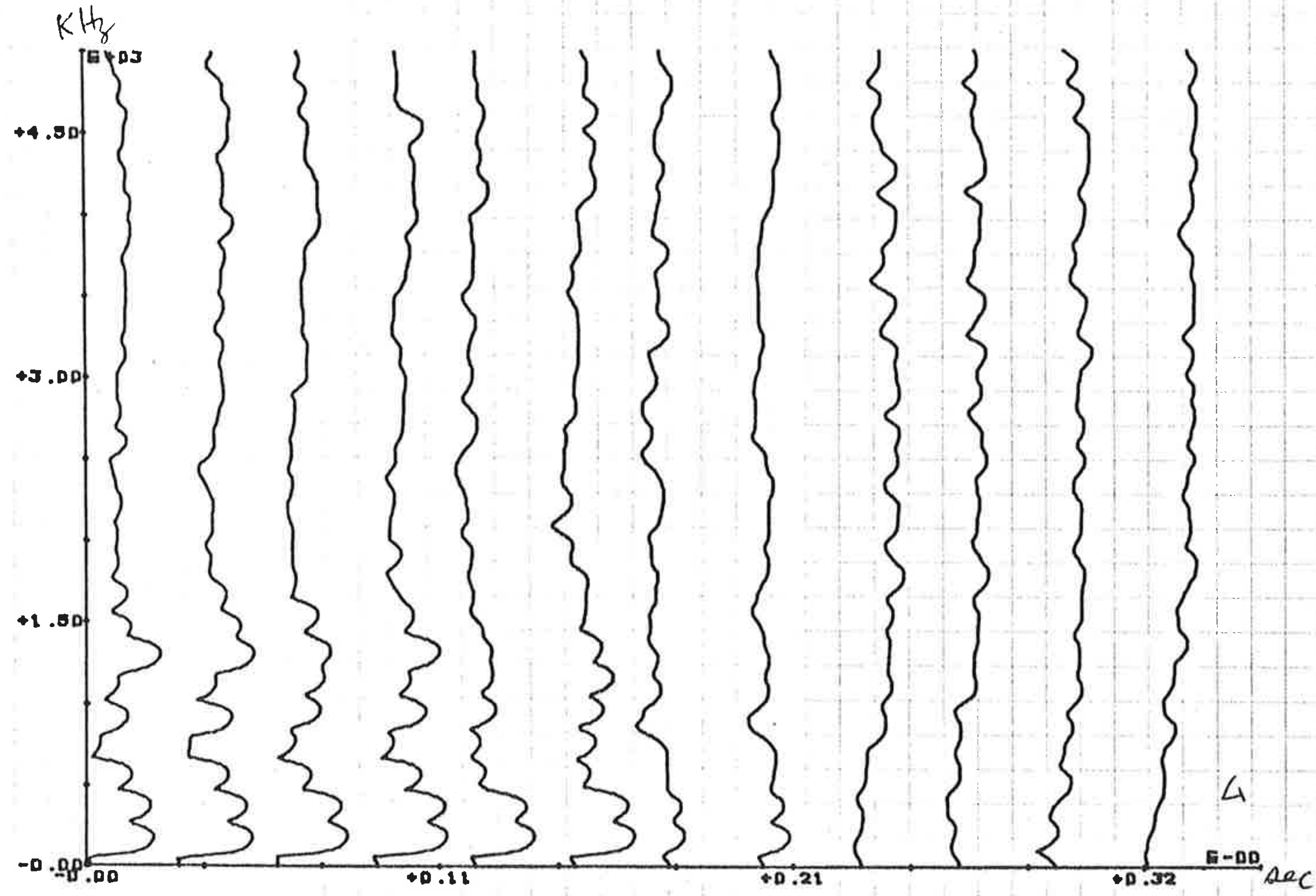

KHz
E+03
+4.80
+3.00
+1.80
-0.00
+0.00
+0.11
+0.21
+0.32
E-00
イ
sec

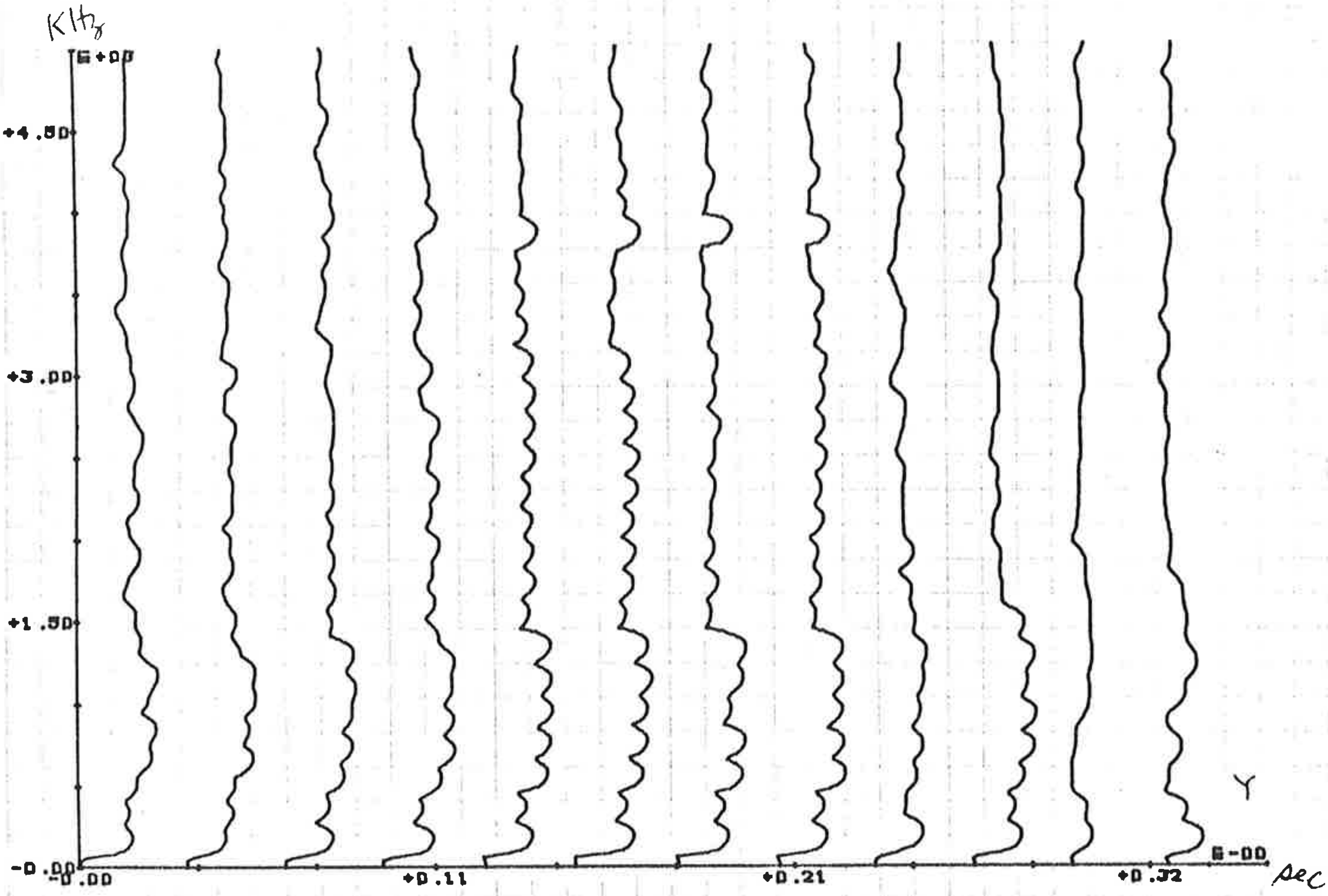

KHz
E+03
+4.80
+3.00
+1.80
-0.00
+0.00
+0.11
+0.21
+0.32
E-00
ア
sec

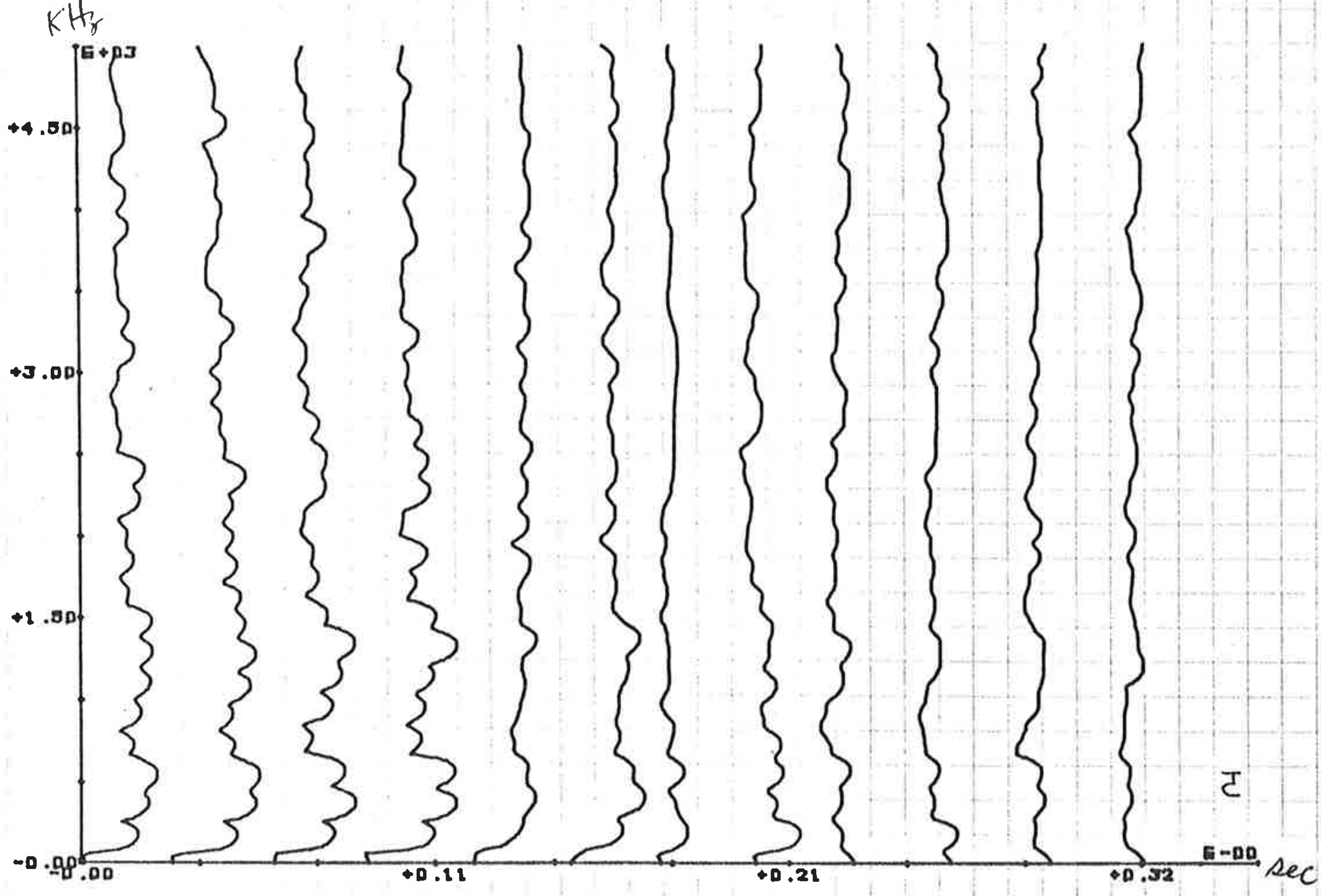

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec

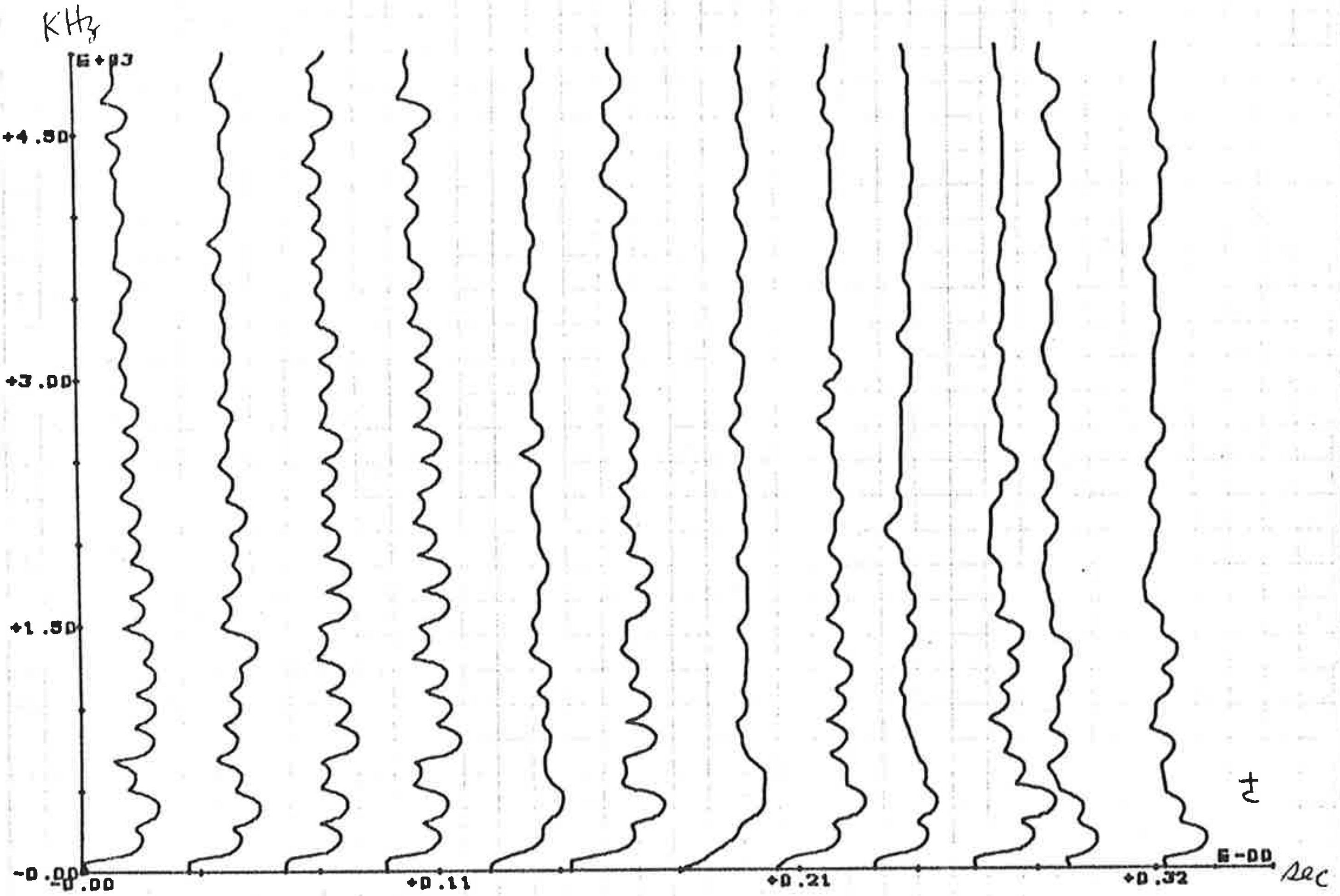

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec
P

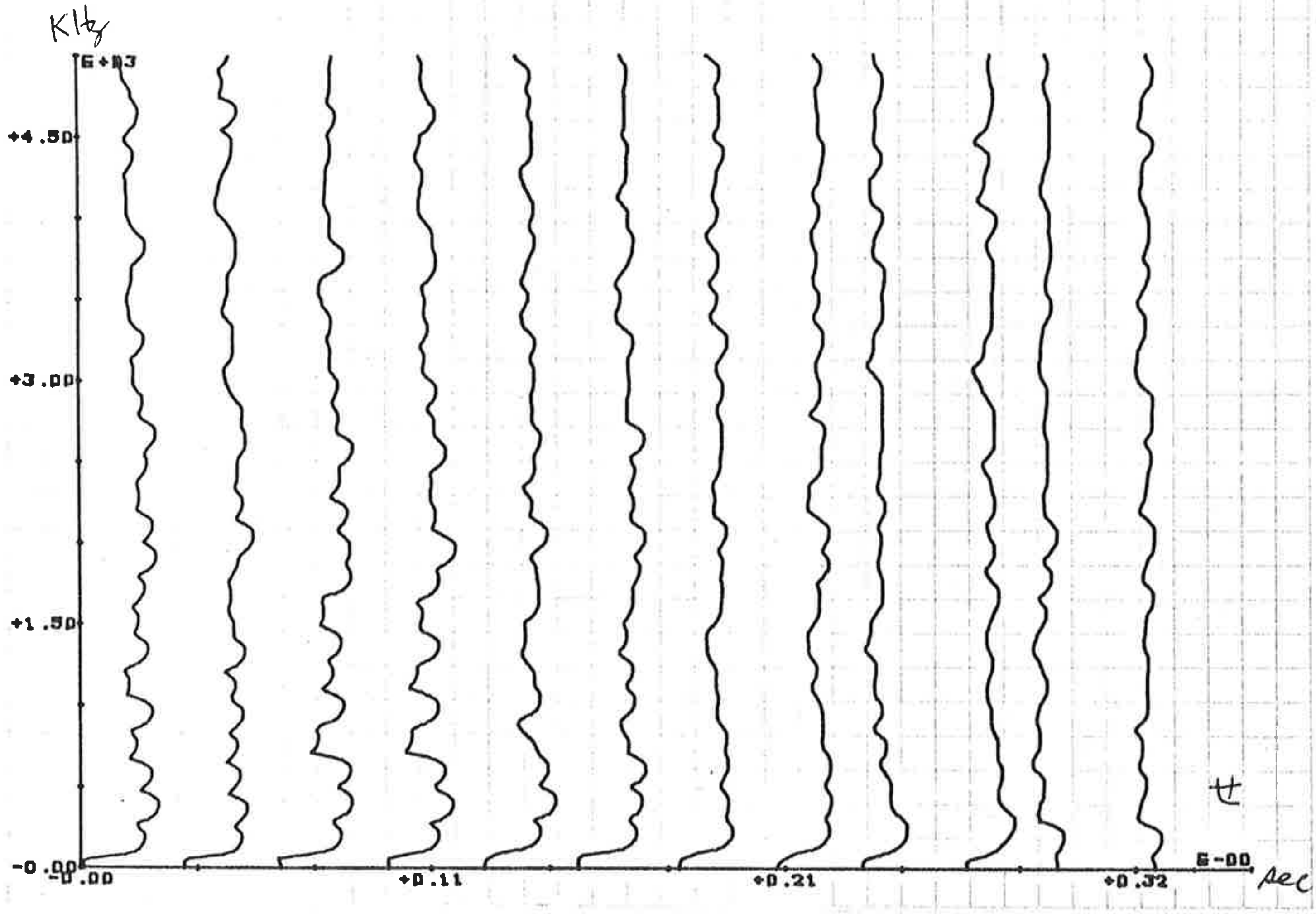

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec
5

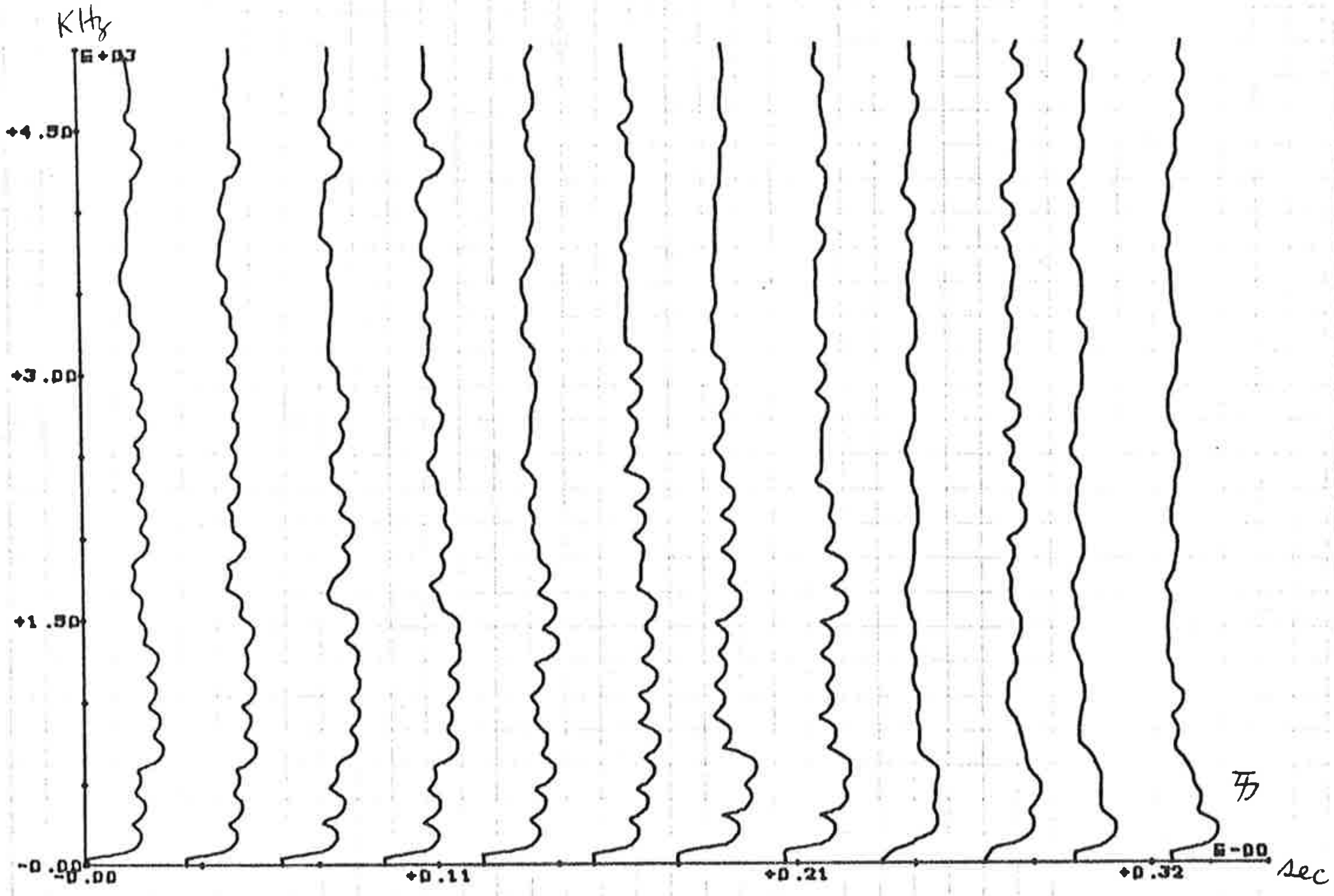

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec
4

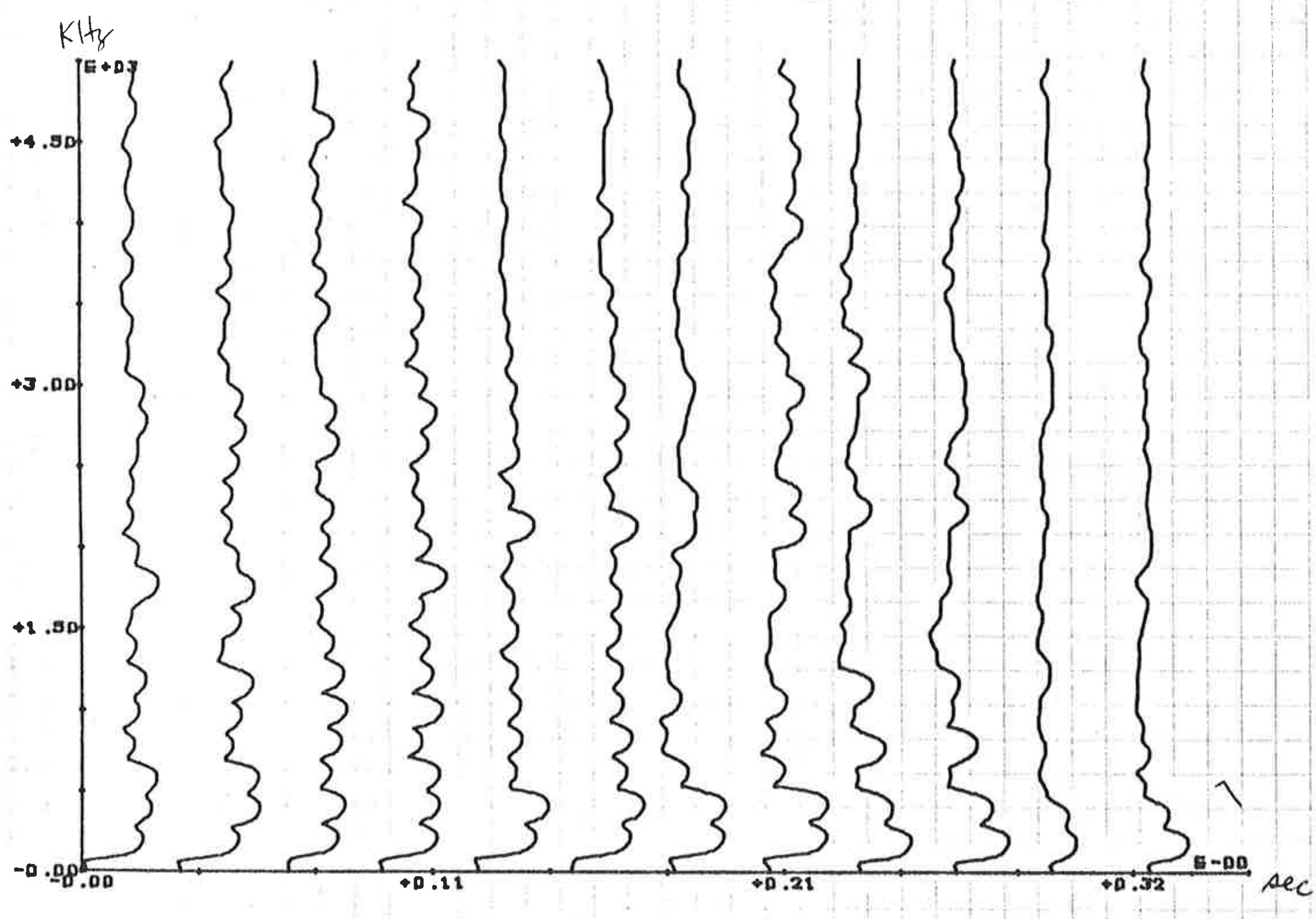

KHz
6+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
6-00
sec
Y

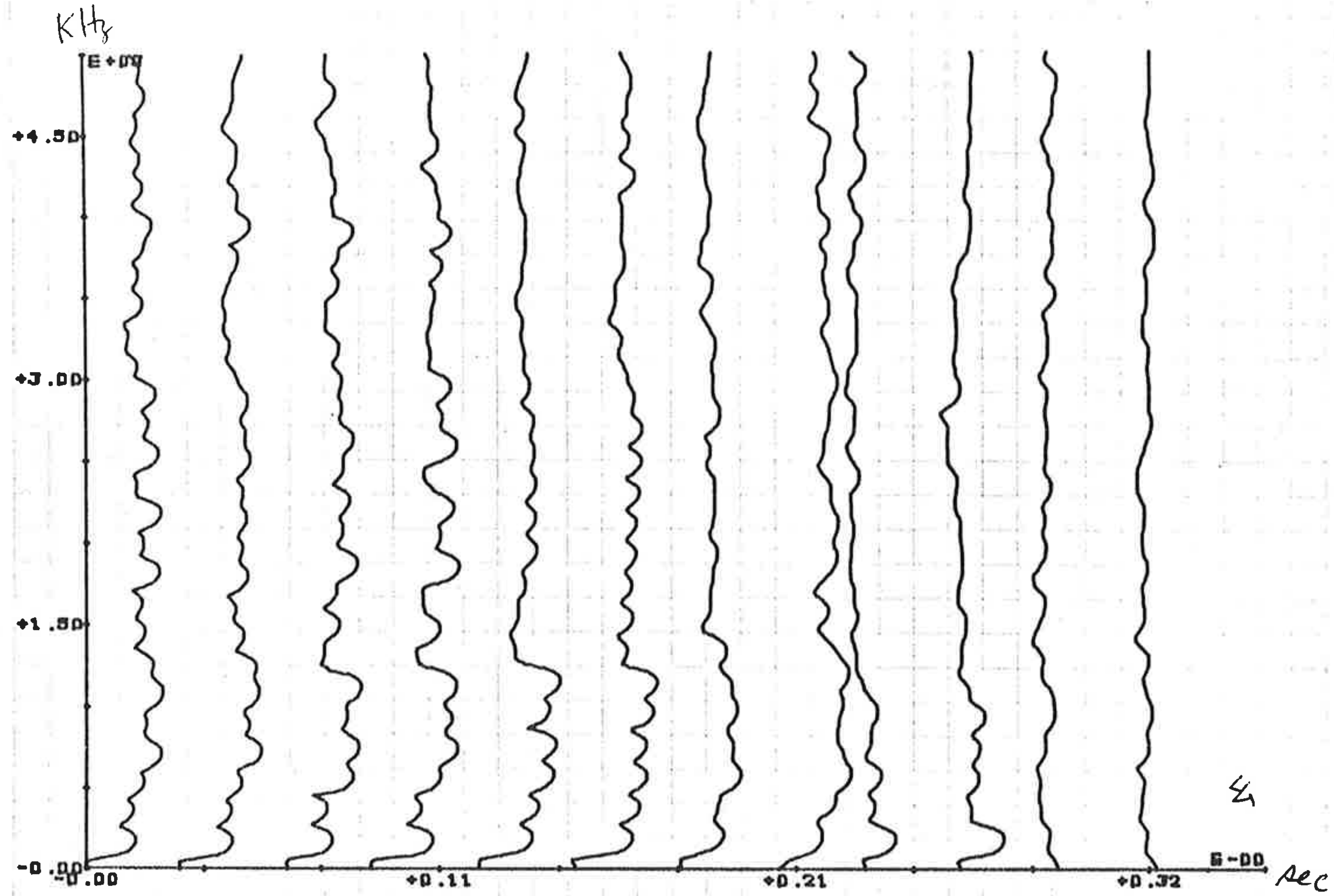

KHz
6+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
6-00
sec

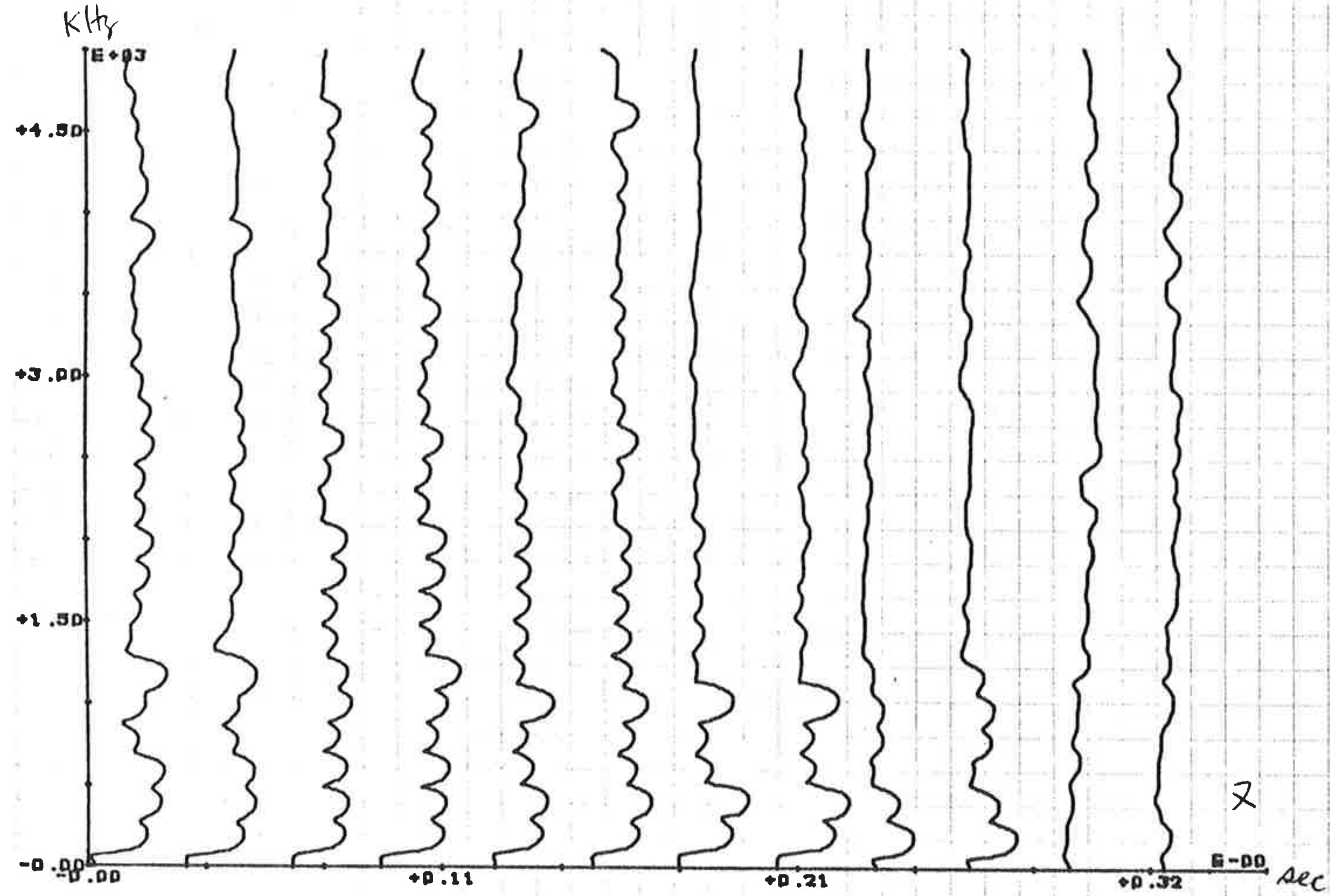

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec
さ

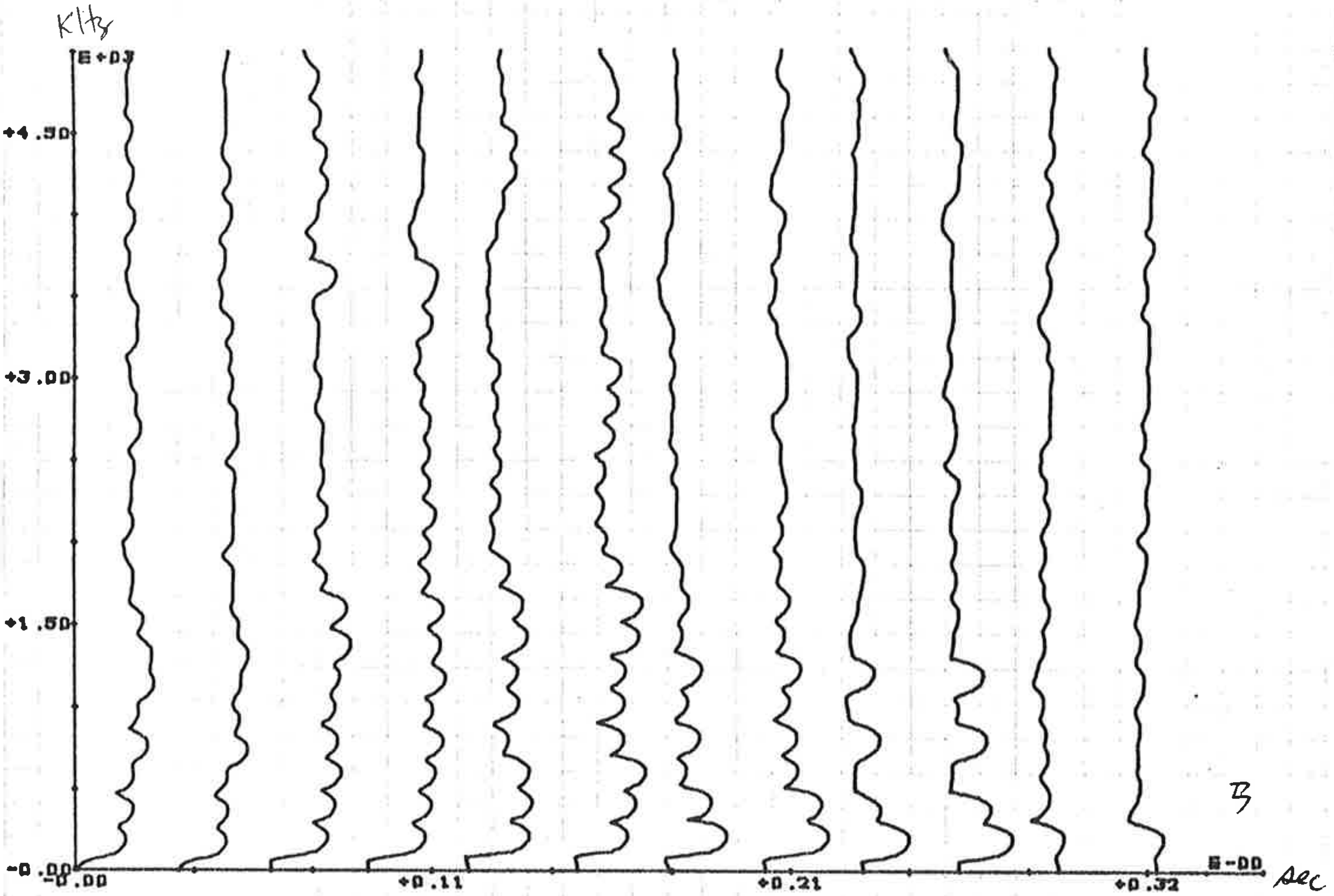

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec
せ

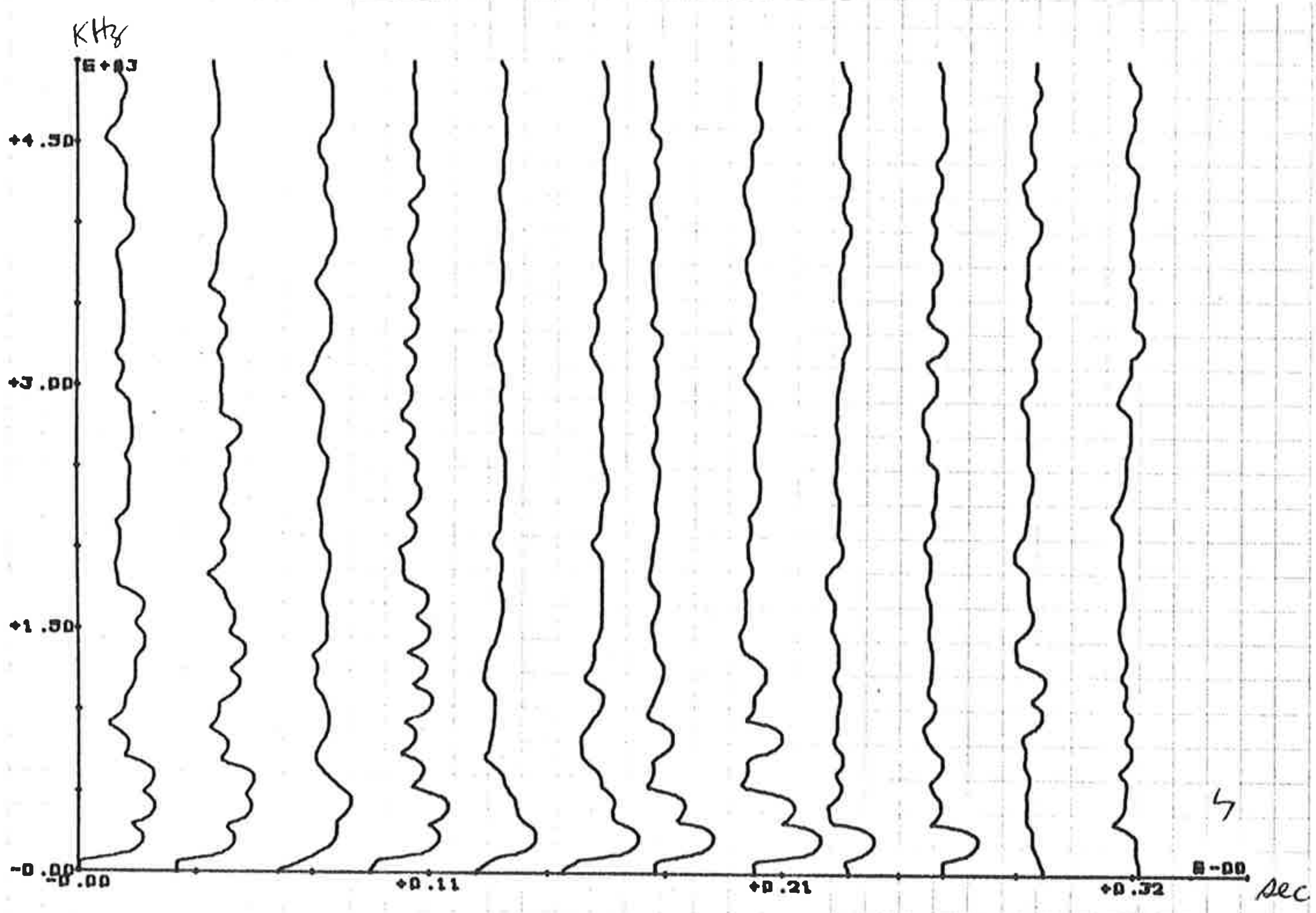

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec

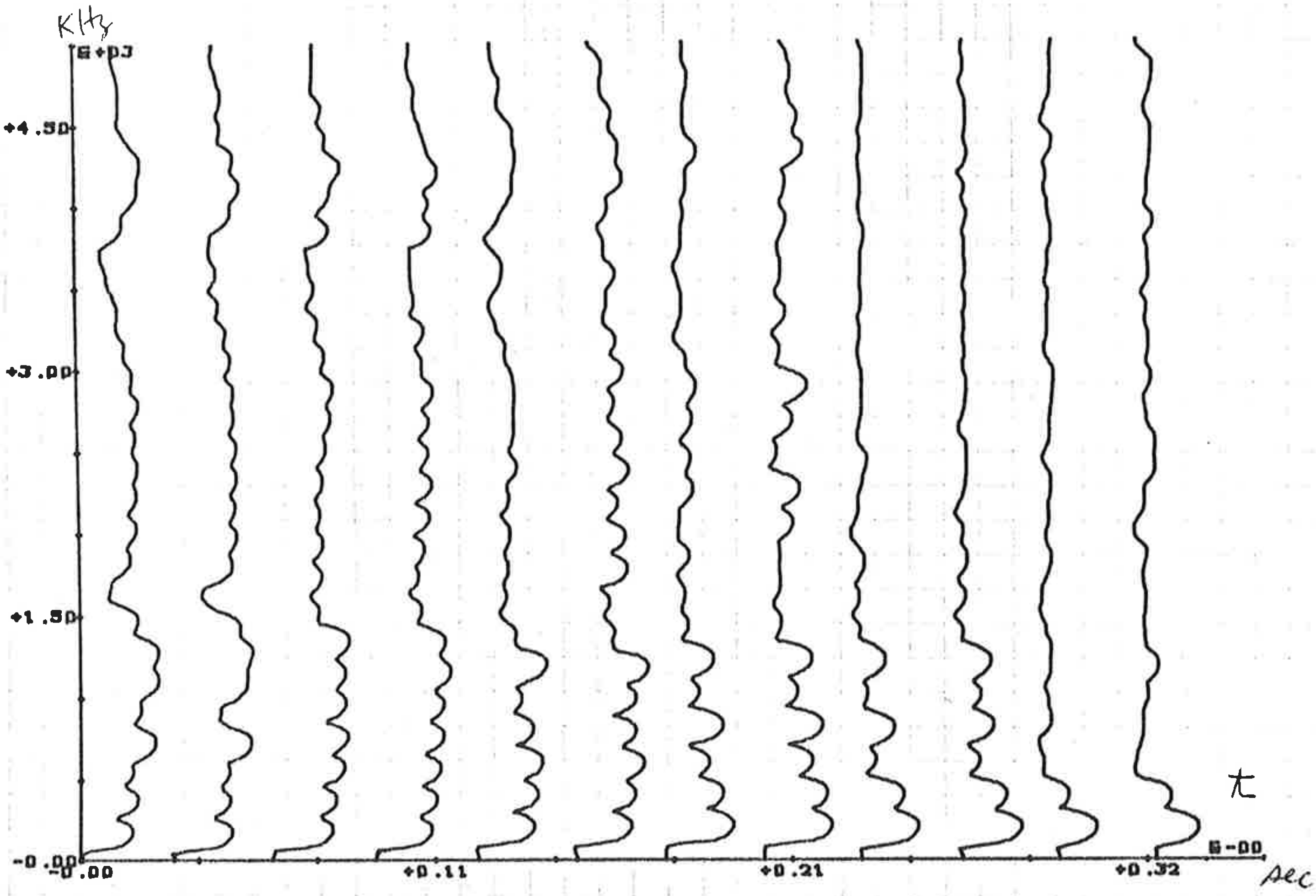

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec

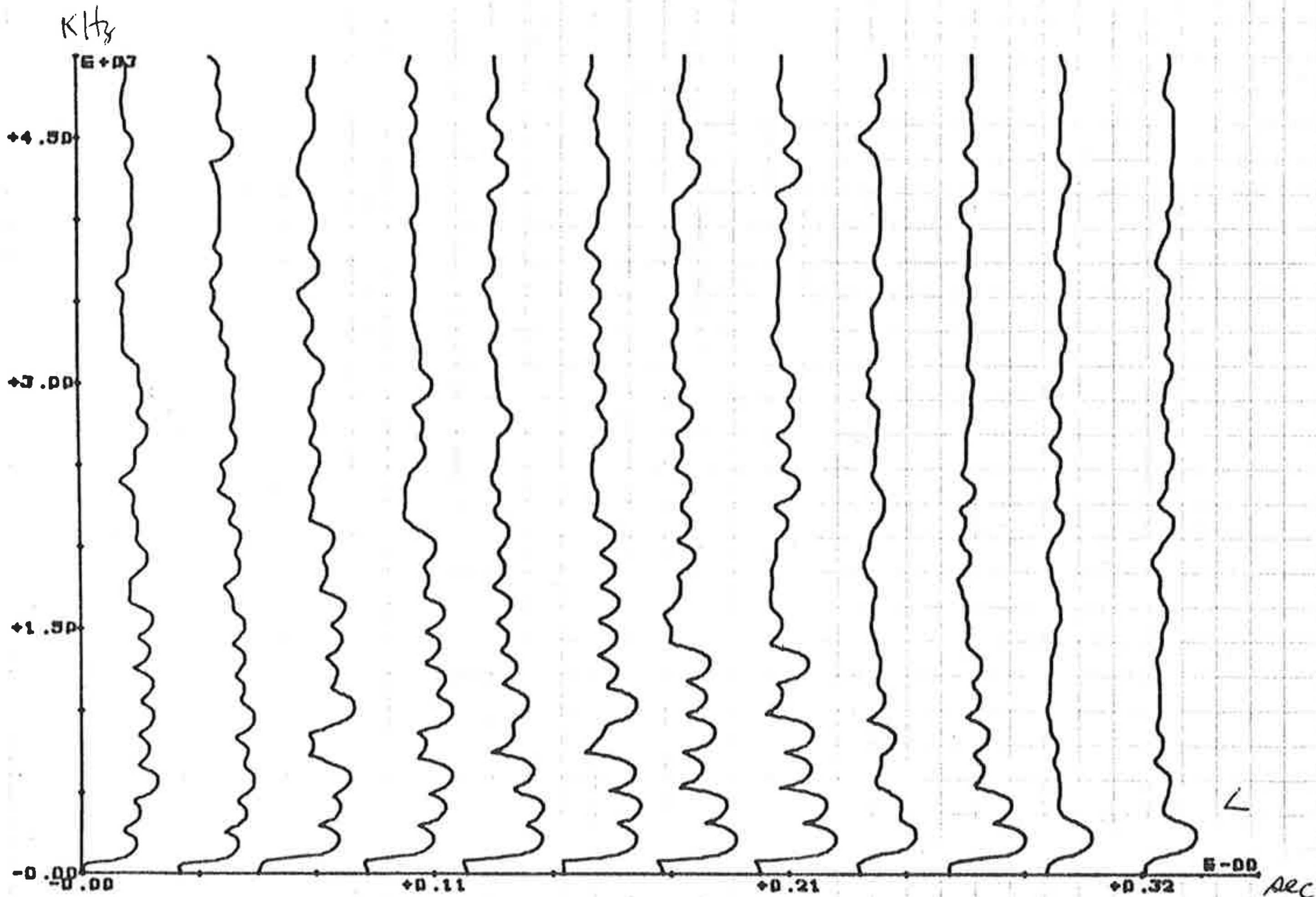

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec

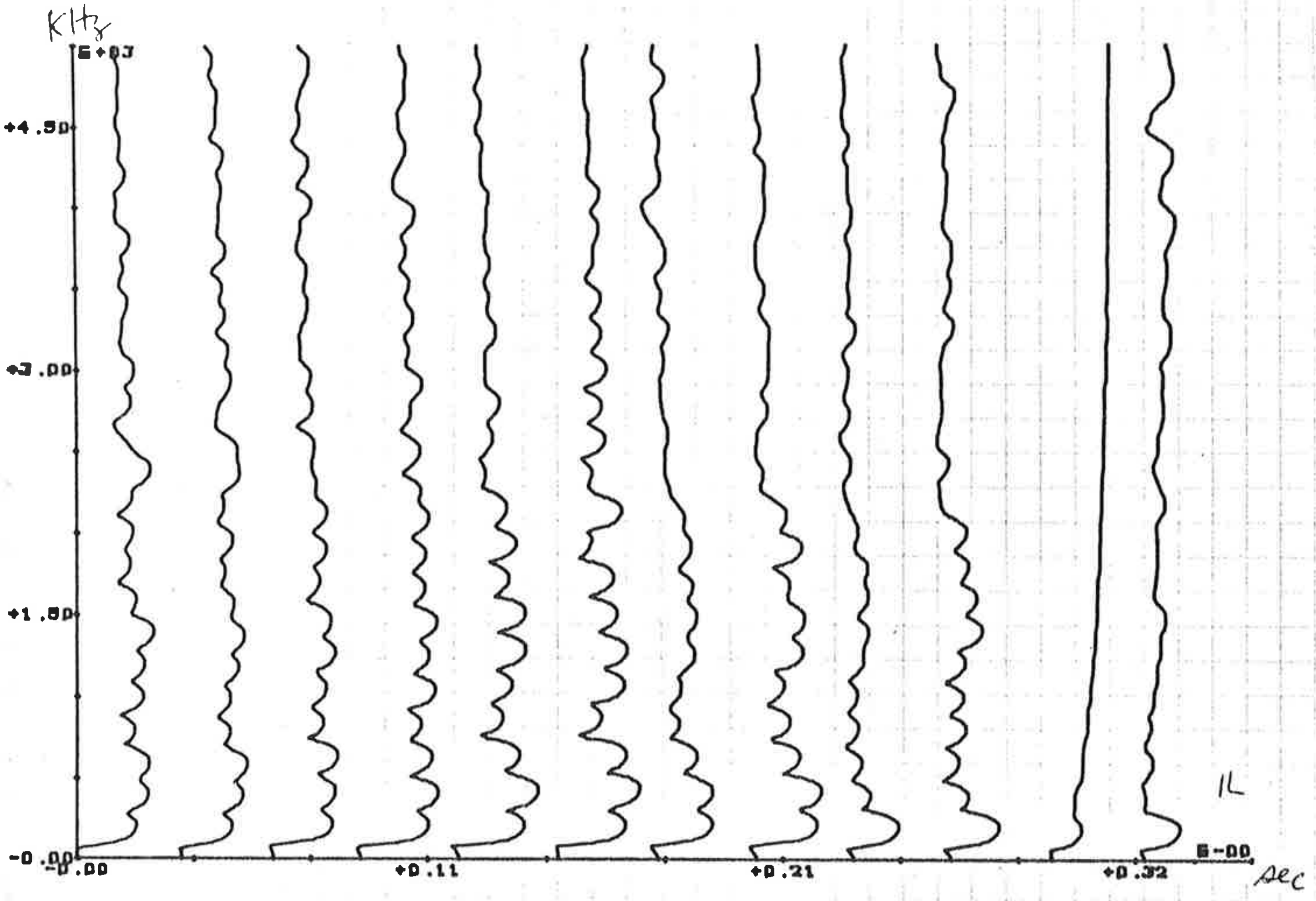

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec
Z

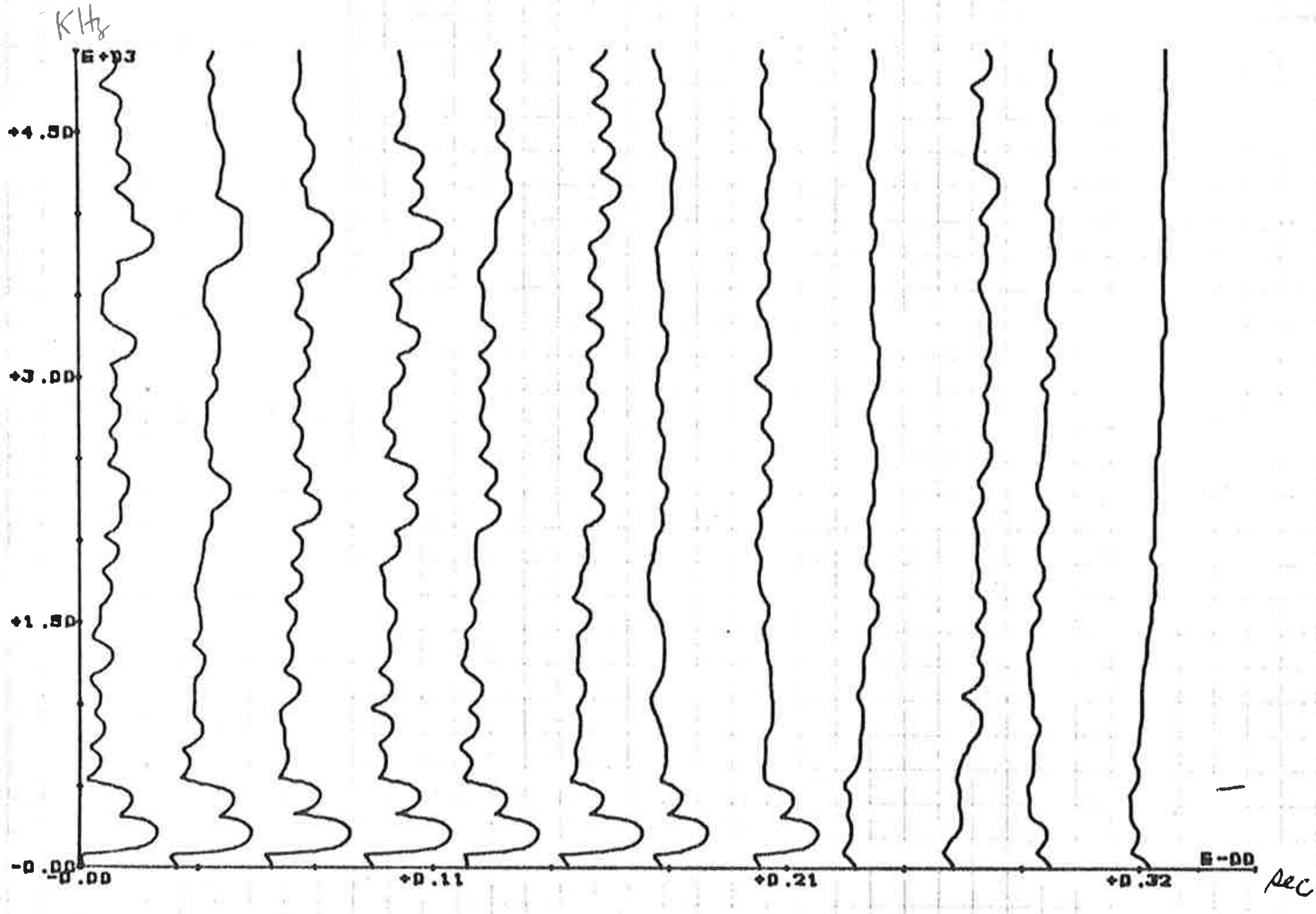

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec
3

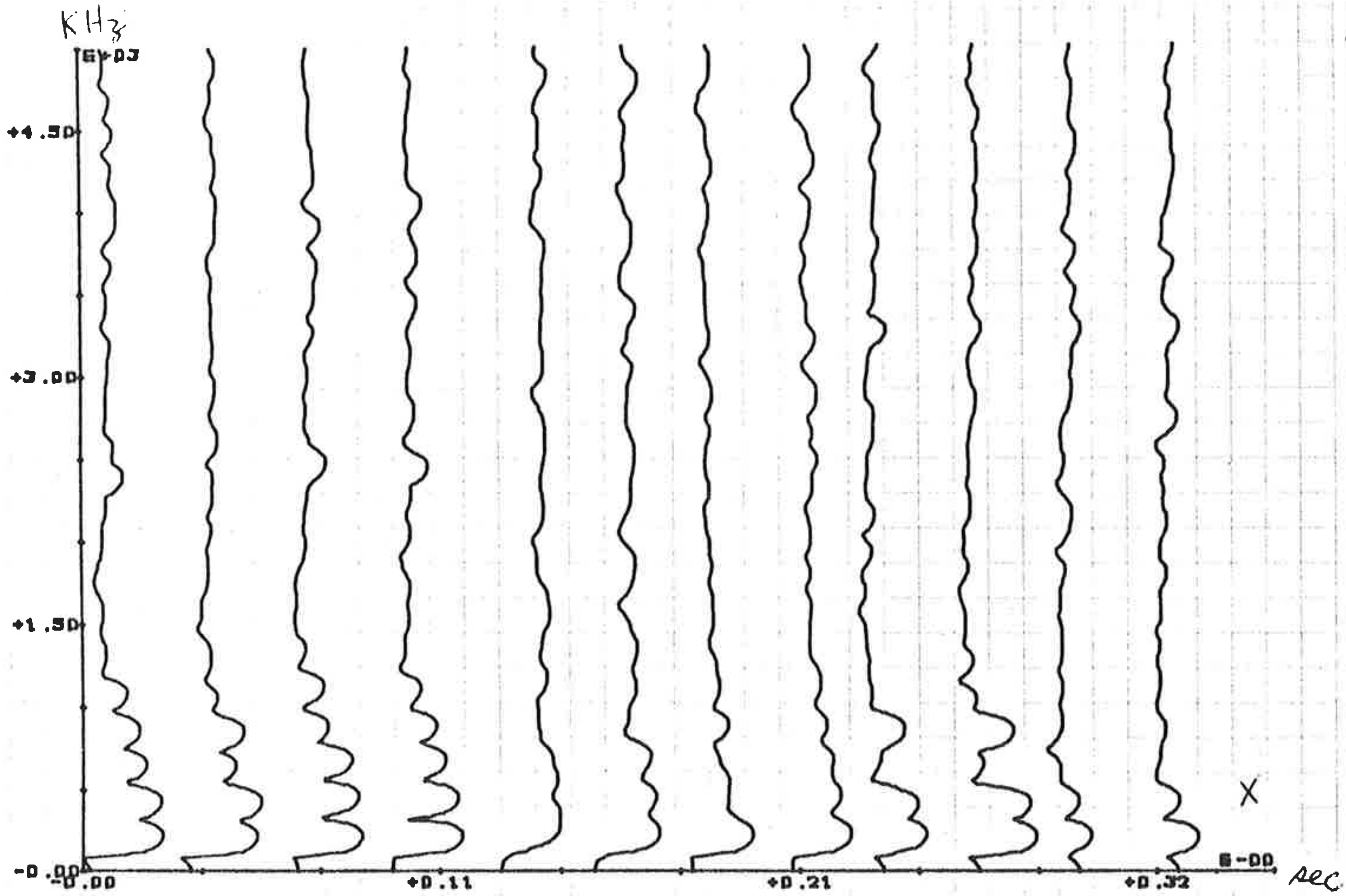

KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec
4

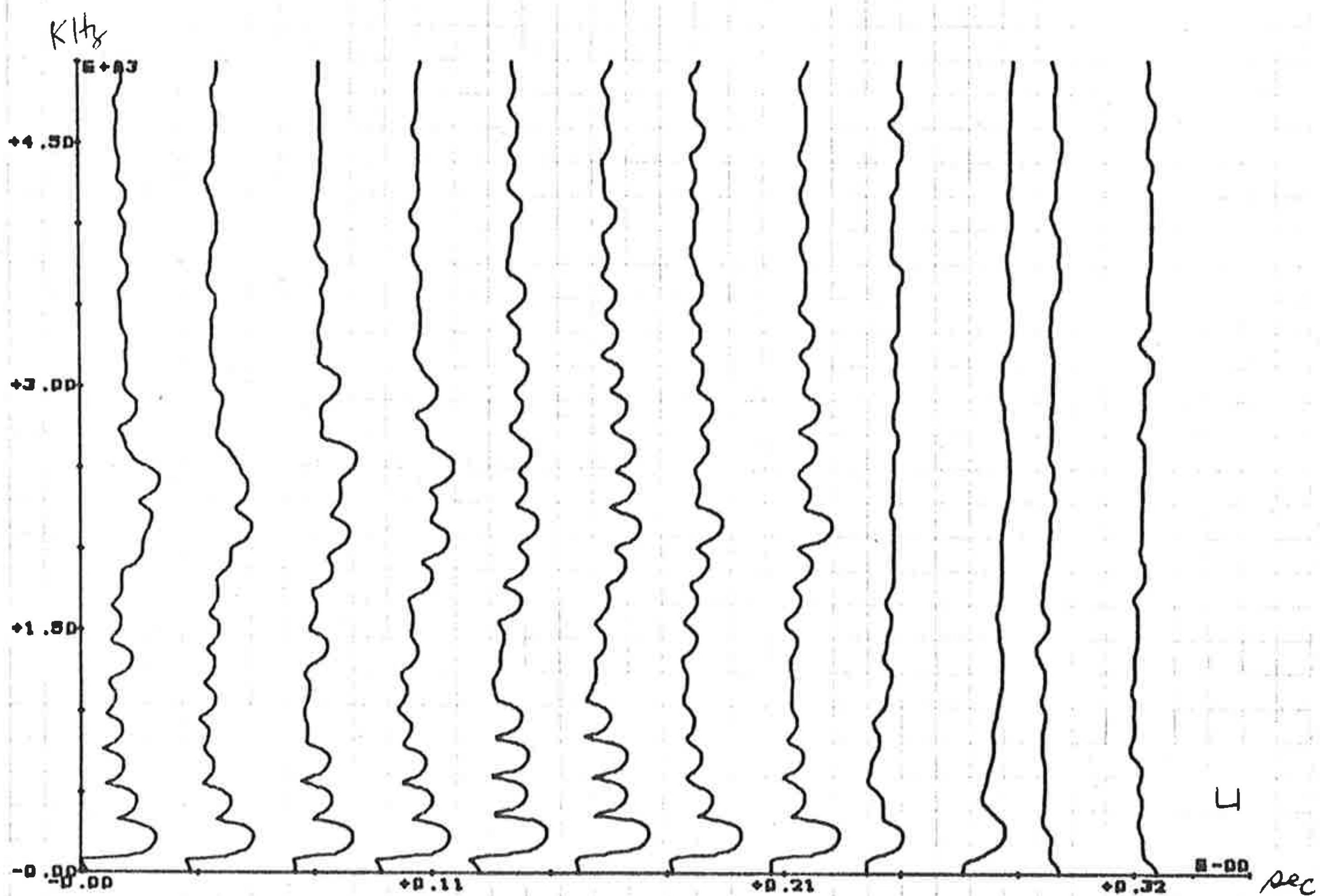

KHz
6+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
6-00
sec
t


KHz
6+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
6-00
sec
L


KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec
IL


KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec


KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec
X


KHz
E+03
+4.50
+3.00
+1.50
-0.00
-0.00
+0.11
+0.21
+0.32
E-00
sec
L

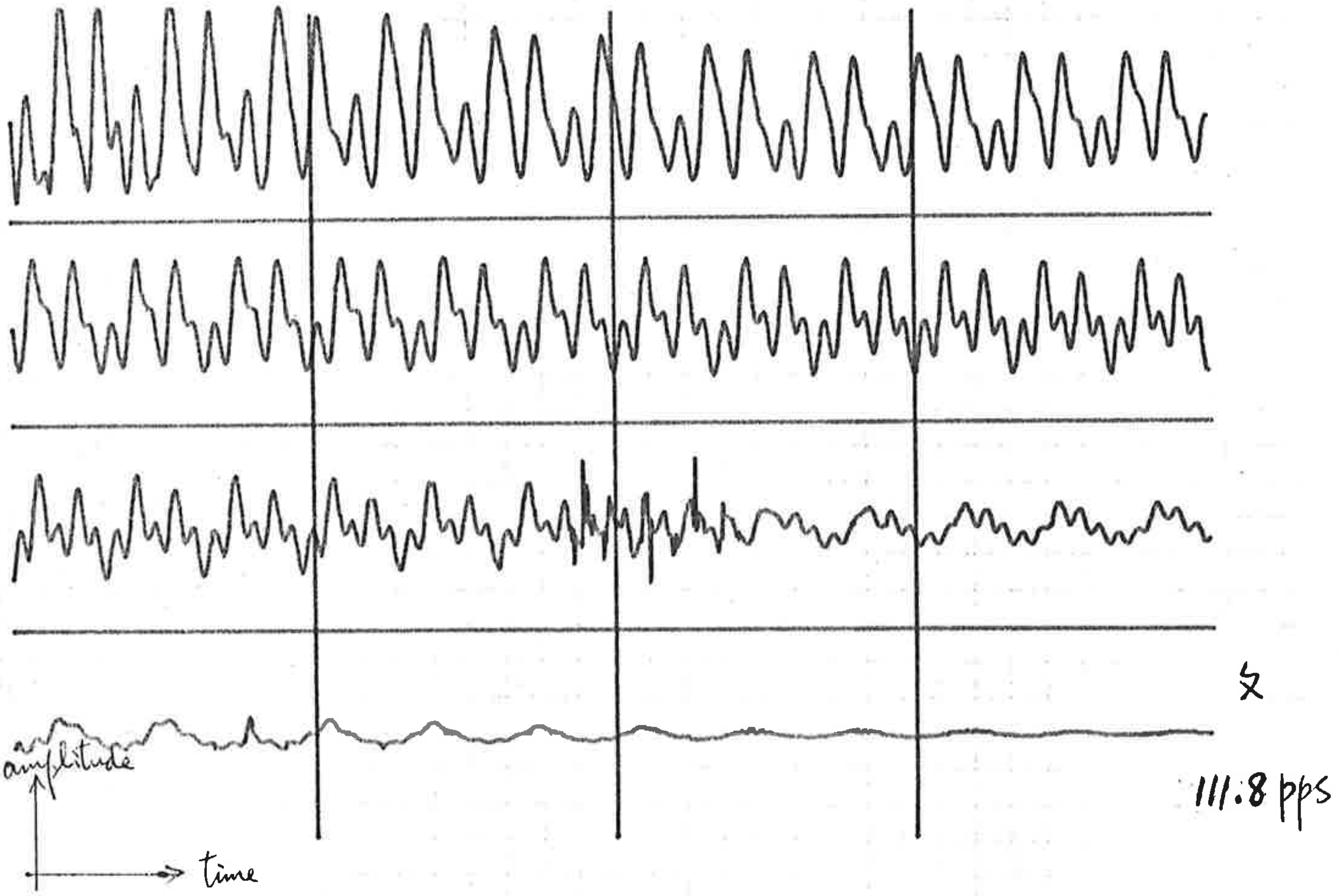
APPENDIX 5

time varying waveforms 18 pages
(i.e. time waveforms)

A rank ordering on a scale of pitchs ranging from low to high is as following:

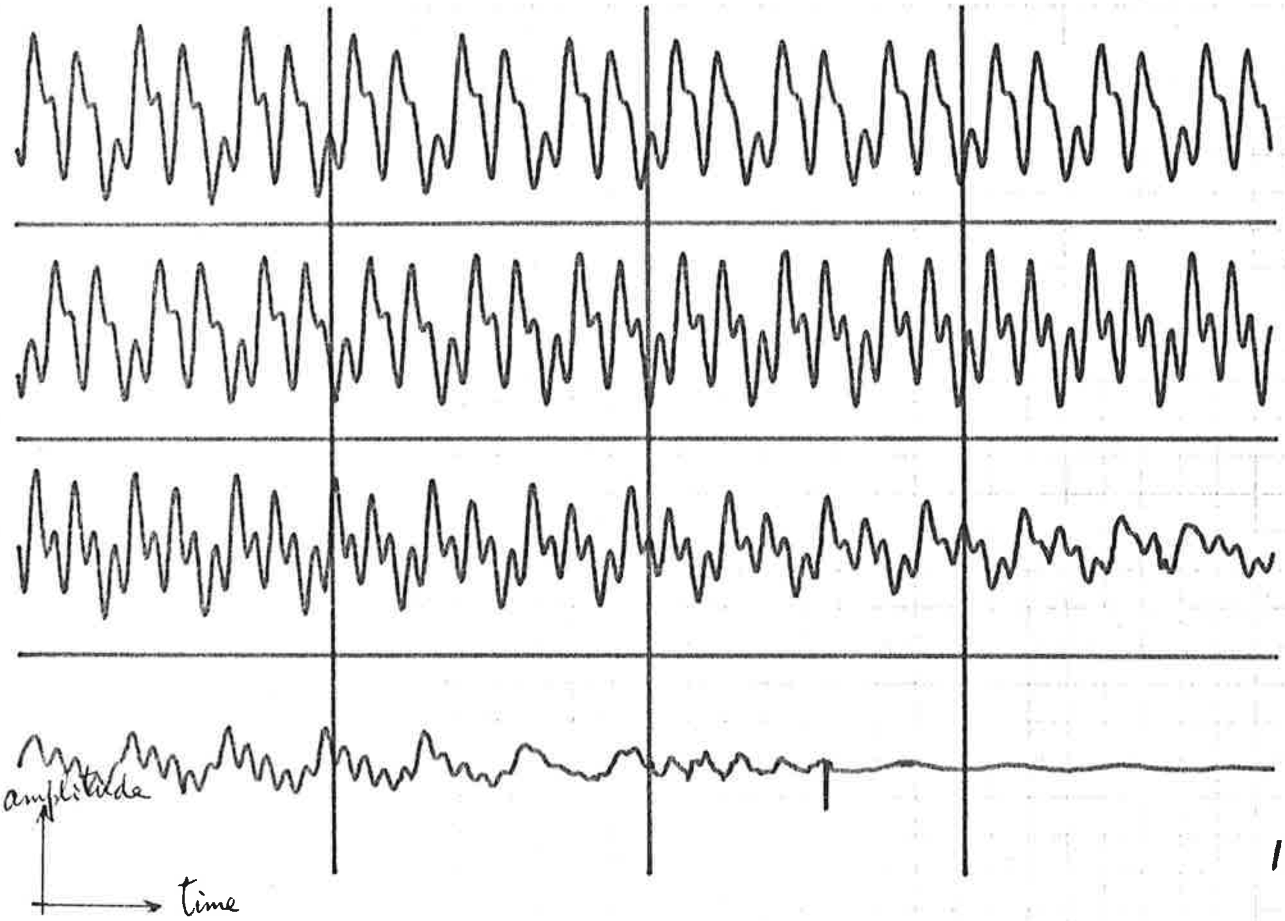
ㄌ, ㄅ(ㄆ), ㄋ, ㄇ(ㄊ), ㄗ(ㄉ), ㄓ(ㄈ), ㄎ(ㄏ), ㄐ, ㄒ, ㄍ, ㄔ, ㄑ, ㄚ, ㄞ(ㄠ, ㄤ),

ㄖ(ㄘ), ㄕ(ㄙ, ㄛ, ㄝ, ㄥ), ㄜ(ㄟ, ㄡ, ㄢ, ㄣ, ㄦ), ㄧ(ㄨ, ㄩ)



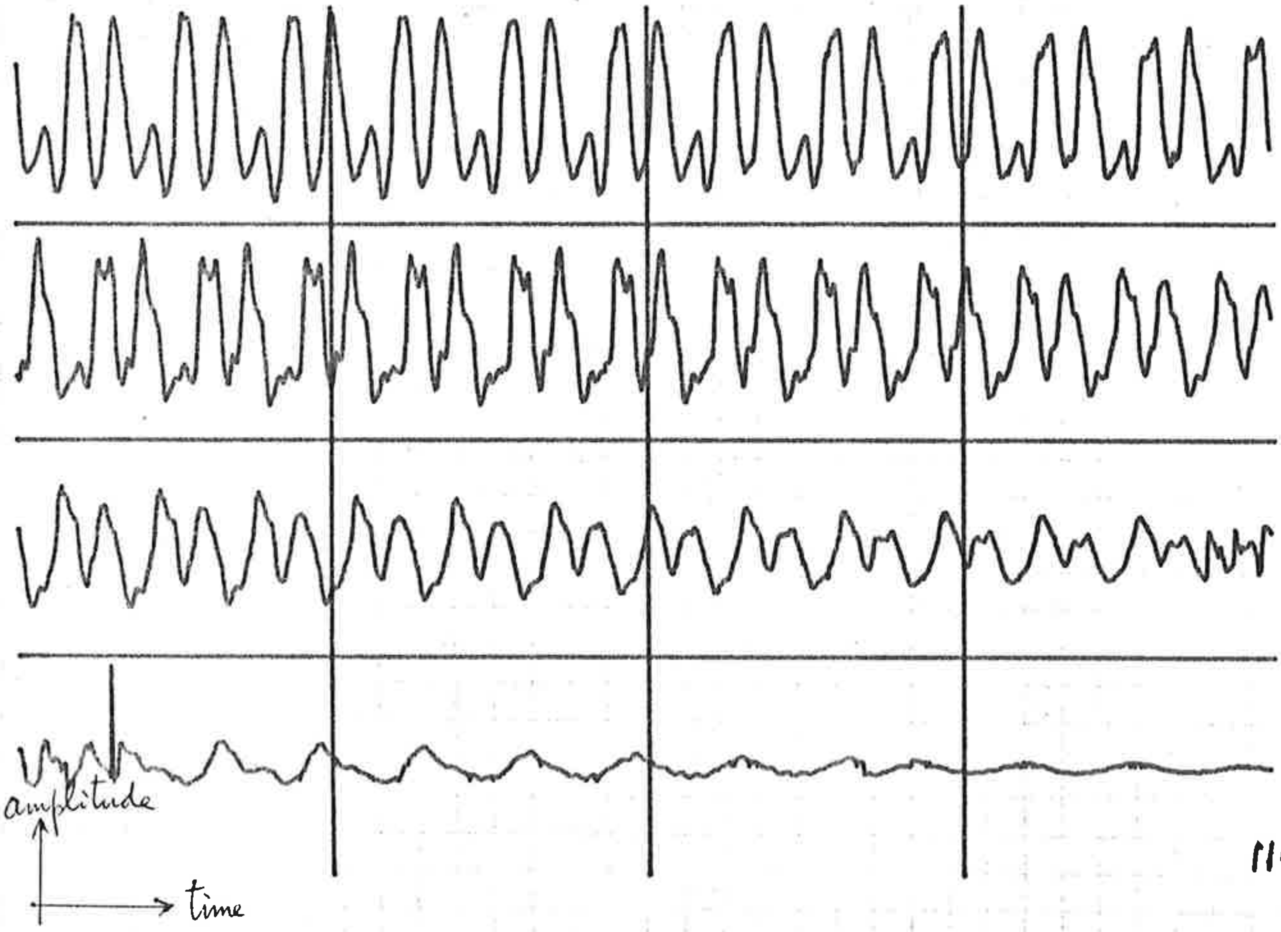
ㄆ
111.8 pps
amplitude
time

ㄇ
113.36 pps
amplitude
time



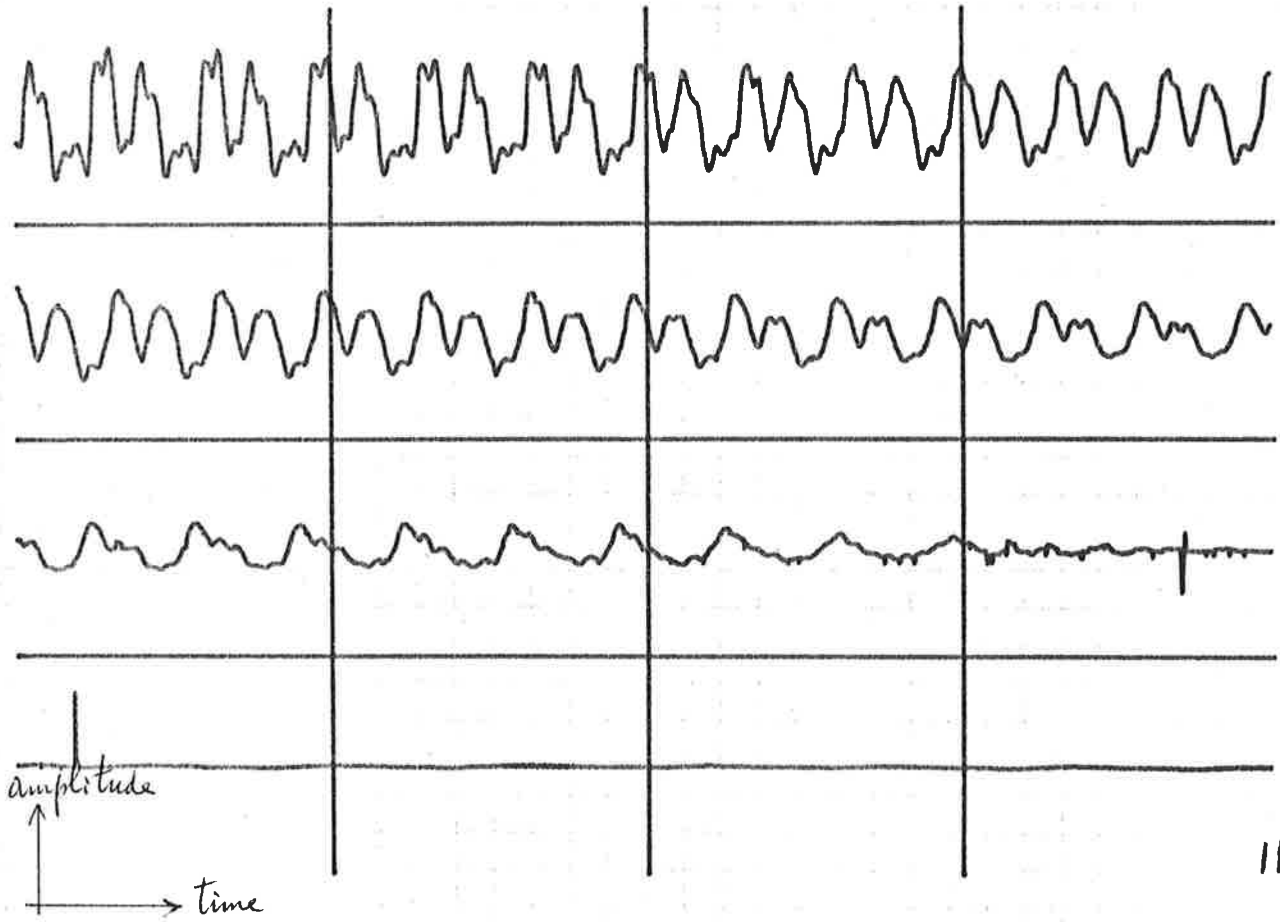
C

115.79 pps



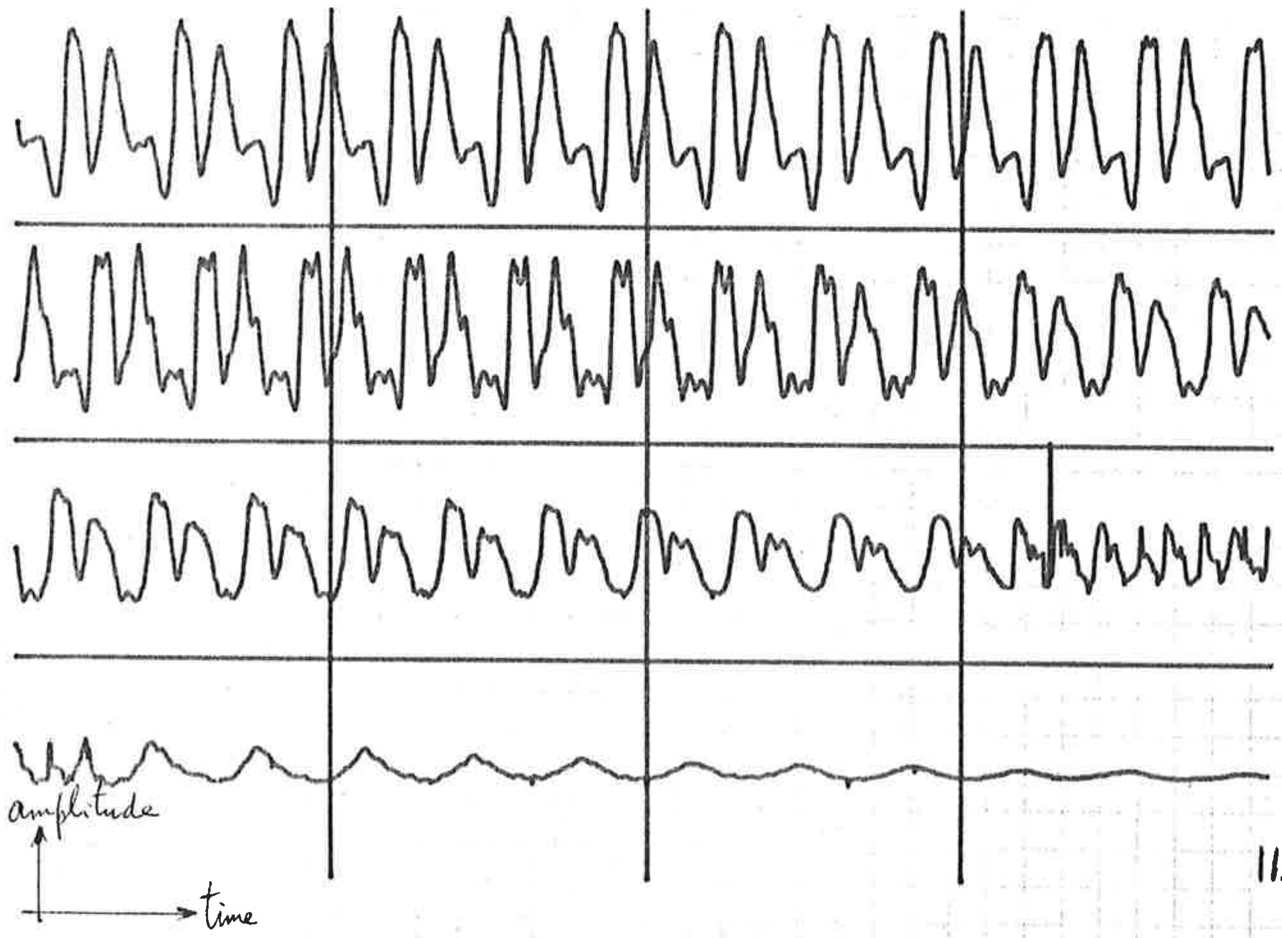
D

114.16 pps



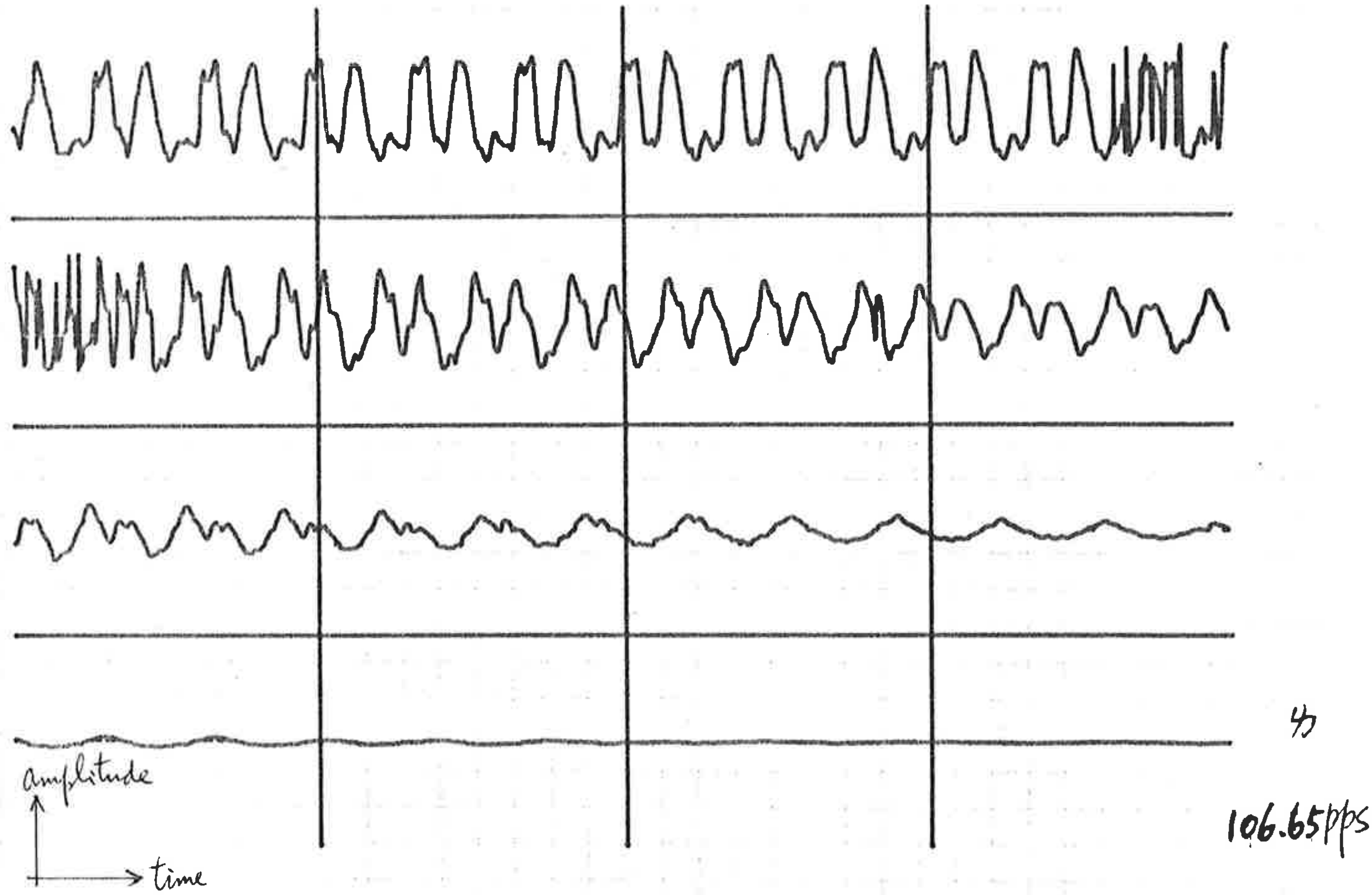
ち

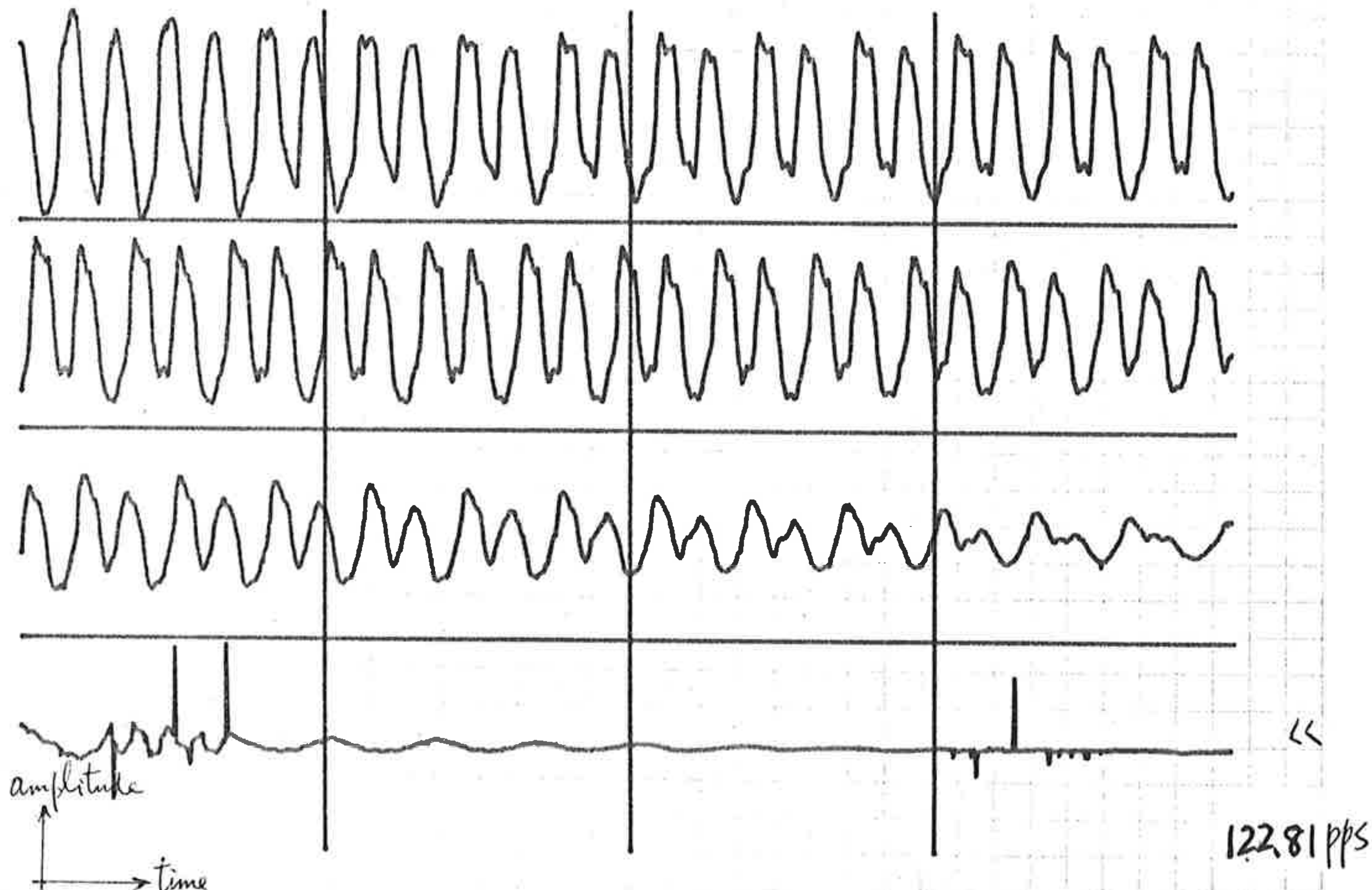
113.36 pps

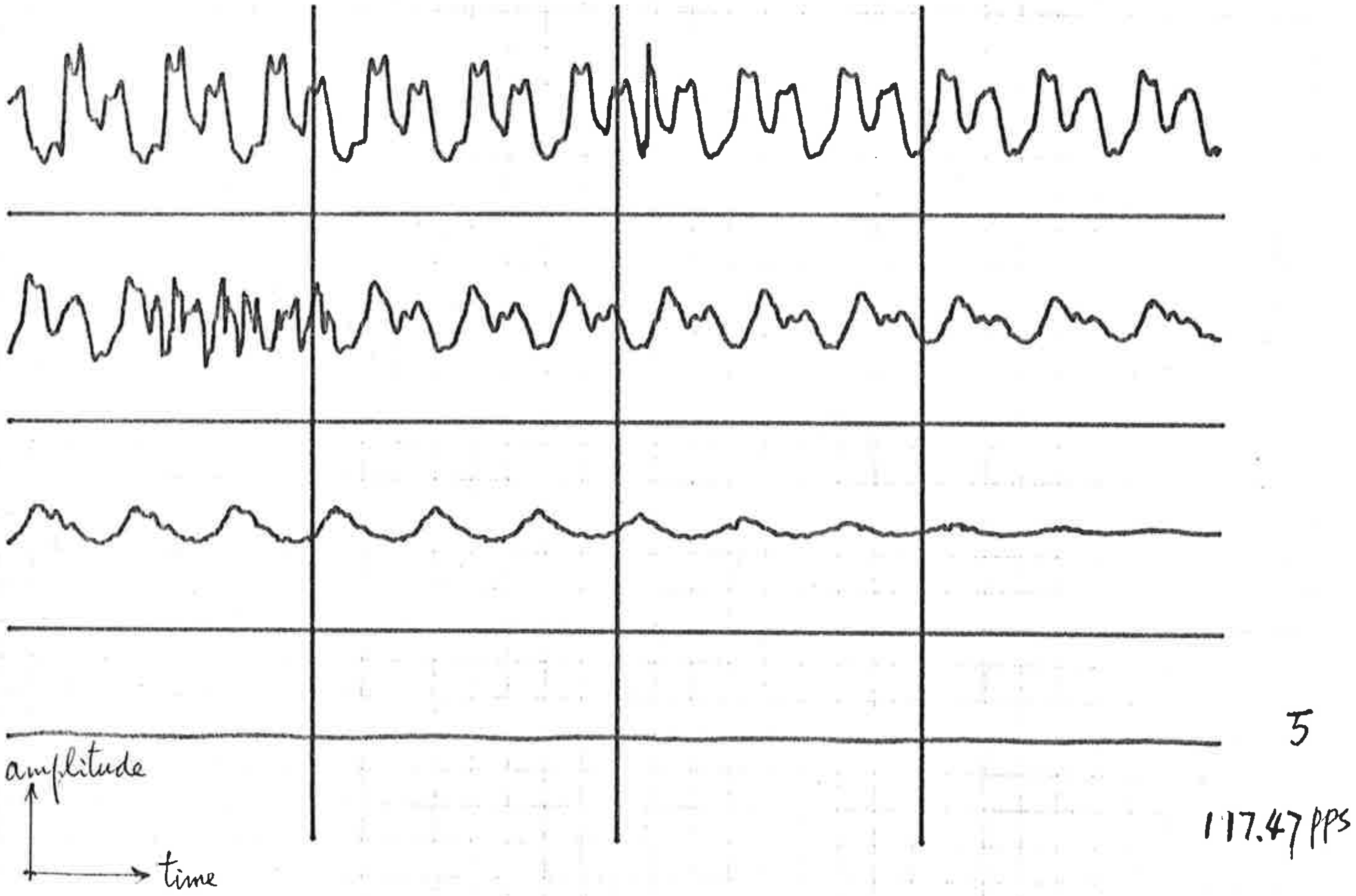


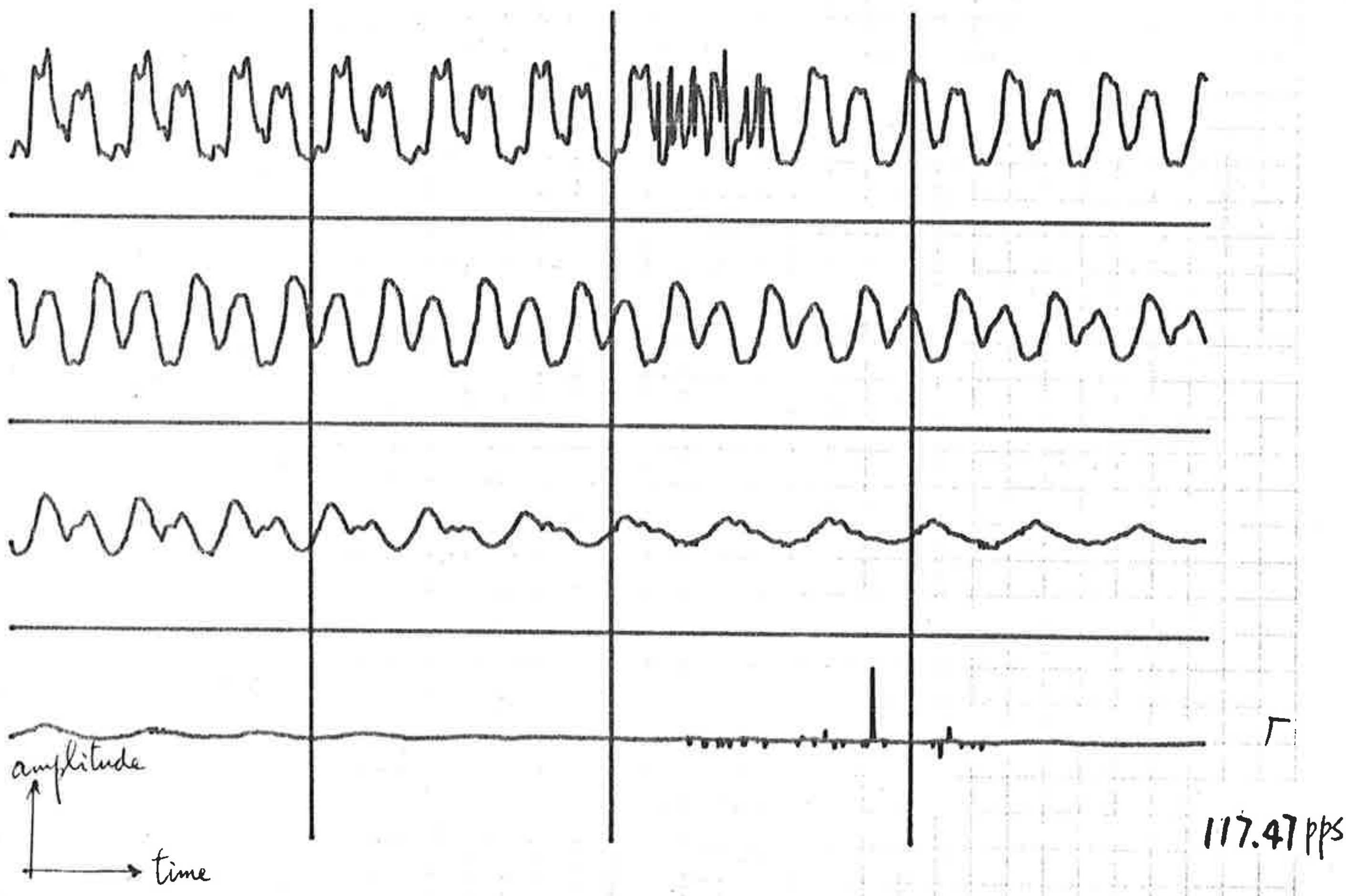
3

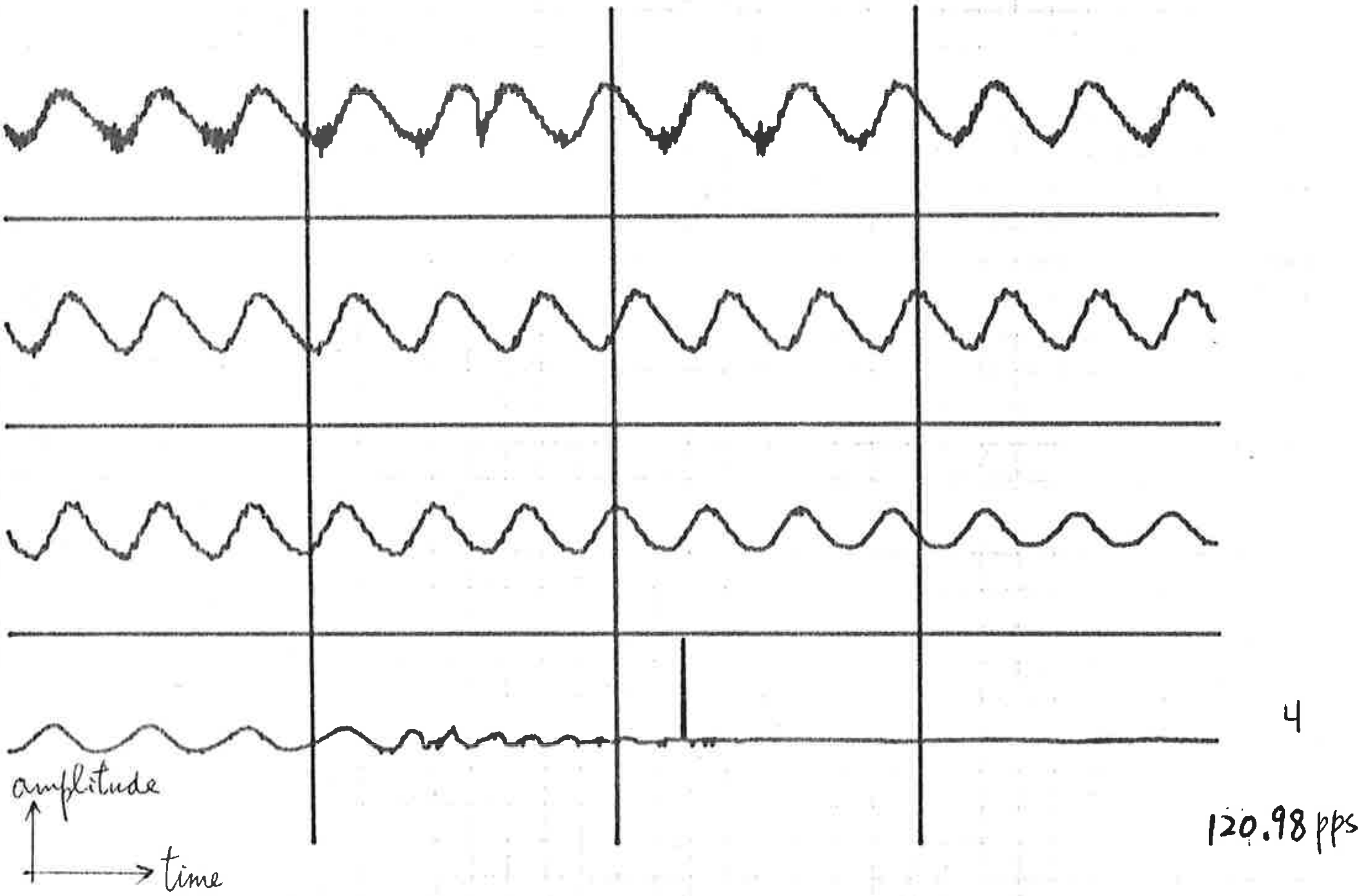
112.57 pps

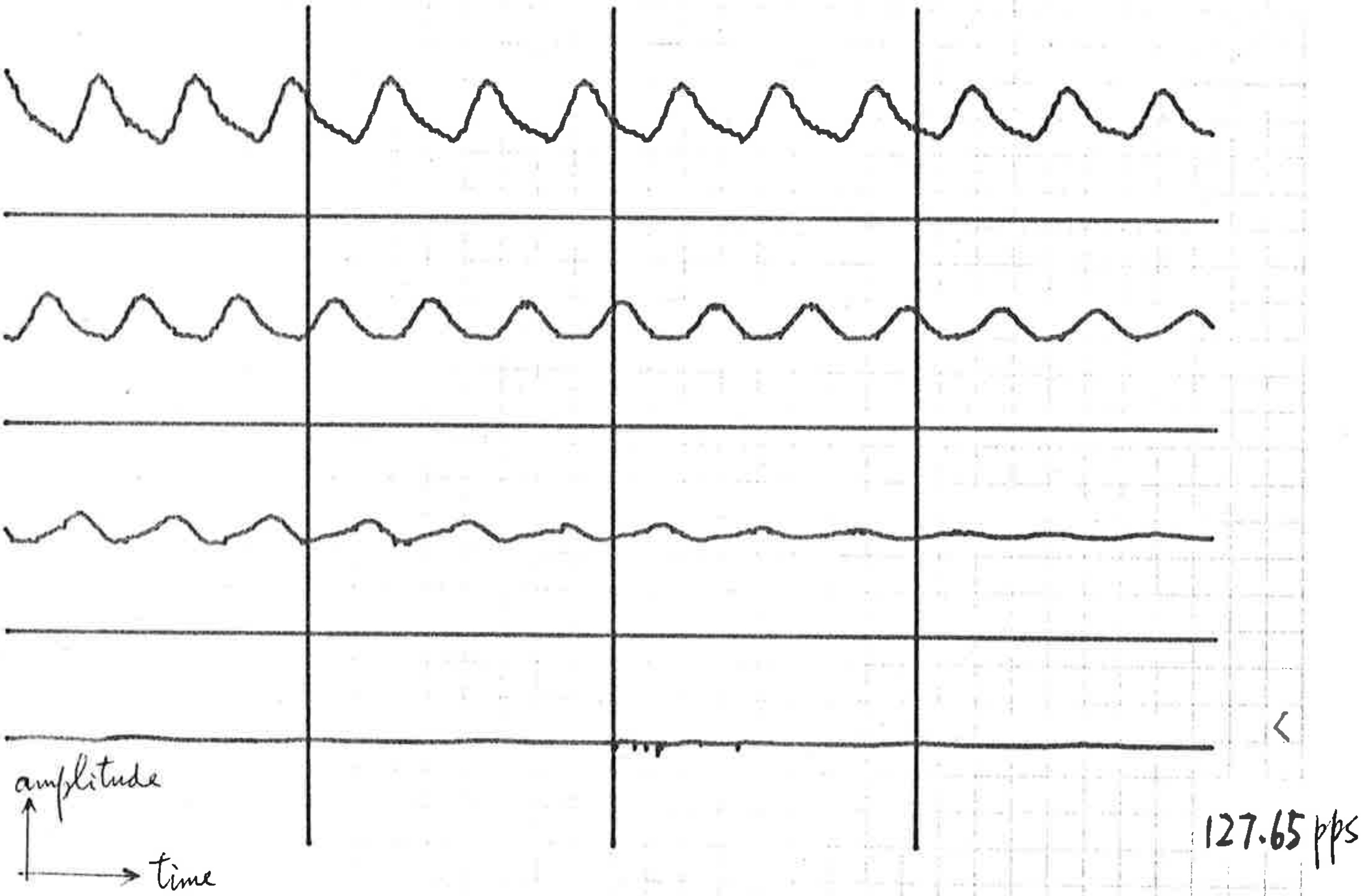

amplitude
time
106.65pps

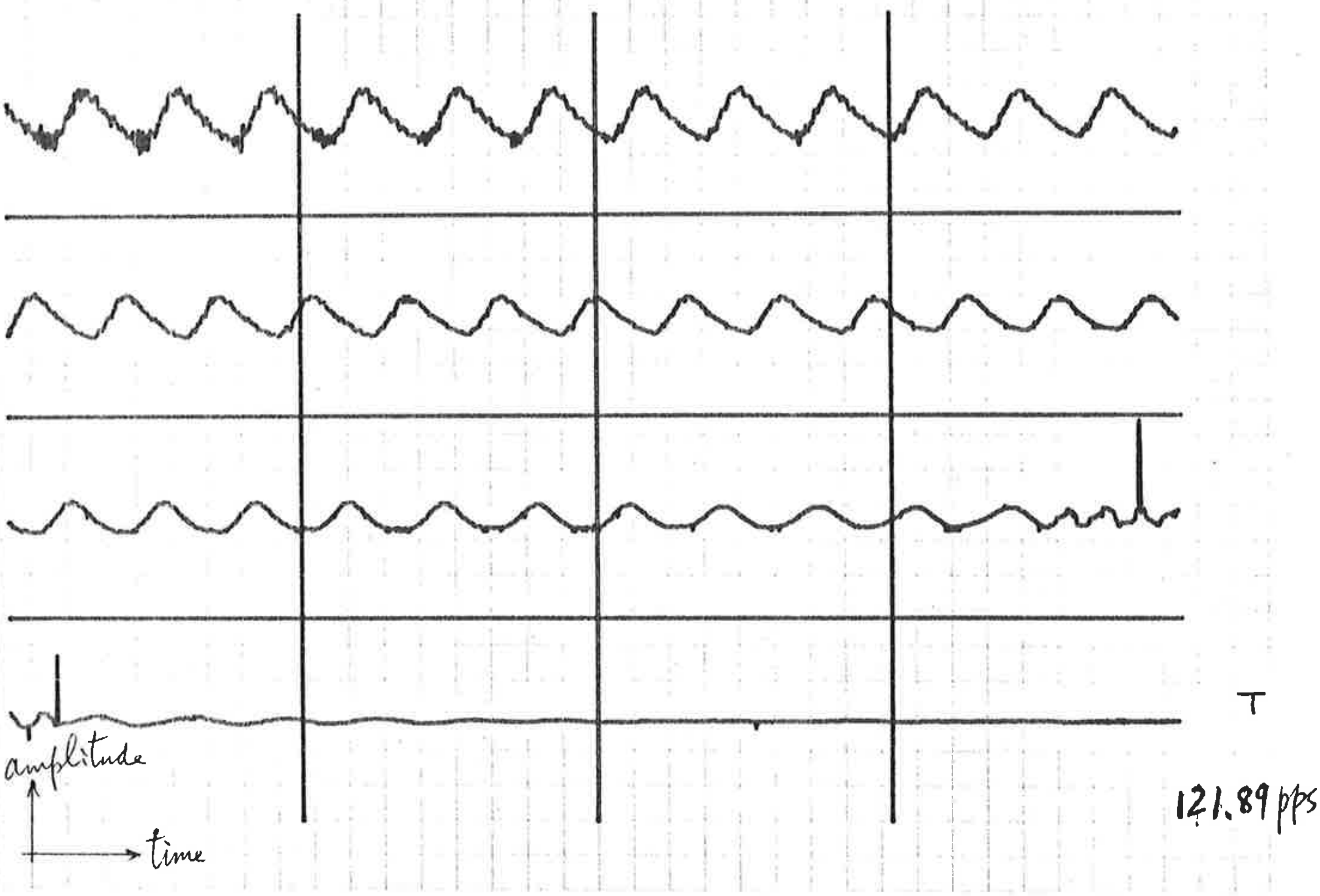

amplitude
time
122.81pps

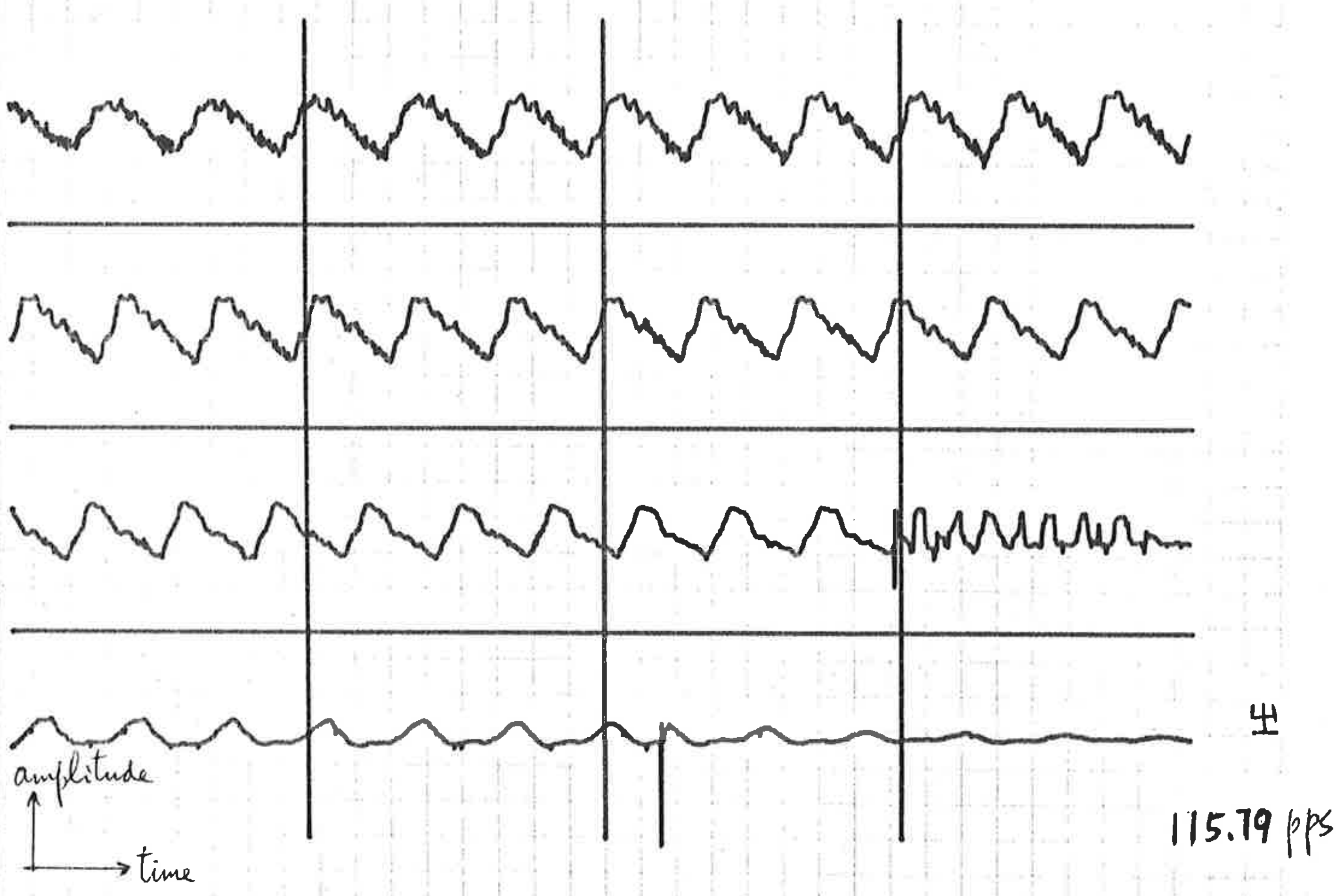

amplitude
time
Б
117.47 pps

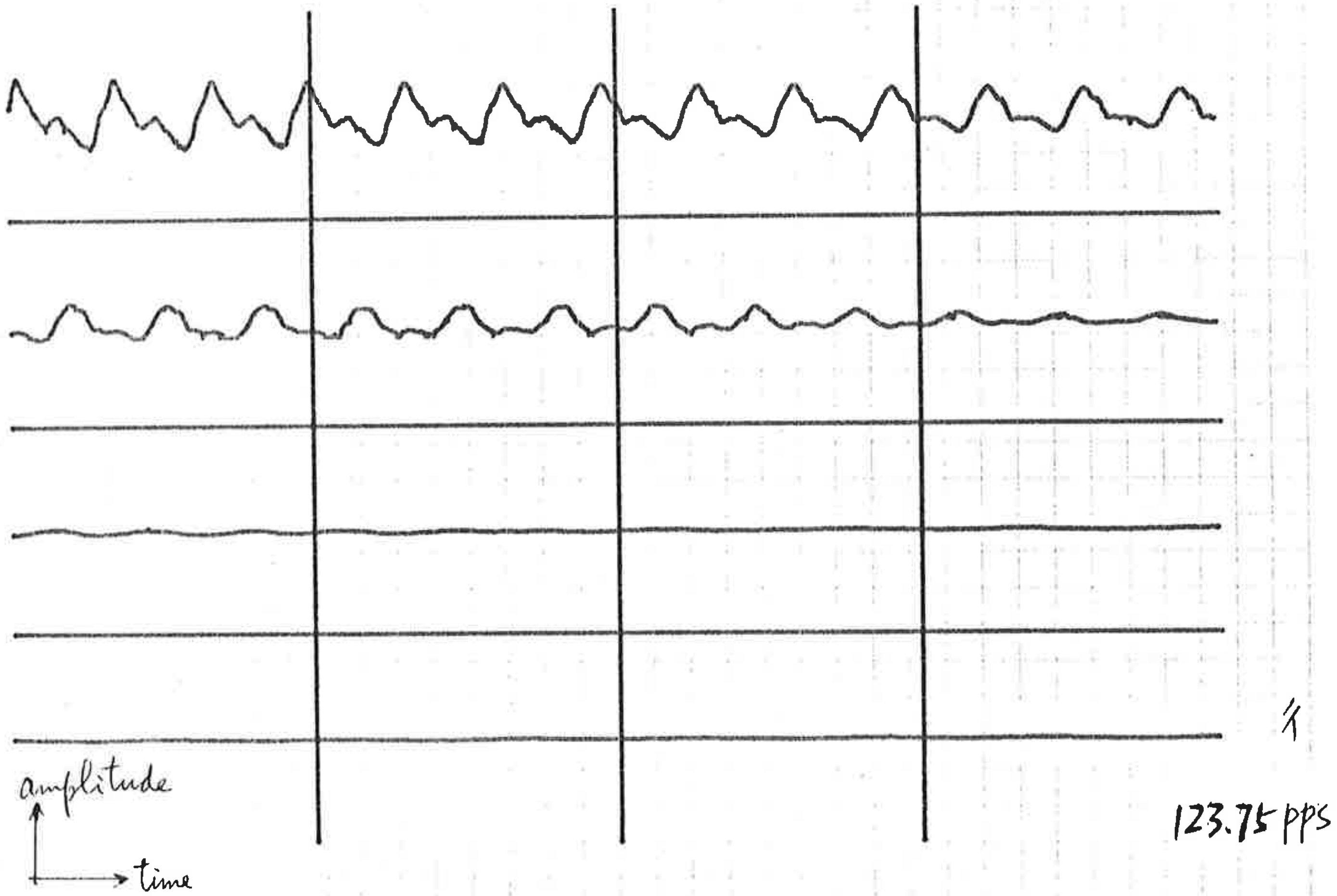

amplitude
time
Г
117.47 pps

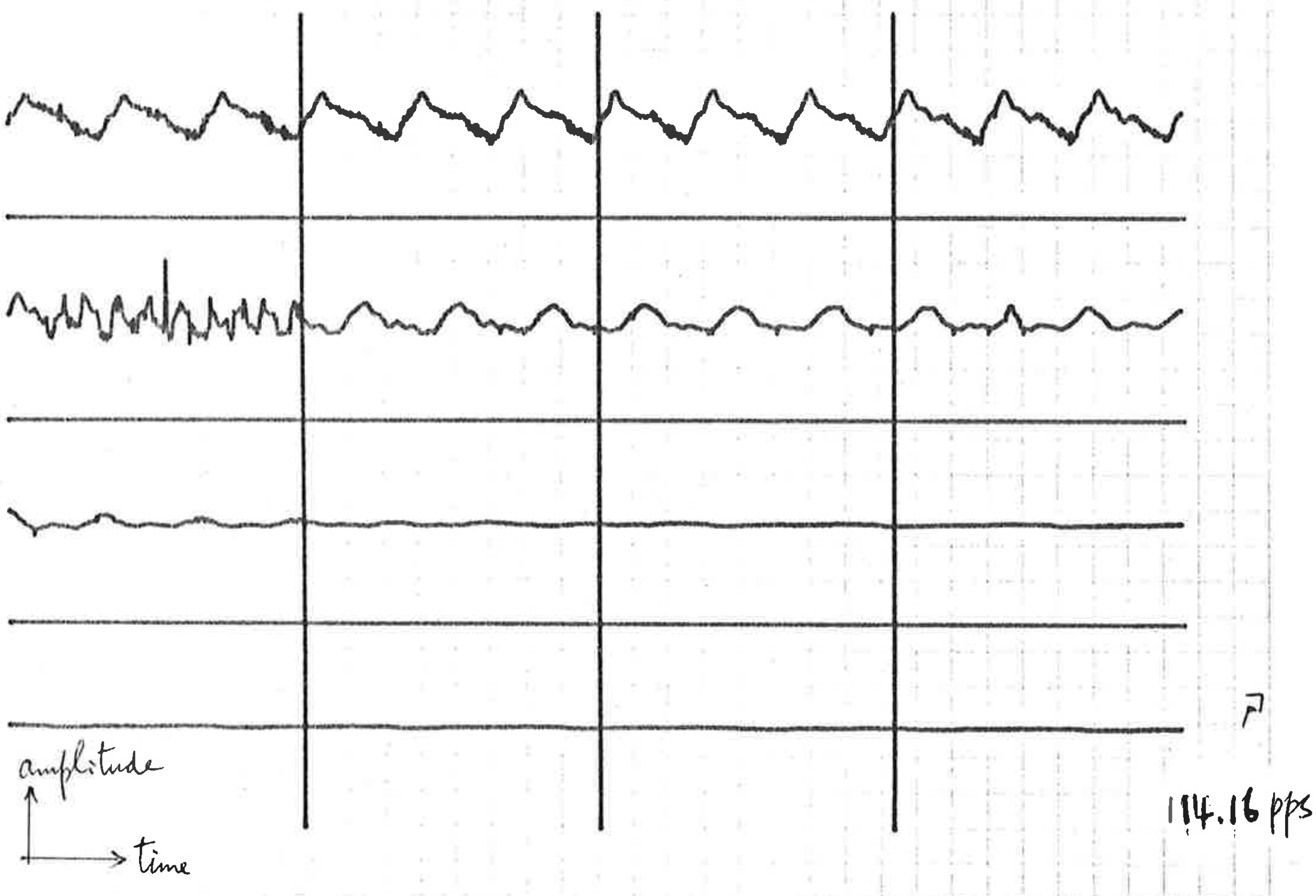

4
amplitude
time
120.98 pps

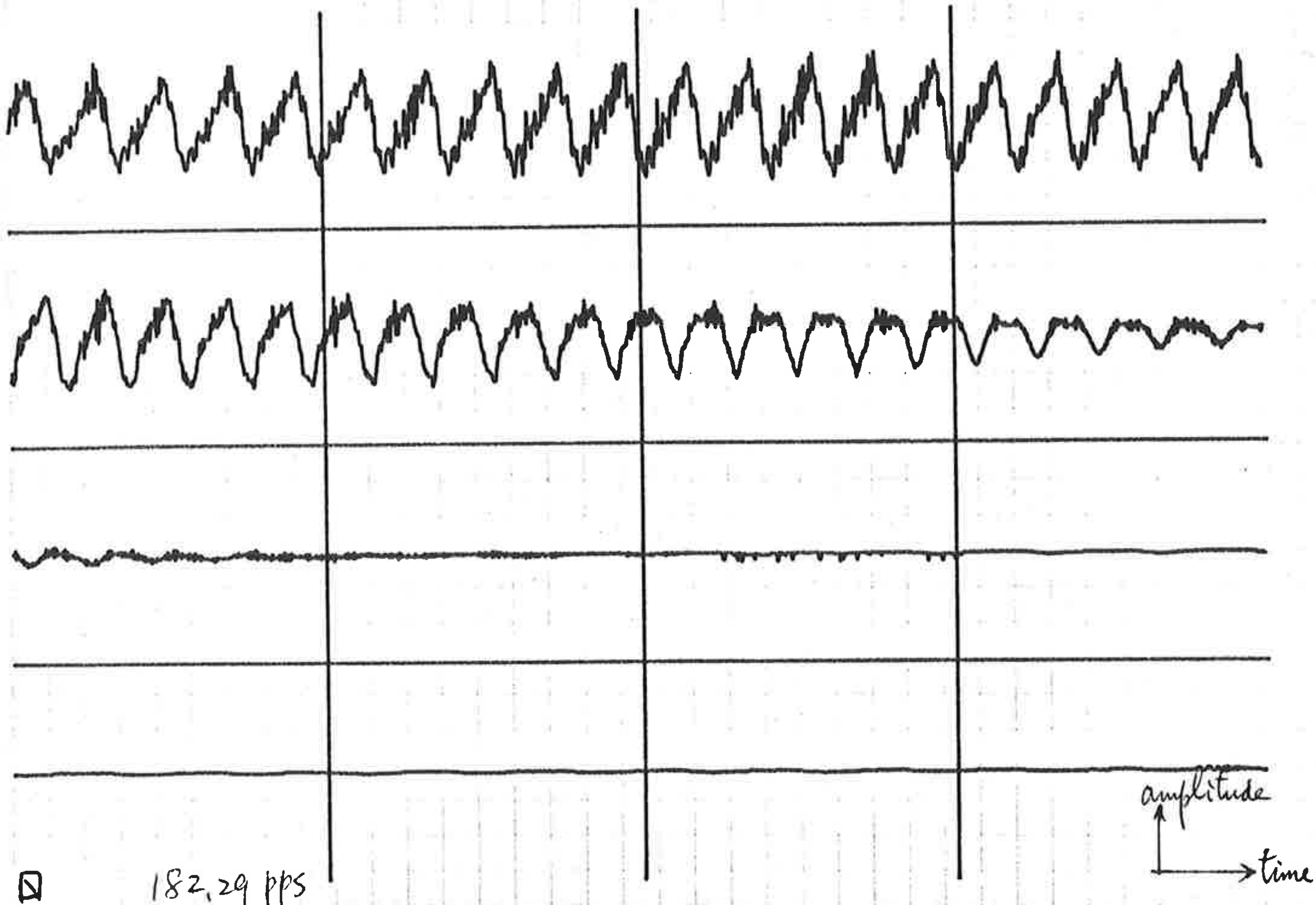

amplitude
time
127.65 pps


amplitude
time
ㄒ
121.89 pps

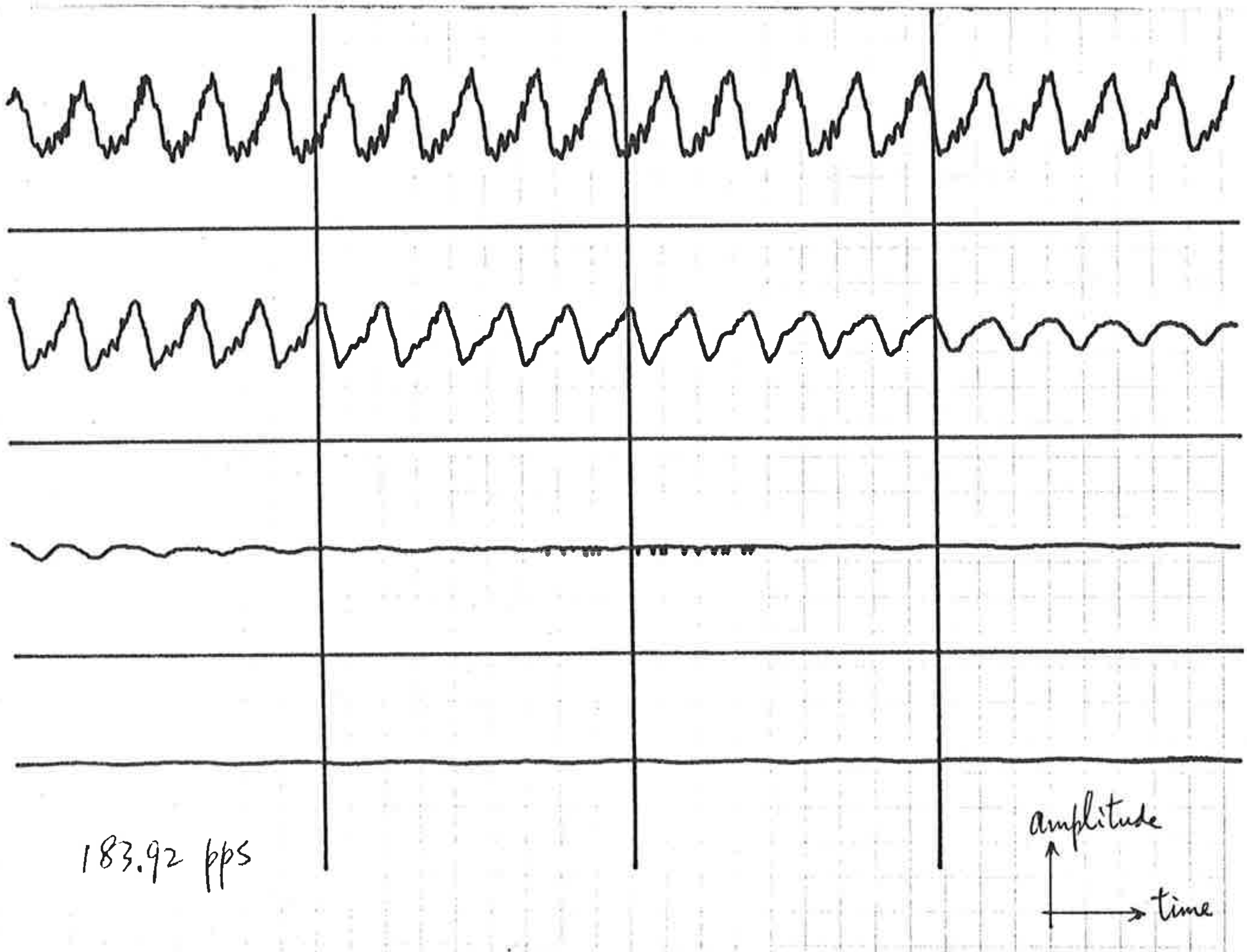

amplitude
time
ㄓ
115.79 pps


イ
amplitude
time
123.75 pps

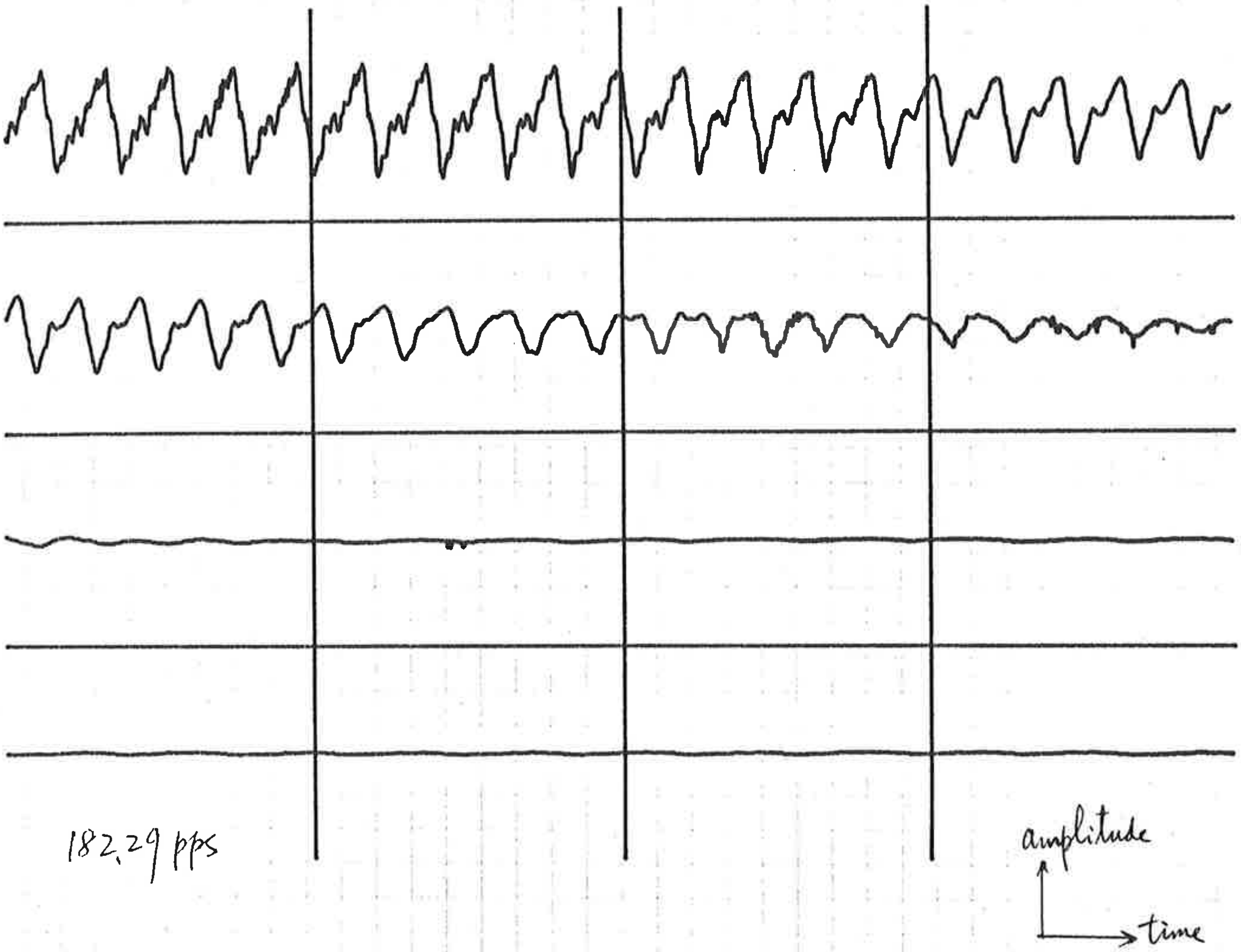

ア
amplitude
time
114.16 pps



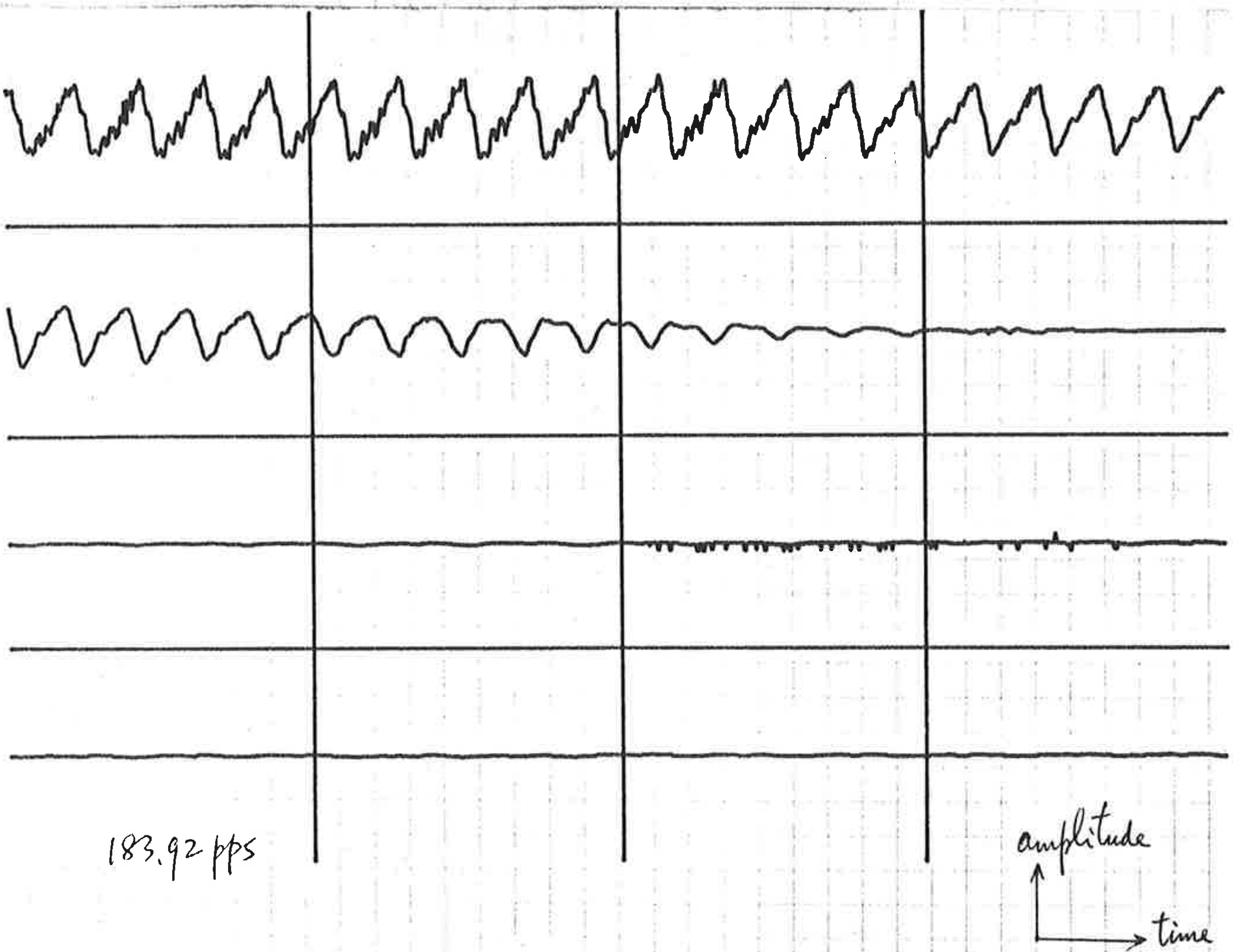
ʊ 182.29 pps



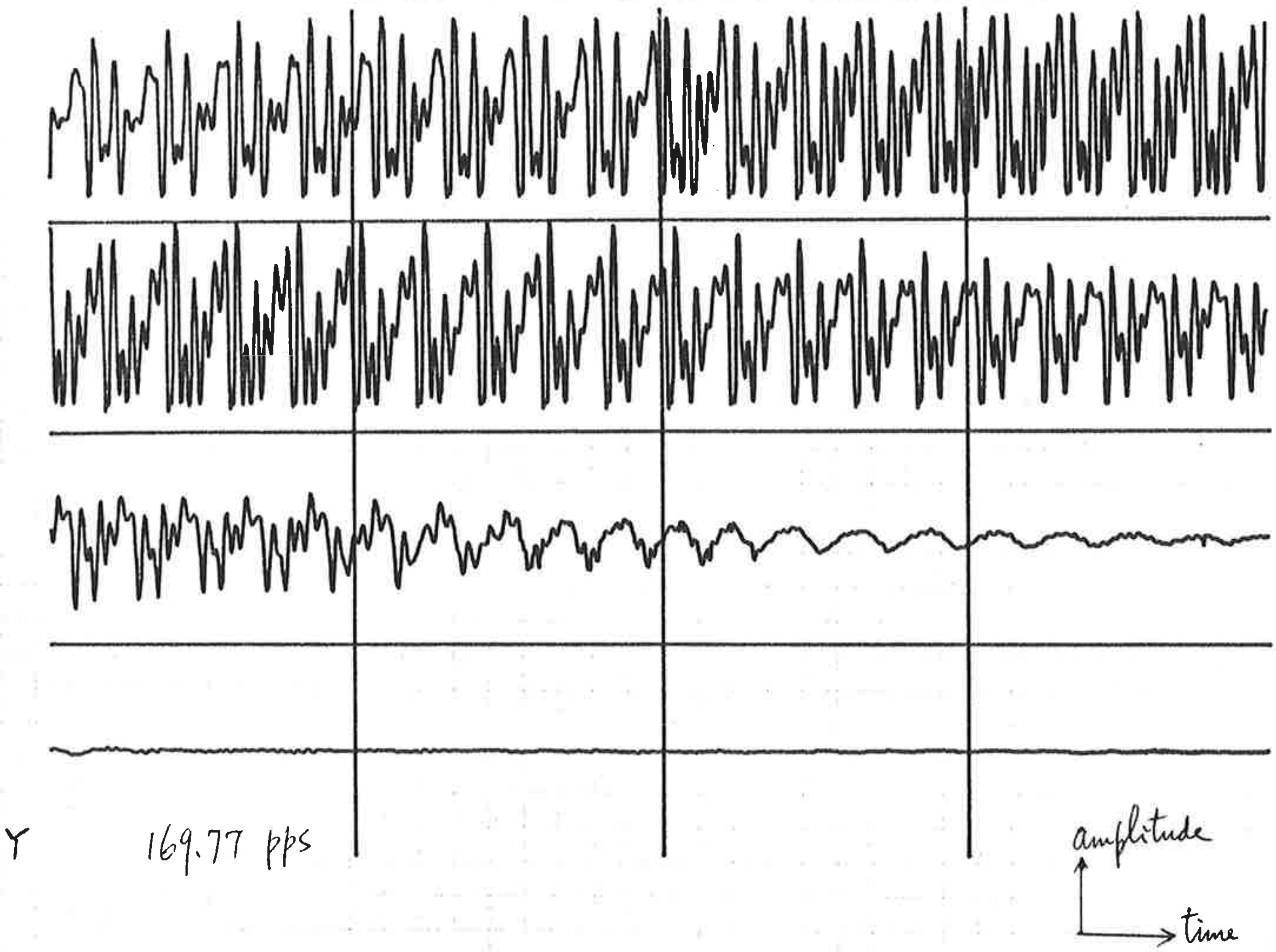
ɒ 183.92 pps

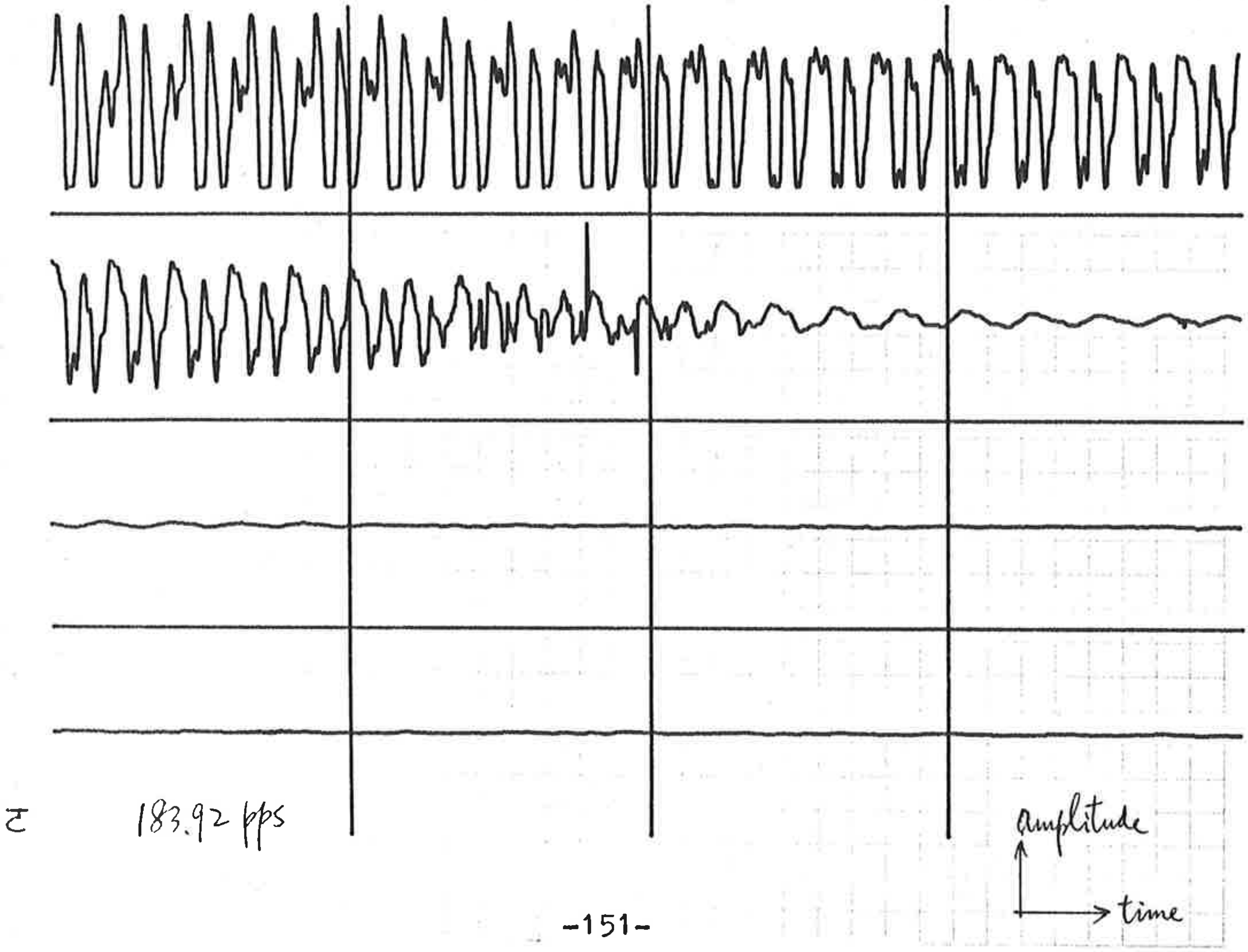


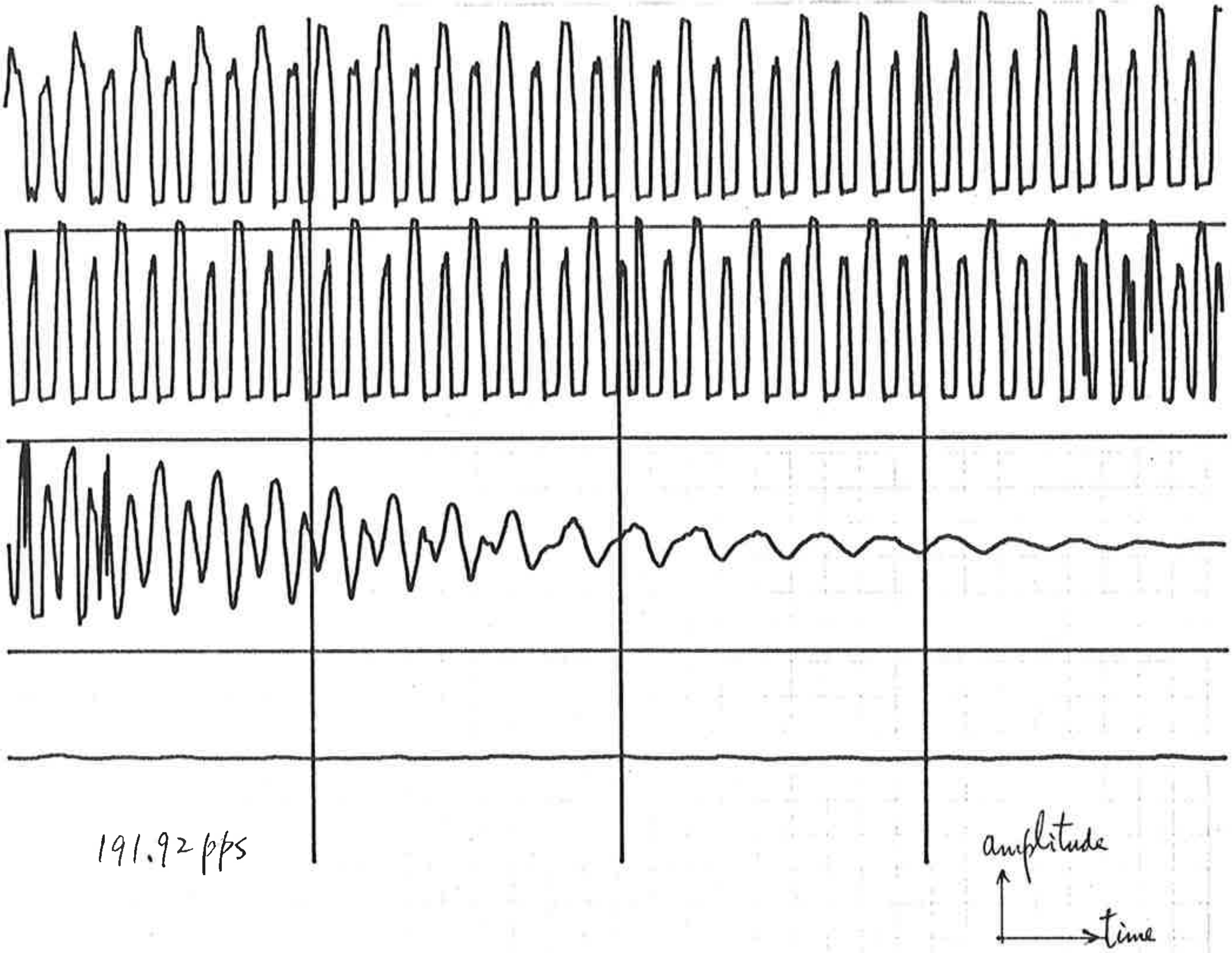
ㄅ 182.29 pps



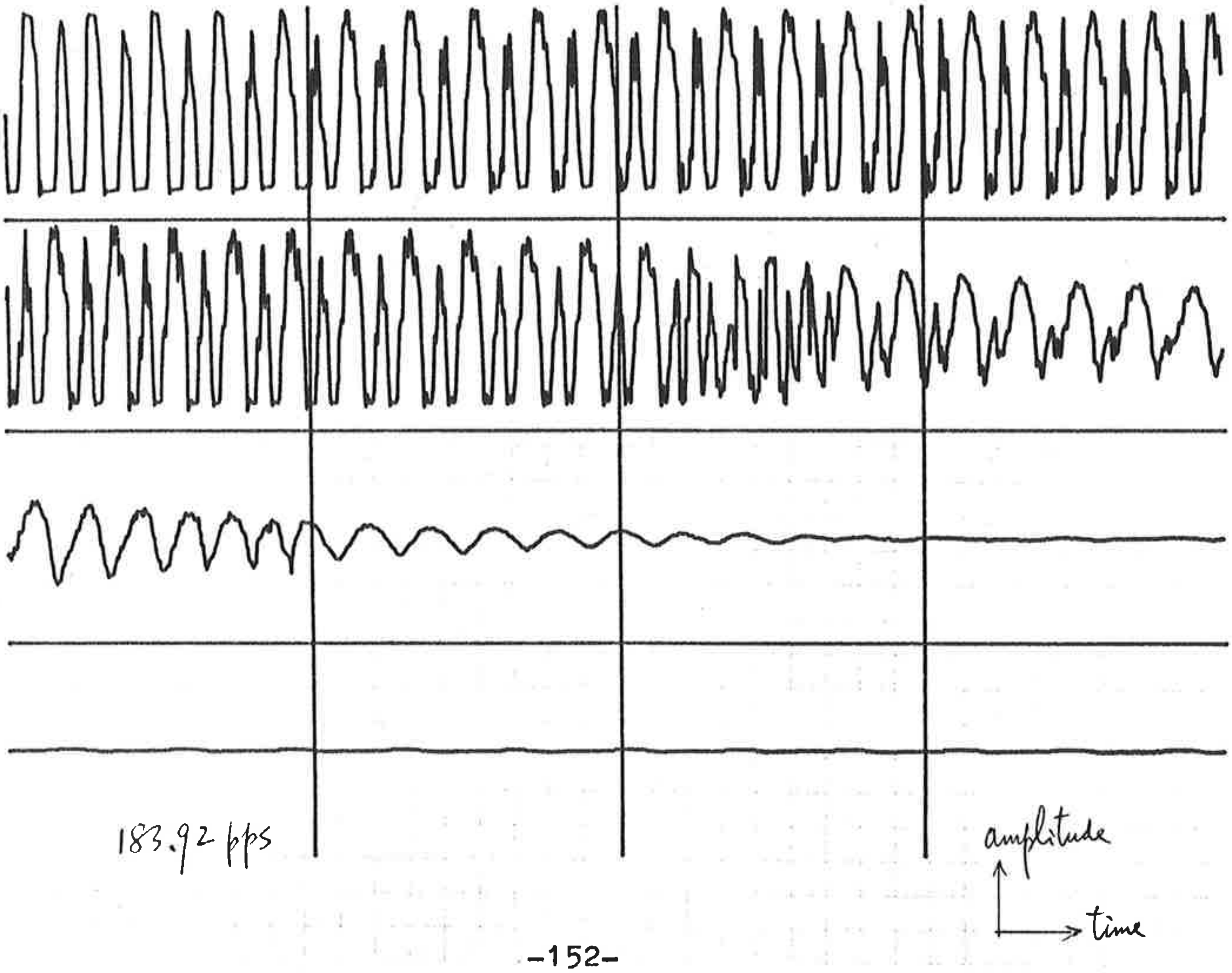
ㄙ 183.92 pps


Y
169.77 pps
amplitude
time

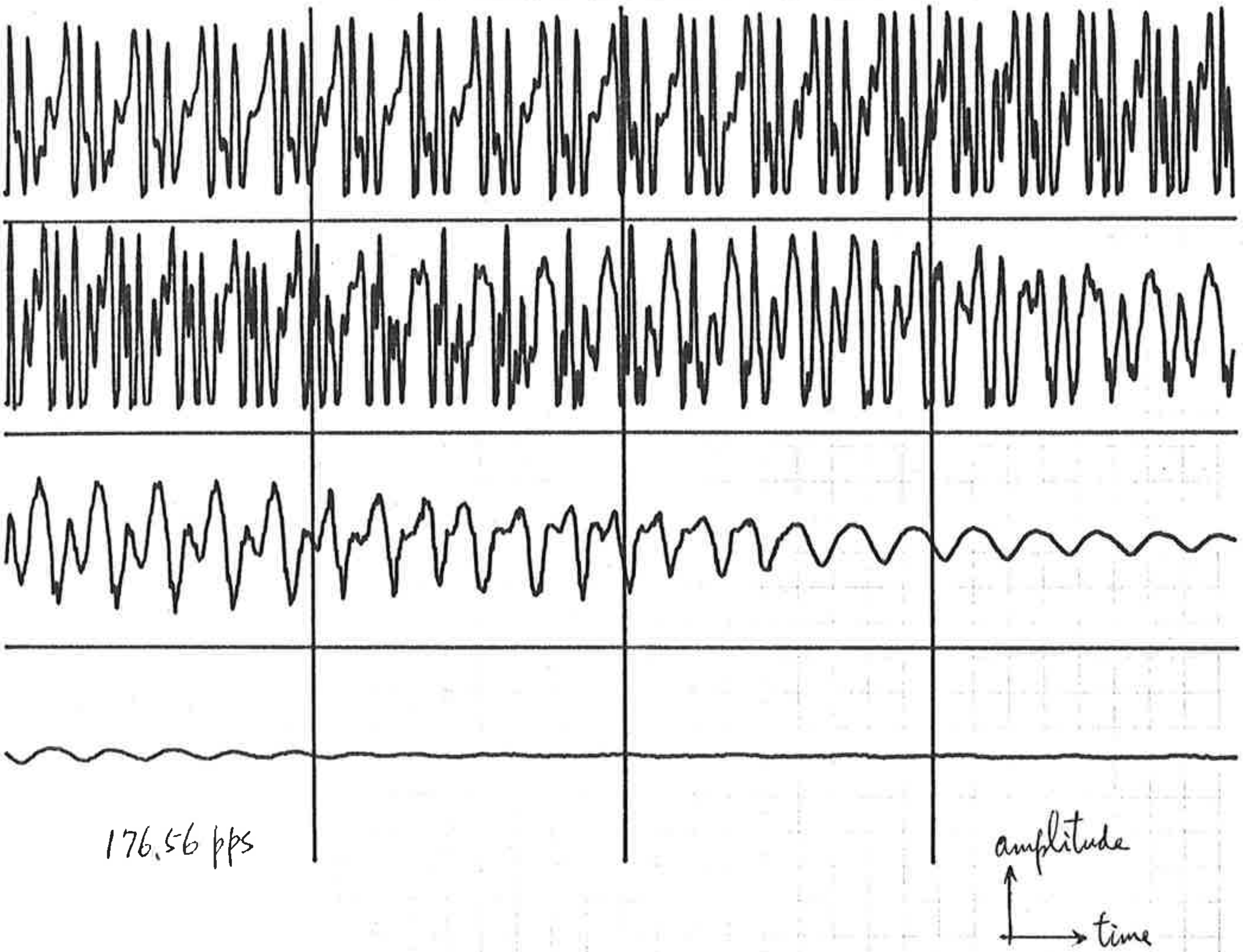

Z
183.92 pps
amplitude
time



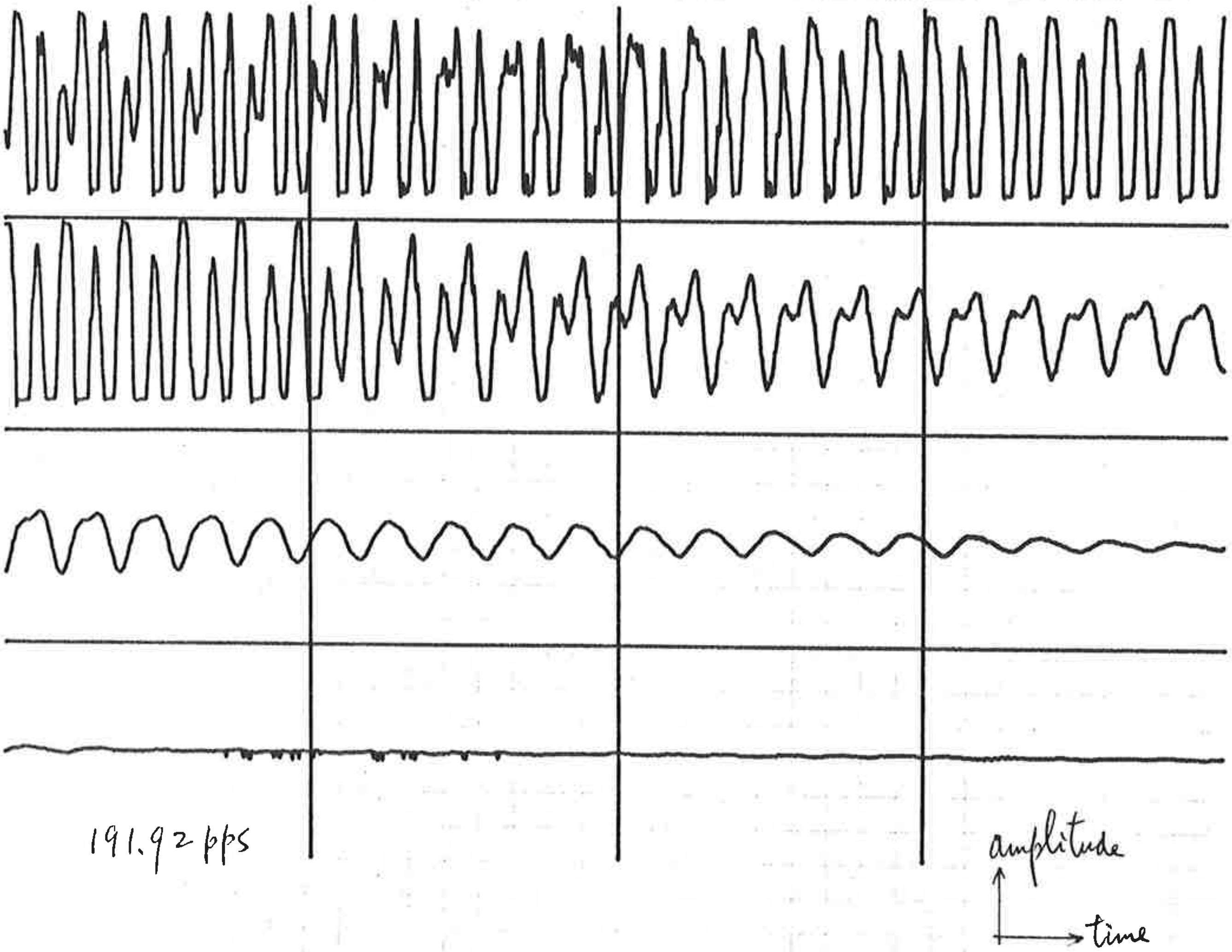
さ 191.92 pps



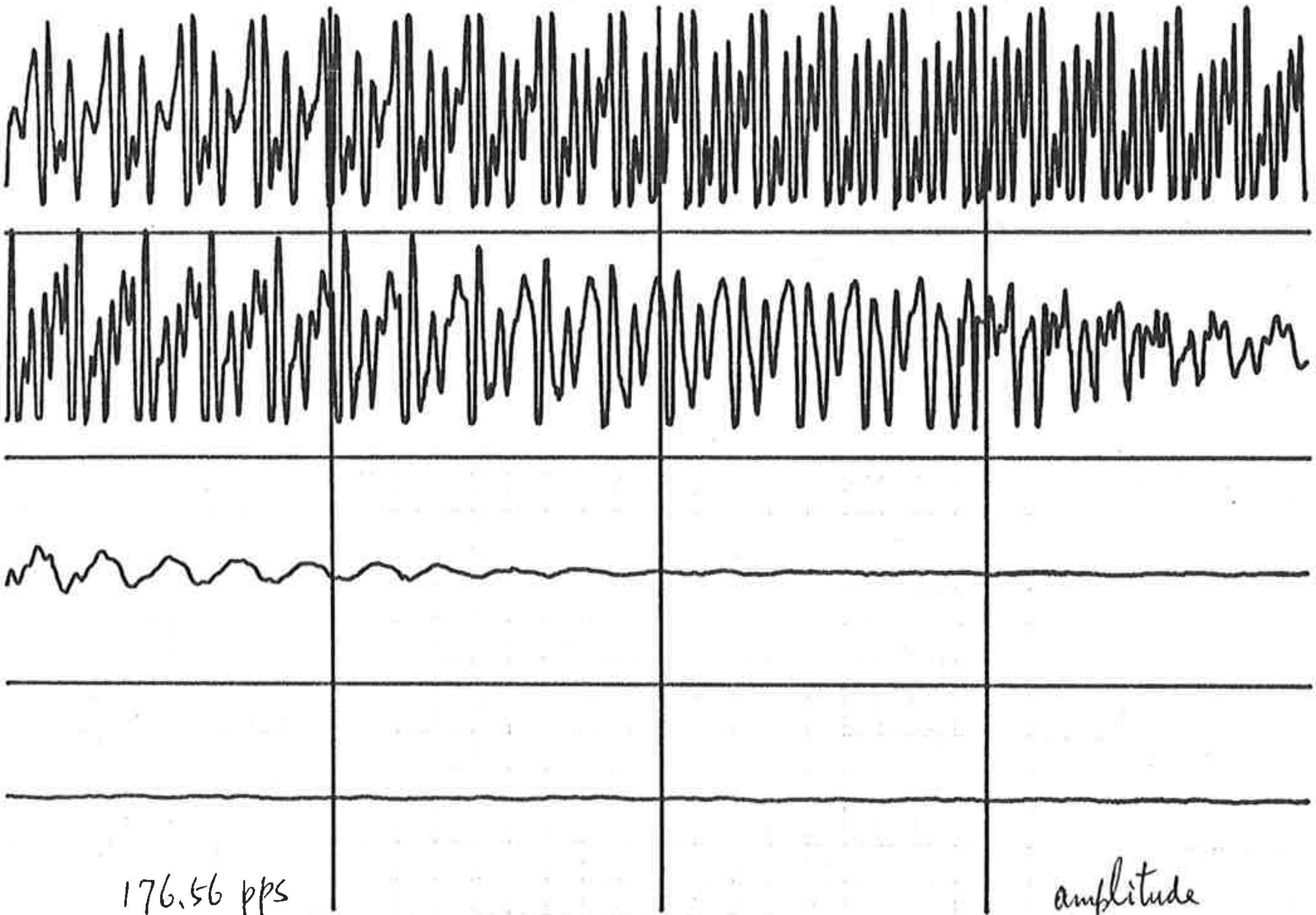
せ 183.92 pps



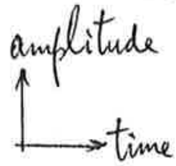
ホ　176.56 pps

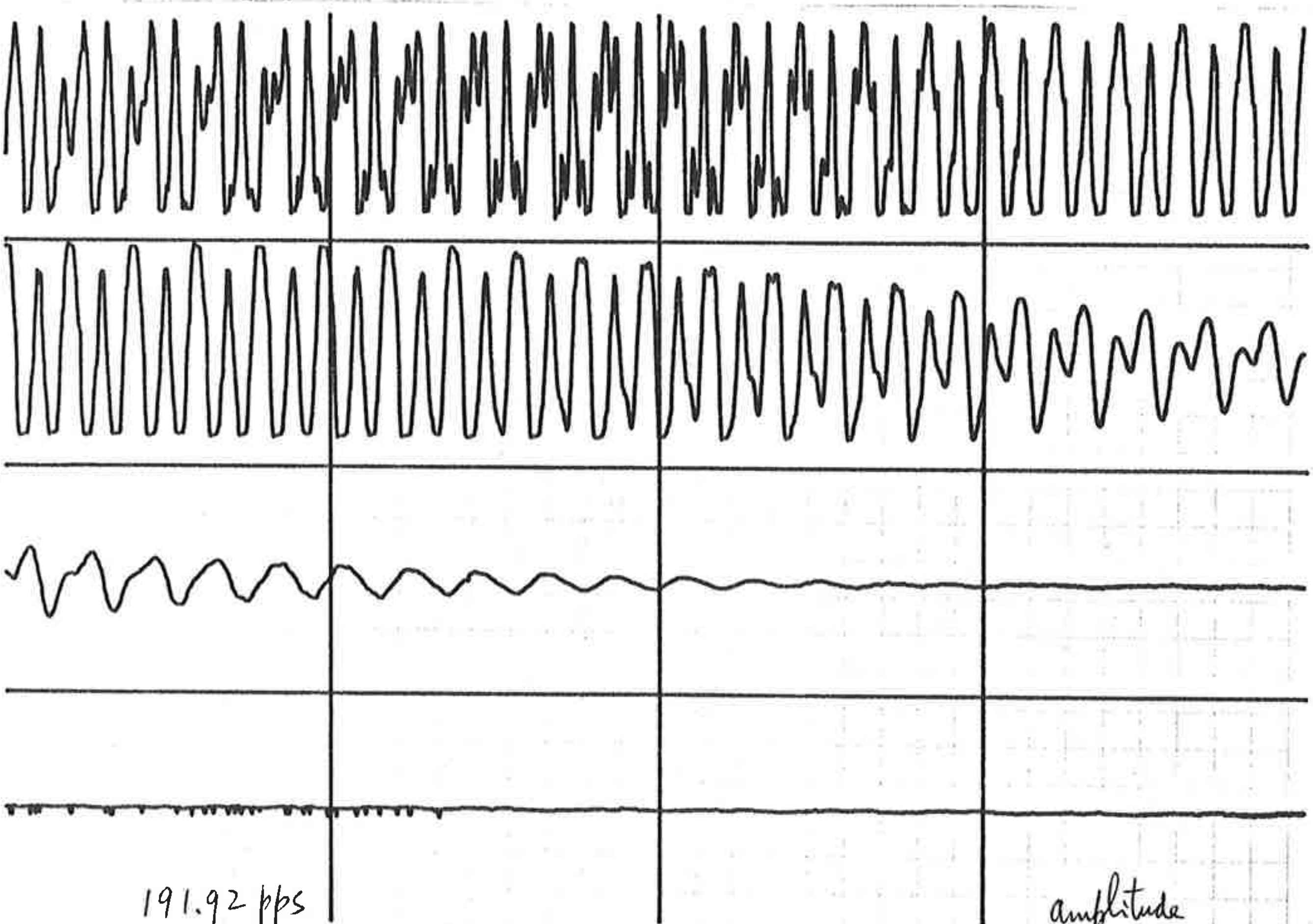


ヘ　191.92 pps

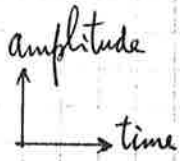


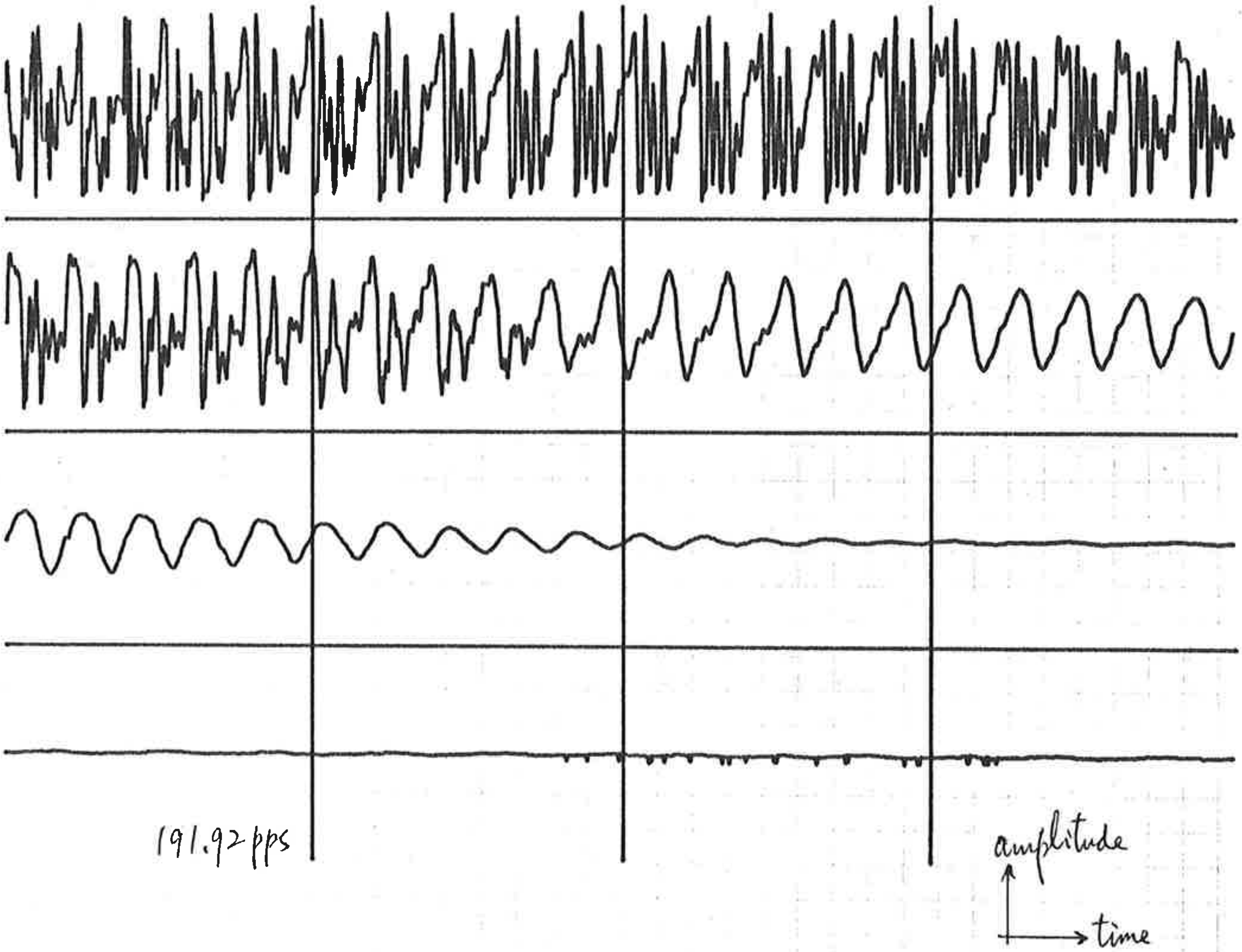
ㄠ 176.56 pps



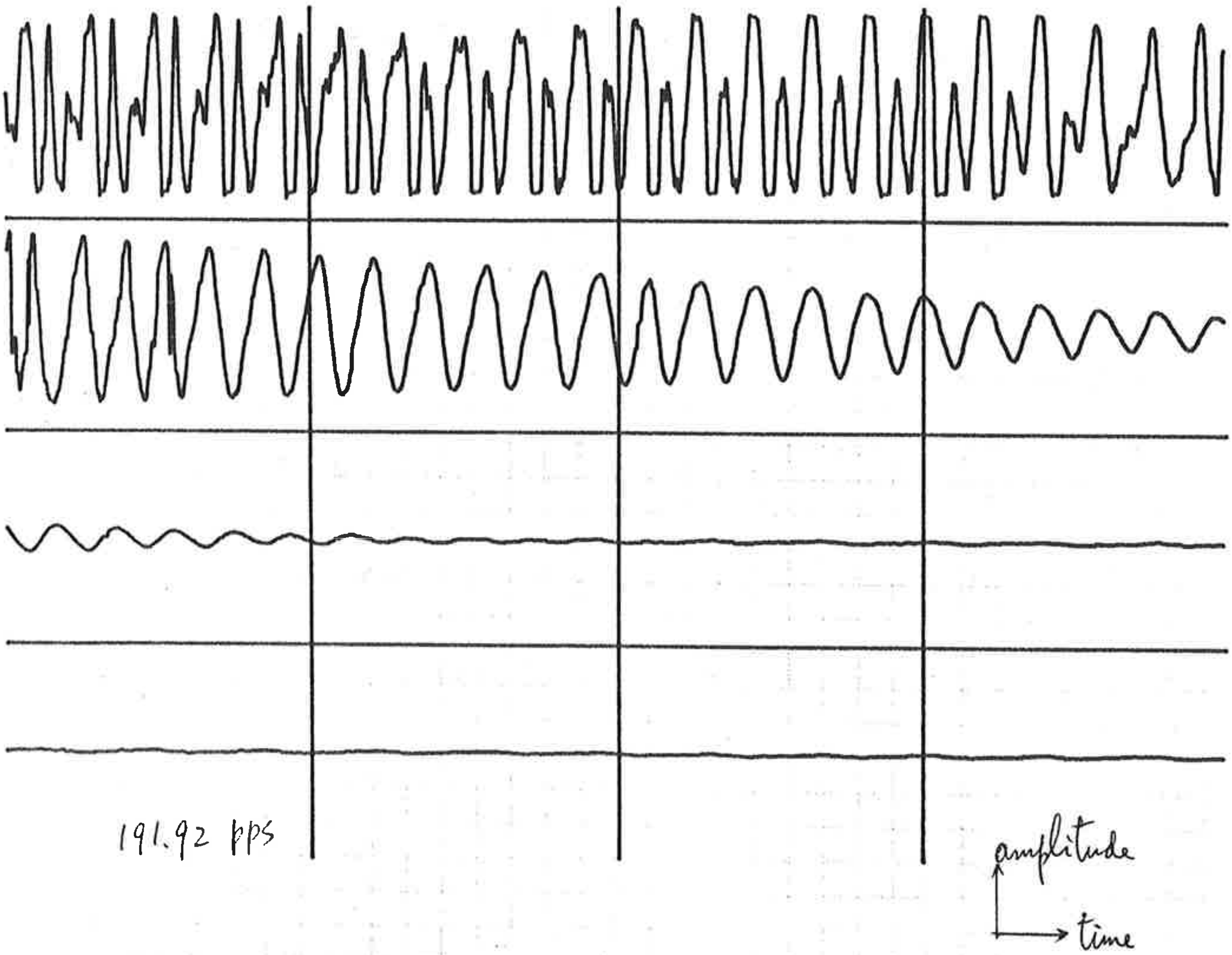


ㄡ 191.92 pps

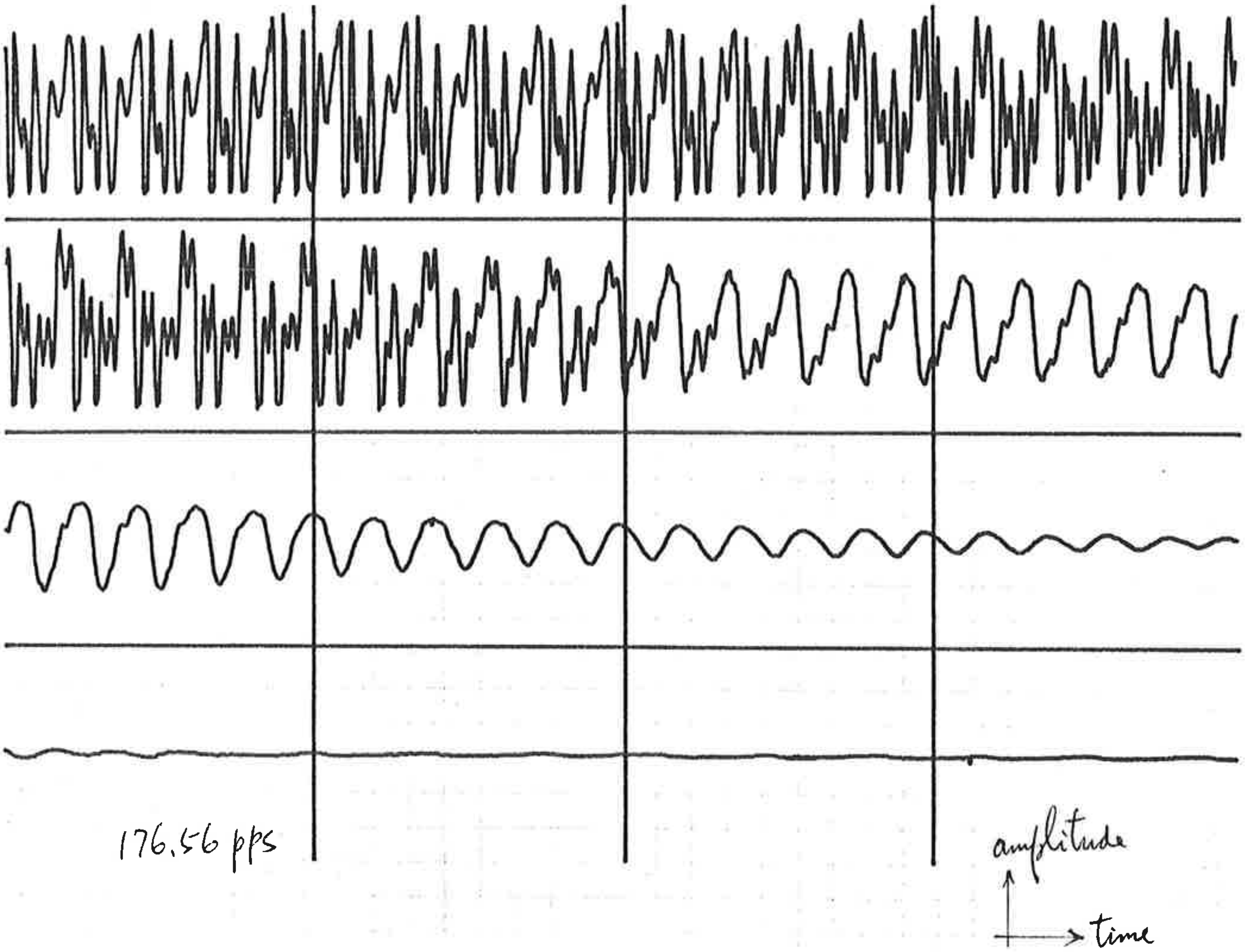


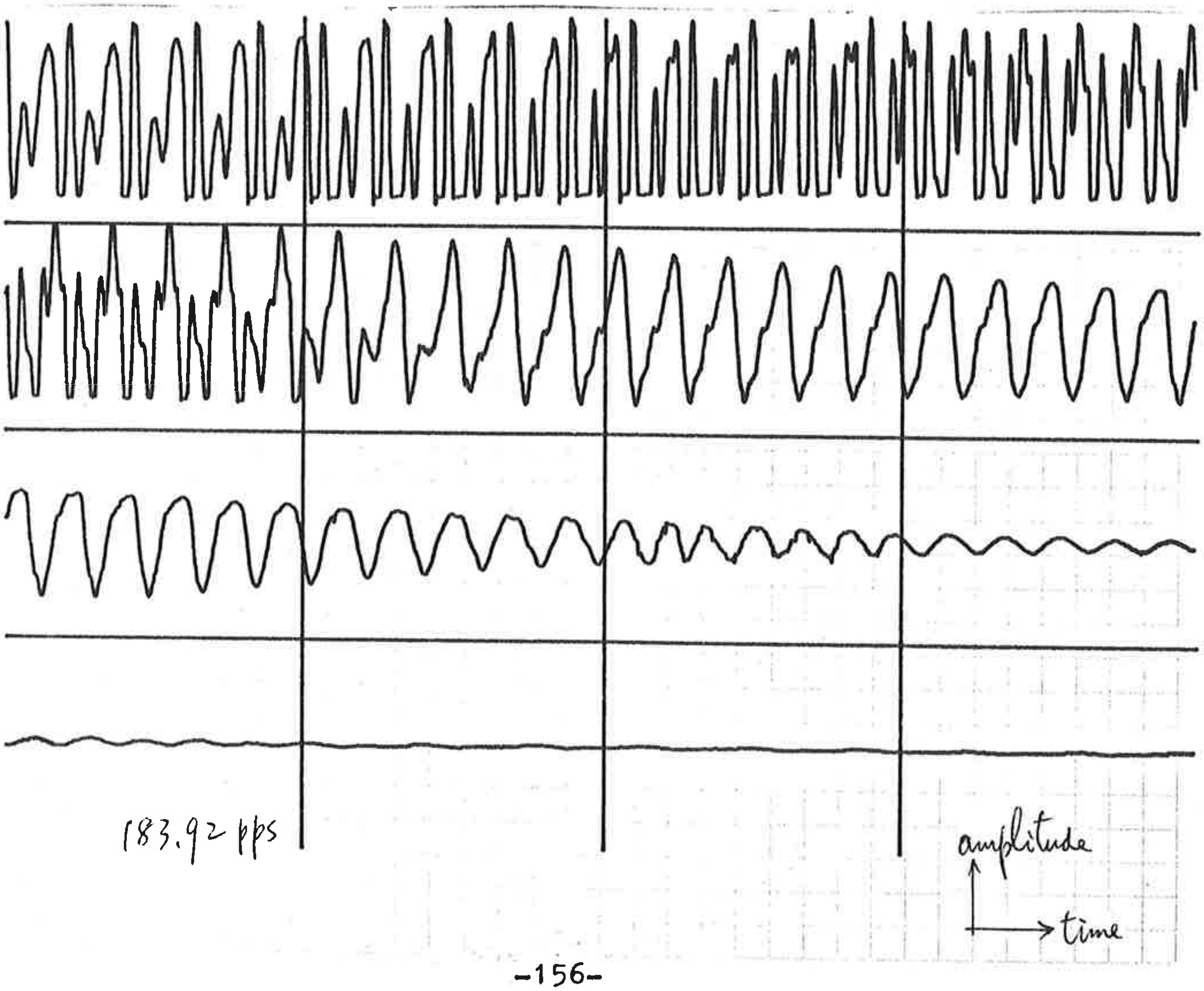


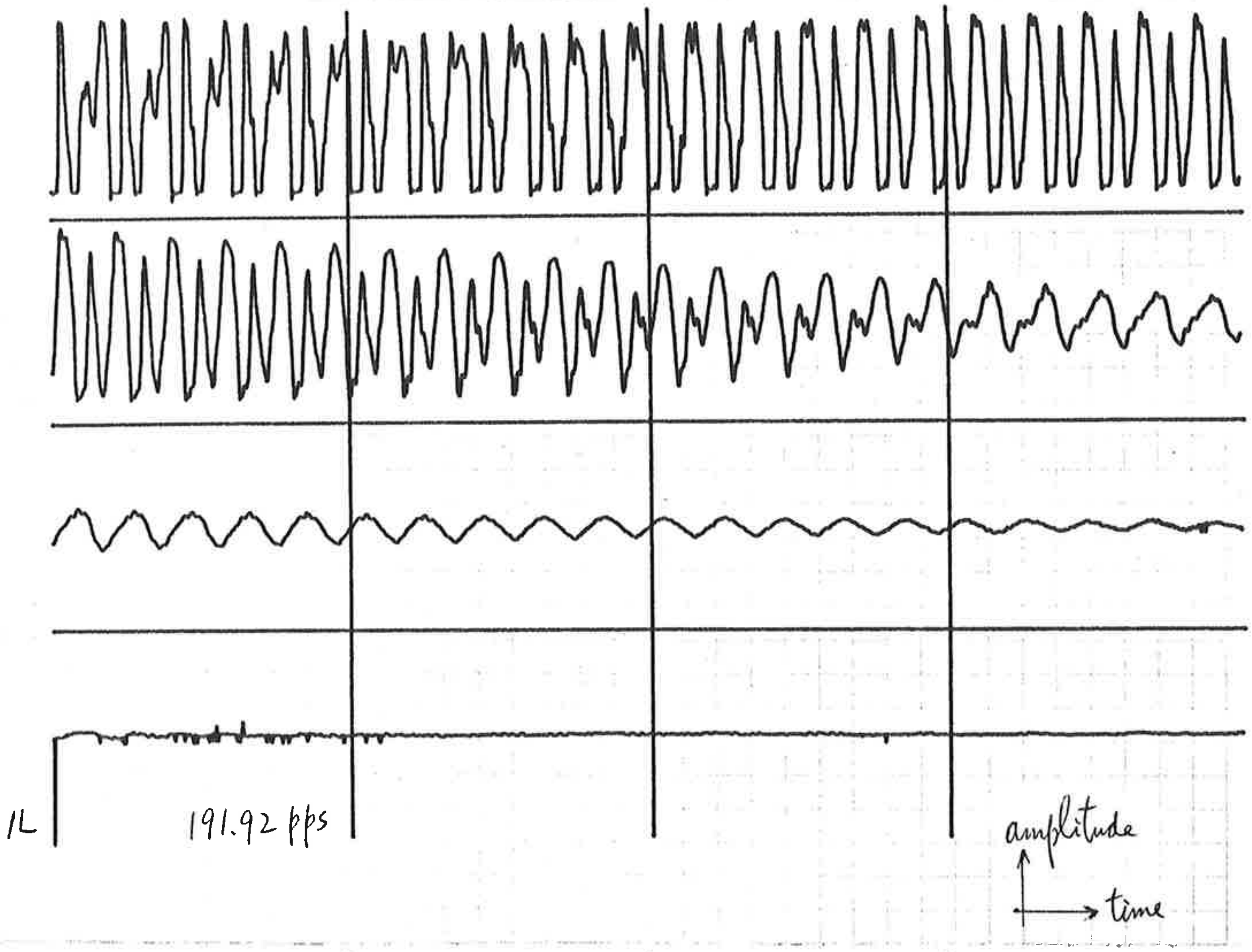
ヨ 191.92 pps

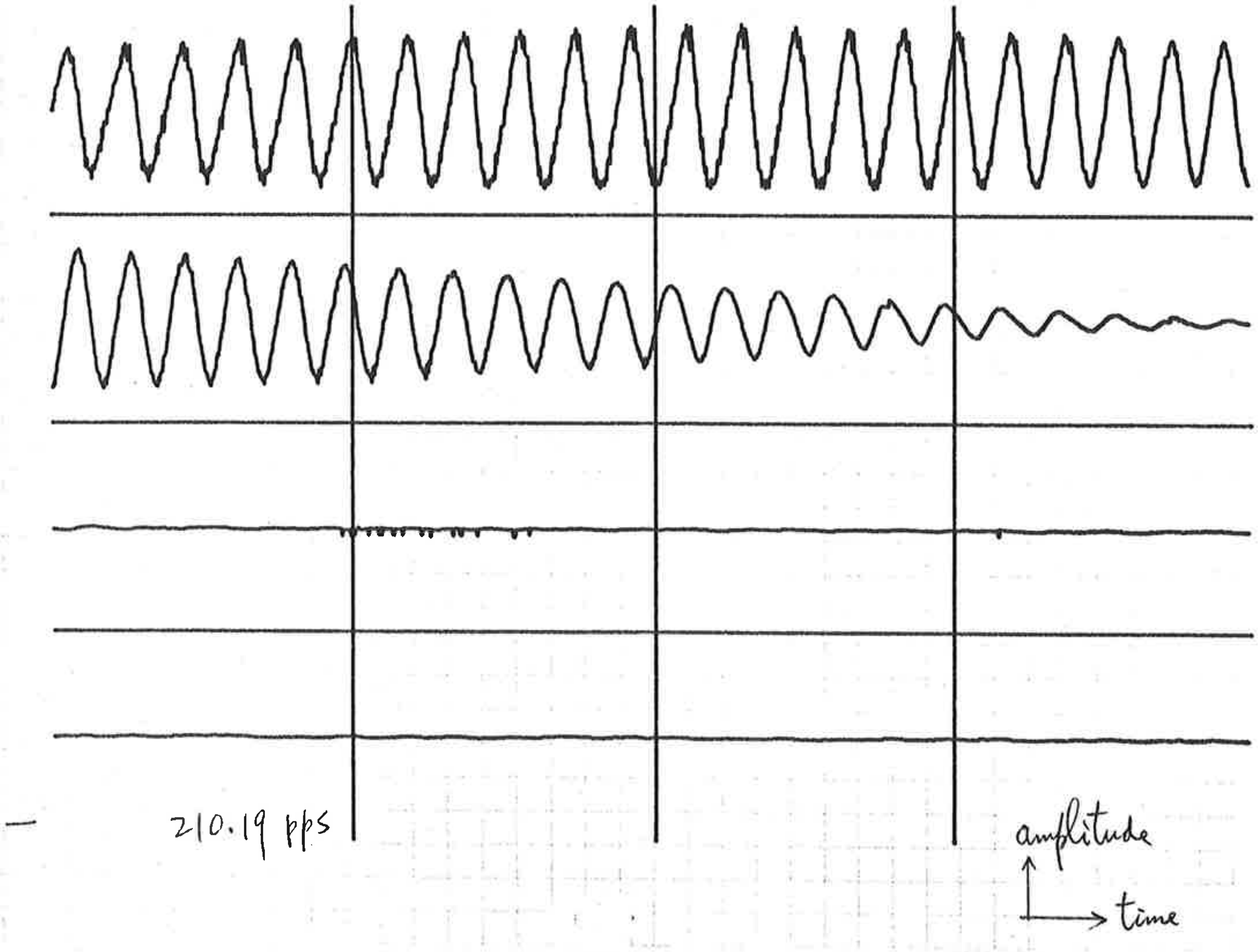


ウ 191.92 pps

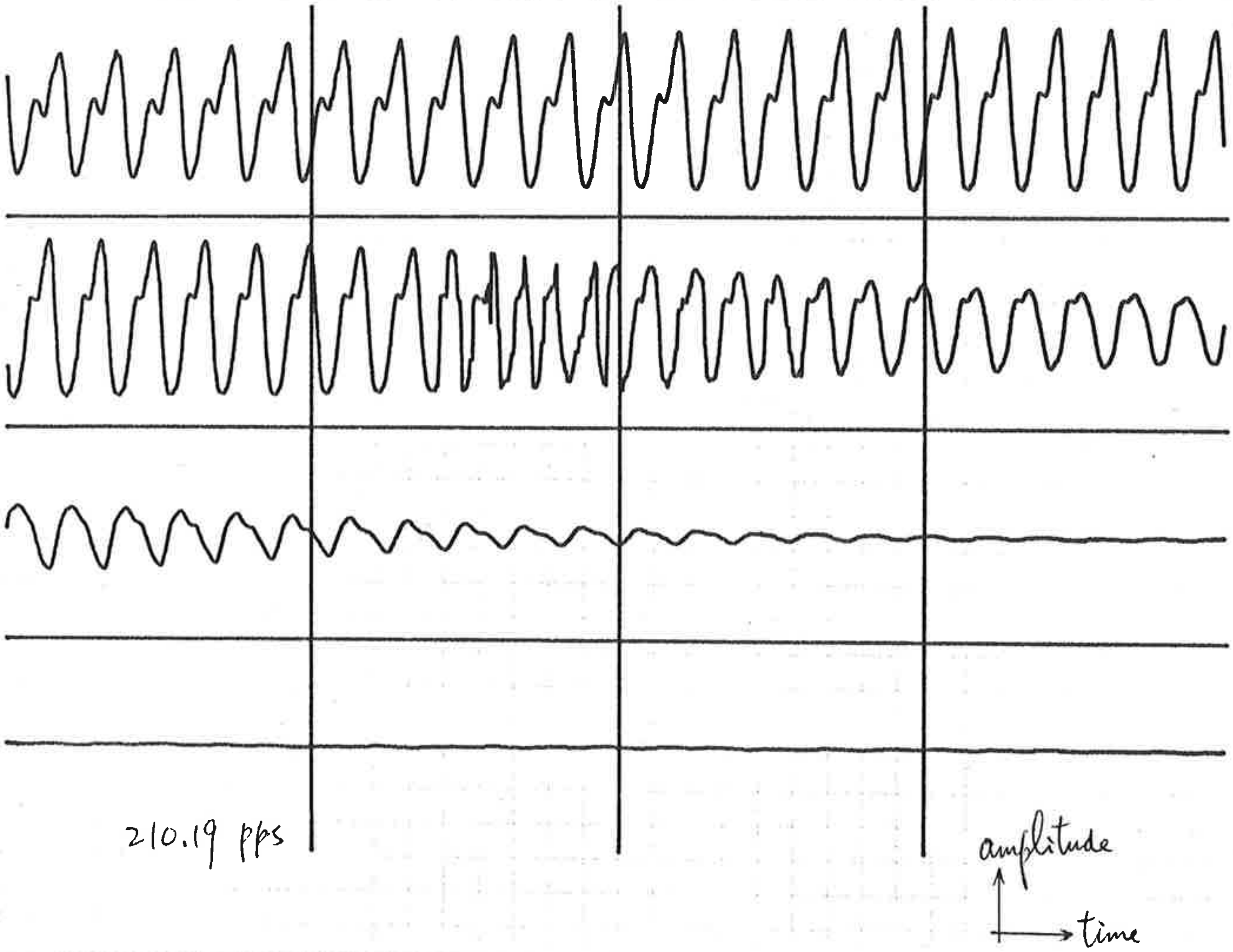
た


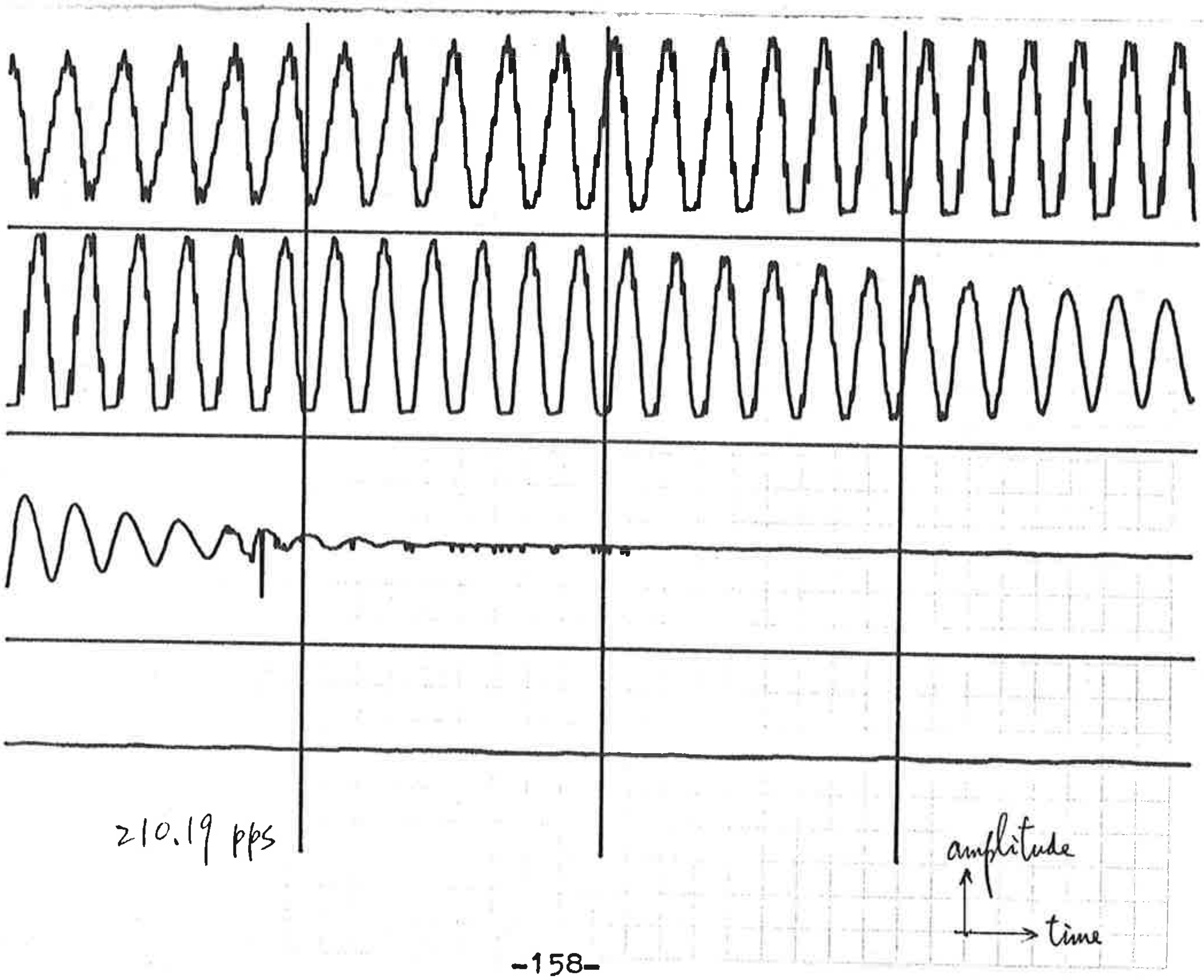
く



1L
191.92 pps
amplitude
time


210.19 pps
amplitude
time

X

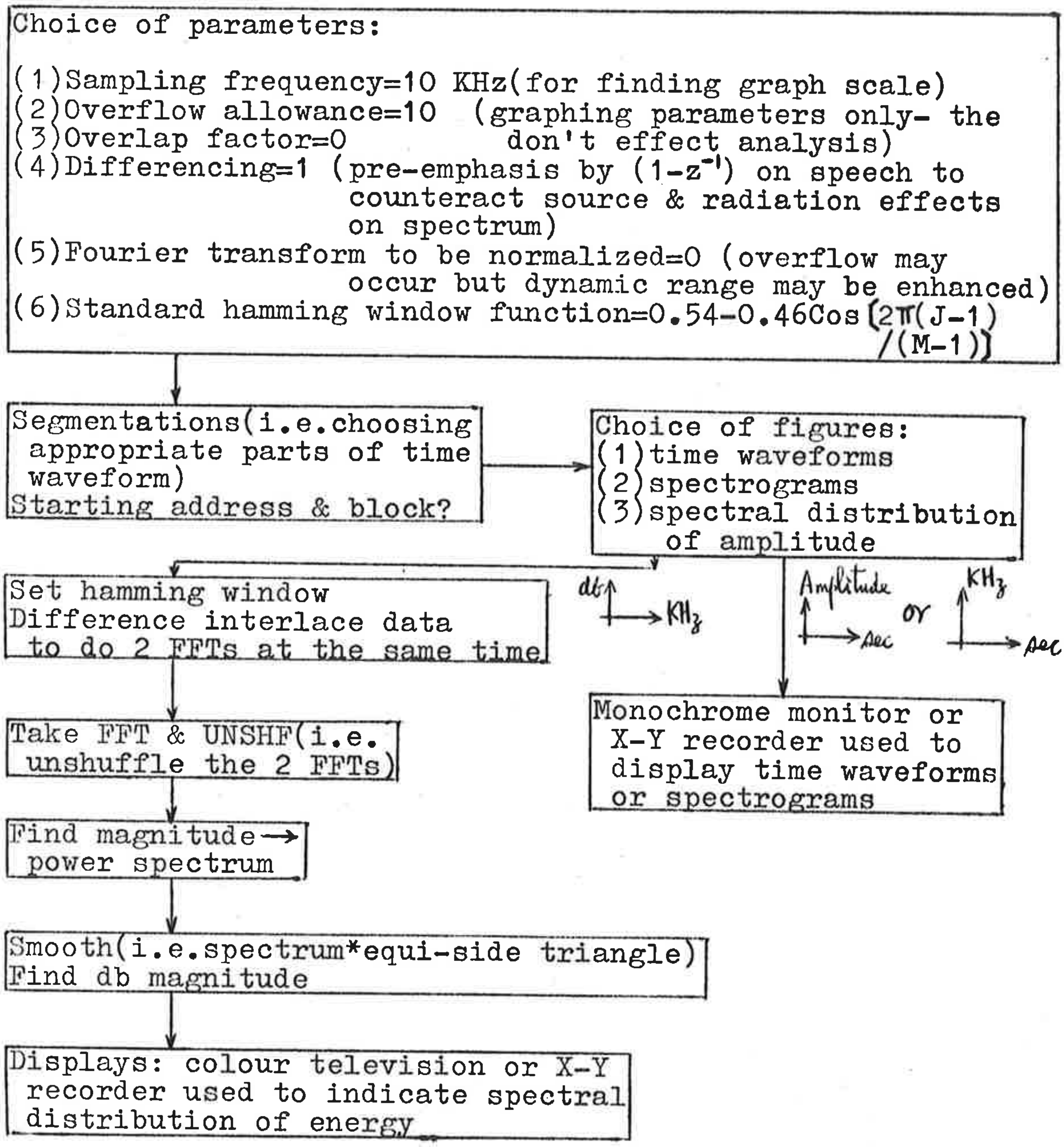




APPENDIX 6

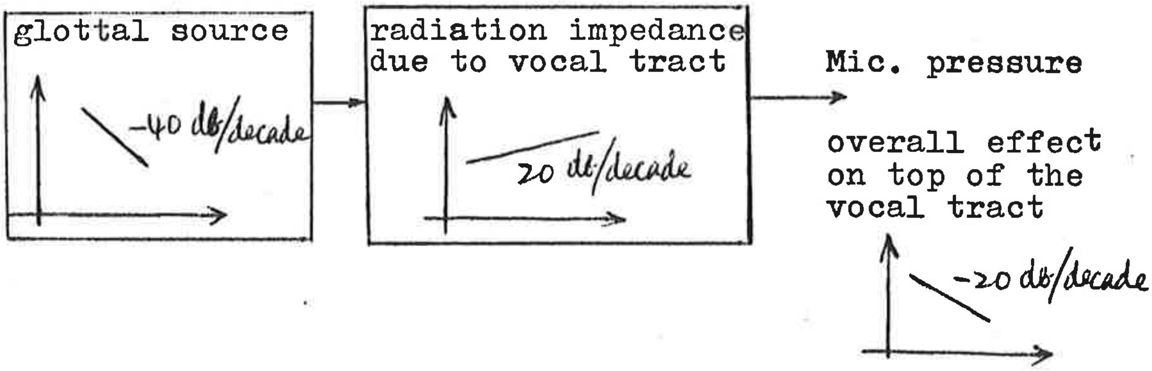
NOVA 2 Computer program- SPGRM   David Fensom (8)*

Block diagram of SPGRM (8)*



Terminology- explanations: (8)*

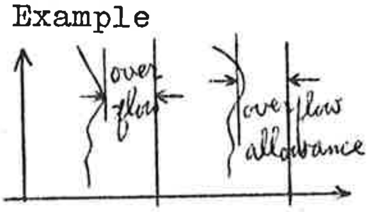
(1) Differencing:1 (i.e. to remove nett 20 db/decade droop causing by vocal organ)



Now linear prediction works best on a near white spectrum so we attempt to remove nett 20 db/decade droop by differentiation.

(2) Overflow allowance:10 (i.e. graphical constant which is numerical overflow to be permitted)

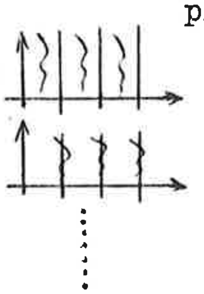
Example



Next spectrum bigger than first doesn't overflow into next graph's region as an allowance has been made for it.

(3) Overlap factor(i.e. proportional overlap number of plots in one segment)

Examples:



overlap factor=0 0%

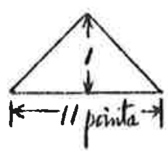
overlap factor=1 50%

⋮

overlap factor=n $\{n/(n+1)\}\cdot 100\%$

(4) Fourier transform to be normalized:0 (i.e. Fourier transforms were not normalized.)

(5) Smoothing: Spectrum is convoluted with equi-side triangle to get smoothed waveform. The figure of the used triangle is as in the left.



```
C         [illegible]
C         THIS PROG. GENERATES A SPECTROGRAPH OF A SECTION OF
C         THE SPEECH WAVEFORM HELD IN FILE DATA.JN OF THE
C         DISK IN DRIVE 1.THE ANALYSIS SPANS 512 DATA POINTS
C         WHICH ARE WINDOWED,FFT'ED,THE MAGNITUDES FOUND AND
C         THEN LUXURIOUSLY SMOOTHED WITH AN 11 POINT TRIANGULAR
C         FUNCTION.LASTLY LOGS ARE TAKEN TO GIVE A DB-LIKE
C         MEASURE.IF MORE THAN ONE SPECTRUM IS TO BE GENERATED
C         THEN SUCCESSIVE ANALYSIS FRAMES OVERLAP 50%.VARIOUS
C         PARAMETERS ARE SOUGHT WHICH CONTROL THE LAYOUT AND SIZE
C         OF THE OUTPUT GRAPHS.FURTHER PARAMETERS CONTROL
C         PRE-EMPHASIS(APPROX +6DB/OCTAVE BY DIFFERENCING
C         CF DIFFERENTIATION) AND WINDOWING OF THE INPUT TIME SERIES.
C         AS WITH ALL PERIODOGRAMMES THERE IS A DIFFICULTY IN
C         INTERPRETATION OF THE RESULTANT SPECTRUM AND THIS IS
C         COMPOUNDED IN THIS CASE BY THE SMOOTHING OPERATION.A
C         LATER PROG "SPECTRA" IS UNSMOOTHED.
C         DAVID FENSOM 12/5/76
          COMMON IMAP(8,0:9),IXORG,IYORG,XGRAD,XCONST
          COMMON YGRAD,YCONST,XB,YB,XT,YT,INTS
          DIMENSION IARRAY(1792),X(256),Y(256),W(9)
          CALL OPEN(2,"$TTI",3,IERR)
          CALL OPEN(3,"$TTO",3,IERR)
          PI=3.14159265
          XB=0.
          YB=0.
          INTS=3
          WRITE (3) " SAMPLING FREQ (HZ) ?"
          READ (2) SFRQ
C
1000      WRITE (3) " CHANGE DISPLAY PARAMETERS ?"
          READ (2) IFLAG
          IF(IFLAG.NE.1)GO TO 22
          WRITE (3) " GRAPH FRACTIONAL SIZE ?(1,2,4,8) "
          READ (2) ISIZE
          WRITE (3) " HARD COPY ? "
          READ (2) IFLAG0
C         THE OVERLAP AND OVERFLOW FACTORS ARE USED WHEN A SEQUENCE OF
C         SPECTRA ARE TO BE PLOTTED ON 1 GRAPH.EACH SPECTRUM IS
C         DRAWN IN A BOX AND IF O'LAP FACTOR=0 THE BOXES SIMPLY
C         SIT ON TOP OF EACH OTHER.IF O'LAP=1 THERE IS A 50% OVERLAP
C         IF O'LAP=N THE OVERLAP=N/(N+1)'THS.THE SCALING USED FOR
C         DRAWING EACH SPECTRUM IS DETERMINED BY THE TOTAL NO
C         OF BOXES REDUCED BY THE COMMON OVERLAPPING AMOUNT AND
C         THE RANGE OF THE 1ST SPECTRUM EVALUATED.TO ALLOW FOR AN
C         ATYPICAL SPECTRUM A +VE OR -VE OVERFLOW ALLOWANCE IS PROVIDED
C         AND IS ADDED TO THE RANGE OF THE 1ST SPECTRUM WHEN THE SCALING IS
C         EVALUATED.
          WRITE (3) " CHANGE IN OVERLAP OR OVERFLOW ALLOWANCE ? "
          READ (2) IFLAG1
          IF(IFLAG1.NE.1)GO TO 22
          WRITE (3) " OVERLAP FACTOR ? "
          READ (2) OLAP
          WRITE (3) " OVERFLOW ALLOWANCE ? "
          READ (2) EXTRA
C
22        WRITE (3) " ANALYSE NEW DATA ? "
          READ (2) IFLAG2
          IF(IFLAG2.NE.1)GO TO 23
          WRITE (3) " VIEW TIME SERIES ? "
          READ (2) IFLAG2
          IF(IFLAG2.EQ.0)GO TO 13
          CALL CHEX(IARRAY(1),IBLGO,IADGO,II,JJ,0)
13        IF(IFLAG0.EQ.1)GO TO 28
          XT=1023.
          YT=767
```

```
          CALL VECUL
          CALL FNEWP
          GO TO 29
28        XT=9999.
          YT=9999.
          CALL VECHP
29        WRITE (3) " NO. OF SPECTRA ?"
          READ (2) INO
C
23        WRITE (3) " CHANGE ANALYSIS ?"
          READ (2) IFLAG3
          IF(IFLAG3.NE.1)GO TO 24
          WRITE (3) " DIFFERENCING ? "
          READ (2) LDIFF
          WRITE (3) " FOURIER TRANSFORM TO BE NORMALISING ? "
          READ (2) INORM
          WRITE (3) " WINDOWING FN C1+C2*COS(2PI(J-1)/(M-1)) "
          WRITE (3) " J=SAMPLE INDEX,M=WINDOW LENGTH(FIXED TO 512 PTS)"
          WRITE (3) " SUCCESSIVE WINDOWS OVERLAP 50% IN ANALYSIS "
          WRITE (3) " C1=0.54,C2=-0.46 IS STANDARD HAMMING "
          WRITE (3) " C1,C2 ? "
          READ (2) C1,C2
C
24        WRITE (3) "<014>"
          CALL FDELY(8)
          WRITE (3,25) OLAP,EXTRA,IBLGO,IADGO,INO
25        FORMAT(1H1,1X,"PARAMETER DETAILS",/,1X,"OVERLAP FACTOR = ",
     1 F3.0,/,1X,"O.FLOW ALLOWANCE = ",F4.1,/,1X,"STARTING BLOCK = ",
     1 I4,5X,"STARTING ADDRESS = ",I3,/,1X,"NO.OF SPECTRA = ",I3,/)
          AMIN=1.E10
          AMAX=0.
          WRITE (3,27) LDIFF,INORM,C1,C2
27        FORMAT(1X,"1 FOR TRUE,0 FOR FALSE  DIFFERENCING?",I1,/,24X,
     1 "NORMALIZING?",I1,/,1X,F4.2,"+(",F5.2,")COS......FOR WINDOW")
C
          IF(INO.EQ.1)GO TO 4
C         MORE THAN ONE SPECTRUM,SET UP AXES
C         NO.OF SECTIONS NEEDED FOR X AXIS=INO+OLAP
          X(1)=0.
C         TIME SCALE APPROX
          X(2)=(FLOAT(INO)+1.)*256./SFRQ
          Y(1)=.4990*SFRQ
          Y(2)=0.
          IJKL=1000*ISIZE+210
          CALL GRAPH(X,Y,2,0.,IJKL)
          FPPS=0.9*(XT-XB)/(FLOAT(INO)+OLAP)
          IF(ISIZE.GE.4)FPPS=FPPS/2.
C         NOW SET UP PROPER FREQ AXIS
4         DO 1 I=1,256
1         Y(I)=(SFRQ/511.)*(FLOAT(I)-1.)
C         GET BLKS
          INOW=1
          DO 5 IMAIN=1,INO,2
          IBL=IBLGO+IMAIN-1
          CALL DPREAD(IBL,5,IARRAY(513))
          CALL DIFF(IARRAY,IADGO,256,LDIFF)
          DO 6 I=1,512
          K=2*I
          J=K-1
C         WINDOW
          FRQ=2.*PI*(FLOAT(I)-1.)/511.
          IARRAY(J)=FLOAT(IARRAY(J))*(C1+C2*COS(FRQ))+.5
6         IARRAY(K)=FLOAT(IARRAY(K))*(C1+C2*COS(FRQ))+.5
C         TRANSFORM
          CALL FFT(-9,INORM,IARRAY)
C         UNSHUFFLE:IBLK2+1=ARRAY ADDR.FOR 2ND BLOCK
```

```
      CALL UNSHF(IARRAY,IBLK2)
C     PREPARE TO WORK ON 1ST UNSHUFFLED BLOCK
      J=0
C     FIND POWER BASED ON 512 PT WINDOW(RECT),FS=10KHZ
      DO 7 I=1,2
      DO 8 L=1,256
      J=J+2
      K=J-1
      ARR1=FLOAT(IARRAY(J))
      ARR2=FLOAT(IARRAY(K))
8     X(L)=(ARR1*ARR1+ARR2*ARR2)/5.12E-2
C
C     SMOOTH
C     SET UP WORKING ARRAY
      W(1)=X(4)
      W(2)=X(3)
      W(3)=X(2)
      W(4)=X(1)
      W(5)=X(1)
      W(6)=X(2)
      W(7)=X(3)
      W(8)=X(4)
      W(9)=X(5)
      DO 10 II=1,256
      DO 11 JJ=1,4
      IJ=10-JJ
11    X(II)=X(II)+FLOAT(JJ)*(W(JJ)+W(IJ))/5.
      IF(X(II).LE.1.)X(II)=1.
      X(II)=10.*ALOG10(X(II))
      IF(INOW.NE.1)GO TO 12
C     FIND MAX AND MIN 1ST PASS
      IF(X(II).GT.AMAX)AMAX=X(II)
      IF(X(II).LT.AMIN)AMIN=X(II)
      IF(II.NE.256)GO TO 16
      WRITE (3,100) AMAX,AMIN
100   FORMAT(1X,"MAX VALUE 1ST PASS = ",F5.1,"DB",/,1X,
     1 "MIN VALUE 1ST PASS = ",F5.1,"DB WRT 51.2MS RECT WINDOW ")
      GO TO 10
12    X(II)=X(II)-AMIN
      IF(II.EQ.256)GO TO 21
16    DO 17 KK=1,8
      IJ=KK+1
17    W(KK)=W(IJ)
      IF(II.GT.251)GO TO 18
      IK=II+5
      W(9)=X(IK)
      GO TO 10
18    IK=10-2*(II-251)
      W(9)=W(IK)
10    CONTINUE
      IF(INOW.NE.1.OR.INO.EQ.1)GO TO 19
      DO 20 II=1,256
20    X(II)=X(II)-AMIN
19    IF(INO.NE.1)GO TO 9
C     SINGLE GRAPH,AXES REVERSED
      IJKL=1000*ISIZE+220
      CALL GRAPH(Y,X,256,0.,IJKL)
      GO TO 5
C     CALC X AXIS MAGNITUDE SCALE
9     IF(INOW.EQ.1)XGRAD=(OLAP+1.)*FPPS/(AMAX-AMIN+EXTRA)
C     STEP TO BE TAKEN ALONG X AXIS
21    XCONST=(FLOAT(INOW)-1.)*FPPS+XB+.5
      IJKL=1000*ISIZE+520
      CALL GRAPH(X,Y,256,0.,IJKL)
C     GO BACK TO A SATISFIED "DO"
      IF(INOW.EQ.INO)GO TO 5
```

```
C        PREPARE FOR 2ND BLOCK OF UNSHUFFLED VALUES
         J=IBLK2
7        INOW=INOW+1
5        CONTINUE
         WRITE (3) " GO AGAIN? TYPE 1 FOR YES "
         READ (2) IFLAG
         IF(IFLAG.EQ.1)GO TO 1000
         STOP
         END
```

APPENDIX 7

CDC Cyber 173 Computer Program with BaHli's method

by A. Y. C. Quan

Abstract of program for determining H(s) from h(t).

'TFIDENT' is the main program. Its different subroutines are as following:

(1)'RATCOF'

Calculates the coefficients b_i of the polynomial $H(s)=b_1+b_2s+b_3s^2+\cdots\cdots$. This subroutine is called by the main program 'TFIDENT'.

(2)'SOLUTON'

Solves the unknowns $p_1, p_2, \cdots, p_n, q_1, q_2, \cdots, q_m$ of the function $H(s)=(p_1+p_2s+p_3s^2+\cdots\cdots+p_ns^{n-1})/(1+q_1s+q_2s^2+\cdots\cdots+q_ms^m)$

(3)'RATFN' and 'MULLER'

Determines the zeros and poles of H(s).

(4)'TIMECON'

Computes the system time constants and arranges them in ascending order.

(5)'TIMEFN'

Computes the function of speech time waveform,h(t) from its transfer function,H(s).

(6)'RUNGS' and 'DER'

These are used by 'TIMEFN' to solve the n first order differential equations

$$\begin{bmatrix} h^{(1)}(t) \\ h^{(2)}(t) \\ \vdots \\ h^{(n)}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -d_n & -d_{n-1} & -d_{n-2} & \cdots & -d_1 \end{bmatrix} \begin{bmatrix} h(t) \\ h^{(1)}(t) \\ \vdots \\ h^{(n-1)}(t) \end{bmatrix}$$

(7)'ERRFN'

Computes the error, $h(t)-h^*(t)$ and calculates the integral of squared error. 'ERRFN' is called by 'TIMEFN'.

(8)'INTEG'

An integration subroutine (called by 'ERRFN') which uses the trapezoidal rule.

(9)'CALSEQ'

Calls the above subroutines in the order (2), (3), (4), and (5).

```
      PROGRAM TFIDENT(INPUT,OUTPUT,RESULT,TAPE1=INPUT,TAPE2=OUTPUT,TAPE3
     *=RESULT)
      DIMENSION F(4000),G(200),T(200),NDELAY(200),Y(26),B(26),A(26,26)
      REAL LAMDA,LENTH1,LENTH2
      COMMON F
      COMMON/BLOCK2/TI(800),NOFUN,ITYPE,IPLOT,XMAX
      READ 11,IEVEN,ITYPE,IPLOT,IPRBS
11    FORMAT(4I2)
      PRINT 158
158   FORMAT(1H1,5(/),10X,* CONTROL CARDS USED-----*)
      PRINT 160,IEVEN,ITYPE,IPLOT,IPRBS
160   FORMAT(//,15X,*IEVEN= *,I2,/,15X,*ITYPE= *,I2,/,15X,*IPLOT= *,
     1  I2,/,15X,*IPRBS,= *,I2)
      PRINT 159
159   FORMAT(5(/),10X,* REQUESTS-------*,///)
      IF(IEVEN)162,161,162
161   PRINT 163
163   FORMAT(10X,* (1)        EQUALLY SPACED DATA SPECIFIED*,//)
      GO TO 164
162   PRINT 1621
1621  FORMAT(10X,* (1)        UNEQUALLY SPACED DATA SPECIFIED*,//)
164   IF(ITYPE)165,166,167
165   PRINT 168
168   FORMAT(10X,*  (2)       NO PLOT OF FUNCTIONS REQUIRED*,///,
     1     10X,    * (3)      NEITHER QIKPLT NOR AUTPLT REQUESTED*//)
      IPLOT=0
      GO TO 178
166   PRINT 169
169   FORMAT(10X,* (2)        2 FUNCTIONS ON THE SAME PLOT REQUESTED*,//)
      GO TO 171
167   PRINT 170
170   FORMAT(10X,* (2)        3 OR MORE FUNCTIONS ON THE SAME PLOT REQUEST
     1ED*,//)
171   IF(IPLOT)172,173,174
172   PRINT 175
175   FORMAT(10X,* (3)        BOTH QIKPLT AND AUTPLT REQUESTED*,//)
      GO TO 178
173   PRINT 176
176   FORMAT(10X,* (3)        ONLY QIKPLT REQUESTED*,//)
      GO TO 178
174   PRINT 177
177   FORMAT(10X,* (3)        ONLY AUTPLT REQUESTED*,//)
178   IF(IPRBS)179,180,179
179   PRINT 182
182   FORMAT(10X,* (4)        DATA OBTAINED IS NOT FROM THE EAI 580 ANALOG
     1UE COMPUTER*)
      GO TO 183
180   PRINT 181
181   FORMAT(10X,* (4)        DATA OBTAINED FROM EAI 580 ANALOGUE COMPUTER
     1*)
183   IF(IEVEN)12,13,12
C FOR EQUALLY SPACED DATA F(I)
13    READ 1 ,M,LAMDA
1     FORMAT(I5,F10.5)
      READ 2,(F(I),I=1,M)
2     FORMAT(16F5.3)
      PRINT 151
```

```
151     FORMAT(1H1,5(/),5X,* INPUT DATA----------------*,
       1//,10X,* DELAY  TIME(SEC)*,12X,* IMPULSE RESPONSE*,/)
        DO 15 I=1,M
        I1=I-1
        TIME=LAMDA*I1
        PRINT 153,I1,TIME,F(I)
153     FORMAT(10X,I5,F10.4,F20.6)
15      CONTINUE
        PRINT 17,M,LAMDA
17      FORMAT(3(/),10X,*NO. OF EQUALLY SPACED DATA POINTS(M)=*,I5,
       1 /,10X,*TIME INCREMENT(LAMDA)=*,F8.5)
        GO TO 14
C FOR UNEQUALLY SPACED DATA G(I)
12      READ 10,MAXDLAY,MM,M,LAMDA
10      FORMAT(3I5,F10.5)
        IF((MAXDLAY+1).LT.M)20,30
20      PRINT 40
40      FORMAT(10(/),10X,*(MAXIMUM DELAY+1) IS LESS THAN THE REQUIRED NO.
       1OF EQUALLY SPACED IMPULSE RESPONSE POINTS----JOB ABORTED*)
        GO TO 51
30      READ 60,(NDELAY(J),J=1,MM)
60      FORMAT(20I4)
        READ 2,(G(J),J=1,MM)
        DO 80 J=1,MM
80      T(J)=LAMDA*NDELAY(J)
C EQUALLY SPACED DATA F(I) OBTAINED FROM UNEQUALLY SPACED DATA G(J)
C BY LINEAR INTERPOLATION
        J=1
        DO 90 I=1,M
        TIME=LAMDA*(I-1)
100     IF(ABS(TIME-T(J)).LE.1.0E-10)110,120
110     F(I)=G(J)
        J=J+1
        GO TO 90
120     IF((TIME-T(J)).GT.1.0E-10)130,140
130     J=J+1
        GO TO 100
140     F(I)=G(J)-((G(J)-G(J-1))/(T(J)-T(J-1)))*(T(J)-TIME)
90      CONTINUE
        PRINT 151
        DO 152 J=1,MM
        PRINT 153,NDELAY(J),T(J),G(J)
152     CONTINUE
        PRINT 18,MAXDLAY,MM,M,LAMDA
18      FORMAT(3(/),10X,*MAXIMUM DELAY(MAXDLAY)=*,I5,
       1 /,10X,*NO. OF UNEQUALLY SPACED DATA POINTS(MM)=*,I5,
       2 /,10X,*NO. OF EQUALLY SPACED POINTS TO BE OBTAINED BY LINEAR INTE
       3RPOLATION(M)=*,I5,
       4 /,10X,*TIME INCREMENT(LAMDA)=*,F8.5)
C SPECIFY THE FOLLOWING VALUES TO CALCULATE THE NEW AMPLITUDE
C SCALING FACTOR BETA2.................
14      IF(IPRBS.NE.0)GO TO 31
        LENTH1=1023.
        LENTH2=1023.0
        FC1=10.0**3
        A1=0.5000
        PM1=2.0
```

```
      GAIN1=10.0
      BETA1=97.5536
      ALPHA=10.0
      FC2=100.0
      A2=0.003
      PM2=2.0
      GAIN2=100.0
      TEMP=(FC2**2.0)*A1*PM1*GAIN1*LENTH1*BETA1
       BETA2=TEMP/((FC1**2.0)*A2*PM2*GAIN2*LENTH2*ALPHA)
      PRINT 7,BETA2
7     FORMAT(1H1,5(/),10X,* VALUE OF AMPLITUDE SCALING FACTOR= *,
     1 F15.6)
      LAMDA=LAMDA*ALPHA
      DO 81 I=1,M
81    F(I)=F(I)*BETA2
C FROM HEREON F(T) IS ACTUALLY H(T)
31    PRINT 34
34    FORMAT(1H1,
     1 * COEFFICIENTS, B(I) OF THE POLYNOMIAL ---------*,//,
     2 *                        2        3 *,/,
     3 * H(S) = B    + B S  + B S   + B S    +  ..........*,/,
     4 *         1      2      3         4 *,3(/))
C ---------CHANGE THE FOLLOWING CARDS IF NECESSARY----------
C NOPLOT=NO. OF DIFFERENT PLOTS EXPECTED
      DO 900 I=1,M
      F(I)=1000.0*F(I)
900   CONTINUE
      NOPLOT=2
      NZERO=10
      NPOLE=12
C  ----------------------END , CHANGE-------------------
      CALL RATCOF(NZERO,NPOLE,N,LAMDA,M,F,B)
C  XMAX=MAXIMUM LENGTH OF GRAPH PAPER EXPECTED IN INCHES
      XMAX=(14.0*NOPLOT)+4.0
      ISTART=0
      NOFUN=1
      DO 35 I=1,N
35    PRINT 36,I,B(I)
36    FORMAT(10X,3H B(, I2, 2H)= ,E15.6,//)
C  ---------FOLLOWING CARDS CHANGED IF NECESSARY----------
      NZERO=10
      NPOLE=12
      CALL CALSEQ(NZERO,NPOLE,B,N,A,Y,M,F,LAMDA,ISTART)
      NZERO=4
      NPOLE=6
      CALL CALSEQ(NZERO,NPOLE,B,N,A,Y,M,F,LAMDA,ISTART)
C  -------------------END,  CHANGE----------
      IF(ITYPE) 51,52,52
52    IF(IPLOT)53,54,53
53    CALL AUTPLT(TI,F,M, NOFUN,11H*TIME(SEC)*,18H*IMPULSE RESPONSE*)
      IF(IPLOT)54,54,51
54    CALL OIKPLT(TI,F,-M,-NOFUN,11H*TIME(SEC)*,18H*IMPULSE RESPONSE*)
51    STOP
      END
```

```
      SUBROUTINE RATCOF(NNM,NND,NMAX,SPACE,M,F,B)
C CALCULATES THE VALUES OF B(K)..ALL INTEGRATIONS BY TRAPEZOIDAL RULE.
C NMAX=MAX. NO. OF UNKNOWNS EXPECTED IN THE TRANSFER FUNCTION H(S)
C                                     2         3         4         5
C  H(S) = B(1) + B(2)S + B(3)S + B(4)S + B(5)S + B(6)S + ..........
      DIMENSION F(1),B(1)
      NMAX=NNM+NND+1
C  CALCULATE B(1)
      PARSM=F(2)
      M1=M-1
      DO 22 I=3,M1
      PARSM=PARSM+F(I)
22    CONTINUE
      B(1)=SPACE*(0.5*(F(1)+F(M))+PARSM)
C NEXT CALCULATE B(K), FOR K=2,NMAX
      DO 30 K=2,NMAX
C THE FOLLOWING 6 CARDS CALCULATES FACTORIAL (K-1) AND STORES THE
C  RESULT IN DEN.
      J=K-1
      JT=J
70    J=J-1
      IF(J.EQ.0)50,60
60    JT=JT*J
      GO TO 70
50    DEN=JT
      K1=K-1
      PARSM=F(2)*(SPACE**K1)
      DO 80 I=3,M1
      I1=I-1
      RI=I1
      PARSM=PARSM+F(I)*((RI*SPACE)**K1)
80    CONTINUE
      RM=M1
      FACT=SPACE*((-1.0)**K1)/DEN
      AREA=0.5*((RM*SPACE)**K1)*F(M)+PARSM
      B(K)=FACT*AREA
30    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CALSEQ(NNM,NND,B,N,A,Y,M,F,XLAMDA,ISTART)
      DIMENSION YN(11),YD(16),ZN(10,2),PD(15,2),A(26,26),TCON(15)
      DIMENSION B(1),Y(1),F(1)
      EQUIVALENCE (PD(15,1),TCON(15))
      COMMON/BLOCK1/YD
      NOSOL=1
      IF(NNM.GE.NND)1,2
1     PRINT 3
3     FORMAT(1H1,5(/),10X,*-----NO. OF POLES IS NOT GREATER THAN THE N
     1O. OF ZEROS--CANNOT PROCEED COMPUTATION------*)
      RETURN
2     CALL SOLUTON(NNM,NND,B,N,A,Y,YN,YD,NOSOL)
      IF(NOSOL.EQ.0)RETURN
      JTEST=1
      CALL RATFN(NNM,NND,YN,YD,ZN,PD,JTEST)
      IF(JTEST.EQ.0) RETURN
C   TEST IF THE REAL PARTS OF THE POLES ARE TOO CLOSE TO THE
C   IMAGINARY AXIS
      K=1
      MAX=NND
      ITEST=1
      DO 5 I=1,MAX
      PD(I,1)=ABS(PD(I,1))
      IF(PD(I,1).LE.1.0E-10)10,15
10    MAX=K
      ITEST=0
      PRINT 20
20    FORMAT(///,5X,* POLE TOO CLOSE TO THE IMAGINARY AXIS,RETURN TO M
     1AIN PROGRAM *)
      GO TO 5
15    K=K+1
5     CONTINUE
      IF(ITEST.EQ.0) RETURN
      CALL TIMECON(NND,TCON)
      NND1=NND+1
      NNM1=NNM+1
      YDTEMP=YD(1)
      DO 30  I=1,NNM1
30    YN(I)=YN(I)/YDTEMP
      DO 60 I=1,NND1
60    YD(I)=YD(I)/YDTEMP
      CALL TIMEFN(NNM,NND,YN,YD,TCON,F,M,XLAMDA,ISTART)
      RETURN
      END
```

```
      SUBROUTINE SOLUTON(NNM,NND,B,N,A,Y,YN,YD,NOSOL)
      DIMENSION Y(1),B(1),YN(1),YD(1),A(26,26)
C  READS IN A(I,J), Y(I) IN THE SET OF N LINEAR EQUATIONS IN
C  N UNKNOWNS WRITTEN IN MATRIX FORM AS--
C             (A)*(X) = (Y)
C  NNM=DEGREE OF THE NUMERATOR POLYNOMIAL
C  NND=DEGREE OF THE DENOMINATOR POLYMIAL
C  N = TOTAL NO. OF UNKNOWNS
      NN1=NNM+1
      NN2=NNM+2
      N=NNM+NND+1
C  SET ALL COEFFICIENTS A(I,J) TO ZERO
      DO 100 I=1,N
      DO 100 J=1,N
      A(I,J)=0.0
100   CONTINUE
C  ENTER THE COEFFICIENTS OF THE NUMERATOR UNKNOWNS
      DO 20 I=1,NN1
20    A(I,I)=-1.00
      I2=2
C  ENTER THE COEFFICIENTS OF THE DENOMINATOR UNKNOWNS
      DO 30 J=NN2,N
      I1=1
      DO 40 I=I2,N
      A(I,J)=B(I1)
      I1=I1+1
40    CONTINUE
      I2=I2+1
30    CONTINUE
C  ENTER THE VALUES FOR Y(I)
      DO 50 I=1,N
50    Y(I)=-B(I)
C  SOLVES THE SET OF N LINEAR EQUATIONS IN N UNKNOWNS
      DO 1 I=1,N
      DO 2 J=I,N
      IF(A(I,J).NE.0.) GO TO 3
2     CONTINUE
      PRINT 4,N
4     FORMAT(15H1NO SOLUTION TO,I3,17H,GIVEN EQUATIONS.)
      NOSOL=0
      RETURN
3      IF(I.NE.J)5,6
C  REVERSES EQUATIONS I AND J
5     DO 7 K=1,N
      B=A(I,K)
      A(I,K)=A(J,K)
      A(J,K)=B
7      CONTINUE
      B=Y(I)
      Y(I)=Y(J)
      Y(J)=B
C DIVIDES EQUATION I BY A(I,I) AND SUBTRACTS EQUATION I (TIMES A(I,J))
C  FROM EACH SUBSEQUENT EQUATION
6      DIV=A(I,I)
      A(I,I)=1.
      Y(I)=Y(I)/DIV
      IF(I.GE.N) GO TO 1
```

```
      MINJ=I+1
      DO 8 J=MINJ,N
      A(I,J)=A(I,J)/DIV
8     CONTINUE
      DO 9 J=MINJ,N
      FACTOR=A(J,I)
      A(J,I)=0.
      DO 10 K=MINJ,N
      A(J,K)=A(J,K)-FACTOR*A(I,K)
10    CONTINUE
      Y(J)=Y(J)-FACTOR*Y(I)
9     CONTINUE
1     CONTINUE
C SOLVES SET OF EQUATIONS WITH COEFFICIENTS THAT ARE ONE ON THE DIAGONAL
C  AND ZERO BELOW THE DIAGONAL
      MAXI=N-1
      DO 11 I=1,MAXI
      J=N-I
      MINK=J+1
      DO 12 K=MINK,N
      Y(J)=Y(J)-A(J,K)*Y(K)
12    CONTINUE
11    CONTINUE
      PRINT 13,N,(I,Y(I),I=1,N)
13    FORMAT(1H1,5X,11HSOLUTION TO,I3,30H SIMULTANEOUS LINEAR EQUATIONS/
     A/5(2H X,I1,4H  = ,E15.8,3X)/4(2H X,I1,4H  = ,E15.8,3X),2H X,I2,3H
     B= ,E15.8/(5(2H X,I2,3H = ,E15.8,3X)))
C  REARRANGES THE COEFFICIENTS OF THE NUMERATOR AND DENOMINATOR
C  POLYMINALS OF THE RATIONAL TRANSFER FUNCTION BEGINNING WITH
C  THE HIGHEST DEGREE TERM. SUITABLE FOR OBTAINING THE ZEROS AND
C  POLES OF H(S)
      N1=N+1
      N2=N+2
      DO 60 I=1,NN1
60    YN(NN2-I)=Y(I)
      DO 70 I=NN2,N
70    YD(N1-I)=Y(I)
      YD(NND+1)=1.0
      RETURN
      END
```

```
      SUBROUTINE RATEN(NN,ND,AN,AD,RN,RD,JTEST)
C     SUBROUTINE RATEN TOGETHER WITH SUBROUTINE MULLER DETERMINES THE
C     ROOTS OF THE POLYNOMIALS
C     NN=DEGREE OF NUMERATOR POLYNOMIAL
C     ND=DEGREE OF DENOMINATOR POLYNOMIAL
C     RN AND RD ARE THE ROOTS OF THE NUMERATOR AND DENOMINATOR POLYMIALS
C     RESPECTIVELY
      DIMENSION AN(1),AD(1),RN(10,2),RD(15,2)
      NN1=NN+1
      ND1=ND+1
      PRINT 3,NN,ND
3      FORMAT(5(/),5X,* TRANSFER FUNCTION WITH *,I2,
     1 *  ZEROS AND *,I2,*  POLES .-------------*,4(/),    5X,
     2 * NUMERATOR POLYNOMIAL, BEGINNING WITH THE HIGHEST DEGREE TERM *)
      PRINT 4,(AN(I),I=1,NN1)
4     FORMAT(10X,E16.6)
      PRINT 5
5     FORMAT( 5(/),5X,
     1 * DENOMINATOR POLYNOMIAL BEGINNING WITH THE HIGHEST DEGREE TERM*)
      PRINT 4,(AD(I),I=1,ND1)
      IF(ABS(AD(1)).LT.1.0E-50) 20,30
20    PRINT 40
40    FORMAT (5(/),5X,*  COEFFICIENT OF HIGHEST DEGREE TERM IN DENOMINAT
     1OR POLYNOMIAL IS TOO SMALL ----  SUBROUTINE MULLER NOT CALLED. *,
     2 /,5X,* ----REARRANGEMENT OF DENOMINATOR POLYNOMIAL COEFFICIENTS
     3 NECESSARY IF SUBROUTINE MULLER IS TO BE CALLED*)
      JTEST=0
      RETURN
30    PRINT 50
50    FORMAT (5(/),5X,
     1 *  ZEROS AND POLES OF THE POLYNOMIALS GIVEN BY THEIR REAL AND IMA
     2GINARY PARTS ARE AS FOLLOWS..- *,'  //   )
      PRINT 6
6     FORMAT(*0ZEROS*)
      CALL MULLER(NN,AN,RN)
      PRINT 7
7     FORMAT(*0POLES*)
      CALL MULLER(ND,AD,RD)
      GAIN=AN(1)/AD(1)
      PRINT 10,GAIN
10    FORMAT(4(/),18X,12H GAIN TERM = ,E14.6)
      RETURN
      END
```

```
      SUBROUTINE MULLER(N1,COF,Z)
C     ROOTS OF A POLYNOMIAL BY THE MULLER METHOD
C     N1 = DEGREE OF POLYNOMIAL
      DIMENSION COE(1),Z(N1,2)
C     COE=COEFFICIENTS OF POLYNOMIAL BEGINNING WITH THE HIGHEST DEG. TERM
C     Z=COMPLEX ROOTS(OUTPUT)
      IF(N1)1,1,3
1     PRINT 2
2     FORMAT(//,20X,4HNONE  )
      RETURN
3     CONTINUE
      N2=N1+1
      N4=0
      I=N1+1
4     IF(COE(I))6,5,6
5     N4=N4+1
      Z(N4,1)=0.0
      Z(N4,2)=0.0
      I=I-1
      IF(N4-N1)4,25,4
6     CONTINUE
7     AXR=0.8
      AXI=0.0
      L=1
      N3=1
      ALP1R=AXR
      ALP1I=AXI
      M=1
      GO TO 33
8     BET1R=TEMR
      BET1I=TEMI
      AXR=0.85
      ALP2R=AXR
      ALP2I=AXI
      M=2
      GO TO 33
9     BET2R=TEMR
       BET2I=TEMI
      AXR=0.9
      ALP3R=AXR
      ALP3I=AXI
      M=3
      GO TO 33
10    BET3R=TEMR
      BET3I=TEMI
11    TE1=ALP1R-ALP3R
      TE2=ALP1I-ALP3I
      TE5=ALP3R-ALP2R
      TE6=ALP3I-ALP2I
      TEM=TE5*TE5+TE6*TE6
      TE3=(TE1*TE5+TE2*TE6)/TEM
      TE4=(TE2*TE5-TE1*TE6)/TEM
      TE7=TE3+1.0
      TE9=TE3*TE3-TE4*TE4
      TE10=2.0*TE3*TE4
      DE15=TE7*BET3R-TE4*BET3I
      DE16=TE7*BET3I+TE4*BET3R
```

```
      TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15
       TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16
      TE7=TE9-1.0
      TE1=TE9*BET2R-TE10*BET2I
      TE2=TE9*BET2I+TE10*BET2R
      TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I
      TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R
      TE15=DE15*TE3-DE16*TE4
      TE16=DE15*TE4+DE16*TE3
      TE1=TE13*TE13-TE14*TE14-4.0* (TE11*TE15-TE12*TE16)
      TE2=2.0*TE13*TE14-4.0*(TE12*TE15+TE11*TE16)
      TEM=SQRT(TE1*TE1+TE2*TE2)
      IF(TE1)12,12,13
 12   TE4=SQRT(0.5*(TEM-TE1))
      TE3=0.5*TE2/TE4
      GO TO 16
 13   TE3=SQRT(0.5*(TEM+TE1))
      IF(TE2)14,15,15
 14   TE3=-TE3
 15   TE4=0.5*TE2/TE3
 16   TE7=TE13+TE3
      TE8=TE14+TE4
      TE9=TE13-TE3
      TE10=TE14-TE4
      TE1=2.0*TE15
      TE2=2.0*TE16
      IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)17,17,18
 17   TE7=TE9
      TE8=TE10
 18   TEM=TE7*TE7+TE8*TE8
      TE3=(TE1*TE7+TE2*TE8)/TEM
      TE4=(TE2*TE7-TE1*TE8)/TEM
      AXR=ALP3R+TE3*TE5-TE4*TE6
      AXI=ALP3I+TE3*TE6+TE4*TE5
      ALP4R=AXR
      ALP4I=AXI
      M=4
      GO TO 33
 19   N6=1
 20   IF(ABS(HELL)+ABS(BELL)-1.E-20)23,23,21
 21   TE7=ABS(ALP3R-AXR)+ABS(ALP3I-AXI)
      IF(TE7/(ABS(AXR)+ABS(AXI))-1.E-7)23,23,22
 22   N3=N3+1
      ALP1R=ALP2R
      ALP1I=ALP2I
      ALP2R=ALP3R
      ALP2I=ALP3I
      ALP3R=ALP4R
      ALP3I=ALP4I
      BET1R=BET2R
      BET1I=BET2I
      BET2R=BET3R
      BET2I=BET3I
      BET3R=TEMR
      BET3I=TEMI
      IF(N3-100)11,23,23
 23   N4=N4+1
```

```
      Z(N4,1) =ALP4R
      Z(N4,2) =ALP4I
      N3=0
24    IF(N4-N1)28,25,25
26    FORMAT(//,10X,10H REAL PART ,10X,14HIMAGINARY PART )
25    PRINT 26
      PRINT 27 ,  ( Z(I,1),Z(I,2) ,I=1,N1)
27    FORMAT (8X, F18.6,3X,F18.6)
      RETURN
28    IF(ABS(Z(N4,2) )-1.E-5)7,7,29
29    GO TO (30,7 ),L
30    AXR=ALP1R
      AXI=-ALP1I
      ALP1I=-ALP1I
      M=5
      GO TO 33
31    BET1R=TEMR
      BET1I=TEMI
      AXR=ALP2R
      AXI=-ALP2I
      ALP2I=-ALP2I
      M=6
      GO TO 33
32    BET2R=TEMR
      BET2I=TEMI
      AXR=ALP3R
      AXI=-ALP3I
      ALP3I=-ALP3I
      L=2
      M=3
33    TEMR=COE(1)
      TEMI=0.0
      DO 34 I=1,N1
      TE1=TEMR*AXR-TEMI*AXI
      TEMI=TEMI*AXR+TEMR*AXI
34    TEMR=     TE1+COE(I+1)
      HELL=TEMR
      BELL=TEMI
35    IF(N4)36,38,36
36    DO 37 I=1,N4
      TEM1=AXR-Z(I,1)
      TEM2=AXI-Z(I,2)
      TE1=TEM1*TEM1+TEM2*TEM2
      TE2=(TEMR*TEM1+TEMI*TEM2)/TE1
      TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1
37    TEMR=TE2
38    GO TO(8,9,10,19,31,32),M
      END
```

```
      SUBROUTINE TIMECON(N,X)
C   ARRANGES THE MAGNITUDE OF THE IMAGINARY PARTS OF THE POLES
C   IN DESCENDING ORDER
C   THE TIME CONSTANTS (TRUNCATED TO TWO SIGNIFICANT DIGITS)
C   CORRESPONDING TO THESE POLES ARE THEN FOUND
      DIMENSION X(N)
C   REARRANGES N NUMBERS IN DESCENDING ORDER IN LOCATIONS
C   X(1),X(2),................,X(N)
C   DETERMINE THE LARGEST NO.
      N1=N-1
      DO 11 I=1,N1
      TE1=X(I)
      I1=I+1
      DO 12 J=I1,N
      IF(TE1.LT.X(J)) TE1=X(J)
12    CONTINUE
C   TE1 CONTAINS THE LARGEST NUMBER. NEXT, DETERMINE ITS LOCATION.
      MAX=N
      K=I
      DO 13 J=1,MAX
      IF(TE1.EQ.X(J)) 14,15
14    MAX=K
      GO TO 13
15    K=K+1
13    CONTINUE
C   LOCATION CONTAINING THE LARGEST NUMBER IS X(K)
C   NEXT, INTERCHANGE X(I) AND X(J)
      TE1=X(I)
      TE2=X(K)
      X(I)=TE2
      X(K)=TE1
11    CONTINUE
      DO 5 J=1,N
5     X(J)=1.0/X(J)
C   TRUNCATES POSITIVE NUMBERS TO K SIGNIFICANT DIGITS
      K=2
      PRINT 1,K
1     FORMAT(5(/),5X,* -----------TIME CONSTANTS TRUNCATED TO *,I2,* SIGNIFI
     1ANT DIGITS (SECS.)-------*,/)
        XLOW=10.**(K-1)
      XHIGH=10.**K
      DO 10 J=1,N
      I=0
      IF(X(J).LE.1.0E-10)20,30
20    PRINT 40,J
40    FORMAT (5X,*  VALUE OF X(*,I2,* ) IS ZERO *,//)
      GO TO 80
30    IF(X(J).GE.XLOW)50,60
60    I=I+1
      X(J)=10.0*X(J)
      GO TO 30
50    IF(X(J).GT.XHIGH)70,80
70    I=I-1
      X(J)=0.1*X(J)
      GO TO 30
C   TRUNCATE X(J)
80    X(J)=AINT(X(J))
```

```
          X(J)=X(J)*(10.0**(-I))
   10     CONTINUE
          PRINT 90,(X(J),J=1,N)
   90     FORMAT (10X,F20.10)
          RETURN
          END
```

```
      SUBROUTINE TIMEEN(NN,N,BC,AC,TC,F,M,XLAMDA,ISTART)
      DIMENSION F(1),TC(1),BC(1),AC(1),X(15),XPRIME(15),A(15,15)
      DIMENSION AINV(15,15),H(800),TBOUND(15),DEL(15),B(15)
      EQUIVALENCE (A,AINV)
      EXTERNAL DER
      DO 10 I=1,N
      DO 20 J=1,N
   20 A(I,J)=0.0
   10 A(I,I)=1.0
      KK=2
   50 K=1
      DO 30 I=KK,N
      A(1,K)=AC(KK)
      K=K+1
   30 CONTINUE
      KK=KK+1
      IF (KK .GT. N) 40,50
   40 NM=N-NN
      NM1=N-NN-1
      K=1
      KK=1
      IF ((NN+1) .LT. N) 32,34
   32 DO 36 I=1,NM1
      B(K)=0.0
      K=K+1
   36 CONTINUE
   34 DO 38 I=NM,N
      B(K)=BC(KK)
      K=K+1
      KK=KK+1
   38 CONTINUE
      PRINT 43
   43 FORMAT(1H1,5(/),5X,* COEFFICIENTS OF NUMERATOR POLYNOMIAL OF TRANS
     1FER FUNCTION BEGINNING WITH THE HIGHEST DEGREE TERM ....*,/)
      PRINT 42, (B(I),I=1,N)
   42 FORMAT (8X,F20.6)
      PRINT 44
   44 FORMAT(5(/),5X,* COEFFICIENTS OF DENOMINATOR POLYNOMIAL OF TRANSF
     1ER FUNCTION BEGINNING WITH THE HIGHEST DEGREE TERM: .....*,/)
      NONE=N+1
      PRINT 42,(AC(I),I=1,NONE)
      PRINT 51
   51 FORMAT(5(/),5X,* MATRIX  [A]  : ........*,/)
      DO 52 I=1,N
      PRINT 53,(A(I,J),J=1,N)
   53 FORMAT(5X,6E20.6)
   52 CONTINUE
      CALL MATRIX(10,N,N,1,A,15,ADET)
      PRINT 54
   54 FORMAT(5(/),5X,* INVERSE OF MATRIX [A]  : .....*,/)
      DO 55 I=1,N
      PRINT 53, (AINV(I,J),J=1,N)
   55 CONTINUE
      DO 71 I=1,N
      X(I)=0.0
      DO 71 K=1,N
   71 X(I)=X(I)+AINV(I,K)*B(K)
```

```
      PRINT 41
41    FORMAT(5(/),5X,* INITIAL STATE VECTOR  [X] ......*,/)
      PRINT 42,(X(I),I=1,N)
      PRINT 57
57    FORMAT(5(/),15X,* TIME CONSTANT    TIME  INCREMENT   TIME BOUND
     1ARY*,/,18X,*(SECS.)                 (SECS.)              (SECS.)*,/)
      INDEX=0
      K=0
      I=1
      T=0.0
      TLIMIT=1.2*XLAMDA*(M-1)
      DO 70 J=1,N
      TBOUND(J)=12.0*TC(J)
      DEL(J)=0.05*TC(J)
      PRINT 75,J,TC(J),DEL(J),TBOUND(J)
75    FORMAT(9X,2H (,I2,1H),4X,F10.5,9X,F10.5,8X,F10.5)
70    CONTINUE
      PRINT 60
60    FORMAT(1H1,5(/),10X, * TIME FUNCTION FROM THE RATIONAL TRANSFER FU
     1NCTION ...............*,5(/))
      PRINT 59
59    FORMAT(11X,* TIME(SEC)     H(T)-CALCULATED   DERIVATIVE OF H(T
     1)*,/,26X,* FROM T.F. MODEL*,/)
80    DELTA=DEL(I)
85    CALL RUNGS(T,DELTA,N,X,XPRIME,DER,INDEX)
      PRINT 90,T,X(1),X(2)
90    FORMAT(F20.5,2E20.6)
      K=K+1
      H(K)=X(1)
      IF(ABS(X(1)) .GE. 1.0E10) 100,110
100   PRINT 120
120   FORMAT(/,5X,* ....H(T) TOO LARGE,  COMPUTATION STOPPED .....*)
      RETURN
110   IF(T .GE. TBOUND(I)) 130,85
130   I=I+1
      IF(T .GT. TLIMIT) GO TO 140
      IF(I .GT. N) 140,80
140   CALL ERRFN(H,F,N,M,XLAMDA,DEL,TBOUND,ISTART)
      RETURN
      END
```

```
      SUBROUTINE DER(T,N,X,XPRIME)
      DIMENSION X(N),XPRIME(N)
      COMMON /BLOCK1/AC(16)
      N1=N-1
      DO 10 I=1,N1
   10 XPRIME(I)=X(I+1)
      XPRIME(N)=-AC(2)*X(N)
      DO 20 I=2,N
   20 XPRIME(N)=XPRIME(N)-AC(I+1)*X(N+1-I)
      RETURN
      END
```



```
      SUBROUTINE RUNGS(X,H,N,Y,YPRIME,NAME,INDEX)
      DIMENSION Y(N),YPRIME(N)
      DIMENSION W(15),Z(15),W1(15),W2(15),W3(15),W4(15)
      IF(INDEX)5,5,1
    1 DO 2 I=1,N
      W1(I)=H*YPRIME(I)
    2 Z(I)=Y(I)+(W1(I)*0.5)
      A=X+(H/2.0)
      CALL NAME (A,N,Z,YPRIME)
      DO 3 I=1,N
      W2(I)=H*YPRIME(I)
    3 Z(I)=Y(I)+(W2(I)*0.5)
      A=X+(H/2.0)
      CALL NAME(A,N,Z,YPRIME)
      DO 4 I=1,N
      W3(I)=H*YPRIME(I)
    4 Z(I)=Y(I)+W3(I)
      A=X+H
      CALL NAME(A,N,Z,YPRIME)
      DO 7 I=1,N
      W4(I)=H*YPRIME(I)
    7 Y(I)=Y(I)+(((2.0*(W2(I)+W3(I)))+W1(I)+W4(I))/6.)
      X=X+H
      CALL NAME(X,N,Y,YPRIME)
      GO TO 6
    5 CALL NAME(X,N,Y,YPRIME)
      INDEX=1
    6 RETURN
      END
```

```
      SUBROUTINE ERREN(H,F,NPOLE,MPOINS,XLAMDA,DELTA,TBOUND,ISTART)
      DIMENSION F(1),H(1),DELTA(1),TBOUND(1),G(800),ERR(800)
      COMMON /BLOCK2/T(800),NOFUN,ITYPE,IPLOT,XMAX
      IF(ISTART .EQ. 0) 1,2
1     MDIV2=MPOINS/2
      DO 3 I=1,MDIV2
      ERR(I)=(0.01*F(I))**2
3     CONTINUE
      MPLUS1=MDIV2+1
      DO 4 I=MPLUS1,MPOINS
      ERR(I)=(0.1*F(I))**2
4     CONTINUE
      CALL INTEG(MPOINS,XLAMDA,ERR,TISE)
      IF(IPLOT .EQ. 0) GO TO 5
      CALL PLOT10(5HNEBN6,5)
      CALL XLIMIT(XMAX)
5     ISTART=1
2     IF(TBOUND(NPOLE) .LT. (XLAMDA*(MPOINS-1))) 10,20
10    PRINT 30
30    FORMAT(1H1,5(/),5X,*  UPPER BOUND OF TIME FOR SYSTEM UNDER TEST
     1EXCEEDS THAT OF THE TRANSFER FUNCTION MODEL,*,/,5X,* .....CANNOT
     2CALCULATE THE INTEGRAL OF SQUARED ERROR,MODIFY PROGRAM ....*)
      RETURN
20    I=1
      J=1
      JJ=1
      JK=1
      TIME=0.0
      G(1)=H(1)
      ERR(1)=H(1)-F(1)
40    TIME=TIME+DELTA(JJ)
      IF(TIME .GE. TBOUND(JJ)) JK=JK+1
50    IF(ABS(TIME-XLAMDA*I) .LE. 1.0E-10) 60,74
60    G(I+1)=H(J+1)
      ERR(I+1)=H(J+1)-F(J+1)
      GO TO 100
74    IF((TIME-XLAMDA*I) .GT. 1.0E-10) 80,90
90    J=J+1
      JJ=JK
      GO TO 40
80    PRINT 82 ,I,J,JJ,H(J),DELTA(JJ),TIME,XLAMDA,F(I+1)
82    FORMAT(3I10,2E15.6,F10.5,2E20.6)
      G(I+1)=H(J)-((H(J+1)-H(J))/DELTA(JJ))*(TIME-I*XLAMDA-DELTA(JJ))
      PRINT 83,I,G(I+1),F(I+1)
83    FORMAT(I20,2E20.6)
100   I=I+1
      IF (I .GT. MPOINS) 110,50
110   DO 130 I=1,MPOINS
130   ERR(I)=ERR(I)*ERR(I)
      CALL INTEG(MPOINS,XLAMDA,ERR,XISE)
      PRINT 58
58    FORMAT(1H1,5(/))
      PRINT 59
59    FORMAT(11X, * TIME(SEC)        H(T)-CALCULATED     H(T)-FROM EXPERI
     1MENTAL*,/,26X,* FROM T.F. MODEL*,11X,*DATA*,13X,* ERROR SQUAR
     2ED*,/)
      DO 996 I=1,MPOINS
```

```
      T(I)=XLAMDA*(I-1)
      PRINT 997,T(I),G(I),F(I),ERR(I)
997   FORMAT(F20.5,2E20.6,6X,F15.8)
996   CONTINUE
      PRINT 140,MDIV2,TISE,XISE
140   FORMAT(5(/),
     1      10X,*INTEGRAL OF SQUARED ERROR CALCULATED ON THE BASIS OF *,
     2/,10X,* 1 PER CENT ERROR FOR THE FIRST*,I4,*EQUALLY SPACED POINTS*
     3 ,/,17X,* AND 10 PER CENT ERROR FOR THE REMAINING POINTS =*,E14.6
     4 ,/,17X,*(FOR COMPARISON PURPOSES) *,
     53(/),10X,*INTEGRAL OF SQUARED ERROR FROM TRANSFER FUNCTION MODEL=
     6*,E14.6)
      IF(ITYPE) 145,150,160
150   IFNEXT=MPOINS+1
      IFFINAL=MPOINS+MPOINS
      DO 170 I=IFNEXT,IFFINAL,1
170   F(I)=G(I-MPOINS)
1111  IF(IPLOT)171,172,171
171   CALL AUTPLT(T,F,MPOINS,2,11H*TIME(SEC)*,18H*IMPULSE RESPONSE*)
      IF(IPLOT) 172,172,145
172   CALL QIKPLT(T,F,-MPOINS,-2,11H*TIME(SEC)*,18H*IMPULSE RESPONSE*)
      GO TO 145
160   NOFUN=NOFUN+1
      IF(NOFUN .LT. 3) IFFINAL=MPOINS
      IFNEXT=IFFINAL+1
      IFFINAL=IFFINAL+MPOINS
      J=1
      DO 180 I=IFNEXT,IFFINAL,1
      F(I)=G(J)
      J=J+1
180   CONTINUE
      GO TO 1111
145   RETURN
      END
```



```
      SUBROUTINE INTEG(M,SPACE,F,XINTEG)
      DIMENSION F(M)
      PARSM=F(2)
      M1=M-1
      DO 10 I=3,M1
10    PARSM=PARSM+F(I)
      XINTEG=SPACE*(0.5*(F(1)+F(M))+PARSM)
      RETURN
      END
```

APPENDIX 8

CDC Cyber 173 Computer Program- MBREC

Referencing (5)* and (24)* we first determine values of the time waveform of speech sound to obtain 128 samples at the rate of 10 KHz and then use the fast Fourier Transform(FFT) to compute spectrum P(w).

Second we compute the autocorrelation function

$$R(i) = \sum_{n=0}^{N-1-i} s_n' \, s_{n+i}' \qquad , \; i \geq 0$$

assuming a rectangular window function

$$s_n' = \begin{cases} s_n w_n , & 0 \leq n \leq N-1 \\ 0 , & \text{otherwise} \end{cases}$$

where s_n is the time waveform of speech sound and w_n is the window function.

Third we solve the normal equations

$$\sum_{k=1}^{p} a_k R(i-k) = -R(i) \;, \qquad 1 \leq i \leq p$$

for p=8 using Durbin's procedure

$$E_0 = R(0)$$
$$k_i = -\left\{R(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)\right\}/E_{i-1}$$
$$a_i^{(i)} = k_i$$
$$a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)} \quad , \qquad 1 \leq j \leq i-1$$
$$E_i = (1-k_i^2)E_{i-1}$$
$$a_j = a_j^{(p)} \quad , \quad 1 \leq j \leq p \qquad \text{for } i = 1, 2, \cdots, p$$

to obtain the coefficients a_k for an all-pole model.

Fourth we compute the all-pole spectrum

$$\hat{P}(w) = G^2/|A(\exp jw)|^2$$

where $G^2 = E_p = R(0) + \sum_{k=1}^{p} a_k R(k)$,

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \quad \text{and} \quad H(z) = G/A(z).$$

$P(w)$ and $\hat{P}(w)$ obtained from first and fourth procedures respectively are demonstrated in the spectral matching.

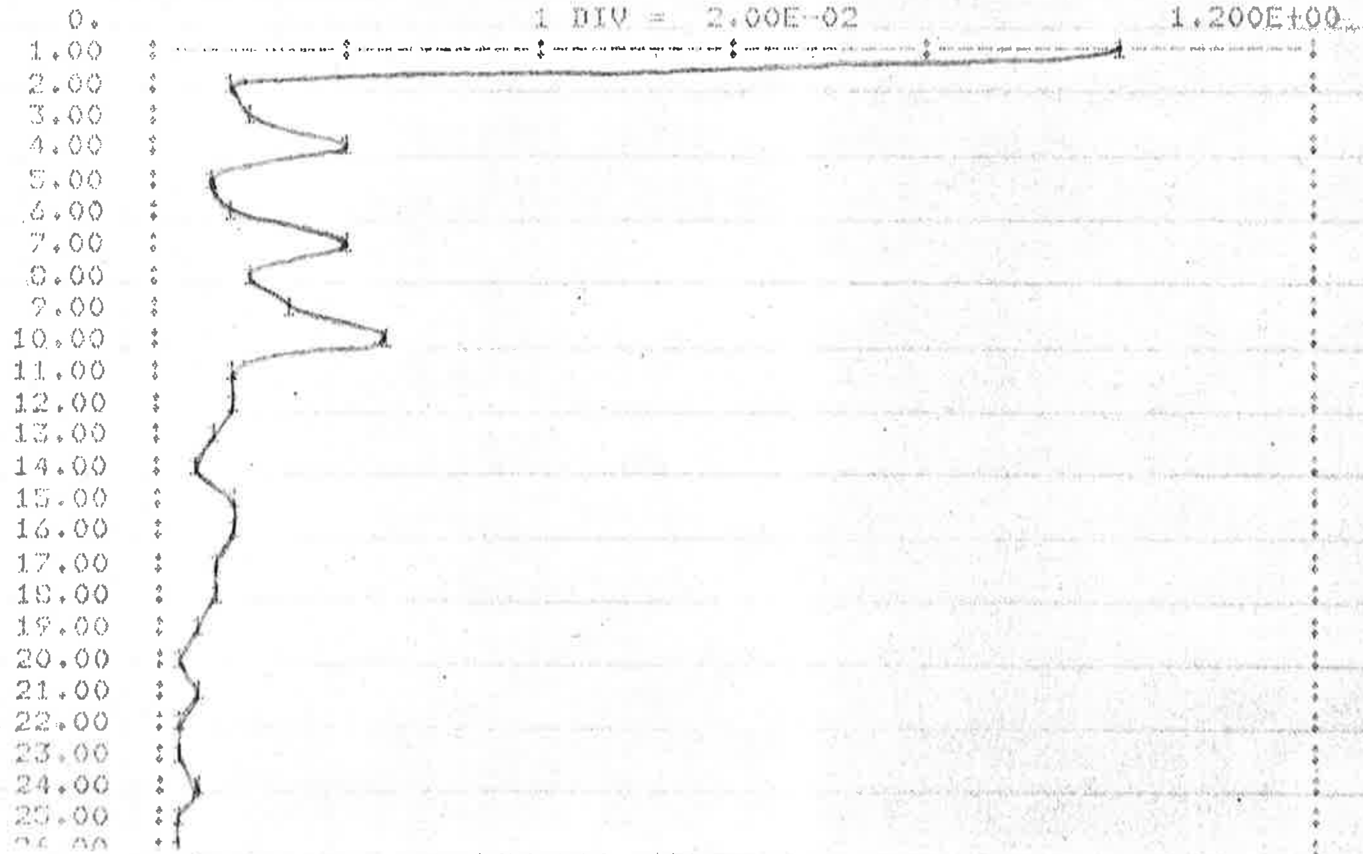
The linear prediction coefficients of speech time-waveform can be used to identify two Chinese speech sounds each time by means of parameter classification.

```
NO SUCH PROGRAM CALL NAME  =  L
COMMAND- edi mbrec

NO SUCH PROGRAM CALL NAME  =  ED
COMMAND- editor

..edi mbrec

..l=a
```

```
      PROGRAM MBREC(INPUT,OUTPUT)
      DIMENSION A(128),B(128),A1(128)
      COMMON R(128),AA(10),P(128)
      N=40
      IP=8
      W=(8.*ATAN(1.))/(128.*1.)
      IQ=128
      DATA(A=0.1,0.18,0.2,0.6,1.5,2.6,3.0,2.9,2.5,2.3,1.6,1.55,
     11.2,1.1,1.05,0.92,0.9,0.95,1.2,1.5,2.0,2.6,2.5,2.4,1.9
     1,1.75,1.4,1.0,0.97,1.0,1.21,1.2,1.15,1.04,1.06,1.18,1.
     13,1.32,1.3,1.1,1.0,0.8,0.2,0.17,0.2,0.3,0.6,1.0,2.0,3.
     11,3.0,2.0,2.2,1.8,1.5,1.3,1.1,0.85,0.87,0.9,1.0,1.05,1
     1.1,2.0,2.5,2.52,2.53,2.42,2.4,1.8,1.4,1.3,0.93,1.0,1.0
     15,1.23,1.2,1.15,1.03,1.01,1.1,1.2,1.3,1.28,1.05,0.8,0.
     17,0.15,0.10,0.19,0.2,0.4,1.0,2.9,3.1,2.95,2.6,2.2,2.0,
     11.1,1.2,1.05,0.87,0.89,0.91,0.75,1.1,1.2,1.8,2.5,2.58,
     12.5,2.35,2.0,1.45,1.2,1.1,0.7,1.0,1.16,1.24,1.18,1.12,
     11.05,1.03,1.02,1.20,1.3)
      DO 2 I=1,128
    2 B(I)=0.
C   NOW CALL UP FFT TO PRODUCE THE SPECTRUM P(W)
      CALL FFT2(A,B,7)
      CALL PLTYHT(60)
      DO 3 I=1,128
      A(I)=SQRT(A(I)**2+B(I)**2)
    3 A1(I)=20.*ALOG10(A(I))
C   COMPUTE THE AUTOCORRELATION
      CALL CORREL(A,R,N)
C    PLOT THE AUTOCORRELATION
      CALL PLT(N,1,R)
C SOLVE THE NORMAL EQUATION
      CALL DURBIN(R,AA,IP)
C   PRINT THE A(K)
      DO 10 I=1,IP
      PRINT 100,I,AA(I)
  100 FORMAT(1X,* AA(*,I2,*) = *,F10.3)
   10 CONTINUE
C    COMPUTE THE SPECTRUM FROM A(K)
      CALL APS(AA,R,IP,W,IQ,P)
      DO 20 I=1,IQ
      P(I)=20.*ALOG10(P(I))
   20 CONTINUE
C  PLOT A(K) SPECTRUM COMPARING WITH THE ORIGINAL SPECTRUM A1
      CALL PLT(128,2,A1,P)
      STOP
      END
      SUBROUTINE FFT2(X,Y,M)
C SUBROUTINE PERFORMANCE COOLEY-TUKEY VERSION OF FAST FOU
C RIER TRANSFORM.  X & Y CONTAIN REAL & IMAGINARY PARTS O
C F IN DATA RESPECTIVELY. 2**M IS THE NO. OF DATA INPUT
      DIMENSION X(1024),Y(1024)
      CALL ORDER(X,Y,M)
      PI=3.141593
      N=2**M
```

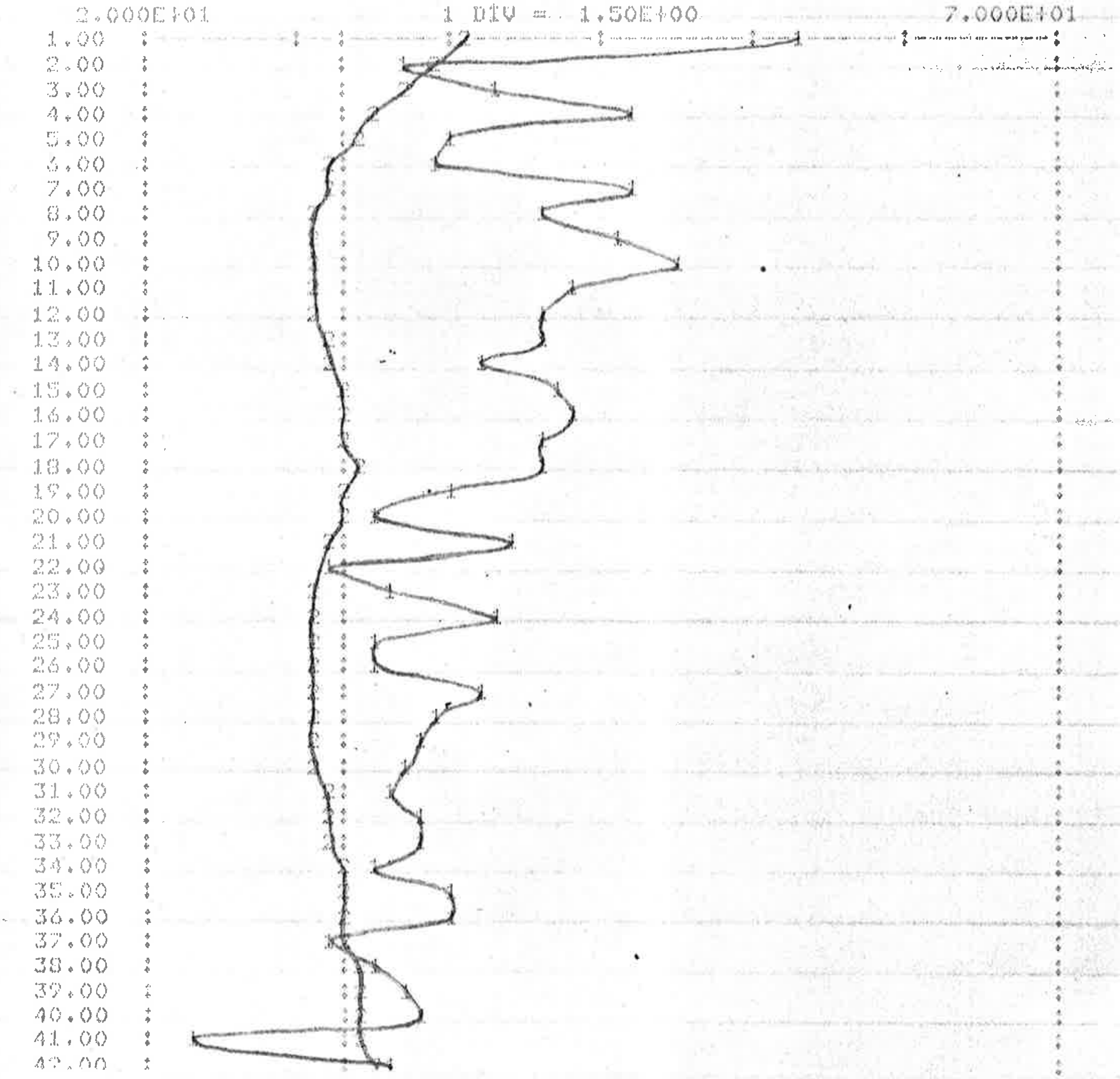
```
      DO 10 I=1,M
      IREP=2**I
      IDISP=IREP/2
      ARG=2.*PI/FLOAT(IREP)
      DO 10 J=1,IDISP
      TWF=FLOAT(J-1)*ARG
      C=COS(TWF)
      S=SIN(TWF)
      DO  10 K=J,N,IREP
      J2=K+IDISP
      T1=C*X(J2)+S*Y(J2)
      T2=-S*X(J2)+C*Y(J2)
      X(J2)=X(K)-T1
      Y(J2)=Y(K)-T2
      X(K)=X(K)+T1
      Y(K)=Y(K)+T2
10    CONTINUE
      RETURN
      END
      SUBROUTINE ORDER(X,Y,M)
C  THIS SUBROUTINE REORDERS THE X & Y ARRAYS I.E. BIT REVERSAL
      DIMENSION X(1024),Y(1024)
      N=2**M
      ND2=N/2
      NM1=N-1
      J=1
      DO 30 I=1,NM1
      IF(I.GE.J) GO TO 10
      T1=X(J)
      X(J)=X(I)
      X(I)=T1
      T2=Y(J)
      Y(J)=Y(I)
      Y(I)=T2
10    K=ND2
20    IF(K.GE.J) GO TO 30
      J=J-K
      K=K/2
      GO TO 20
30    J=J+K
      RETURN
      END
      SUBROUTINE CORREL(A,R,N)
      DIMENSION A(128),R(128)
      DO 1 I=1,N
      R(I)=0.
      NN=N-I+1
      DO 2 J=1,NN
      K=J+I-1
      R(I)=A(J)*A(K)+R(I)
2     CONTINUE
1     CONTINUE
C   NORMALIZE CORRELATION
      R1=R(1)
      DO 3 I=1,N
      R(I)=R(I)/R1
3     CONTINUE
      RETURN
      END
      SUBROUTINE DURBIN(R,AA,IP)
      DIMENSION R(128),AA(10)
      E=R(1)
      DO 1 I=1,IP
      SUM=0.
      IP1=I-1
      IF(IP1.LT.1) GO TO 5
```

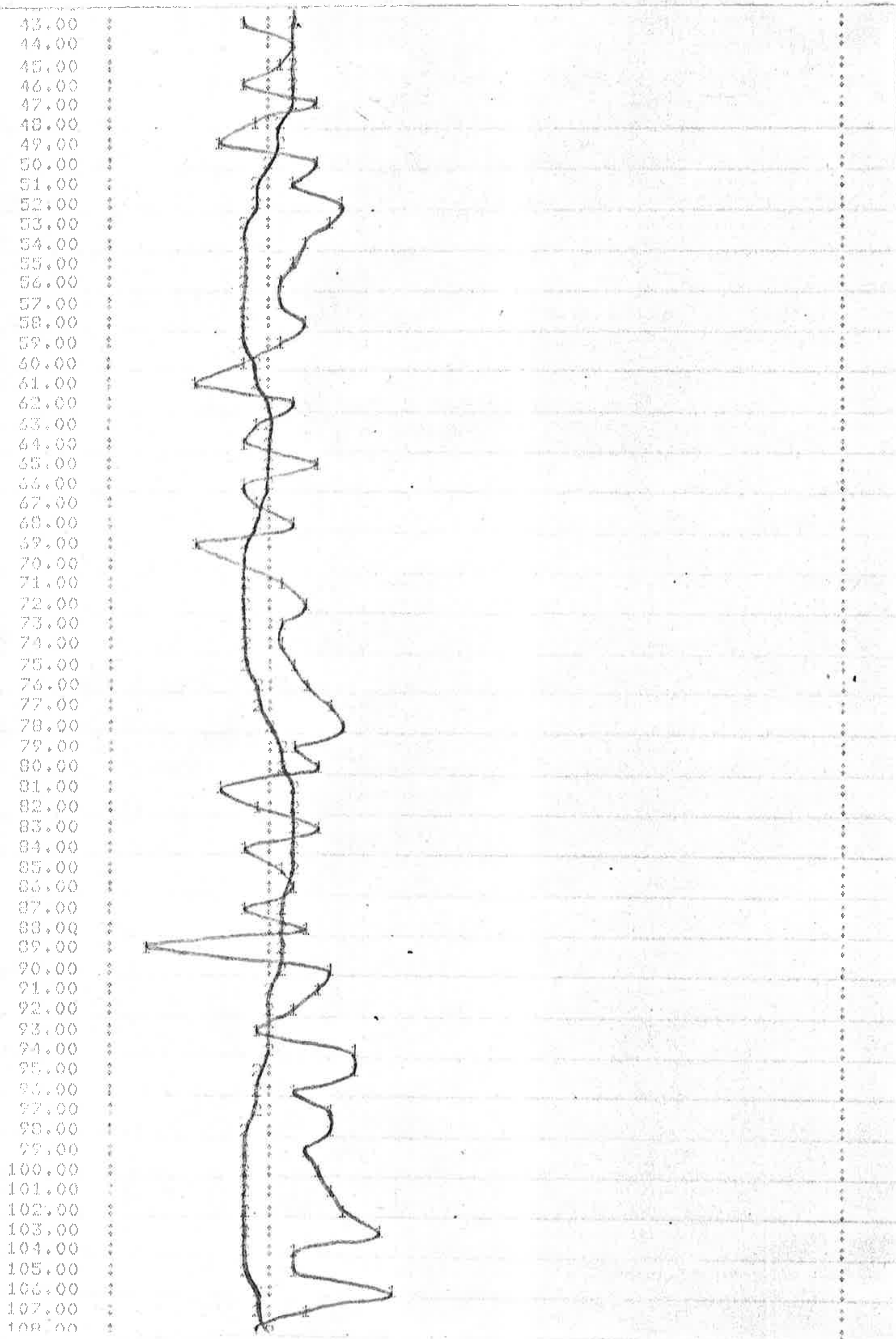
```
      DO 2 J=1,IP1
2     SUM=AA(J)*R(I-J+1)+SUM
5     CONTINUE
      AK=-(R(I+1)+SUM)/E
      AA(I)=AK
      IF(IP1.LT.1) GO TO 6
      DO 3 J=1,IP1
3     AA(J)=AA(J)+AK*AA(I-J)
6     CONTINUE
      E=(1.-(AK*AK))*E
1     CONTINUE
      RETURN
      END
      SUBROUTINE APS(AA,R,IP,W,IQ,P)
      DIMENSION AA(10),R(128),P(128)
      DIMENSION A11(10),A2(10)
      SUM=0.
      DO 1 I=1,IP
1     SUM=AA(I)*R(I+1)+SUM
      GSRD=R(1)+SUM
      DO 3 J=1,IQ
      DW=W*FLOAT(J)
      SUM1=0.
      SUM2=0.
      DO 2 K=1,IP
      A11(K)=AA(K)*COS(DW*K)
      A2(K)=AA(K)*SIN(DW*K)
      SUM1=A11(K)+SUM1
2     SUM2=A2(K)+SUM2
      SUM1=1.+SUM1
      DEM=(SUM1*SUM1)+(SUM2*SUM2)
      P(J)=GSRD/DEM
3     CONTINUE
      RETURN
      END
..run,ftn

  45000B   CM STORAGE USED
    .765 CP SECONDS COMPILATION TIME
```

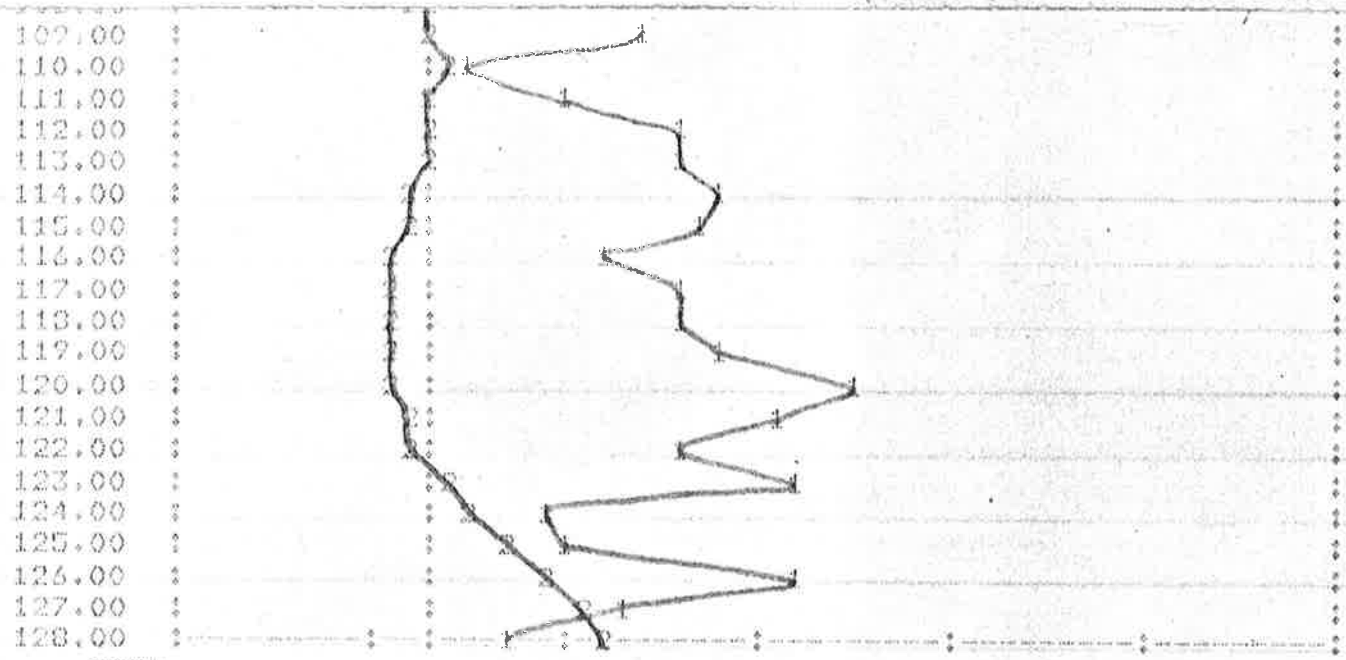


CM LWA+1 = 17041B, LOADER USED 32000B

AA(1) = -.037
AA(2) = -.042
AA(3) = -.157
AA(4) = -.012
AA(5) = -.010
AA(6) = .142
AA(7) = -.048
AA(8) = -.101




43.00
44.00
45.00
46.00
47.00
48.00
49.00
50.00
51.00
52.00
53.00
54.00
55.00
56.00
57.00
58.00
59.00
60.00
61.00
62.00
63.00
64.00
65.00
66.00
67.00
68.00
69.00
70.00
71.00
72.00
73.00
74.00
75.00
76.00
77.00
78.00
79.00
80.00
81.00
82.00
83.00
84.00
85.00
86.00
87.00
88.00
89.00
90.00
91.00
92.00
93.00
94.00
95.00
96.00
97.00
98.00
99.00
100.00
101.00
102.00
103.00
104.00
105.00
106.00
107.00



STOP
.631 CP SECONDS EXECUTION TIME

APPENDIX 9

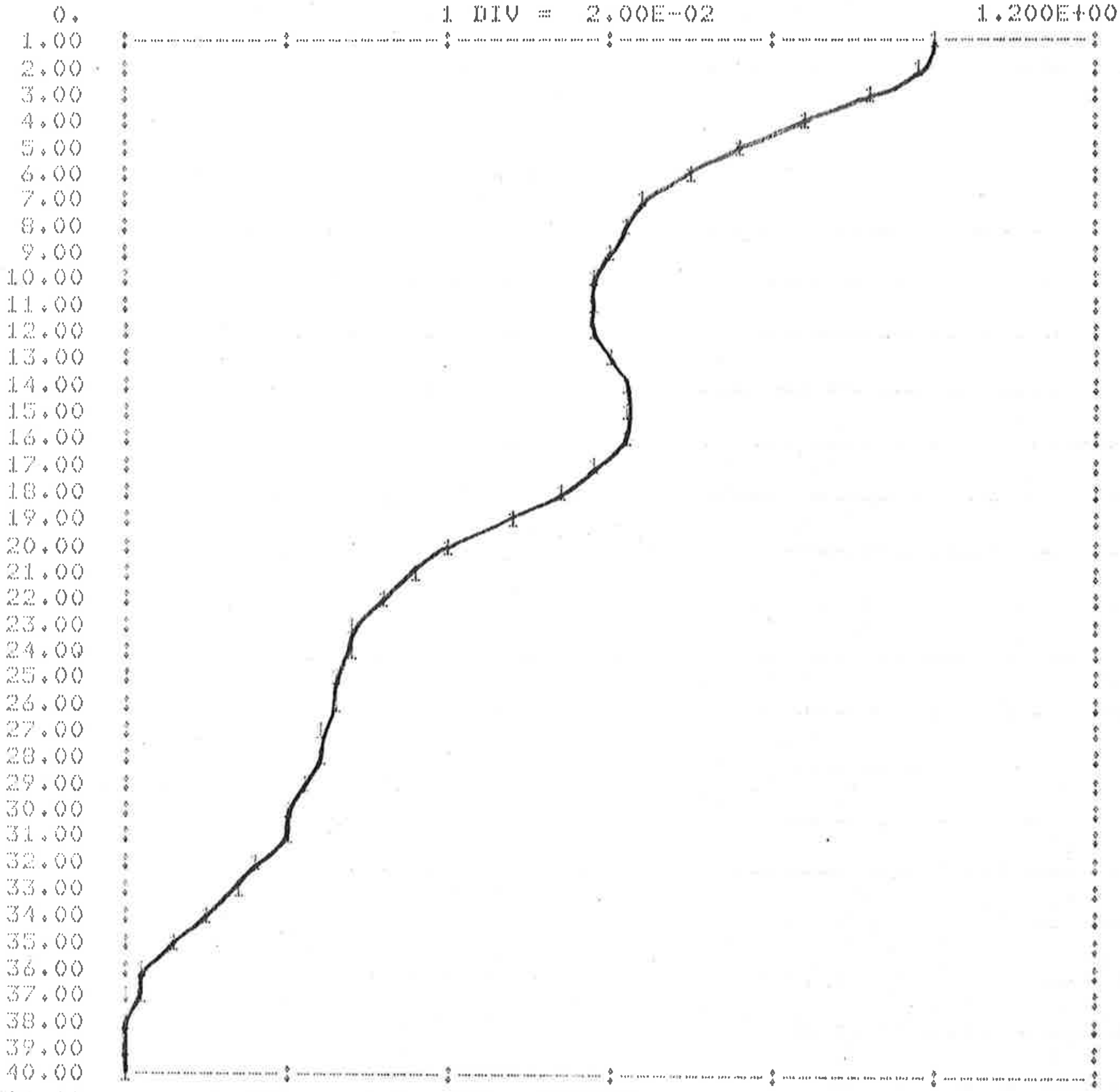
CDC Cyber 173 Computer Program- QBREC

Referring to Wakita's method(25)* we first calculate the autocorrelation coefficients R then the reflection coefficients REFL with Durbin's algorithm(5)*. From those reflection coefficients REFL we can count the area functions AREA(24)*. Equations (3), (8), (9) and (22) are utilized in this procedure.

Reflection coefficients or area functions of vocal-tract tube model can be accompanied with parameter classification to identify two Chinese speech sounds each time.

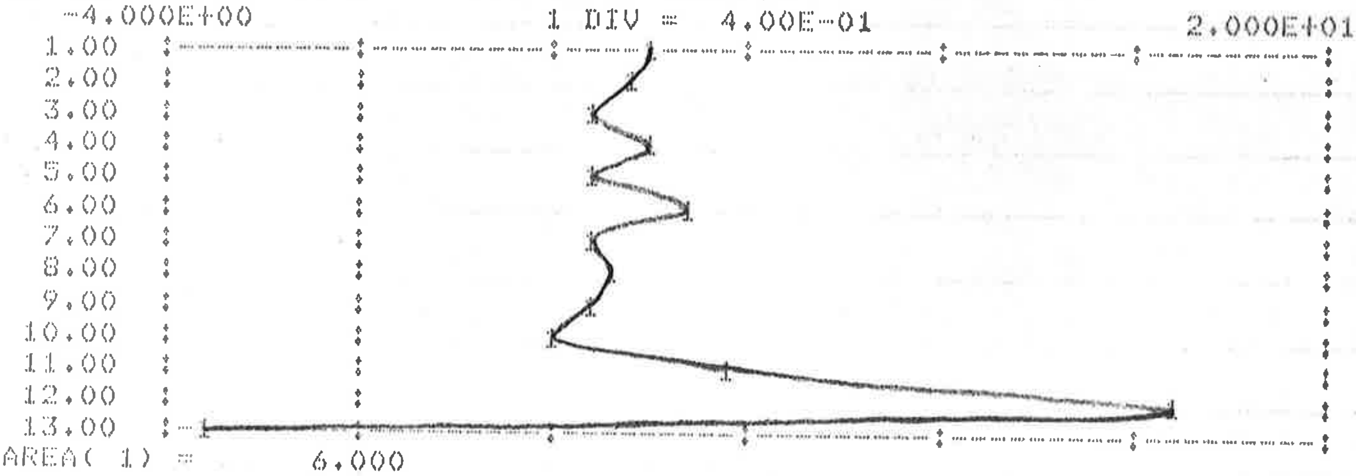
```
      PROGRAM QBREC(INPUT,OUTPUT)
      DIMENSION A(128),R(128),AA(12),AREA(13)
      N=40
      IP=12
      DATA(A=0.1,0.18,0.2,0.6,1.5,2.6,3.0,2.9,2.5,2.3,1.6,1.55,
     11.2,1.1,1.05,0.92,0.9,0.95,1.2,1.5,2.0,2.6,2.5,2.4,1.9
     11,1.75,1.4,1.0,0.97,1.0,1.21,1.2,1.15,1.04,1.06,1.18,1.
     13,1.32,1.3,1.1,1.0,0.8,0.2,0.19,0.2,0.3,0.6,1.0,2.0,3.
     11,3.0,2.8,2.2,1.8,1.5,1.3,1.1,0.85,0.87,0.9,1.0,1.05,1
     1.1,2.0,2.5,2.52,2.53,2.42,2.4,1.8,1.4,1.3,0.93,1.0,1.0
     15,1.23,1.2,1.15,1.03,1.01,1.1,1.2,1.3,1.28,1.05,0.8,0.
     17,0.15,0.18,0.19,0.2,0.4,1.0,2.9,3.1,2.95,2.6,2.2,2.0,
     11.4,1.2,1.05,0.87,0.89,0.91,0.95,1.1,1.2,1.8,2.5,2.58,
     12.5,2.35,2.0,1.45,1.2,1.1,0.9,1.0,1.16,1.24,1.18,1.12,
     11.05,1.03,1.02,1.28,1.3)
       CALL PLTYHT(60)
C     COMPUTE THE AUTOCORRELATION
       CALL CORREL(A,R,N)
C     PLOT THE AUTOCORRELATION
       CALL PLT(N,1,R)
C SOLVE THE NORMAL EQUATION
       CALL DURBIN(R,AA,IP)
C  PRINT THE A(K)  I.E. REFLECTION COEFFICIENTS
       DO 10 I=1,IP
       PRINT 100,I,AA(I)
100    FORMAT(1X,* AA(*,I2,*) = *,F10.3,3X,* I.E. REFL. COEFFI.
     1 *)
10     CONTINUE
C  CALCULATE THE AREA FUNCTIONS       AREA
       SM=6.
       AREA(1)=SM
       DO 60 J=1,IP
       JJ=IP+1-J
       SM=SM*(1.+AA(JJ))/(1.-AA(JJ))
60     AREA(J+1)=SM
       CALL PLT(13,1,AREA)
       DO 999 I=1,13
       PRINT 99,I,AREA(I)
99     FORMAT(1X,* AREA(*,I2,*) = *,F10.3)
999    CONTINUE
       STOP
       END
       SUBROUTINE CORREL(A,R,N)
       DIMENSION A(128),R(128)
       DO 1 I=1,N
       R(I)=0.
       NN=N-I+1
       DO 2 J=1,NN
       K=J+I-1
       R(I)=A(J)*A(K)+R(I)
2      CONTINUE
1      CONTINUE
C     NORMALIZE CORRELATION
       R1=R(1)
       DO 3 I=1,N
       R(I)=R(I)/R1
3      CONTINUE
       RETURN
       END
       SUBROUTINE DURBIN(R,AA,IP)
```

```
      DIMENSION R(128),AA(10)
      E=R(1)
      DO 1 I=1,IP
      SUM=0.
      IP1=I-1
      IF(IP1.LT.1) GO TO 5
      DO 2 J=1,IP1
2     SUM=AA(J)*R(I-J+1)+SUM
5     CONTINUE
      AK=-(R(I+1)+SUM)/E
      AA(I)=AK
      IF(IP1.LT.1) GO TO 6
      DO 3 J=1,IP1
3     AA(J)=AA(J)+AK*AA(I-J)
6     CONTINUE
      E=(1.-(AK*AK))*E
1     CONTINUE
      RETURN
      END
..run,ftn
```




```
AA( 1) =      -1.462    I.E. REFL. COEFFI.
  45000B   CM STORAGE USED
     .411 CP SECONDS COMPILATION TIME
  CM LWA+1 = 15400B, LOADER USED  30700B
```

```
AA( 2) =      .366    I.E. REFL. COEFFI.
AA( 3) =      .322    I.E. REFL. COEFFI.
AA( 4) =     -.116    I.E. REFL. COEFFI.
AA( 5) =     -.030    I.E. REFL. COEFFI.
AA( 6) =      .043    I.E. REFL. COEFFI.
AA( 7) =     -.169    I.E. REFL. COEFFI.
AA( 8) =      .170    I.E. REFL. COEFFI.
AA( 9) =     -.113    I.E. REFL. COEFFI.
AA(10) =      .122    I.E. REFL. COEFFI.
AA(11) =     -.068    I.E. REFL. COEFFI.
AA(12) =     -.048    I.E. REFL. COEFFI.
```




```
AREA( 1) =      6.000
AREA( 2) =      5.455
AREA( 3) =      4.760
AREA( 4) =      6.088
AREA( 5) =      4.857
AREA( 6) =      6.842
AREA( 7) =      4.862
AREA( 8) =      5.302
AREA( 9) =      4.996
AREA(10) =      3.958
AREA(11) =      7.718
AREA(12) =     16.640
AREA(13) =     -3.122
     STOP
        .171 CP SECONDS EXECUTION TIME
..100=$program  abrec(input,output)
110=$dimension a(128),r(128),aa(128),area(128)
```

APPENDIX 10

Experimental results of recognition processes using a CDC Digital Computer Cyber 173

Experimental results of recognition with the aid of CDC Digital Computer Cyber 173 are shown as following:

A. Recognitions with linear prediction coefficients

Pairs of sounds	Percentage of recognition			
ㄅ, ㄇ	ㄅ	100%	ㄇ	30%
ㄅ, ㄈ	ㄅ	100%	ㄈ	100%
ㄅ, ㄉ	ㄅ	100%	ㄉ	100%
ㄅ, ㄊ	ㄅ	100%	ㄊ	100%
ㄅ, ㄋ	ㄅ	100%	ㄋ	100%
ㄅ, ㄌ	ㄅ	100%	ㄌ	100%
ㄅ, ㄏ	ㄅ	100%	ㄏ	100%
ㄅ, ㄑ	ㄅ	100%	ㄑ	100%
ㄅ, ㄔ	ㄅ	100%	ㄔ	100%
ㄅ, ㄕ	ㄅ	100%	ㄕ	100%
ㄅ, ㄖ	ㄅ	100%	ㄖ	100%
ㄅ, ㄚ	ㄅ	100%	ㄚ	100%
ㄅ, ㄛ	ㄅ	100%	ㄛ	100%
ㄅ, ㄜ	ㄅ	100%	ㄜ	100%
ㄅ, ㄞ	ㄅ	100%	ㄞ	100%
ㄅ, ㄟ	ㄅ	100%	ㄟ	100%
ㄅ, ㄠ	ㄅ	100%	ㄠ	100%
ㄅ, ㄢ	ㄅ	100%	ㄢ	100%
ㄅ, ㄤ	ㄅ	100%	ㄤ	100%
ㄅ, ㄣ	ㄅ	100%	ㄣ	100%
ㄅ, ㄥ	ㄅ	100%	ㄥ	100%
ㄅ, ㄧ	ㄅ	100%	ㄧ	100%
ㄅ, ㄨ	ㄅ	100%	ㄨ	100%

ㄈ, ㄉ	ㄈ	100%	ㄉ	100%
ㄈ, ㄊ	ㄈ	100%	ㄊ	100%
ㄈ, ㄋ	ㄈ	100%	ㄋ	100%
ㄈ, ㄌ	ㄈ	100%	ㄌ	100%
ㄈ, ㄏ	ㄈ	100%	ㄏ	100%
ㄈ, ㄑ	ㄈ	100%	ㄑ	100%
ㄈ, ㄔ	ㄈ	100%	ㄔ	100%
ㄈ, ㄕ	ㄈ	100%	ㄕ	100%
ㄈ, ㄖ	ㄈ	100%	ㄖ	100%
ㄈ, ㄚ	ㄈ	100%	ㄚ	100%
ㄈ, ㄛ	ㄈ	100%	ㄛ	100%
ㄈ, ㄜ	ㄈ	100%	ㄜ	95%
ㄈ, ㄞ	ㄈ	100%	ㄞ	100%
ㄈ, ㄟ	ㄈ	100%	ㄟ	100%
ㄈ, ㄠ	ㄈ	100%	ㄠ	100%
ㄈ, ㄢ	ㄈ	100%	ㄢ	100%
ㄈ, ㄤ	ㄈ	100%	ㄤ	100%
ㄈ, ㄣ	ㄈ	100%	ㄣ	100%
ㄈ, ㄥ	ㄈ	100%	ㄥ	100%
ㄈ, ㄧ	ㄈ	100%	ㄧ	100%
ㄈ, ㄨ	ㄈ	100%	ㄨ	100%
ㄇ, ㄈ	ㄇ	100%	ㄈ	100%
ㄇ, ㄉ	ㄇ	100%	ㄉ	100%
ㄇ, ㄊ	ㄇ	100%	ㄊ	100%
ㄇ, ㄋ	ㄇ	95%	ㄋ	100%
ㄇ, ㄌ	ㄇ	95%	ㄌ	100%
ㄇ, ㄏ	ㄇ	100%	ㄏ	100%

ㄇ, ㄑ	ㄇ	100%	ㄑ	100%
ㄇ, ㄔ	ㄇ	100%	ㄔ	100%
ㄇ, ㄕ	ㄇ	15%	ㄕ	100%
ㄇ, ㄖ	ㄇ	95%	ㄖ	100%
ㄇ, ㄚ	ㄇ	100%	ㄚ	100%
ㄇ, ㄛ	ㄇ	100%	ㄛ	100%
ㄇ, ㄜ	ㄇ	25%	ㄜ	100%
ㄇ, ㄞ	ㄇ	100%	ㄞ	100%
ㄇ, ㄟ	ㄇ	95%	ㄟ	100%
ㄇ, ㄠ	ㄇ	95%	ㄠ	100%
ㄇ, ㄢ	ㄇ	100%	ㄢ	100%
ㄇ, ㄤ	ㄇ	100%	ㄤ	100%
ㄇ, ㄣ	ㄇ	95%	ㄣ	100%
ㄇ, ㄥ	ㄇ	100%	ㄥ	100%
ㄇ, ㄧ	ㄇ	15%	ㄧ	100%
ㄇ, ㄨ	ㄇ	100%	ㄨ	100%
ㄉ, ㄊ	ㄉ	100%	ㄊ	100%
ㄉ, ㄋ	ㄉ	100%	ㄋ	100%
ㄉ, ㄌ	ㄉ	100%	ㄌ	100%
ㄉ, ㄏ	ㄉ	100%	ㄏ	100%
ㄉ, ㄑ	ㄉ	100%	ㄑ	100%
ㄉ, ㄔ	ㄉ	100%	ㄔ	100%
ㄉ, ㄕ	ㄉ	100%	ㄕ	100%
ㄉ, ㄖ	ㄉ	100%	ㄖ	100%
ㄉ, ㄚ	ㄉ	100%	ㄚ	100%
ㄉ, ㄛ	ㄉ	100%	ㄛ	100%
ㄉ, ㄜ	ㄉ	100%	ㄜ	100%
ㄉ, ㄞ	ㄉ	100%	ㄞ	100%

ㄉ, ㄟ	ㄉ	100%	ㄟ	100%
ㄉ, ㄠ	ㄉ	100%	ㄠ	100%
ㄉ, ㄢ	ㄉ	100%	ㄢ	100%
ㄉ, ㄤ	ㄉ	100%	ㄤ	100%
ㄉ, ㄣ	ㄉ	100%	ㄣ	100%
ㄉ, ㄥ	ㄉ	100%	ㄥ	100%
ㄉ, ㄧ	ㄉ	100%	ㄧ	100%
ㄉ, ㄨ	ㄉ	100%	ㄨ	100%
ㄊ, ㄋ	ㄊ	100%	ㄋ	100%
ㄊ, ㄌ	ㄊ	100%	ㄌ	100%
ㄊ, ㄏ	ㄊ	100%	ㄏ	100%
ㄊ, ㄑ	ㄊ	100%	ㄑ	100%
ㄊ, ㄔ	ㄊ	100%	ㄔ	100%
ㄊ, ㄕ	ㄊ	100%	ㄕ	100%
ㄊ, ㄖ	ㄊ	100%	ㄖ	100%
ㄊ, ㄚ	ㄊ	100%	ㄚ	100%
ㄊ, ㄛ	ㄊ	100%	ㄛ	100%
ㄊ, ㄜ	ㄊ	100%	ㄜ	100%
ㄊ, ㄞ	ㄊ	100%	ㄞ	95%
ㄊ, ㄟ	ㄊ	100%	ㄟ	100%
ㄊ, ㄠ	ㄊ	100%	ㄠ	100%
ㄊ, ㄢ	ㄊ	100%	ㄢ	100%
ㄊ, ㄤ	ㄊ	100%	ㄤ	100%
ㄊ, ㄣ	ㄊ	100%	ㄣ	100%
ㄊ, ㄥ	ㄊ	100%	ㄥ	100%
ㄊ, ㄧ	ㄊ	100%	ㄧ	100%
ㄊ, ㄨ	ㄊ	100%	ㄨ	100%
ㄋ, ㄌ	ㄋ	100%	ㄌ	100%

ㄋ, ㄏ	ㄋ	100%	ㄏ	100%
ㄋ, ㄑ	ㄋ	100%	ㄑ	100%
ㄋ, ㄔ	ㄋ	100%	ㄔ	100%
ㄋ, ㄕ	ㄋ	100%	ㄕ	100%
ㄋ, ㄖ	ㄋ	100%	ㄖ	100%
ㄋ, ㄚ	ㄋ	100%	ㄚ	100%
ㄋ, ㄛ	ㄋ	100%	ㄛ	100%
ㄋ, ㄜ	ㄋ	100%	ㄜ	100%
ㄋ, ㄞ	ㄋ	100%	ㄞ	100%
ㄋ, ㄟ	ㄋ	100%	ㄟ	100%
ㄋ, ㄠ	ㄋ	100%	ㄠ	100%
ㄋ, ㄢ	ㄋ	100%	ㄢ	100%
ㄋ, ㄤ	ㄋ	100%	ㄤ	100%
ㄋ, ㄣ	ㄋ	100%	ㄣ	100%
ㄋ, ㄥ	ㄋ	100%	ㄥ	95%
ㄋ, ㄧ	ㄋ	100%	ㄧ	100%
ㄋ, ㄨ	ㄋ	100%	ㄨ	100%
ㄌ, ㄍ	ㄌ	100%	ㄍ	100%
ㄌ, ㄑ	ㄌ	100%	ㄑ	100%
ㄌ, ㄔ	ㄌ	100%	ㄔ	100%
ㄌ, ㄕ	ㄌ	100%	ㄕ	100%
ㄌ, ㄖ	ㄌ	100%	ㄖ	100%
ㄌ, ㄚ	ㄌ	100%	ㄚ	100%
ㄌ, ㄛ	ㄌ	100%	ㄛ	100%
ㄌ, ㄜ	ㄌ	100%	ㄜ	100%
ㄌ, ㄞ	ㄌ	100%	ㄞ	100%
ㄌ, ㄟ	ㄌ	100%	ㄟ	100%
ㄌ, ㄠ	ㄌ	100%	ㄠ	100%
ㄌ, ㄢ	ㄌ	100%	ㄢ	100%

ㄉ, ㄤ	ㄉ	100%	ㄤ	100%
ㄉ, ㄣ	ㄉ	100%	ㄣ	100%
ㄉ, ㄥ	ㄉ	100%	ㄥ	100%
ㄉ, ㄧ	ㄉ	100%	ㄧ	100%
ㄉ, ㄨ	ㄉ	100%	ㄨ	100%
ㄏ, ㄑ	ㄏ	100%	ㄑ	100%
ㄏ, ㄔ	ㄏ	100%	ㄔ	100%
ㄏ, ㄕ	ㄏ	100%	ㄕ	100%
ㄏ, ㄖ	ㄏ	100%	ㄖ	100%
ㄏ, ㄚ	ㄏ	100%	ㄚ	100%
ㄏ, ㄛ	ㄏ	100%	ㄛ	100%
ㄏ, ㄜ	ㄏ	100%	ㄜ	100%
ㄏ, ㄞ	ㄏ	100%	ㄞ	100%
ㄏ, ㄟ	ㄏ	100%	ㄟ	100%
ㄏ, ㄠ	ㄏ	100%	ㄠ	100%
ㄏ, ㄢ	ㄏ	100%	ㄢ	100%
ㄏ, ㄤ	ㄏ	100%	ㄤ	100%
ㄏ, ㄣ	ㄏ	100%	ㄣ	100%
ㄏ, ㄥ	ㄏ	100%	ㄥ	100%
ㄏ, ㄧ	ㄏ	100%	ㄧ	100%
ㄏ, ㄨ	ㄏ	100%	ㄨ	100%
ㄑ, ㄔ	ㄑ	100%	ㄔ	100%
ㄑ, ㄒ	ㄑ	100%	ㄒ	100%
ㄑ, ㄖ	ㄑ	100%	ㄖ	100%
ㄑ, ㄚ	ㄑ	100%	ㄚ	100%
ㄑ, ㄛ	ㄑ	100%	ㄛ	100%
ㄑ, ㄜ	ㄑ	100%	ㄜ	100%
ㄑ, ㄞ	ㄑ	100%	ㄞ	100%

ㄑ, ㄟ	ㄑ	100%	ㄟ	100%
ㄑ, ㄠ	ㄑ	100%	ㄠ	100%
ㄑ, ㄢ	ㄑ	100%	ㄢ	100%
ㄑ, ㄤ	ㄑ	100%	ㄤ	100%
ㄑ, ㄣ	ㄑ	100%	ㄣ	100%
ㄑ, ㄥ	ㄑ	100%	ㄥ	100%
ㄑ, ㄧ	ㄑ	100%	ㄧ	100%
ㄑ, ㄨ	ㄑ	100%	ㄨ	100%
ㄔ, ㄕ	ㄔ	100%	ㄕ	100%
ㄔ, ㄖ	ㄔ	100%	ㄖ	100%
ㄔ, ㄚ	ㄔ	100%	ㄚ	100%
ㄔ, ㄛ	ㄔ	100%	ㄛ	100%
ㄔ, ㄜ	ㄔ	100%	ㄜ	100%
ㄔ, ㄞ	ㄔ	100%	ㄞ	100%
ㄔ, ㄟ	ㄔ	100%	ㄟ	100%
ㄔ, ㄠ	ㄔ	100%	ㄠ	100%
ㄔ, ㄢ	ㄔ	100%	ㄢ	100%
ㄔ, ㄤ	ㄔ	100%	ㄤ	100%
ㄔ, ㄣ	ㄔ	100%	ㄣ	100%
ㄔ, ㄥ	ㄔ	100%	ㄥ	100%
ㄔ, ㄧ	ㄔ	100%	ㄧ	100%
ㄔ, ㄨ	ㄔ	100%	ㄨ	100%
ㄕ, ㄖ	ㄕ	100%	ㄖ	100%
ㄕ, ㄚ	ㄕ	100%	ㄚ	100%
ㄕ, ㄛ	ㄕ	100%	ㄛ	100%
ㄕ, ㄜ	ㄕ	100%	ㄜ	100%
ㄕ, ㄞ	ㄕ	100%	ㄞ	100%
ㄕ, ㄟ	ㄕ	100%	ㄟ	100%

ㄗ, ㄠ	ㄗ	100%	ㄠ	100%
ㄗ, ㄢ	ㄗ	100%	ㄢ	100%
ㄗ, ㄤ	ㄗ	100%	ㄤ	100%
ㄗ, ㄣ	ㄗ	100%	ㄣ	100%
ㄗ, ㄥ	ㄗ	100%	ㄥ	100%
ㄗ, ㄧ	ㄗ	100%	ㄧ	100%
ㄗ, ㄨ	ㄗ	100%	ㄨ	100%
ㄖ, ㄚ	ㄖ	100%	ㄚ	100%
ㄖ, ㄛ	ㄖ	100%	ㄛ	100%
ㄖ, ㄜ	ㄖ	100%	ㄜ	100%
ㄖ, ㄞ	ㄖ	100%	ㄞ	100%
ㄖ, ㄟ	ㄖ	100%	ㄟ	100%
ㄖ, ㄠ	ㄖ	100%	ㄠ	100%
ㄖ, ㄢ	ㄖ	100%	ㄢ	100%
ㄖ, ㄤ	ㄖ	100%	ㄤ	100%
ㄖ, ㄣ	ㄖ	100%	ㄣ	100%
ㄖ, ㄥ	ㄖ	100%	ㄥ	100%
ㄖ, ㄧ	ㄖ	100%	ㄧ	100%
ㄖ, ㄨ	ㄖ	100%	ㄨ	100%
ㄚ, ㄛ	ㄚ	100%	ㄛ	100%
ㄚ, ㄜ	ㄚ	100%	ㄜ	100%
ㄚ, ㄞ	ㄚ	100%	ㄞ	100%
ㄚ, ㄟ	ㄚ	100%	ㄟ	100%
ㄚ, ㄠ	ㄚ	100%	ㄠ	100%
ㄚ, ㄢ	ㄚ	100%	ㄢ	100%
ㄚ, ㄤ	ㄚ	100%	ㄤ	100%
ㄚ, ㄣ	ㄚ	100%	ㄣ	100%
ㄚ, ㄥ	ㄚ	100%	ㄥ	100%

ㄚ, ㄧ	ㄚ	100%	ㄧ	100%
ㄚ, ㄨ	ㄚ	100%	ㄨ	100%
ㄛ, ㄜ	ㄛ	100%	ㄜ	100%
ㄛ, ㄞ	ㄛ	100%	ㄞ	100%
ㄛ, ㄠ	ㄛ	100%	ㄠ	100%
ㄛ, ㄢ	ㄛ	100%	ㄢ	100%
ㄛ, ㄤ	ㄛ	100%	ㄤ	100%
ㄛ, ㄣ	ㄛ	100%	ㄣ	100%
ㄛ, ㄥ	ㄛ	100%	ㄥ	100%
ㄛ, ㄧ	ㄛ	100%	ㄧ	100%
ㄛ, ㄨ	ㄛ	100%	ㄨ	100%
ㄜ, ㄞ	ㄜ	100%	ㄞ	100%
ㄜ, ㄟ	ㄜ	100%	ㄟ	100%
ㄜ, ㄠ	ㄜ	100%	ㄠ	100%
ㄜ, ㄢ	ㄜ	100%	ㄢ	100%
ㄜ, ㄤ	ㄜ	100%	ㄤ	100%
ㄜ, ㄣ	ㄜ	100%	ㄣ	100%
ㄜ, ㄥ	ㄜ	100%	ㄥ	100%
ㄜ, ㄧ	ㄜ	100%	ㄧ	100%
ㄜ, ㄨ	ㄜ	100%	ㄨ	100%
ㄞ, ㄟ	ㄞ	100%	ㄟ	100%
ㄞ, ㄠ	ㄞ	95%	ㄠ	100%
ㄞ, ㄢ	ㄞ	100%	ㄢ	100%
ㄞ, ㄤ	ㄞ	100%	ㄤ	100%
ㄞ, ㄣ	ㄞ	95%	ㄣ	100%
ㄞ, ㄥ	ㄞ	100%	ㄥ	100%
ㄞ, ㄧ	ㄞ	100%	ㄧ	100%

ㄞ, ㄨ	ㄞ	95%	ㄨ	100%
ㄟ, ㄠ	ㄟ	100%	ㄠ	100%
ㄟ, ㄢ	ㄟ	100%	ㄢ	100%
ㄟ, ㄤ	ㄟ	100%	ㄤ	100%
ㄟ, ㄣ	ㄟ	100%	ㄣ	100%
ㄟ, ㄥ	ㄟ	100%	ㄥ	100%
ㄟ, ㄧ	ㄟ	100%	ㄧ	100%
ㄟ, ㄨ	ㄟ	100%	ㄨ	100%
ㄠ, ㄢ	ㄠ	100%	ㄢ	100%
ㄠ, ㄣ	ㄠ	100%	ㄣ	100%
ㄠ, ㄥ	ㄠ	100%	ㄥ	100%
ㄠ, ㄧ	ㄠ	100%	ㄧ	100%
ㄠ, ㄨ	ㄠ	100%	ㄨ	100%
ㄢ, ㄤ	ㄢ	100%	ㄤ	100%
ㄢ, ㄣ	ㄢ	100%	ㄣ	100%
ㄢ, ㄥ	ㄢ	100%	ㄥ	100%
ㄢ, ㄧ	ㄢ	100%	ㄧ	100%
ㄢ, ㄨ	ㄢ	100%	ㄨ	100%
ㄤ, ㄣ	ㄤ	100%	ㄣ	100%
ㄤ, ㄥ	ㄤ	100%	ㄥ	100%
ㄤ, ㄧ	ㄤ	100%	ㄧ	100%
ㄤ, ㄨ	ㄤ	100%	ㄨ	100%
ㄣ, ㄥ	ㄣ	100%	ㄥ	100%
ㄣ, ㄧ	ㄣ	95%	ㄧ	100%
ㄣ, ㄨ	ㄣ	100%	ㄨ	100%
ㄥ, ㄧ	ㄥ	100%	ㄧ	100%
ㄥ, ㄨ	ㄥ	100%	ㄨ	100%
ㄧ, ㄨ	ㄧ	100%	ㄨ	100%

ㄅ, ㄆ	ㄅ	100%	ㄆ	100%
ㄅ, ㄓ	ㄅ	100%	ㄓ	100%
ㄆ, ㄓ	ㄆ	100%	ㄓ	100%
ㄌ, ㄎ	ㄌ	100%	ㄎ	100%
ㄍ, ㄎ	ㄍ	100%	ㄎ	100%
ㄏ, ㄐ	ㄏ	100%	ㄐ	100%
ㄐ, ㄒ	ㄐ	50%	ㄒ	50%
ㄐ, ㄓ	ㄐ	50%	ㄓ	50%
ㄒ, ㄓ	ㄒ	50%	ㄓ	50%
ㄛ, ㄟ	ㄛ	50%	ㄟ	50%
ㄛ, ㄡ	ㄛ	50%	ㄡ	50%
ㄟ, ㄡ	ㄟ	50%	ㄡ	50%
ㄑ, ㄕ	ㄑ	50%	ㄕ	50%
ㄎ, ㄏ	ㄎ	50%	ㄏ	50%
ㄖ, ㄗ	ㄖ	50%	ㄗ	0%
ㄖ, ㄘ	ㄖ	50%	ㄘ	50%
ㄖ, ㄙ	ㄖ	50%	ㄙ	50%
ㄗ, ㄘ	ㄗ	50%	ㄘ	0%
ㄗ, ㄙ	ㄗ	0%	ㄙ	50%
ㄘ, ㄙ	ㄘ	50%	ㄙ	50%
ㄜ, ㄝ	ㄜ	50%	ㄝ	50%
ㄠ, ㄤ	ㄠ	50%	ㄤ	50%
ㄧ, ㄩ	ㄧ	50%	ㄩ	50%
ㄣ, ㄦ	ㄣ	50%	ㄦ	50%

B. Recognitions with vocal tract areas

Pairs of sounds	Percentage of recognition			
ㄅ, ㄇ	ㄅ	100%	ㄇ	100%
ㄅ, ㄈ	ㄅ	100%	ㄈ	100%
ㄅ, ㄉ	ㄅ	100%	ㄉ	100%
ㄅ, ㄊ	ㄅ	100%	ㄊ	100%
ㄅ, ㄋ	ㄅ	100%	ㄋ	100%
ㄅ, ㄍ	ㄅ	100%	ㄍ	100%
ㄅ, ㄏ	ㄅ	100%	ㄏ	100%
ㄅ, ㄑ	ㄅ	100%	ㄑ	100%
ㄅ, ㄔ	ㄅ	100%	ㄔ	100%
ㄅ, ㄕ	ㄅ	100%	ㄕ	100%
ㄅ, ㄖ	ㄅ	100%	ㄖ	100%
ㄅ, ㄚ	ㄅ	100%	ㄚ	100%
ㄅ, ㄛ	ㄅ	100%	ㄛ	100%
ㄅ, ㄜ	ㄅ	100%	ㄜ	100%
ㄅ, ㄞ	ㄅ	100%	ㄞ	100%
ㄅ, ㄟ	ㄅ	100%	ㄟ	100%
ㄅ, ㄠ	ㄅ	100%	ㄠ	100%
ㄅ, ㄢ	ㄅ	100%	ㄢ	100%
ㄅ, ㄤ	ㄅ	100%	ㄤ	100%
ㄅ, ㄣ	ㄅ	100%	ㄣ	100%
ㄅ, ㄥ	ㄅ	100%	ㄥ	100%
ㄅ, ㄧ	ㄅ	100%	ㄧ	100%
ㄅ, ㄨ	ㄅ	100%	ㄨ	100%
ㄇ, ㄈ	ㄇ	100%	ㄈ	100%

ㄇ, ㄉ	ㄇ	100%	ㄉ	100%
ㄇ, ㄊ	ㄇ	100%	ㄊ	100%
ㄇ, ㄋ	ㄇ	100%	ㄋ	100%
ㄇ, ㄌ	ㄇ	100%	ㄌ	100%
ㄇ, ㄏ	ㄇ	100%	ㄏ	100%
ㄇ, ㄑ	ㄇ	100%	ㄑ	100%
ㄇ, ㄔ	ㄇ	100%	ㄔ	100%
ㄇ, ㄕ	ㄇ	100%	ㄕ	100%
ㄇ, ㄖ	ㄇ	100%	ㄖ	100%
ㄇ, ㄚ	ㄇ	90%	ㄚ	100%
ㄇ, ㄛ	ㄇ	100%	ㄛ	100%
ㄇ, ㄜ	ㄇ	100%	ㄜ	100%
ㄇ, ㄞ	ㄇ	100%	ㄞ	100%
ㄇ, ㄟ	ㄇ	100%	ㄟ	100%
ㄇ, ㄠ	ㄇ	100%	ㄠ	100%
ㄇ, ㄢ	ㄇ	100%	ㄢ	100%
ㄇ, ㄤ	ㄇ	100%	ㄤ	100%
ㄇ, ㄣ	ㄇ	100%	ㄣ	100%
ㄇ, ㄥ	ㄇ	100%	ㄥ	100%
ㄇ, ㄧ	ㄇ	100%	ㄧ	100%
ㄇ, ㄨ	ㄇ	100%	ㄨ	100%
ㄈ, ㄉ	ㄈ	100%	ㄉ	100%
ㄈ, ㄊ	ㄈ	100%	ㄊ	100%
ㄈ, ㄋ	ㄈ	100%	ㄋ	100%
ㄈ, ㄌ	ㄈ	100%	ㄌ	100%
ㄈ, ㄏ	ㄈ	100%	ㄏ	100%
ㄈ, ㄑ	ㄈ	100%	ㄑ	100%
ㄈ, ㄔ	ㄈ	100%	ㄔ	100%

ㄈ, ㄗ	ㄈ	100%	ㄗ	100%
ㄈ, ㄖ	ㄈ	100%	ㄖ	100%
ㄈ, ㄚ	ㄈ	100%	ㄚ	100%
ㄈ, ㄛ	ㄈ	100%	ㄛ	100%
ㄈ, ㄜ	ㄈ	100%	ㄜ	100%
ㄈ, ㄞ	ㄈ	100%	ㄞ	100%
ㄈ, ㄟ	ㄈ	100%	ㄟ	100%
ㄈ, ㄠ	ㄈ	100%	ㄠ	100%
ㄈ, ㄢ	ㄈ	100%	ㄢ	100%
ㄈ, ㄤ	ㄈ	100%	ㄤ	100%
ㄈ, ㄣ	ㄈ	100%	ㄣ	100%
ㄈ, ㄥ	ㄈ	100%	ㄥ	100%
ㄈ, ㄧ	ㄈ	100%	ㄧ	100%
ㄈ, ㄨ	ㄈ	100%	ㄨ	100%
ㄉ, ㄊ	ㄉ	100%	ㄊ	100%
ㄉ, ㄋ	ㄉ	100%	ㄋ	100%
ㄉ, ㄓ	ㄉ	100%	ㄓ	100%
ㄉ, ㄏ	ㄉ	100%	ㄏ	100%
ㄉ, ㄑ	ㄉ	100%	ㄑ	100%
ㄉ, ㄔ	ㄉ	100%	ㄔ	100%
ㄉ, ㄗ	ㄉ	100%	ㄗ	100%
ㄉ, ㄖ	ㄉ	100%	ㄖ	100%
ㄉ, ㄚ	ㄉ	100%	ㄚ	100%
ㄉ, ㄛ	ㄉ	100%	ㄛ	100%
ㄉ, ㄜ	ㄉ	100%	ㄜ	100%
ㄉ, ㄞ	ㄉ	100%	ㄞ	100%
ㄉ, ㄟ	ㄉ	100%	ㄟ	100%
ㄉ, ㄠ	ㄉ	100%	ㄠ	100%

ㄉ, ㄢ	ㄉ	100%	ㄢ	100%
ㄉ, ㄤ	ㄉ	100%	ㄤ	100%
ㄉ, ㄣ	ㄉ	100%	ㄣ	100%
ㄉ, ㄥ	ㄉ	100%	ㄥ	100%
ㄉ, ㄧ	ㄉ	100%	ㄧ	100%
ㄉ, ㄨ	ㄉ	100%	ㄨ	100%
ㄊ, ㄋ	ㄊ	100%	ㄋ	100%
ㄊ, ㄌ	ㄊ	100%	ㄌ	100%
ㄊ, ㄏ	ㄊ	100%	ㄏ	100%
ㄊ, ㄑ	ㄊ	100%	ㄑ	100%
ㄊ, ㄔ	ㄊ	100%	ㄔ	100%
ㄊ, ㄕ	ㄊ	100%	ㄕ	100%
ㄊ, ㄖ	ㄊ	100%	ㄖ	100%
ㄊ, ㄚ	ㄊ	100%	ㄚ	100%
ㄊ, ㄛ	ㄊ	100%	ㄛ	100%
ㄊ, ㄜ	ㄊ	100%	ㄜ	100%
ㄊ, ㄞ	ㄊ	100%	ㄞ	100%
ㄊ, ㄟ	ㄊ	100%	ㄟ	100%
ㄊ, ㄠ	ㄊ	100%	ㄠ	100%
ㄊ, ㄢ	ㄊ	100%	ㄢ	100%
ㄊ, ㄤ	ㄊ	100%	ㄤ	100%
ㄊ, ㄣ	ㄊ	100%	ㄣ	100%
ㄊ, ㄥ	ㄊ	100%	ㄥ	100%
ㄊ, ㄧ	ㄊ	100%	ㄧ	100%
ㄊ, ㄨ	ㄊ	100%	ㄨ	100%
ㄋ, ㄌ	ㄋ	100%	ㄌ	100%
ㄋ, ㄏ	ㄋ	100%	ㄏ	100%
ㄋ, ㄑ	ㄋ	100%	ㄑ	100%

ㄋ, ㄔ	ㄋ	100%	ㄔ	100%
ㄋ, ㄕ	ㄋ	100%	ㄕ	100%
ㄋ, ㄖ	ㄋ	100%	ㄖ	100%
ㄋ, ㄚ	ㄋ	100%	ㄚ	100%
ㄋ, ㄛ	ㄋ	100%	ㄛ	100%
ㄋ, ㄜ	ㄋ	100%	ㄜ	100%
ㄋ, ㄞ	ㄋ	100%	ㄞ	100%
ㄋ, ㄟ	ㄋ	100%	ㄟ	100%
ㄋ, ㄠ	ㄋ	100%	ㄠ	95%
ㄋ, ㄢ	ㄋ	100%	ㄢ	100%
ㄋ, ㄤ	ㄋ	100%	ㄤ	100%
ㄋ, ㄣ	ㄋ	100%	ㄣ	100%
ㄋ, ㄥ	ㄋ	100%	ㄥ	70%
ㄋ, ㄧ	ㄋ	100%	ㄧ	100%
ㄋ, ㄨ	ㄋ	100%	ㄨ	100%
ㄌ, ㄏ	ㄌ	100%	ㄏ	100%
ㄌ, ㄑ	ㄌ	100%	ㄑ	100%
ㄌ, ㄔ	ㄌ	100%	ㄔ	100%
ㄌ, ㄕ	ㄌ	100%	ㄕ	100%
ㄌ, ㄖ	ㄌ	100%	ㄖ	100%
ㄌ, ㄚ	ㄌ	100%	ㄚ	100%
ㄌ, ㄛ	ㄌ	100%	ㄛ	100%
ㄌ, ㄜ	ㄌ	100%	ㄜ	100%
ㄌ, ㄞ	ㄌ	100%	ㄞ	100%
ㄌ, ㄟ	ㄌ	100%	ㄟ	100%
ㄌ, ㄠ	ㄌ	100%	ㄠ	100%
ㄌ, ㄢ	ㄌ	100%	ㄢ	100%
ㄌ, ㄤ	ㄌ	100%	ㄤ	100%

ㄌ, ㄣ	ㄌ	100%	ㄣ	100%
ㄌ, ㄥ	ㄌ	100%	ㄥ	100%
ㄌ, ㄧ	ㄌ	100%	ㄧ	100%
ㄌ, ㄨ	ㄌ	100%	ㄨ	100%
ㄏ, ㄑ	ㄏ	100%	ㄑ	100%
ㄏ, ㄔ	ㄏ	100%	ㄔ	100%
ㄏ, ㄕ	ㄏ	100%	ㄕ	100%
ㄏ, ㄖ	ㄏ	100%	ㄖ	100%
ㄏ, ㄚ	ㄏ	100%	ㄚ	100%
ㄏ, ㄛ	ㄏ	100%	ㄛ	100%
ㄏ, ㄜ	ㄏ	100%	ㄜ	100%
ㄏ, ㄞ	ㄏ	100%	ㄞ	100%
ㄏ, ㄟ	ㄏ	100%	ㄟ	100%
ㄏ, ㄠ	ㄏ	100%	ㄠ	100%
ㄏ, ㄢ	ㄏ	100%	ㄢ	100%
ㄏ, ㄤ	ㄏ	100%	ㄤ	100%
ㄏ, ㄣ	ㄏ	100%	ㄣ	100%
ㄏ, ㄥ	ㄏ	100%	ㄥ	100%
ㄏ, ㄧ	ㄏ	100%	ㄧ	100%
ㄏ, ㄨ	ㄏ	100%	ㄨ	100%
ㄑ, ㄔ	ㄑ	100%	ㄔ	100%
ㄑ, ㄒ	ㄑ	100%	ㄒ	100%
ㄑ, ㄖ	ㄑ	100%	ㄖ	100%
ㄑ, ㄚ	ㄑ	100%	ㄚ	100%
ㄑ, ㄛ	ㄑ	100%	ㄛ	100%
ㄑ, ㄜ	ㄑ	100%	ㄜ	100%
ㄑ, ㄞ	ㄑ	100%	ㄞ	100%
ㄑ, ㄟ	ㄑ	100%	ㄟ	100%

ㄑ, ㄠ	ㄑ	100%	ㄠ	100%
ㄑ, ㄢ	ㄑ	100%	ㄢ	100%
ㄑ, ㄤ	ㄑ	100%	ㄤ	100%
ㄑ, ㄣ	ㄑ	100%	ㄣ	100%
ㄑ, ㄥ	ㄑ	100%	ㄥ	100%
ㄑ, ㄧ	ㄑ	100%	ㄧ	100%
ㄑ, ㄨ	ㄑ	100%	ㄨ	100%
ㄔ, ㄕ	ㄔ	100%	ㄕ	100%
ㄔ, ㄖ	ㄔ	100%	ㄖ	100%
ㄔ, ㄚ	ㄔ	100%	ㄚ	100%
ㄔ, ㄛ	ㄔ	100%	ㄛ	100%
ㄔ, ㄜ	ㄔ	100%	ㄜ	100%
ㄔ, ㄞ	ㄔ	100%	ㄞ	100%
ㄔ, ㄟ	ㄔ	100%	ㄟ	100%
ㄔ, ㄠ	ㄔ	100%	ㄠ	100%
ㄔ, ㄢ	ㄔ	100%	ㄢ	100%
ㄔ, ㄤ	ㄔ	100%	ㄤ	100%
ㄔ, ㄣ	ㄔ	100%	ㄣ	100%
ㄔ, ㄥ	ㄔ	100%	ㄥ	100%
ㄔ, ㄧ	ㄔ	100%	ㄧ	100%
ㄔ, ㄨ	ㄔ	100%	ㄨ	100%
ㄕ, ㄖ	ㄕ	100%	ㄖ	100%
ㄕ, ㄚ	ㄕ	100%	ㄚ	100%
ㄕ, ㄛ	ㄕ	100%	ㄛ	100%
ㄕ, ㄜ	ㄕ	100%	ㄜ	100%
ㄕ, ㄞ	ㄕ	100%	ㄞ	100%
ㄕ, ㄟ	ㄕ	100%	ㄟ	100%
ㄕ, ㄠ	ㄕ	100%	ㄠ	100%

ㄗ, ㄢ	ㄗ	100%	ㄢ	100%
ㄗ, ㄤ	ㄗ	100%	ㄤ	100%
ㄗ, ㄣ	ㄗ	100%	ㄣ	100%
ㄗ, ㄥ	ㄗ	100%	ㄥ	100%
ㄗ, ㄧ	ㄗ	100%	ㄧ	100%
ㄗ, ㄨ	ㄗ	100%	ㄨ	100%
ㄖ, ㄚ	ㄖ	100%	ㄚ	100%
ㄖ, ㄛ	ㄖ	100%	ㄛ	100%
ㄖ, ㄜ	ㄖ	100%	ㄜ	100%
ㄖ, ㄞ	ㄖ	100%	ㄞ	100%
ㄖ, ㄟ	ㄖ	100%	ㄟ	100%
ㄖ, ㄠ	ㄖ	100%	ㄠ	95%
ㄖ, ㄢ	ㄖ	100%	ㄢ	100%
ㄖ, ㄤ	ㄖ	100%	ㄤ	100%
ㄖ, ㄣ	ㄖ	100%	ㄣ	100%
ㄖ, ㄥ	ㄖ	100%	ㄥ	95%
ㄖ, ㄧ	ㄖ	100%	ㄧ	100%
ㄖ, ㄨ	ㄖ	100%	ㄨ	100%
ㄚ, ㄛ	ㄚ	100%	ㄛ	100%
ㄚ, ㄜ	ㄚ	100%	ㄜ	100%
ㄚ, ㄞ	ㄚ	100%	ㄞ	100%
ㄚ, ㄟ	ㄚ	100%	ㄟ	100%
ㄚ, ㄠ	ㄚ	100%	ㄠ	100%
ㄚ, ㄢ	ㄚ	100%	ㄢ	100%
ㄚ, ㄤ	ㄚ	100%	ㄤ	100%
ㄚ, ㄣ	ㄚ	100%	ㄣ	100%
ㄚ, ㄥ	ㄚ	100%	ㄥ	100%
ㄚ, ㄧ	ㄚ	100%	ㄧ	100%

ㄚ, ㄨ	ㄚ	100%	ㄨ	100%
ㄛ, ㄜ	ㄛ	100%	ㄜ	100%
ㄛ, ㄞ	ㄛ	100%	ㄞ	100%
ㄛ, ㄠ	ㄛ	100%	ㄠ	100%
ㄛ, ㄢ	ㄛ	95%	ㄢ	100%
ㄛ, ㄤ	ㄛ	100%	ㄤ	100%
ㄛ, ㄣ	ㄛ	100%	ㄣ	100%
ㄛ, ㄥ	ㄛ	100%	ㄥ	100%
ㄛ, ㄧ	ㄛ	100%	ㄧ	100%
ㄛ, ㄨ	ㄛ	100%	ㄨ	100%
ㄜ, ㄞ	ㄜ	100%	ㄞ	100%
ㄜ, ㄟ	ㄜ	100%	ㄟ	100%
ㄜ, ㄠ	ㄜ	100%	ㄠ	100%
ㄜ, ㄢ	ㄜ	100%	ㄢ	100%
ㄜ, ㄤ	ㄜ	100%	ㄤ	100%
ㄜ, ㄣ	ㄜ	100%	ㄣ	100%
ㄜ, ㄥ	ㄜ	100%	ㄥ	100%
ㄜ, ㄧ	ㄜ	100%	ㄧ	100%
ㄜ, ㄨ	ㄜ	100%	ㄨ	100%
ㄞ, ㄟ	ㄞ	100%	ㄟ	100%
ㄞ, ㄠ	ㄞ	100%	ㄠ	100%
ㄞ, ㄢ	ㄞ	100%	ㄢ	100%
ㄞ, ㄤ	ㄞ	100%	ㄤ	100%
ㄞ, ㄣ	ㄞ	100%	ㄣ	100%
ㄞ, ㄥ	ㄞ	100%	ㄥ	100%
ㄞ, ㄧ	ㄞ	100%	ㄧ	100%
ㄞ, ㄨ	ㄞ	100%	ㄨ	100%

ㄟ, ㄠ	ㄟ	100%	ㄠ	100%
ㄟ, ㄢ	ㄟ	100%	ㄢ	100%
ㄟ, ㄤ	ㄟ	100%	ㄤ	100%
ㄟ, ㄣ	ㄟ	100%	ㄣ	100%
ㄟ, ㄥ	ㄟ	100%	ㄥ	100%
ㄟ, ㄧ	ㄟ	100%	ㄧ	100%
ㄟ, ㄨ	ㄟ	100%	ㄨ	100%
ㄠ, ㄢ	ㄠ	100%	ㄢ	100%
ㄠ, ㄣ	ㄠ	100%	ㄣ	100%
ㄠ, ㄥ	ㄠ	100%	ㄥ	100%
ㄠ, ㄧ	ㄠ	100%	ㄧ	100%
ㄠ, ㄨ	ㄠ	100%	ㄨ	100%
ㄢ, ㄤ	ㄢ	100%	ㄤ	100%
ㄢ, ㄣ	ㄢ	100%	ㄣ	100%
ㄢ, ㄥ	ㄢ	100%	ㄥ	100%
ㄢ, ㄧ	ㄢ	100%	ㄧ	100%
ㄢ, ㄨ	ㄢ	100%	ㄨ	100%
ㄤ, ㄣ	ㄤ	100%	ㄣ	100%
ㄤ, ㄥ	ㄤ	100%	ㄥ	100%
ㄤ, ㄧ	ㄤ	100%	ㄧ	100%
ㄤ, ㄨ	ㄤ	100%	ㄨ	100%
ㄣ, ㄥ	ㄣ	100%	ㄥ	100%
ㄣ, ㄧ	ㄣ	100%	ㄧ	100%
ㄣ, ㄨ	ㄣ	100%	ㄨ	100%
ㄅ, ㄓ	ㄅ	100%	ㄓ	100%
ㄆ, ㄓ	ㄆ	100%	ㄓ	100%
ㄉ, ㄍ	ㄉ	100%	ㄍ	100%
ㄉ, ㄎ	ㄉ	100%	ㄎ	100%

ㄍ, ㄎ	ㄍ	100%	ㄎ	100%
ㄏ, ㄐ	ㄏ	100%	ㄐ	100%
ㄥ, ㄧ	ㄥ	100%	ㄧ	95%
ㄥ, ㄨ	ㄥ	95%	ㄨ	100%
ㄧ, ㄨ	ㄧ	95%	ㄨ	100%
ㄐ, ㄒ	ㄐ	50%	ㄒ	50%
ㄐ, ㄓ	ㄐ	50%	ㄓ	50%
ㄒ, ㄓ	ㄒ	50%	ㄓ	50%
ㄛ, ㄟ	ㄛ	50%	ㄟ	50%
ㄛ, ㄡ	ㄛ	50%	ㄡ	50%
ㄟ, ㄡ	ㄟ	50%	ㄡ	50%
ㄑ, ㄕ	ㄑ	50%	ㄕ	50%
ㄎ, ㄏ	ㄎ	50%	ㄏ	50%
ㄖ, ㄗ	ㄖ	50%	ㄗ	0%
ㄖ, ㄘ	ㄖ	50%	ㄘ	50%
ㄖ, ㄙ	ㄖ	50%	ㄙ	50%
ㄗ, ㄘ	ㄗ	0%	ㄘ	50%
ㄗ, ㄙ	ㄗ	0%	ㄙ	50%
ㄘ, ㄙ	ㄘ	50%	ㄙ	50%
ㄜ, ㄝ	ㄜ	50%	ㄝ	50%
ㄠ, ㄤ	ㄠ	50%	ㄤ	50%
ㄧ, ㄩ	ㄧ	50%	ㄩ	50%
ㄣ, ㄦ	ㄣ	50%	ㄦ	50%

APPENDIX 11

CDC Cyber 173 Computer Program- VBREC

Any single valued function f(x) that has a finite number of discontinuities may be represented by a Fourier series in the interval of $(-\pi,\pi)$, or in the interval of (-1,1). For a function f(x) defined in the interval of (-1,1), the trigonometric form of a Fourier series is defined as

$$f(x)=a_0/2 + \sum_{n=1}^{\infty} a_n \mathrm{Cos}(n\pi x/l) + \sum_{n=1}^{\infty} b_n \mathrm{Sin}(n\pi x/l)$$

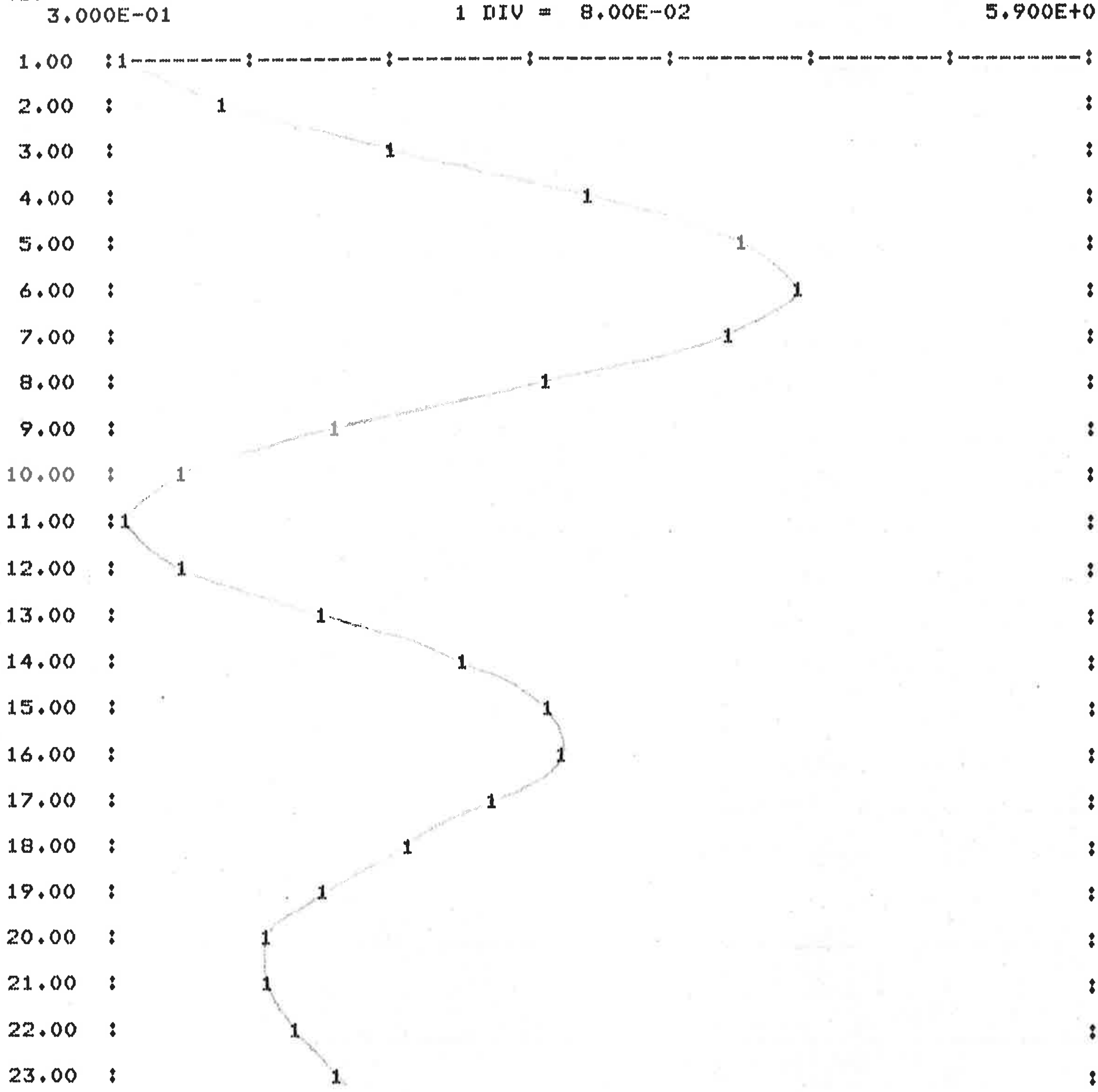
where a_n and b_n are the Fourier coefficients; the period is 2l, which corresponds to 2π and that f(x+2l)=f(x).

In this example first 8 terms of the function of Fourier series are used in the best fit of experimentally measured values of $(x_1,y_1),(x_2,y_2),\ldots\ldots,$and (x_{30},y_{30}). We adopt the number 0.65 as the value of half period l. Finally I make use of Eq.(56) in section D of II.2 and Eq.(19) in Appendix 12 to obtain first 8 coefficients (i.e. $a_0/2, a_1, a_2, a_3, b_1, b_2, b_3$ and b_4) of the required function of Fourier series. Those 8 coefficients are considered as the features applied in the method of parameter classification.

```
      PROGRAM VBREC(INPUT,OUTPUT)
C   LEAST SQURES CURVE FITTING APPLIED TO SPEECH RECOGNITION
      DIMENSION X(30),Y(30),F(30,16),FT(16,30),A(16,17),B(16),C
     1(16)
C   READ IN THE NUMBER OF C
      DATA(M=8)
C   READ IN THE NUMBER OF DATA POINTS
      DATA(N=30)
C   READ X VALUES OF DATA POINTS
      DATA(X=0,.04483,.08966,.13449,.17931,.22414,.26898,.31381,.3586
     12,.40345,.44828,.49311,.53795,.58278,.62762,.67245,.71724,
     1.76207,.8069,.85173,.89655,.94138,.98622,1.03105,1.0759,1.
     112073,1.16556,1.21039,1.25523,1.30006)
C   READ Y VALUES OF DATA POINTS
      DATA(Y=.3,.5,1.25,4.1,4.52,4.05,3.15,2.4,1.8,1.3,.38,1,1.85,2.5
     1,4.18,4,3.45,2.5,1.35,1.32,1.3,1.25,1.17,1.45,2.2,2.05,1.4
     1,.7,.37,.4)
C   DEFINE THE  FUNCTIONS
      F1(X)=1.
      F2(X)=COS(3.1415926536*X/.65)
      F3(X)=COS(2.*3.1415926536*X/.65)
      F4(X)=COS(3.*3.1415926536*X/.65)
      F5(X)=SIN(3.1415926536*X/.65)
      F6(X)=SIN(2.*3.1415926536*X/.65)
      F7(X)=SIN(3.*3.1415926536*X/.65)
      F8(X)=SIN(4.*3.1415926536*X/.65)
C   GENERATE THE F MATRIX
      DO 4 I=1,N
      F(I,1)=F1(X(I))
      F(I,2)=F2(X(I))
      F(I,3)=F3(X(I))
      F(I,4)=F4(X(I))
      F(I,5)=F5(X(I))
      F(I,6)=F6(X(I))
      F(I,7)=F7(X(I))
      F(I,8)=F8(X(I))
4     CONTINUE
C   GENERATE THE TRANSPOSE OF THE F MATRIX
      DO 5 I=1,N
      DO 5 J=1,M
      FT(J,I)=F(I,J)
5     CONTINUE
C   DETRIME COEFFICIENT MATRIX A OF SIMULTANEOUS EQUATION SYSTEM
      CALL  MATMPY(FT,F,A,M,N,M)
C   DETERMINE THE COLUMN OF CONSTANTS FOR SIMULTANEOUS EQUATION S
      CALL MATMPY(FT,Y,B,M,N,1)
      DO 6 I=1,M
      A(I,M+1)=B(I)
6     CONTINUE
C   DETERMINE C VALUES BY SOLVING SIMULTANEOUS EQUATIONS USING
C     CHOLESKY METHOD
      MP1=M+1
      CALL CHLSKY(A,M,MP1,C)
C   WRITE OUT THE C VALUES
      PRINT 7
7     FORMAT(4X,* C(1) THROUGH C(M) *)
      DO 11 I=1,M
      PRINT 12,I,C(I)
12    FORMAT(1X,* C(*,I1,*) = *,E14.7)
11    CONTINUE
```

```
C  PLOT THE SPEECH WAVEFORM Y(X)
       CALL PLTYHT(70)
       DO 14 I=1,N
       Y(I)=C(1)*F(I,1)+C(2)*F(I,2)+C(3)*F(I,3)+C(4)*F(I,4)+C(5)*F(I,5
      1)+C(6)*F(I,6)+C(7)*F(I,7)+C(8)*F(I,8)
14     CONTINUE
       CALL PLT(N,1,Y)
       STOP
       END
       SUBROUTINE MATMPY(A,B,C,M,N,L)
C  DETERMINES MATRIX C AS PRODUCT OF A AND B MATRICES
       DIMENSION A(16,30),B(30,16),C(16,17)
       DO 2 I=1,M
       DO 2 J=1,L
       C(I,J)=0.
       DO 2 K=1,N
       C(I,J)=C(I,J)+A(I,K)*B(K,J)
2      CONTINUE
       RETURN
       END
       SUBROUTINE CHLSKY(A,N,M,X)
       DIMENSION A(16,17),X(16)
C  CALCULATE THE FIRST ROW OF UPPER UNIT TRIANGULAR MATRIX
       DO 3 J=2,M
       A(1,J)=A(1,J)/A(1,1)
3      CONTINUE
C  CALCULATE OTHER ELEMENTS OF U AND L MATRICES
       DO 8 I=2,N
       J=I
       DO 5 II=J,N
       SUM=0.
       JM1=J-1
       DO 4 K=1,JM1
       SUM=SUM+A(II,K)*A(K,J)
4      CONTINUE
       A(II,J)=A(II,J)-SUM
5      CONTINUE
       IP1=I+1
       DO 7 JJ=IP1,M
       SUM=0.
       IM1=I-1
       DO 6 K=1,IM1
       SUM=SUM+A(I,K)*A(K,JJ)
6      CONTINUE
       A(I,JJ)=(A(I,JJ)-SUM)/A(I,I)
7      CONTINUE
8      CONTINUE
C  SOLVE FOR X(I) BY BACK SUBSTITUTION
       X(N)=A(N,N+1)
       L=N-1
       DO 10 NN=1,L
       SUM=0.
       I=N-NN
       IP1=I+1
       DO 9 J=IP1,N
       SUM=SUM+A(I,J)*X(J)
9      CONTINUE
       X(I)=A(I,M)-SUM
10     CONTINUE
       RETURN
       END
..run,ftn
```

```
 45000B   CM STORAGE USED
     .599 CP SECONDS COMPILATION TIME
  C(1) THROUGH C(M)
C(1) =    .1820659E+01
C(2) =   -.6886829E-01
C(3) =   -.1866830E+00
C(4) =   -.1173550E+01
C(5) =    .4776330E+00
C(6) =    .7137102E+00
C(7) =   -.8870041E-03
C(8) =   -.1870204E+00
```



```
 CM LWA+1 = 17706B, LOADER USED  33200B
24.00  :                  1                                                  :
25.00  :                   1                                                 :
26.00  :                1                                                    :
27.00  :           1                                                         :
28.00  :     1                                                               :
29.00  :1                                                                    :
30.00  :1--------:---------:---------:---------:---------:---------:---------:
    STOP
      .188 CP SECONDS EXECUTION TIME
```

APPENDIX 12

CHOLESKY'S METHOD (26)*

by

M.L. James, G.M. Smith, and J.C. Wolford

Cholesky's method was used in the solution of curve fitting mentioned in section D of Chapter II.2.

Cholesky's method, also known as Crout's method, a method of matrix decomposition, and factorization, is more economical of computer time than other elimination methods.

For illustrative purposes, let us discuss the solution of three simultaneous equations in three unknowns. The set is represented by the matrix equation

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \end{Bmatrix} \qquad (1)$$

If we let $\underset{\sim}{A}$ represent the coefficient matrix, $\{X\}$ the column matrix of the unknowns, and $\{C\}$ the column matrix of the constants, we can replace Eq. (1) by

$$\underset{\sim}{A}\{X\} - \{C\} = 0 \qquad (2)$$

If we could reduce the system of equations to an equivalent system of the form

$$\begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} \qquad (3)$$

they could readily be solved by back substitution as in Gaussian elimination. The upper unit triangular matrix in (3) will be represented by $\underset{\sim}{u}$. Then Eq.(3) can be written as $\underset{\sim}{u}\{X\} - \{D\} = 0 \qquad (4)$

Suppose there exists a lower triangular matrix

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix}$$

which we represent by $\underset{\sim}{L}$ and which has the following property :When we premultiply the left side of Eq(4) by $\underset{\sim}{L}$, it will give us the left side of Eq.(2). The existence of such a matrix $\underset{\sim}{L}$ hasn't been proved at this point, but in the discussion which follows we demonstrate its existence whenever matrix $\underset{\sim}{A}$ is nonsingular. In accordance with the above property of $\underset{\sim}{L}$, we have

$$\underset{\sim}{L}\ (\underset{\sim}{u}\{X\} - \{D\}) = \underset{\sim}{A}\{X\} - \{C\} \qquad (5)$$

Therefore, $\underset{\sim}{L}\,\underset{\sim}{u} = \underset{\sim}{A} \quad (6)$ and $\underset{\sim}{L}\{D\} = \{C\} \quad (7)$

Equations (6) and (7) may be combined into the single matrix equation

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \left[\begin{array}{ccc|c} 1 & u_{12} & u_{13} & d_1 \\ 0 & 1 & u_{23} & d_2 \\ 0 & 0 & 1 & d_3 \end{array}\right] = \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & C_1 \\ a_{21} & a_{22} & a_{23} & C_2 \\ a_{31} & a_{32} & a_{33} & C_3 \end{array}\right] \qquad (8)$$

For the sake of convenience, let the C's be represented by a_{i4} and the d's by u_{i4} to obtain Eq.(8) as

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \qquad (9)$$

The l_{ij} and u_{ij} elements may be determined from Eq.(9). The augmented $\underset{\sim}{u}$ matrix of Eq.(9) is the augmented matrix of the equivalent triangular set given by Eq.(3). There-fore, with the u_{ij} determined, the unknown x_i values may be determined by back substitution.

From the rules of matrix multiplication, we note that the first column of matrix $\underset{\sim}{L}$ is identical with the first column of the augmented $\underset{\sim}{A}$ matrix. That is,

$$l_{i1} = a_{i1} \qquad (10)$$

We also note that $l_{11} u_{12} = a_{12}$, $l_{11} u_{13} = a_{13}$, $l_{11} u_{14} = a_{14}$ Therefore, we can obtain the first row of the u matrix as the first row of $\underset{\sim}{A}$ divided by l_{11} (or a_{11}). That is,

$$u_{1j} = a_{1j}/l_{11} \qquad (11)$$

We have determined the first column of $\underset{\sim}{L}$ and the first row of $\underset{\sim}{u}$. We can now proceed to determine the second column of $\underset{\sim}{L}$, followed by the second row of $\underset{\sim}{u}$.

From matrix multiplication, we have

$$l_{21} u_{12} + l_{22} \cdot 1 = a_{22}$$

$$l_{31} u_{12} + l_{32} \cdot 1 = a_{32}$$

from which we obtain

$$\begin{aligned} l_{22} &= a_{22} - l_{21} \cdot u_{12} \\ l_{32} &= a_{32} - l_{31} \cdot u_{12} \end{aligned} \qquad (12)$$

Having the second column of $\underset{\sim}{L}$, we get the second row of $\underset{\sim}{u}$ from

$$l_{21} \cdot u_{13} + l_{22} u_{23} = a_{23}$$

$$l_{21} \cdot u_{14} + l_{22} u_{24} = a_{24}$$

or

$$\begin{aligned} u_{23} &= (a_{23} - l_{21} u_{13})/l_{22} \\ u_{24} &= (a_{24} - l_{21} u_{14})/l_{22} \end{aligned} \qquad (13)$$

Next we get the third column of $\underset{\sim}{L}$ followed by the third row of $\underset{\sim}{u}$. The third column of $\underset{\sim}{L}$ is obtained from

$$l_{31}u_{13} + l_{32}u_{23} + l_{33} = a_{33}$$

or

$$l_{33} = a_{33} - (l_{31}u_{13} + l_{32}u_{23}) \qquad (14)$$

The element u_{34}, the only element in the third row of $\underset{\sim}{u}$ except the 1, is obtained from

$$l_{31}u_{14} + l_{32}u_{24} + l_{33}u_{34} = a_{34}$$

or

$$u_{34} = \{a_{34} - (l_{31}u_{14} + l_{32}u_{24})\}/l_{33} \qquad (15)$$

The general equations for the elements of the $\underset{\sim}{L}$ and augmented $\underset{\sim}{u}$ matrices, for n equations in unknowns, are

$$\begin{cases} l_{i1} = a_{i1} & \text{for} \begin{cases} i = 1,2,\dots,n \\ j = 1 \end{cases} \\ u_{1j} = a_{1j}/a_{11} & \text{for} \begin{cases} j = 1,2,\dots,n+1 \\ i = 1 \end{cases} \\ l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj} & \text{for} \begin{cases} j = 2,3,\dots,n \\ i = j,j+1,\dots,n \\ \text{(for each value of } j) \end{cases} \\ u_{ij} = (a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj})/l_{ii} & \text{for} \begin{cases} i = 2,3,\dots,n \\ j = i+1,i+2,\dots,n+1 \\ \text{(for each value of } i) \end{cases} \end{cases} \qquad (16)$$

The back-substitution formulas are

$$\begin{cases} x_n = u_{n,n+1} \\ x_i = u_{i,n+1} - \sum_{j=i+1}^{n} u_{ij}x_j & \text{for } i = n-1, n-2, \dots, 1 \end{cases} \qquad (17)$$

If we make sure that a_{11} in the original coefficient matrix is nonzero, then the divisions of Eq.(16) will

always be defined since the l_{ii} values will be nonzero. This may be seen by noting that $\underset{\sim}{L}\,\underset{\sim}{u} = \underset{\sim}{A}$ and therefore the determinant of $\underset{\sim}{L}$ times the determinant of $\underset{\sim}{u}$ equals the determinant of $\underset{\sim}{A}$. That is, $|\underset{\sim}{L}|\,|\underset{\sim}{u}| = |\underset{\sim}{A}|$ (18)
We are assuming independent equations, so the determinant of $\underset{\sim}{A}$ is nonzero. Therefore, the determinant of $\underset{\sim}{L}$ must be nonzero. Since the determinant of a triangular matrix is the product of the main diagonal elements, the l_{ii} elements are all nonzero.

The original a_{ij} values of the augmented $\underset{\sim}{A}$ matrix are first stored in the computer. Then, calling both the u_{ij} and l_{ij} elements simply a_{ij}, the old a_{ij} values are replaced with new a_{ij} values. Then the first of Eq.(16) is automatically satisfied with no-computer execution. The other equations become

$$a_{1j} = a_{1j}/a_{11} \qquad \text{for } j=2,3,\ldots\ldots,n+1$$

$$a_{ij} = a_{ij} - \sum_{k=1}^{j-1} a_{ik}a_{kj} \qquad \text{for } \begin{cases} j=2,3,\ldots\ldots,n \\ i=j,j+1,\ldots,n \\ \quad (\text{for each } j \text{ value}) \end{cases}$$

(19)

$$a_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} a_{ik}a_{kj}\right)/a_{ii} \qquad \text{for } \begin{cases} i=2,3,\ldots\ldots,n \\ j=i+1,i+2,\ldots\ldots,n+1 \\ \quad (\text{for each } i \text{ value}) \end{cases}$$