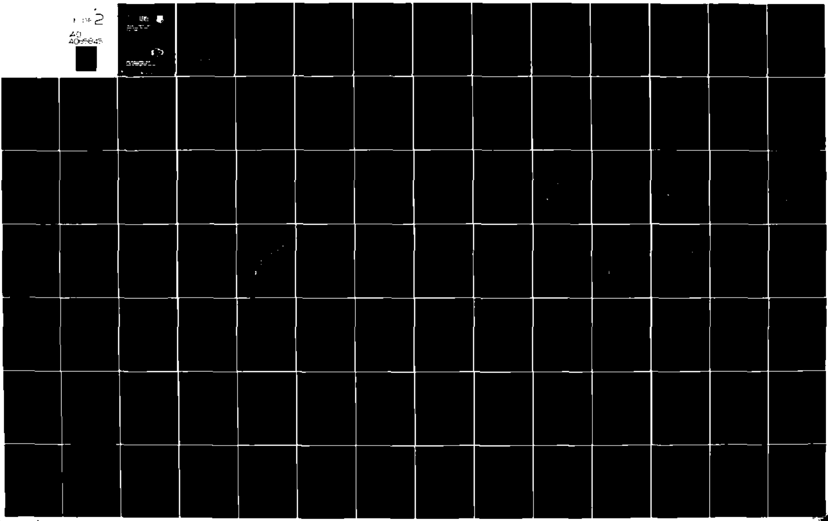
AD-A085 845 ITT DEFENSE COMMUNICATIONS DIV SAN DIEGO CA F/G 9/4
SPEAKER AUTHENTICATION OPERATIONAL TEST AND EVALUATION.(U)
APR 80 E H WRENCH F30602-78-C-0324
UNCLASSIFIED RADC-TR-80-64 NL

1 OF 2
AD
A085845



ADA085845

RADC-TR-80-64
Final Technical Report
April 1980

LEVEL II

12

SPEAKER AUTHENTICATION OPERATIONAL TEST AND EVALUATION

ITT Defense Communications Division

Dr. Edwin H. Wrench, Jr.

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC
ELECTE
JUN 30 1980
S D

[illegible]

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-80-64 has been reviewed and is approved for publication.

APPROVED:

JEFFREY P. WOODARD, 1/Lt, USAF
Project Engineer

APPROVED:

OWEN R. LAWTER, Colonel, USAF
Chief, Intelligence & Reconnaissance Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

[illegible]

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. ~~REPORT NUMBER~~ (18) RADC-TR-80-64 / 2. GOVT ACCESSION NO. AD-A085845	3. RECIPIENT'S CATALOG NUMBER
4. ~~TITLE~~ (and Subtitle) (6) SPEAKER AUTHENTICATION OPERATIONAL TEST AND EVALUATION (9)	5. ~~TYPE OF REPORT & PERIOD COVERED~~ Final Technical Report, 22 Sep 78 — 6 Nov 79
	6. ~~PERFORMING ORG. REPORT NUMBER~~ N/A
7. AUTHOR(s) Dr. Edwin H. Wrench, Jr. (10)	8. ~~CONTRACT OR GRANT NUMBER(s)~~ (15) F30602-78-C-0324
9. ~~PERFORMING ORGANIZATION NAME AND ADDRESS~~ ITT Defense Communications Division 9999 Business Park Avenue San Diego CA 92131 (16)	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 31011GF 70550730 (17)
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRAA) Griffiss AFB NY 13441 (11)	12. ~~REPORT DATE~~ Apr[redacted]80
	13. ~~NUMBER OF PAGES~~ 157
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same (12) 157	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Jeffrey P. Woodard, 1/Lt, USAF (IRAA)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Speaker Authentication	Speech Processing
Markel's Technique	Speaker Identification
Pfeifer's Technique	Linear Predictive Coding

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The long term goal of this program is to implement a realtime speaker recognition system that can be operated by untrained personnel. The objective of the present contract was to demonstrate the feasibility of using speaker recognition techniques to aid in the identification of unknown speakers. The contract was conducted in two phases. The first was the algorithm selection phase, in which various speaker recognition techniques were implemented and tested to determine (Cont'd)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

411-0-

Item 20 (Cont'd)

the optimal technique to satisfy the specific program requirements. The second phase of the program was to implement the selected technique in a laboratory demonstration system, and to develop a convenient, easy to use operator interface to the system.

The most difficult task was to develop a text independent speaker recognition technique that would achieve a high level of recognition accuracy with only 10 seconds of training data for each talker. Previous text independent speaker recognition techniques had been reported that had achieved the required accuracies, but all had used several minutes of traning data to generate the speaker reference models. In addition, recognition was accomplished in most of these studies by using at least one minute of unknown speech. A serious question that needed to be investigated was whether or not the required accuracy could be maintaining when the training data was reduced to 10 seconds, and the unknown speech was limited to less than one minute.

All requirements for the contract were met or exceeded. An algorithm has been developed and tested using an ITTDCD speaker recognition test bed capability. The resulting algorithm achieves excellent recognition rates for all speakers when used with limited amounts of speech for both the reference models and unknowns. In addition, this algorithm has been implemented in a realtime speaker recognition demonstration system and achieves similar high recognition scores. The realtime demonstration system has proven to be easy to operate with little or no instruction. An operator can generate a model using "live" speech, document the model with pertinent speaker data, and use the model for realtime speaker recognition, all within less than a minute.

EVALUATION

This effort resulted in a successful laboratory demonstration of a speaker identification system. The speaker identification system performed with greater than 90% accuracy for a group of 30 male, American talkers. These encouraging results were obtained although as little as 10 seconds of both unknown and reference speech data were used. The identification was performed automatically, on-line, and in real-time, with band-limited, text-independent speech. At signal-to-noise ratios of 15db, identification accuracies were reduced by only about 10%.

The speaker identification system was implemented with an interactive operator interface so that untrained personnel could use the system with a minimum of instruction.

The results of this effort indicate that this type of speaker identification technology should be further developed so that it can be integrated in future operational systems with other automated speech technologies, such as keyword and language identification. These future systems will provide an operator with a real-time automated capability for the processing and analysis of speech signals.

Jeffrey P. Woodard

JEFFREY P. WOODARD
Project Engineer

Accession For
NTIS GRA&I ☑
DDC TAB □
Unannounced □
Justification______
By______
Distribution/
Availability Codes

Dist	Avail and/or special
A	

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1: INTRODUCTION AND SUMMARY

1.1 INTRODUCTION

This is the final report for RADC's Speaker Authentication OT & E contract conducted by ITT Defense Communications Division. The long term goal of the program is to implement a realtime speaker recognition system that can be operated by untrained personnel. The objective of the present contract was to demonstrate the feasibility of using speaker recognition techniques to aid in the identification of unknown speakers. The contract was conducted in two phases. The first was the algorithm selection phase, in which various speaker recognition techniques were implemented and tested to determine the optimal technique to satisfy the specific program requirements. The second phase of the program was to implement the selected technique in a laboratory demonstration system, and to develop a convenient, easy to use operator interface to the system.

This report is organized in four chapters. The first chapter is a summary of the present speaker authentication effort and contains all the major results and conclusions. The remaining three chapters discuss these efforts in detail. Chapter 2 describes the algorithm selection phase, Chapter 3 discusses the implementation of the realtime speaker recognition system, and Chapter 4 contains the conclusions of this effort and recommendations. The appendix contains experimental results.

1.2 PROGRAM OVERVIEW

This work in speaker recognition was conducted at the ITT Defense Communications Division (ITTDCD) facility in San Diego, California under contract to the Air Force's Rome Air Development Center. The objective of the contract was to demonstrate the feasibility of using speaker recognition techniques to aid in the identification of unknown speakers.

The reauirements for anv soeaker recognition system develooed under this contract include the followinq:

1. Recoqnize unknown talkers from a set of up to 30 known speakers in a text independent environment
2. Achieve 95% recognition on 90% of the speakers using 10 seconds of training soeech, and 5 to 60 seconds of unknown speech.
3. Generate and document speaker reference models.
4. Permit ooeration by untrained personnel with minimum instruction.

The most difficult task was to develop a text independent speaker recognition techniaue that would achieve 95% recoqnition accuracy with only 10 seconds of traininq data for each talker. Previous text independent speaker recognition techniaues had been reported that had achieved the reauired accuracies, but all had used several minutes of training data to generate the speaker reference models. In addition, recognition was accomplished in most of these studies by using at least one minute of unknown speech. A serious auestion that needed to be investigated was whether or not the reauired accuracy could be maintained when the training data was reduced to 10 seconds, and the unknown speech was limited to less than one minute.

The approach was to divide the effort into two parts. The first phase was to investiqate speaker recognition techniques to determine their performance under the conditions of limited amounts of input speech. The second phase of the program was to implement the best technique in a laboratory demonstration system and investigate some of the human factors considerations of the ooerator interface.

ITTDCD has had a long term qoal in the area of speaker recognition and verification, and therefore has developed a realtime capability for speaker recoqnition using a high speed, special purpose signal processor. During the second phase of this contract, an operator interface was developed and combined with the realtime speaker recognition capability. This operator interface was designed to be easy to use by untrained personnel.

1.3 ALGORITHM SELECTION STUDY

The first phase of the program was to identify a text independent, closed set, speaker recognition algorithm that would achieve high recognition accuracy with very limited reference data. Previous studies had reported high recognition rates, but all had used several minutes of speech for the references.

Two speaker recognition techniques were implemented and tested during the algorithm selection phase of the contract. The first technique was originally developed by Markel [1]. ITTDCD had tested Markel's technique under a previous government contract and achieved excellent results with ten minutes of reference data. The second technique was originally implemented by Pfeifer [2] under an RADC contract. This second technique was suggested by the sponsor as a candidate for implementation.

Markel's technique, uses ten linear prediction coder (LPC) reflection coefficients as speaker recognition features. The features are averaged over the entire recognition period, and the average feature vector is then compared with the stored talker models. The recognized talker is the one whose model is most similar to the unknown speech.

Pfeifer's technique also uses reflection coefficients as speaker recognition features. The difference between the two techniques is that Pfeifer's algorithm does not average the features before comparing them with the stored models, but rather makes a decision as to the talker identity for every speech frame. The final recognition decision is then made by determining which model compares best with the unknown for the majority of the frames during the recognition period.

Both speaker recognition techniques were implemented using the ITTDCD speaker recognition test bed. The test bed contains a realtime speaker recognition capability, but is flexible enough it permit the rapid implementation of new algorithms. It is implemented on two processors, the PDP-11/60 and a high speed signal processor developed by ITTDCD (the

Quintrell RAM). The Quintrell RAM is programed to execute the majority of the computationally intensive signal processing tasks required for speaker recognition. The PDP-11/60 portion of the test bed permits the use of high level languages for controlling the Quintrell processor, and provides disk storage facilities for the speaker data bases and results.

The two techniques evaluated under the contract were tested using a portion of the ARPA speaker recognition data base. This data base consists of 10 three minute interviews from 17 different talkers. The interviews with each talker were conducted at one week intervals for 10 weeks. The data used for these experiments came from the fifth and sixth interviews for each talker.

Experiments were conducted to investigate the performance of the two algorithms under a variety of conditions, all of which involved the use of limited amounts of input speech both for model generation and recognition. The majority of the testing used 10 or 20 seconds of speech for the reference models, and recognition used one to 40 seconds of speech.

The results for both Markel's and Pfeifer's techniques when used with 10 and 20 seconds of reference data are shown in figure. 1.1. The results indicate that Markel's algorithm performs better than Pfeifer's for applications where the durations of both the reference data and the model data are limited. The performance of Markel's technique was 96% when used with 20 second models and 40 second unknowns. The performance of Markel's algorithm was also evaluated when the signal to noise ratio of the input speech was reduced to 15 dB. The speaker recognition accuracy decreased by less than 10%.

Additional experiments were performed with the speaker recognition test bed to investigate recognition algorithms that combined the frame averaging of Markel's technique with the majority decision of Pfeifer's technique. The results indicate that the more averaging that is done before the recognition features are compared with the reference models, the better the system performance. Therefore Markel's technique was chosen as the algorithm to be implemented in the realtime speaker recognition system.

COMPARISON OF MARKEL AND PFEIFER'S TECHNIQUES

10 and 20 SECOND MODELS, INDIVIDUAL COVARIANCE MATRICES

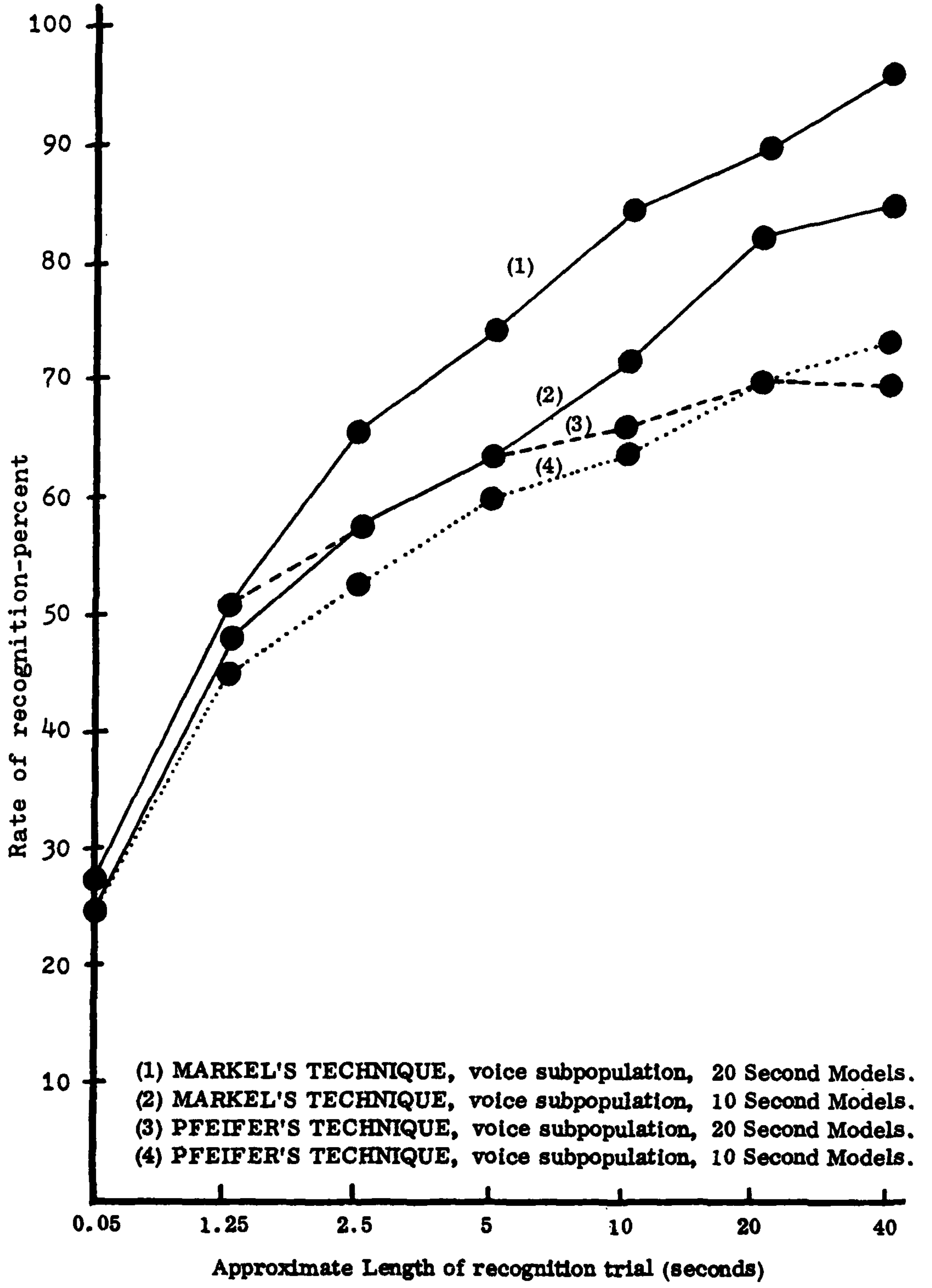


FIGURE 1.1

1.4 SPEAKER RECOGNITION LABORATORY DEMONSTRATION SYSTEM

Markel's algorithm was chosen for the laboratory realtime demonstration system based on the results obtained from the algorithm selection studies. The realtime speaker recognition capability of the test bed was integrated with an operator interface that provides a very flexible, easy to use system. The operator is prompted by the system at every phase of the operation with the options that are available at that time. The operator simply selects the desired operating mode (model generation, recognition, etc), and indicates the action to be taken via simple commands.

The system is capable both of generating models and of recognizing up to 30 talkers simultaneously and in realtime. The system has two display modes, one for talker similarity, and the other for recognition confidence.

Limited testing of the realtime speaker recognition system was conducted under the contract. A thirty talker data base was generated by recording speakers from commercial television. Speaker models were generated with 10 and 20 seconds of the recorded speech for each talker. Different 10 and 20 second segments from each talker were then used as unknowns.

The results are very encouraging. As table 1.1 indicates, the best recognition rate was 100% correct for 10 second models and 20 second unknowns. It is somewhat surprising that the 10 second models performed better than the 20 second models, however, it must be remembered that this was an extremely small test. Only one recognition trial was run for each speaker. Further testing is required to adequately estimate the system performance.

Table 1.1: REALTIME SPEAKER RECOGNITION RESULTS

	10 second models	20 second models
10 second unknowns	93% (28 correct out of 30)	90% (27 correct out of 30)
20 second unknowns	100% (30 correct out of 30)	97% (29 correct out of 30)

1.5 CONCLUSIONS

All requirements for the contract have been met or exceeded. An algorithm has been developed and tested using an ITTDCD speaker recognition test bed capability. The resulting algorithm achieves recognition rates in excess of 90% for all speakers when used with limited amounts of speech for both the reference models and the unknowns. In addition, this algorithm has been implemented in a realtime speaker recognition demonstration system and achieves similar high recognition scores. The realtime demonstration system has proven to be easy to operate with little or no instruction. An operator can generate a model using "live" speech, document the model with pertinent speaker data, and use the model for realtime speaker recognition, all within less than a minute. The findings of the study are summarized in the following paragraphs.

1. Markel's technique was shown to be well-suited for speaker recognition systems operating with limited amounts of speech for both model generation and reccgnition. In addition, the study proved that Markel's technique performs better than the Pfeifer technique under these conditions.

2. Recognition accuracies close to 95% were obtained with Markel's

technique using a 17 talker data base. The models were generated with 20 seconds and 40 seconds of unknown speech.

3. Recognition accuracies for noisy speech (approximately 15 Db signal to noise ratio) were shown to decrease less than 10% when compared with the original noise free results for the same input speech. These tests also used Markel's technique with 17 talkers.

4. A realtime laboratory speaker recognition capability was integrated with a convenient, easy to use operator interface to produce a realtime speaker recognition demonstration system. The realtime processing is done in an ITTDCD developed, high speed signal processing unit (the Quintrell RAM). The operator interface is implemented in a PDP 11/60.

5. The realtime capability of the speaker recognition demonstration system was tested by generating models and performing recognition in realtime on a thirty speaker data base. Limited testing of the system showed recognition accuracies of 97% when using 20 second models and 20 second unknown speech samples. An accuracy of 100% was demonstrated when using 10 second models and 20 second unknowns (see Table 1.1).

Chapter 2: ALGORITHM SELECTION

2.1 INTRODUCTION

The first step in the development of the speaker authentication system was to select a speaker recognition technique and to refine the algorithm to meet the goals of the program. Several recognition techniques have been described in the open literature, but none of them had been tested under the conditions of interest in this program. In particular, the contract requirement to generate models using as little as 10 seconds of speech had not been addressed previously. Because of the uncertainty in the performance of these algorithms, the program was divided into two phases, an algorithm selection phase, and a system development phase. The algorithm selection study is described in this chapter.

Two speaker recognition techniques were implemented and tested during the algorithm selection phase of the contract. The first technique was originally developed by Markel[1]. ITTDCD had tested Markel's technique under a previous government contract and achieved excellent results. The second technique was originally implemented by Pfeifer[2] under an RADC contract. This second technique was suggested by RADC as a candidate for implementation.

2.2 SPEAKER RECOGNITION ALGORITHMS

The class of speaker recognition techniques applicable to this problem are referred to as text independent, closed set recognizers. These techniques are designed to choose, from a set of known talkers, the candidate whose speech most closely matches the unknown speech segment. Text independent recognition implies there is no constraint on the content of the unknown speech to be analyzed nor on the speech segment used for the model.

A speaker recognition technique must operate in two modes, a model generation mode, and a recognition mode. Before speakers can be identified, models must be generated that characterize each talkers voice, Recognition is then performed by comparing these previously generated models with an unknown utterance, and making a decision as to the identity of the speaker.

The majority of text independent closed set speaker recognition techniques can be modeled as shown in figure 2.1. For a generalized speaker recognition system such as the one shown, the following steps are performed.

1. The input speech is digitized.

2. A parametric representation for the speech is generated. Parameters in general use include spectral coefficients, cepstral coefficients, and linear predictive coding (LPC) reflection coefficients. The amount of time represented by the speech segment required to generate one complete set of speech parameters is refered to as the frame period, and the set of parameters generated during that time is refered to as a frame.

3. The parameters or frames are then passed to a subpopulation filter that retains only those frames that have the particular attributes selected as important for distinguishing talkers. Subpopulations that have been been used in previous investigations include; all speech, all voiced speech, all vowels, all transitional speech, and all nasals.

4. The next step in a generalized recognition system is to further process the frames contained in the selected subpopulation to generate speaker recognition features. Features are generated by averaging frames, performing a transformation of the original speech parameters using principle component analysis, or combinations of both averaging and transformation.

GENERALIZED SPEAKER AUTHENTICATION SYSTEM

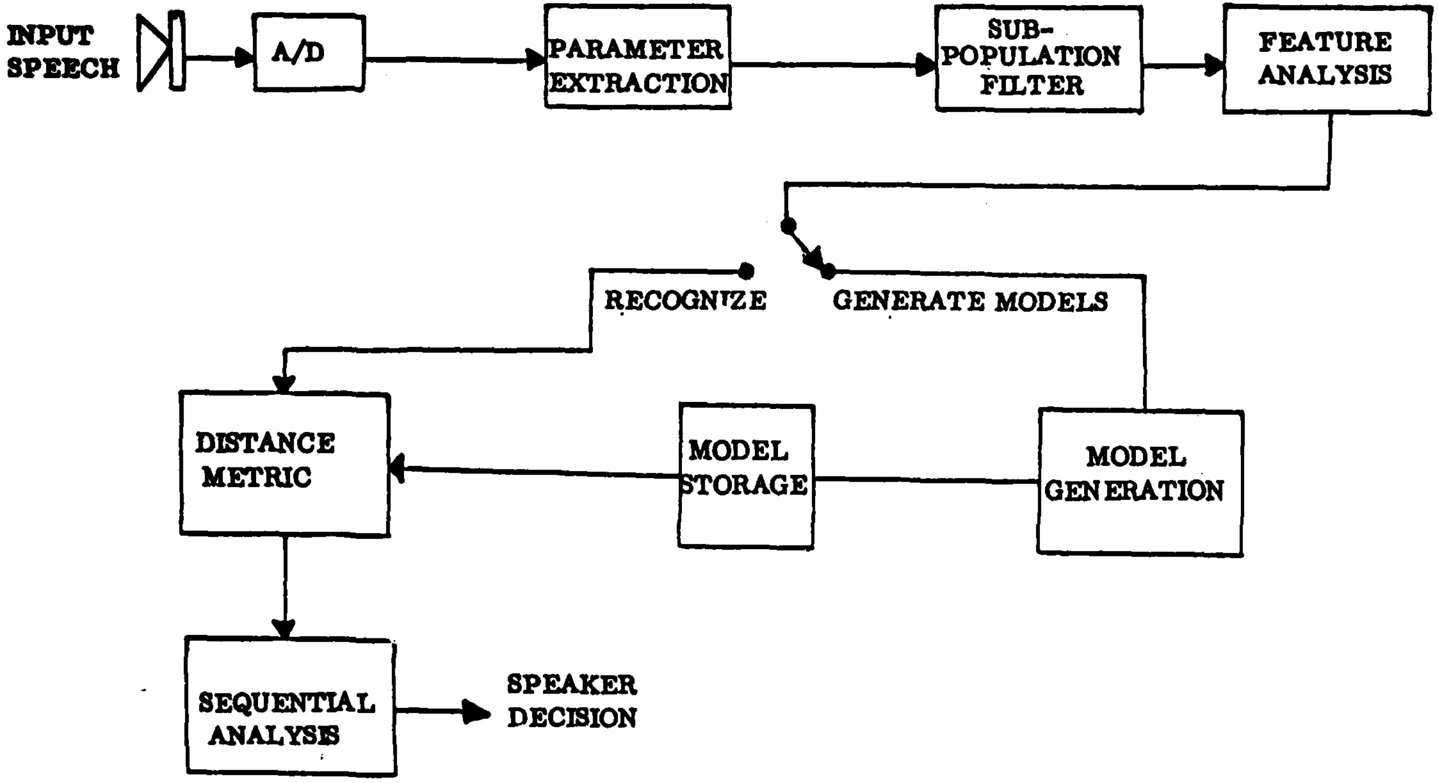


FIGURE 2.1

5. In the model generation mode, the speaker recognition features are processed to extract statistical parameters that characterize the talker. These statistical parameters, such as mean feature vectors and covariance matrices for the features, are stored as talker models.

6. In the recognition mode, a distance metric is used to determine the similarity between the unknown feature vectors, and the stored speaker models. Distance metrics that have been investigated previously include the Euclidean, the weighted Euclidean, and the Mahlanobis metric.

7. In the recognition mode, the similarity scores determined by the distance metric are then analyzed by a sequential analysis process that examines the time sequence of similarity measures, and makes a determination as to the speaker identity.

2.3 ITTDCD SPEAKER RECOGNITION TEST BED

Both speaker recognition techniques were implemented and evaluated using a speaker recognition test bed developed at ITTDCD under IR&D funding. The speaker recognition test bed incorporates a number of processing capabilities that permit the easy implementation of speaker recognition algorithms. The test bed was also developed to permit the rapid processing of the extremely large data bases that are required for the evaluation of algorithm performance.

The speaker recognition test bed is resident on two processors, the PDP-11/60, and the Quintrell high speed signal processor. The PDP-11/60 is used as a controller for the processing, and for mass storage of the data base and the processed data. The Quintrell is used to implement the various high speed signal processing routines required for speaker recognition algorithms. Table 2.1 contains a list of the signal processing routines available in the test bed.

Table 2.1. SIGNAL PROCESSING ROUTINES IMPLEMENTED IN THE SPEAKER RECOGNITION TEST BED

LPC-10 analysis with pitch, using live data from the A/D
LPC-10 analysis with pitch, using stored digital data
LPC-10 synthesis from stored reflection coefficients and pitch
Bandpass filter analysis
Pseudo formant analysis using stored reflection coefficients
Distance metrics:
    Euclidean
    Weighted Euclidean
    Mahlanobis
FFT's using stored digital signals

Using test bed software, the PDP-11 can send data to and receive data from the Quintrell, and direct the Quintrell to execute any of the functions in Table 2.1. Specific speaker recognition algorithms are implemented by specifying a string of processing commands to the Quintrell, and identifying the files where data is to be obtained and stored. In addition, functions not currently available in the Quintrell can be coded in Unix "C" and used to supplement the Quintrell's capabilities.

The Quintrell portion of the speaker recognition test bed consists of a control program, or executive, and a series of modular programs that perform the signal processing functions shown in Table 2.1. The Quintrell executive communicates with the PDP-11 over a DMA interface. It receives and decodes commands from the PDP-11, loads the necessary data from the PDP, directs the execution of the specified function, and returns any generated data to the PDP-11.

The test bed is thus a vehicle for developing and testing speaker recognition algorithms in a high level language. It also provides a high speed implementation of these algorithms so that large amounts of data can be efficiently processed in resonable periods of time.

To supplement the routines already available in the test bed, additional Unix "C" programs were written. These included the subpopulation filters and the model generator.

2.3.1 Subpopulation Filter

Two subpopulations were investigated under this contract, the voiced speech subpopulation, and the vowel subpopulation. Markel originally used the voiced speech subpopulation in his studies. However, since Pfeifer used the vowel subpopulation in his work with good results, both subpopulations were investigated. Two vowel subpopulation filters were used. The first, the single vowel subpopulation filter, locates the vowel nuclei and extracts a single vowel frame for each nucleus. This filter produced so few frames when used with 10 and 20 second utterances, that difficulty was encountered in generating models. Therefore, a second filter, the multi-vowel subpopulation filter, was also implemented. The multi-vowel filter extracts three frames from the center of each vowel nucleus, and therefore produces three times as many output frames as the original filter. The basic algorithm is the same for both filters

The subpopulation filters are describe below in detail. To extract the input frames that belong to the various subpopulations, the speech signal is first differentiated into voiced and unvoiced subpopulations. A subset of the voiced population, vocalic nuclei, is then separated. Since programs for subpopulation filtering are not currently part of the speaker recognition test bed, these routines were written in Unix "C" and executed on the PDP-11. The methods for separating the various subpopulations are described below.

Voiced Speech: The voice/unvoiced subpopulations are derived by the voicing detector in the LPC analysis routines. This voicing detector uses an energy measure with a number of adaptive energy thresholds, and zero crossing analysis to make its decisions. It also incorporates smoothing and isolated error correction to the voicing decision. The algorithm is briefly outlined below.

An initial voicing decision is made based on the following energy related parameters:

1. The low-passed speech energy of the present frame.
2. The low-passed speech energy of the previous frame.
3. The updated background noise energy, defined as:

$$N_t = 15/16\ N_{t-1} + 1/16\ P_t$$

where

P_t is the power of the low pass signal

and

N_t is the noise power.

Also, N_t is constrained to be greater than $V_t/32$

where V_t is defined as

$$V_t = (63/64\ V_{t-1} + P_t)$$

and

V_t is only updated during frames judged to be voiced during the primary stage.

Three decisions are allowed in the initial stage:

1) Definitely voiced,
2) Tentatively voiced,
3) Unvoiced.

A secondary stage of the voicing algorithm is then used to refine the initial voicing decision. This refinement is based on the following:

1) k_1 and k_2, the first and second reflection coefficients, are used in the secondary stage of the voicing algorithm. If the frame is marked as voiced during the primary state and $100(k_1 + k_2) < -80$ then the frame is converted to unvoiced.

2) The number of zero crossings of the full band speech is also used in the secondary voicing decision. If the numbe: of zero crossings in the frame of 180 samples is greated than 52, a voiced decision at the primary stage is converted to unvoiced.

3) Finally, a three frame nonlinear smoothing function is applied to the voicing decision.

This algorithm is the same as is currently used by the DOD LPC-10 secure voice system, and has been thoroughly tested under various signal conditions.

Vowel Nuclei: The vowel subpopulation is obtained in two steps. First, all non-steady state (transitionals) frames are eliminated from the voiced population leaving only vowels and some sonorants. Then the vocalic nuclei are separated from the remaining sonorants. The procedure is as follows.

A voiced speech frame is determined to be transitional if the following is true

$$F_n^* > \overline{F}_n$$

where F_n^* is the distance between the $(n-1)^{th}$ and the n^{th} frame, defined as

$$F_n^* = \sum_{i=1}^{10} (f_{i,n} - f_{i,n-1})^2$$

and $\overline{F}_n$ is the average distance between frames, calculated as

$$\bar{F}_n = 1/(n-1) \cdot \sum_{i=1}^{n-1} (F_i^*)$$

where

$f_{i,n}$ is the i^{th} speech parameter at the n^{th} frame in time.

The definition of the voiced transitional is rather ad hoc. The algorithm does eliminate most transitional areas without much error. The weakness in the process is that the remaining voiced speech is not accurately defined as steady state speech, but is a mixture of steady state sounds and weakly transitional speech. This was not considered to be a serious problem since the steady state subpopulation was not used in developing recognition features but was only used as a candidate population for the vocalic subpopulations.

The second step in locating vocalic nuclei separates vowels from sonorant sounds. The n^{th} analysis frame is defined as a vocalic nucleus if the voiced speech frame at time n is a local maxima of the signal power function. More precisely, if

P_i is the signal power for the i^{th} frame then

$$P_{n-2} < P_{n-1} < P_n$$

and

$$P_{n+2} < P_{n+1} < P_n$$

and

$$P_n > \bar{P}_n$$

where

$$\bar{P}_n = (31/32) \cdot \bar{P}_{n-1} + (1/31) \cdot P_n$$

for n and n-1 voiced frames.

If the condition is met, then the nth frame is said to be in the region of a vocalic nucleus. The actual speech frame used in the single vowel subpopulation is the frame m such that

$$F^*_m \text{ is a minimum for } n-2 < m < n+2$$

where F^*_m is the rate of change defined above.

The multi-vowel subpopulation filter includes not only the mth frame, but also the m-1th, and the m+1th frames.

2.3.2 Model Generation

The second routine written in Unix "C" to supplement the speaker recognition test bed is the model generator. Each model is generated from a set of speaker recognition features derived from the LPC-10 analysis of the speech signal. The model generator routine calculates the mean vector and covariance matrix from the desired input speaker data (the reflection coefficients in the appropriate subpopulation). A standard mathematics routine is then used to invert the matrix. Two types of covariance matrices were studied.

Individual Covariance: Individual covariance matrices were generated for each speaker in the data base using only those feature vectors that were known to originate from that speaker. Models of this type incorporate speaker dependent information into the weighting matrix of the model.

Pooled Covariance: Pooled covariance models contain the covariance between features generated across all speakers. The mean vector for each speaker model is derived using data from each individual speaker, but the covariance matrix is generated by "pooling" the data from all speakers. A single covariance matrix is calculated for the "pooled" data. This type of model does not exploit any speaker dependent information that may be contained in the individual covariance matrices. The model does have the advantage that a relatively few frames from each speaker when pooled together, result in enough frames to adequately estimate the covariance matrix.

2.3.3 Data Base

The data base used in this study is a subset of the ARPA data base. It is made up of 17 speakers, 11 males and 6 females. All 17 of the speakers were adults, ranging in age from their early 20's to late their 30's. None of the speakers had a distinguishing accent or regional dialect. Two sessions that were recorded one week apart were used.

Originally, each recorded session was approximately 20 minutes in length. The end result of tape editing produced three minute segments for each of the 17 talkers. The editing removed long pauses, laughter, and non-speech sounds whenever possible.

2.4 MARKEL'S SPEAKER RECOGNITION TECHNIQUE

A block diagram of Markel's speaker recognition technique is shown in figure 2.2. Markel's technique is a subset of the generalized speaker recognition system described in Section 2.2. The functional blocks in the technique are as follows:

1. LPC-10 analysis is performed on the input speech. The speech parameters used for Markel's technique are the ten LPC reflection coefficients.

2. The next block is the subpopulation filter. Markel's original implementation used only the voiced subpopulation. For this contract, two subpopulations were tested with Markel's technique, the vowel subpopulation, and the voiced speech subpopulation.

MARKEL'S SPEAKER AUTHENTICATION TECHNIQUE

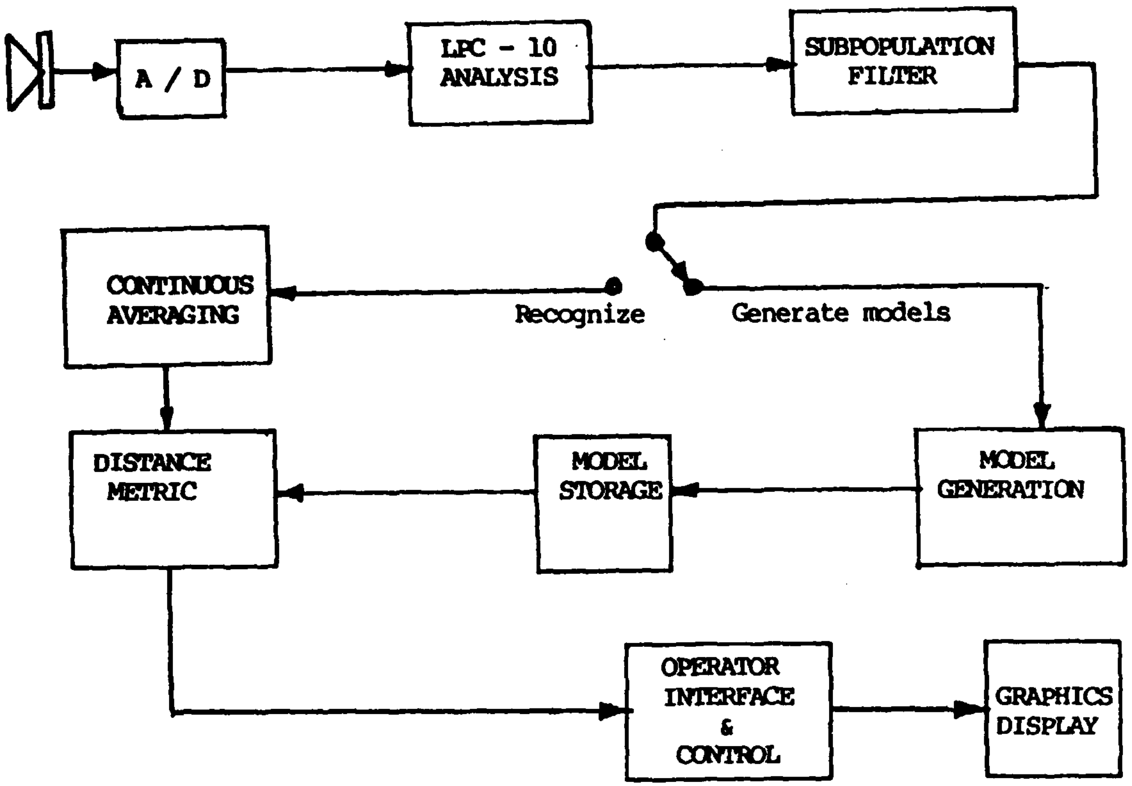


FIGURE 2.2

3. The models used for Markel's technique are generated by computing the averages for each of the ten reflection coefficients, and the associated covariance matrix.

4. Markel's technique averages the unknown frame data before calculating a distance between the feature vector and each model in the reference set. The model with minimum distance to the feature vector is selected as the speaker for that recognition trial.

5. The distance metric for Markel's technique is the Mahlanobis metric. This metric is defined as follows:

$$D = (\bar{F} - \bar{M})\ [W]^{-1}\ (\bar{F} - \bar{M}) \qquad \text{Eq. 2.1}$$

where $\bar{F}$ is the average coefficient vector,
$\bar{M}$ is the mean vector from a model,
and $[W]^{-1}$ is the inverse covariance matrix from a model.

The metric outputs the "winner" (the model closest to the unknown speech) for each input frame.

2.4.1 Experimental Design with Markel's Technique

The ability of Markel's technique to perform speaker recognition was evaluated using five different experiments. They are shown in Table 2.2.

Table 2.2: EXPERIMENTS WITH MARKEL'S TECHNIQUE

1. Voiced Speech Subpopulation with Individual Covariance Matrices
2. Voiced Speech Subpopulation with Pooled Covariance Matrices
3. Single vowels with Individual Covariance Matrices
4. Multiple vowels with Individual Covariance Matrices
5. Multiple vowels with Pooled Covariance Matrices

2.4.1.1 LPC-10 Analysis

The first step was to perform LPC-10 analysis on the digitized data base. The digitized speech is sent to Quintrell over the DMA interface. The ten reflection coefficients, the frame number, the pitch, and the power for each analysis frame are computed in the Quintrell and returned to the PDP-11 for storage. An LPC data base was thus created for use in evaluating Markel's technique.

2.4.1.2 Subpopulation Filtering

The next step in processing the data was to filter the LPC-10 frames using the subpopulation filter routines described in the preceding section. This resulted in three new data bases:

1. The voiced subpopulation
2. The single vowel subpopulation
3. The multi-vowel subpopulation

Recognition experiments were then run with each of these three subpopulation data bases. The experimental procedure was virtually identical for all five experiments. The processing of the voiced subpopulation is described below.

2.4.1.3 Model Generation

First, models were generated from the appropriate portion of the voiced subpopulation data. Two recorded sessions of speech, sessions numbered 5 and 6, recorded within one weeks time, were examined in the experiments. The models were generated from the two sessions to determine the effect of using aged models in the process of speaker authentication. An aged model refers to models generated at a time other than the test session, not the physical age of the speaker.

Another variable in generating models, is the number of feature vectors used to create the model. The models used in testing Markel's technique are listed in Table 2.3 below.

Table 2.3: DATA USED FOR GENERATING DIFFERENT MODEL TYPES

1) Model I:
Feature vectors from the first 10 seconds of the test set, session 5;
2) Model II:
Feature vectors from the first 10 seconds of session 6;
3) Model III:
Feature vectors from the first 20 seconds of the test set, session 5;
4) Model IV:
Feature vectors from the first 20 seconds of session 6;
5) Model V:
Feature vectors from all 3 minutes of the test set, session 5;
6) Model VI:
Feature vectors from all 3 minutes of session 6;

For each of the five experiments using the various subpopulations, models were generated from the feature vectors belonging to the subpopulations found in the first 10 seconds, 20 seconds and 3 minutes of voiced speech. At the time of subpopulation filtering, values were saved representing the time slice from which the features were derived to permit the selection of the correct feature vectors for model generation. In this manner, it was possible to compare results for different populations of feature vectors with respect to the same time slices of the speech signal in both the model and the test sets.

The contract asked for models to be generated with as little as ten seconds of data. Models I and II were generated with ten seconds of data and used to determine recognition accuracy with this limited speech. Models II and IV were generated using approximately 20 seconds of speech, and used to determine if more speech in the model would significantly improve performance.

Models I and III are of most interest for this contract since they are generated using speech from the same time period as the unknowns. Models II and IV are generated with data one week older than the unknowns, and were chosen to measure the effects of the model age on system performance.

Model V was generated using the same data as used to generate the unknowns, and is therefore total unrealistic. Its only value is to provide an upper bound on system performance.

Model VI was generated using 3 minutes of speech recorded one week later than the unknowns. These models closely resemble the models originally used by Markel, and were used to validate the performance of the ITTDCD implementation of the algorithm. Since the contract required much shorter speech segments be used for model generation, model VI does not apply to this contract.

2.4.1.4 Frame Averaging

The next step in the experiments was to average the data. The averaging is done over various block sizes. The blocks were set up for recognition trials of 40 ,20, 10, 5, 2.5, 1.25, and .05 seconds. The blocks contained 800, 400 ,200, 100, 50, 25, and 1 voiced frames respectively.

The blocking and averaging procedure is identical for all the subpopulations. In order to compare the performance of the various subpopulations, the blocking is always done so the frames from the same portion of the speech always end up in the same block. For example, the 40 second blocks for the vowels are generated using the same input speech as the 40 second blocks for the vowels are generated using the same inout speech as the 40 second blocks for the voiced speech. However, the number of frame in each block is no longer fixed as it was with the voice frames. Only the vowel frames that were contained in the original 800 voiced frame block are included in the corresponding vowel block.

2.4.1.5 Distance Metric

The final step in the experiments to evaluate Markel's technique was to calculate the distance metric on the average frames (one for each recognition trial). The model with the minimum distance to the unknown average frame is selected as the talker. The percentage of correct recognitions over all the recognition trials is then tabulated.

2.4.2 Experimental Results of Markel's Technique

The results achieved by Markel's technique are very good. Although the technique was original tested by Markel using models generated from several minutes of speech, the results obtained in this effort indicate that acceptable performance can be obtained using models generated from 10-20 seconds of speech.

The performance of the 10 second models (model type I) for various lengths of unknown speech is shown in figure 2.3. The recognition accuracy for the voiced speech subpopulation (curves 1 and 2) is approximately 15% better than the multi-vowel subpopulation (curves 3 and 4). The performance of the single vowel subpopulation (curve 5) is approximately 20% less than the multi-vowel. The poor performance of the single vowels (curve 5) is probably due to the small number of frames available for generating the models.

The results for the ten second models indicate little difference between models using the individual covariance matrix and the pooled covariance matrix (curves 1 vs 2 and 3 vs 4). The performance of the system continues to increase as the length of the unknown speech segments increases. For a 40 second unknown, the best recognition rate is

FIVE IMPLEMENTATIONS OF MARKEL'S TECHNIQUE

200 FRAME (10 SECONDS) MODELS

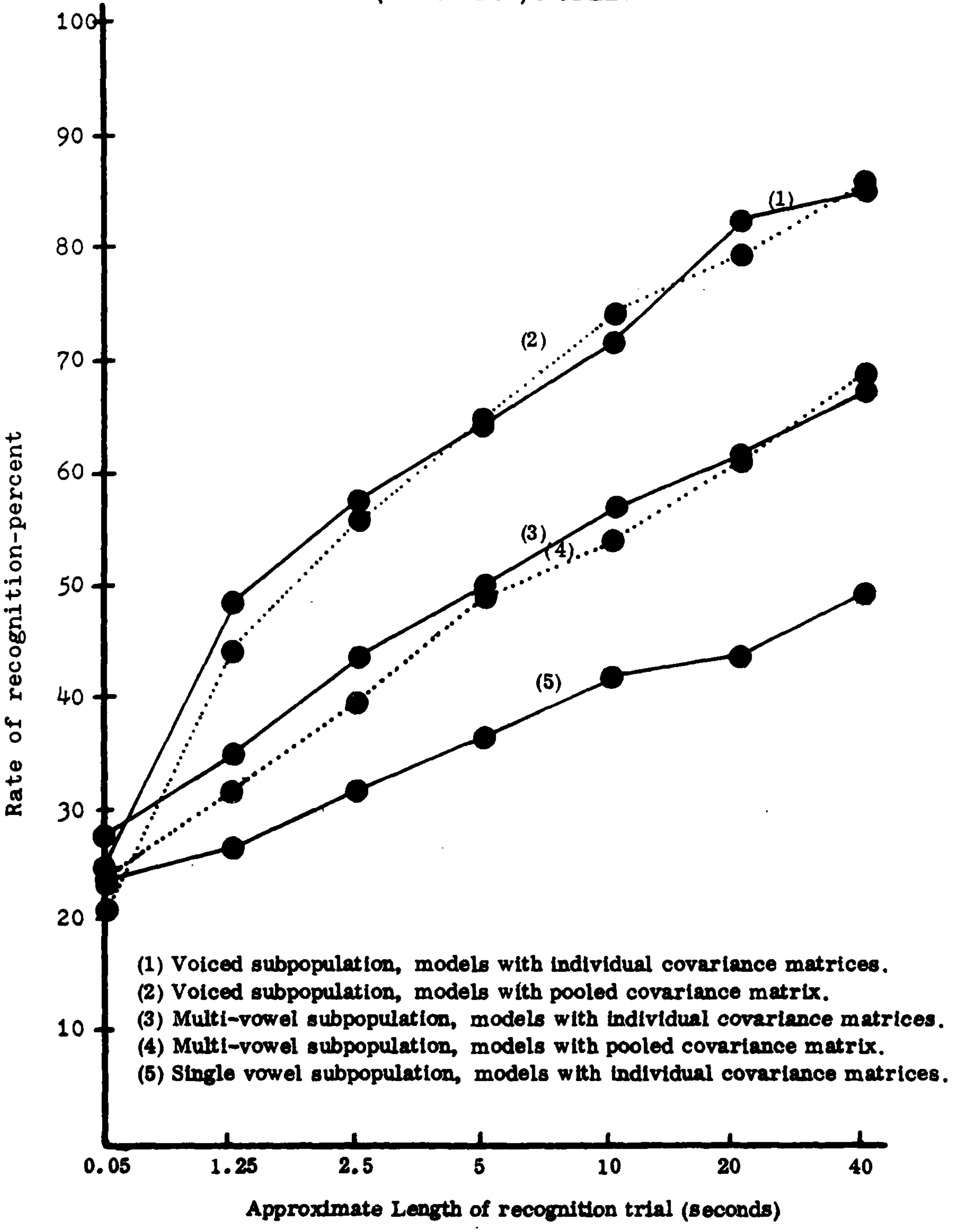


FIGURE 2.3

anproximately 86%.

Fiqure 2.4 shown the results for Markel's techniciue using 20 seconds of speech to generate the models (type III models). The performance imoroves bv approximately 10% over that of the 10 second models. The recoqnition accuracy with 40 second unknowns is approximately 95%.

There is a significant difference in the performance of the pooled covariance versus the individual covariance models for the vowel subpopulations. This is probably due to the fact that when 20 seconds of speech are pooled to generate the model's covariance matrix, enough frames are available to obtain a resonable estimate of the covariance. From the 10 second models, and the individual covariance models at 20 seconds, the number of frames available is still not adequate to estimate the covariances.

The problem of limited data for generating the model is responsible for a change in the implementation of Markel's algorithm. As shown in fiqure 2.2, the models are generated from the individual frames in the subpopulation. In Markel's original study, the models were generated from speech that was averaqed in to the same block lengths as the unknown speech used in the recoqnition trials. The use of averaged data blocks to generate the models does not affect the mean vector in the model, but does affect the covariance matrix. Also, number of frames required to generate a model increases siqnificantly when averaged frames are used .

As part of this contract, an experiment was run to determine how the performance of the recognition system is effected by the averaging of the data before model generation. Figure 2.5 is a block diagram showing where the averaging is done in the model generation process. Five different models were produced using three minutes of speech recorded one week after the unknowns. Markel's technique was run using the voiced subpopulation and these five model types. The results of the test are shown in figure 2.6. The curves indicate that the less averaging that is done before generating the models, the better the performance. This is probably due to the large number of frames required to adequately estimate the covariance

FIVE IMPLEMENTATIONS OF MARKEL'S TECHNIQUE

400 FRAME (20 SECONDS) MODELS

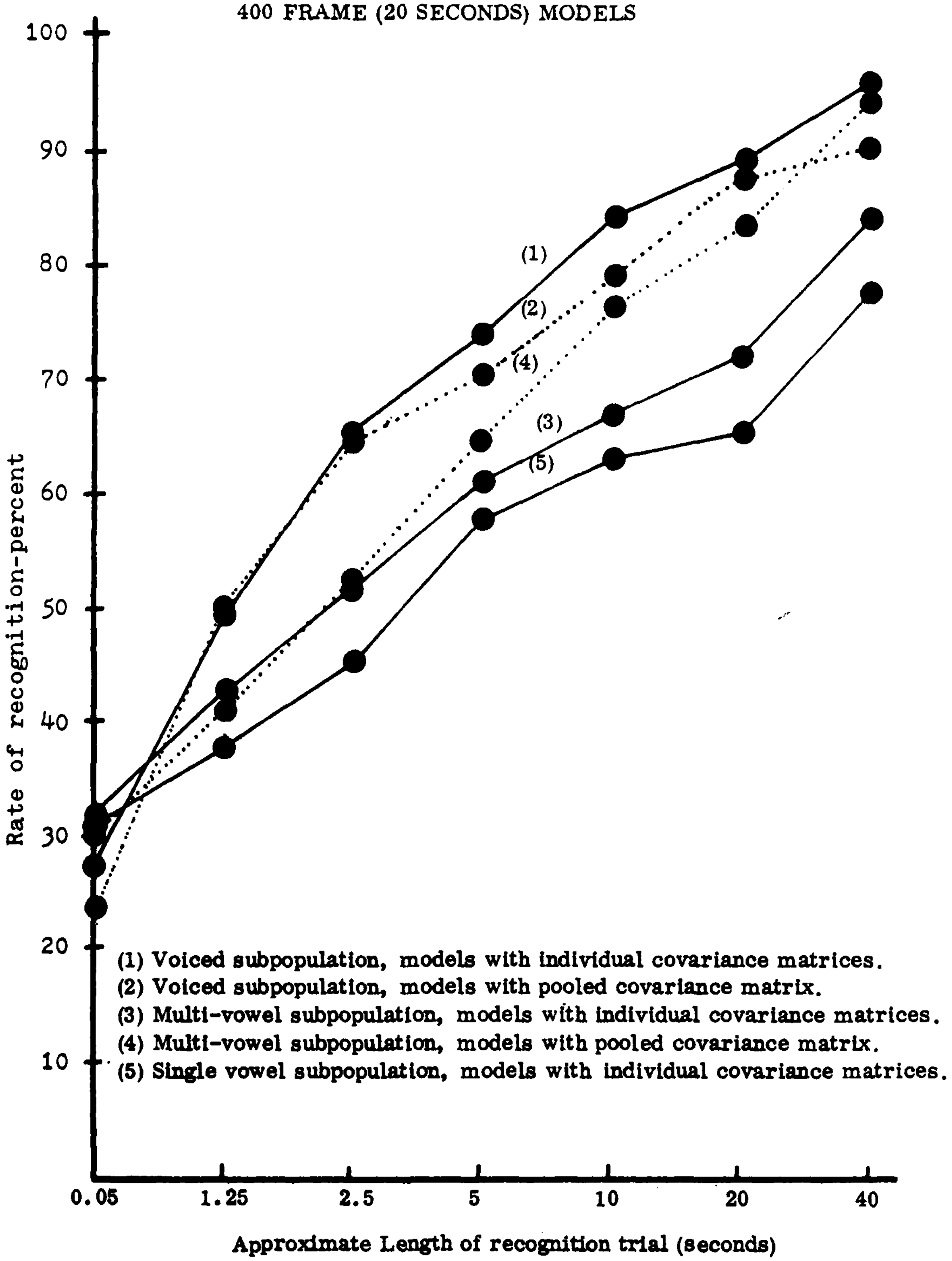


FIGURE 2.4

MODEL GENERATION

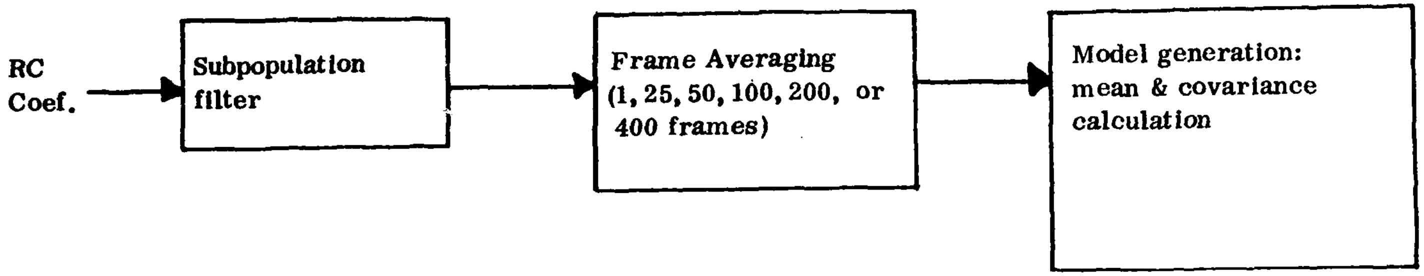


FIGURE 2.5

EFFECTS OF FRAME AVERAGING BEFORE MODEL GENERATION
Markel's Technique: 3 minute models

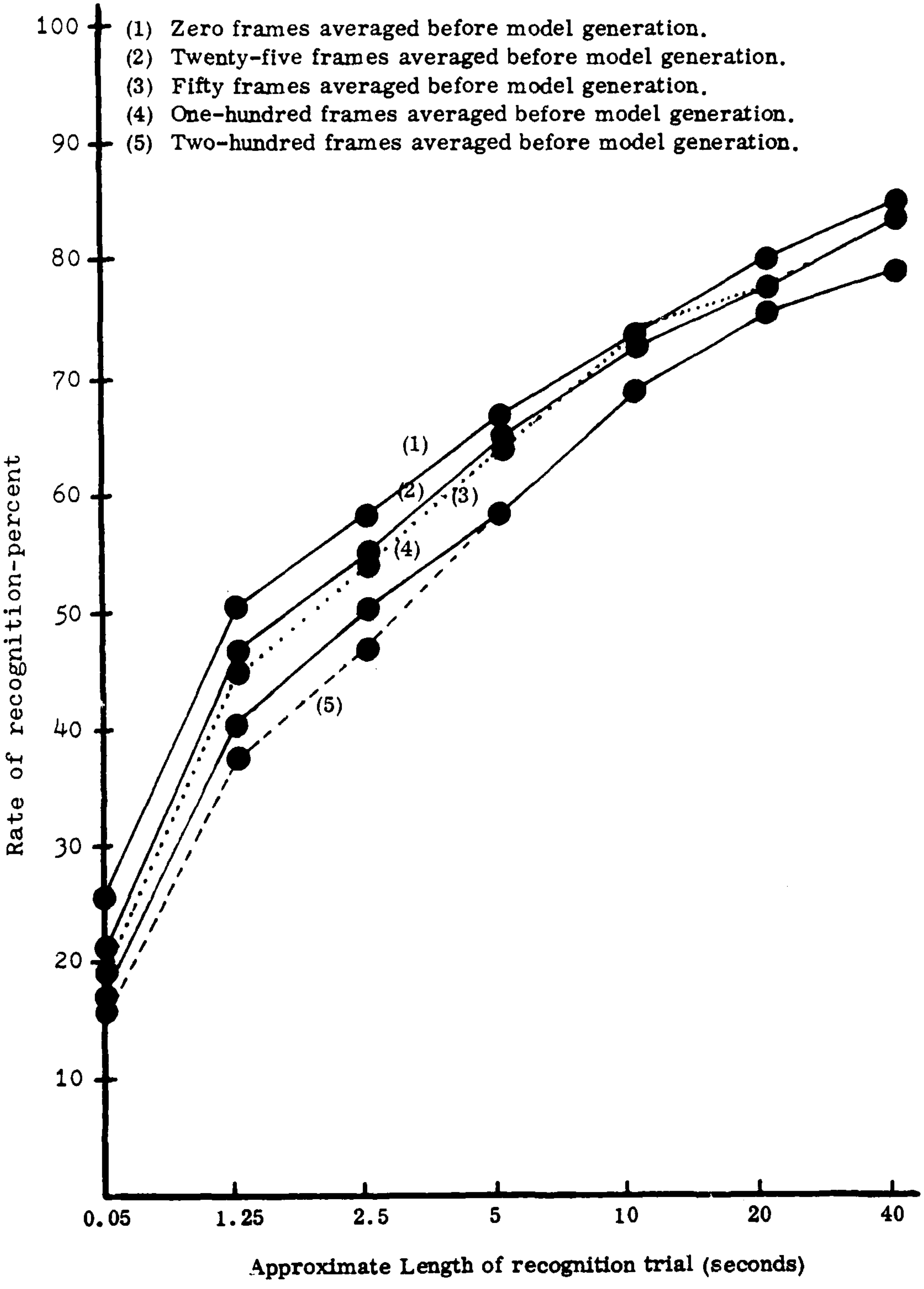


FIGURE 2.6

matrix. When the frames are averaged before the model is generated, there are not enough averaged frames to accurately estimate the covariances. Even though the covariances that are estimated in the no-averaging case are not the covariances of the data used in the distance metric (the unknown data in the distance metric is averaged data), this matrix still performs better in the model than the correct, but inaccurately estimated covariance matrix (determined from averaged data). This is very fortuitous since the contract requires ten second models (~200 frames), and if averaging were required, the number of averaged frames would not be enough to estimate the covariance matrix at all.

One additional experiment was conducted with Markel's technique to determine the effect of a moderate amount of noise on the recognition performance. The best subpopulation (voiced speech) from the above experiments was used. The data base was contaminated by adding white noise. The signal to noise ratio was reduced to approximately 15 dB as follows.

1. The RMS power of the original data base was measured. Since the original data was relatively noise free, this RMS measurement was assumed to be the signal power.

2. A white noise data base was generated, and the noise power scaled to be 15 dB less than the signal power (as measured above).

3. The scaled noise was then added to the original signal.

The results for Markel's technique with the noisy data are shown in figures 2.7 and 2.8, for the 10 and 20 second models respectively. The system performance is degraded by less than 10% in all cases by the noise. For the 10 second models, the performance was actually better in some case with the noisy data. This may be due to the fact that low level voiced frames, classified as voiced in the noise free data, are classified as non-voiced in the noisy data. These low level frames may not be characteristic of the speaker, and therefore performance improves slightly when they are eliminated. Secondly, the number of recognition trials

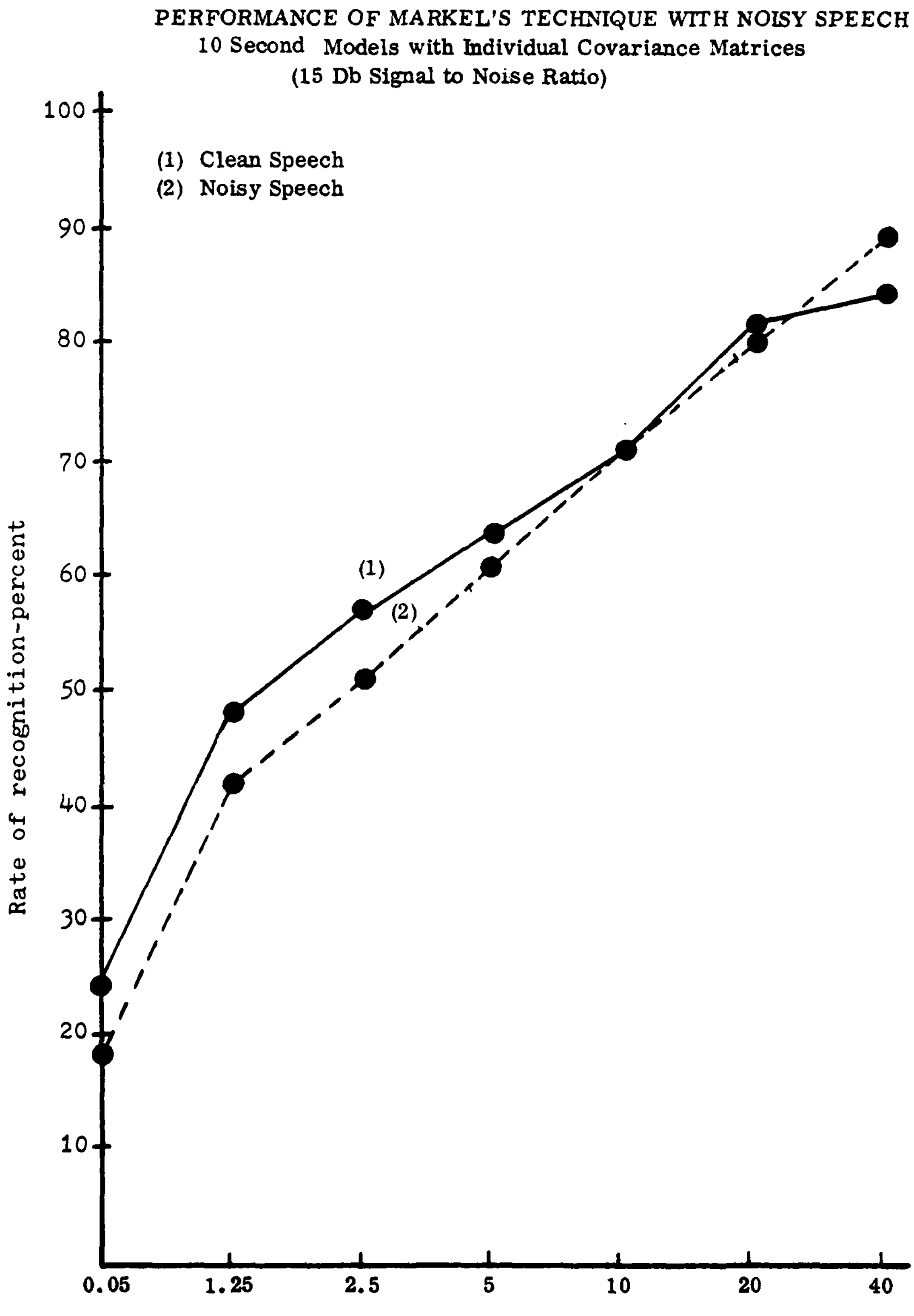

PERFORMANCE OF MARKEL'S TECHNIQUE WITH NOISY SPEECH
10 Second Models with Individual Covariance Matrices
(15 Db Signal to Noise Ratio)
(1) Clean Speech
(2) Noisy Speech
(1)
(2)
Rate of recognition-percent
100
90
80
70
60
50
40
30
20
10
0.05
1.25
2.5
5
10
20
40
Approximate Length of recognition trial (seconds)

FIGURE 2.7

PERFORMANCE OF MARKEL'S TECHNIQUE WITH NOISY SPEECH
20 Second Models with Individual Covariance Matrices
(15Db Signal to Noise Ratio)

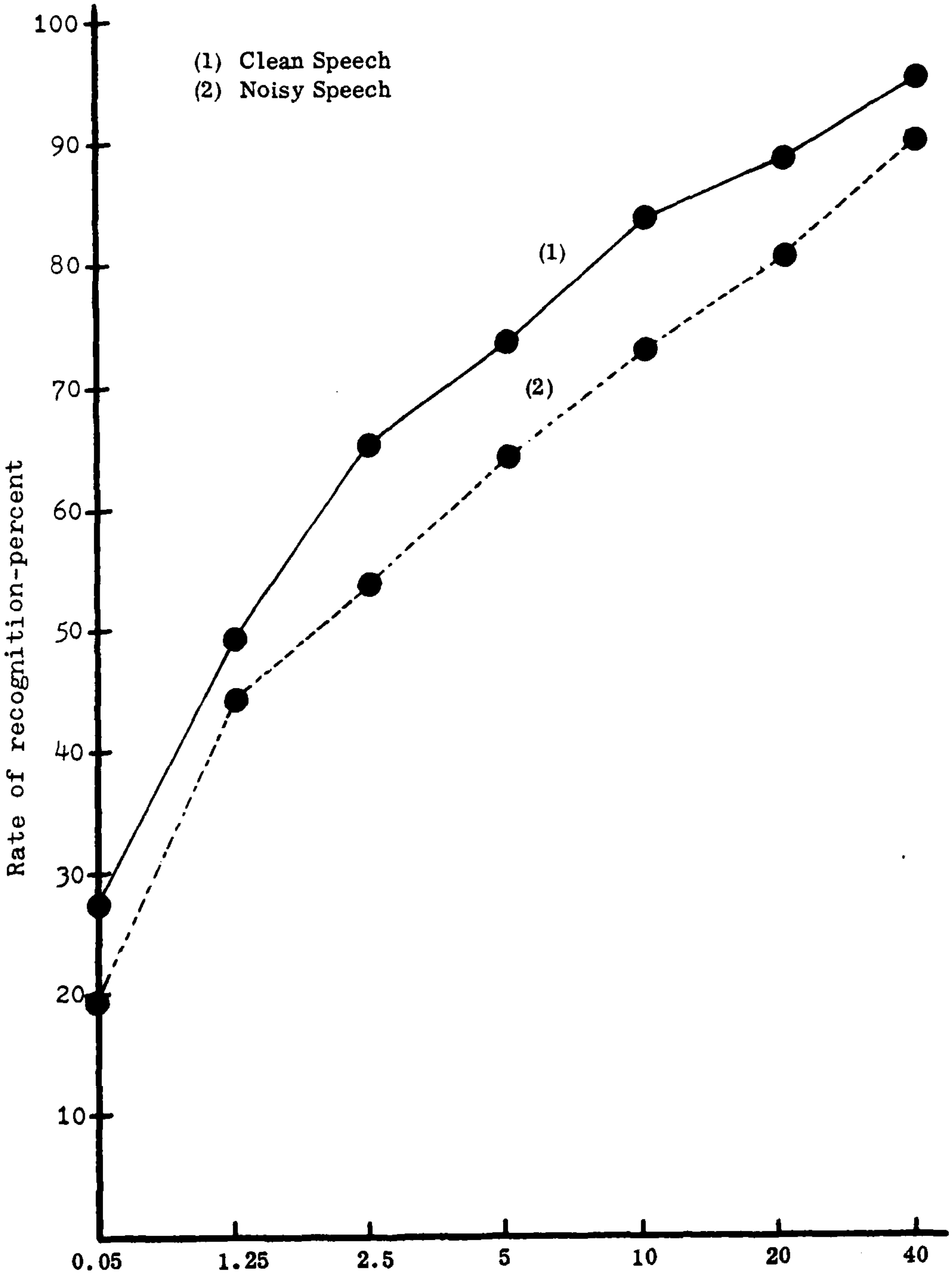


FIGURE 2.8

conducted for the 40 second unknowns is only four or five per talker, and therefore there can be significant variance in the measured performance.

The performance of all the models in table 2.3 is shown in figure 2.9 for the voiced speech subpopulation and session five used as the unknown. The complete results for all of the experiments with Markel's technique are in the appendices. As expected, model V (model generated from the unknown speech) performed best, and demonstrates that if extremely good models are available, 100% recognition can be obtain for certain talker sets. Model III (20 seconds from the same session as the unknown) has the second best performance. Model III is superior to model VI (3 minute models made from data recorded one week after the unknown), indicating that changes in talker characteristics take place over periods as short as one week. The performance of models II and IV (10 and 20 seconds taken one week apart from the unknown) perform much worse than the corresponding models generated at the same time as the unknowns. This is also indicative of changes in the speaker's characteristics over time.

2.5 PFEIFER'S SPEAKER RECOGNITION TECHNIQUE

A block diagram of Pfeifer's speaker recognition technique is shown in figure 2.10. Pfeifer's technique is very similar to Markel's speaker recognition technique described above. The main difference between the two is that Pfeifer does not average the frames passed by the subpopulation filter before calculating the distance metric. Instead, the distance metric is calculated on each individual frame in the subpopulation. The actual speaker decision in Pfeifer's technique is made with a majority vote over the sequence of winners produced by the distance metric.

COMPARISON OF 6 MODEL TYPES: MARKEL'S TECHNIQUE - VOICED SPEECH

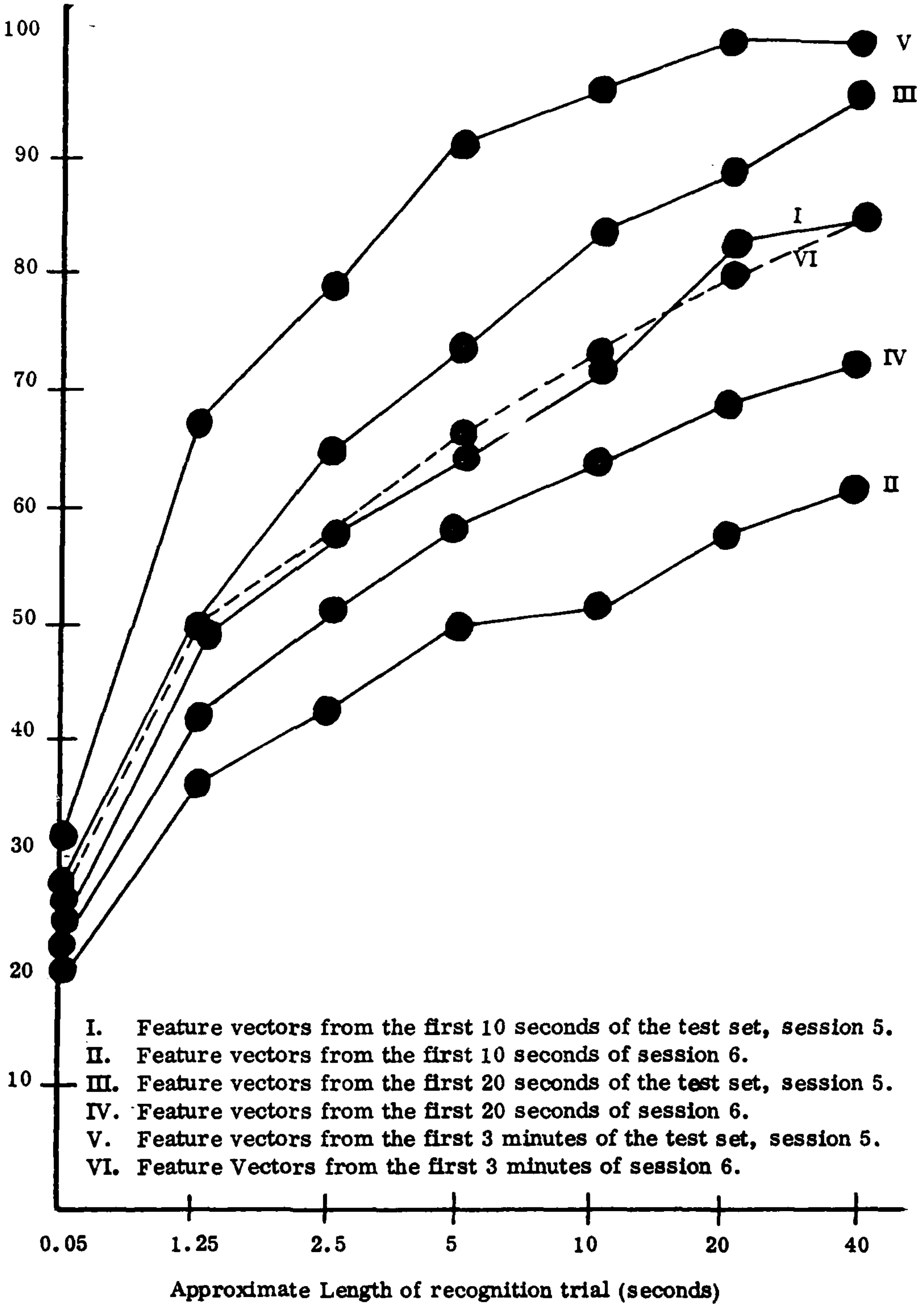


FIGURE 2.9

PFEIFER'S SPEAKER AUTHENTICATION TECHNIQUE

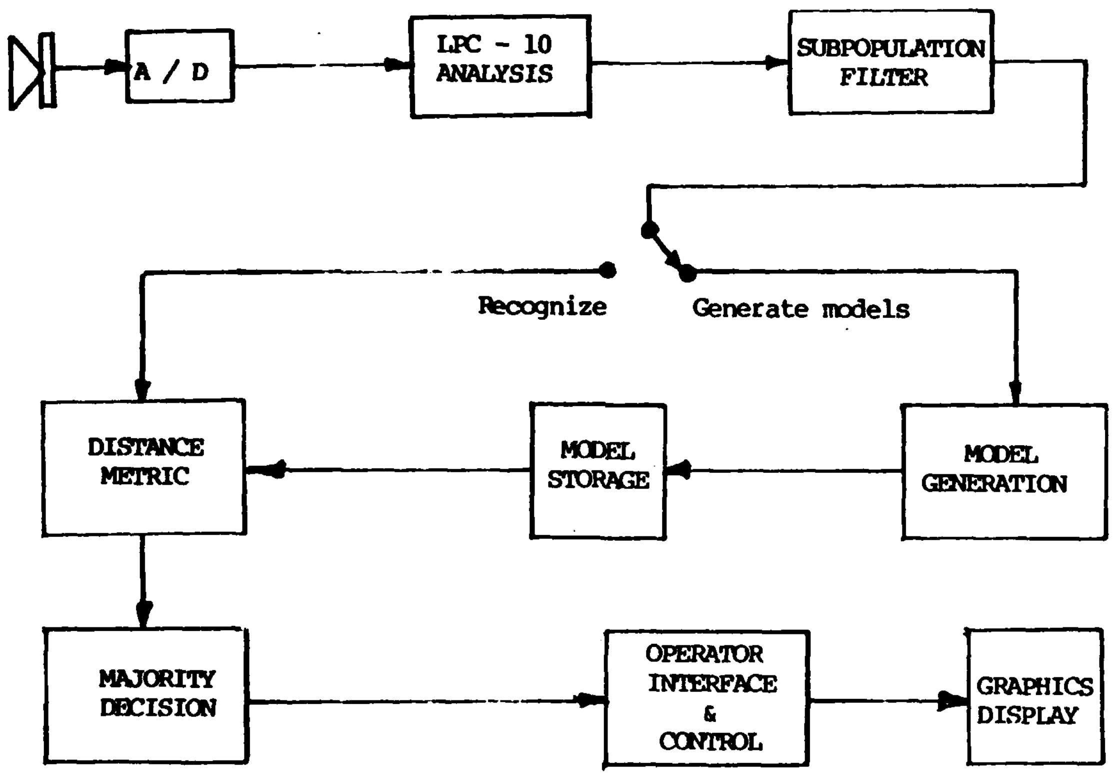


FIGURE 2.10

In Pfeifer's original implementation, an acceptance, and rejection threshold are set and models with fewer wins than the rejection level are removed from contention. When the accumulated wins for a model exceeds the acceptance threshold, the corresponding speaker is selected as the unknown talker. This sequential analysis technique therefore allows the recognition to be completed in a variable amount of time that depends on how well the unknown speech matches one of the models. For the algorithm implementation in this study, a fixed amount of "time for recognition" was used for each recognition trial, and therefore the acceptance and rejection threshold were not used. Since the recognition had to be completed in a fixed time, there was an upper limit on the amount of speech that could be processed. Speaker decisions had to be made at the end of the period even if the acceptance threshold had not been reached. On the other hand, if the acceptance threshold was reached prior to the end of the recognition period, the inclusion of the remainder of the unknown data (out to the end of the recognition period) should only improve the results. Therefore, it was concluded that using fixed lengths for the recognition trials would produce an accurate estimate of the technique's performance under the conditions of interest.

A comparison of the implementations of Markel's and Pfeifer's techniques is presented below:

1. The first two functional blocks are identical for Pfeifer's and Markel's technique. LPC-10 analysis is performed on the input speech. The speech parameters used are the ten LPC reflection coefficients.

2. The next block is the subpopulation filter. Pfeifer's original implementation used the voiced speech subpopulation. In this implementation, the same two subpopulations used with Markel's technique were tested with Pfeifer's technique; the vowel subpopulation, and the voiced speech subpopulation.

3. The models used for both Markel's and Pfeifer's technique are generated by computing the averages for each of the ten reflection coefficients, and the associated covariance matrix.

4. A major difference between Pfeifer's and Markel's technique is in the recognition feature extraction. Markel's technique averages the unknown frame data before calculating a distance between the feature vector and each model in the reference set. Pfeifer's technique calculates the distance metric on each individual frame in the subpopulation with no averaging.

5. The distance metric for both Pfeifer's and Markel's technique is the Mahlanobis metric, as defined in equation 2.1. The metric outputs the "winner" (the model closest to the unknown speech) for each input frame.

6. The next functional block in Pfeifer's technique is the sequential analysis routine. This routine examines the sequence of winners produced by the distance metric, and identifies the unknown talker as the model with the majority of the winners over the unknown utterance. This is different than Markel's technique in that Markel only looks at one distance metric calculated for the averaged frame for the unknown utterance.

2.5.1 Implementation of Pfeifer's Recognition Technique

Pfeifer's speaker recognition technique was implemented as a subset of Markel's technique using the speaker recognition test bed. Only one new routine was written to implement Pfeifer's technique. The implementation of Markel's technique described in section 2.3.1 was run using one frame averaging before the distance metric. This corresponds exactly to Pfeifer's technique, and the output of Markels implementation with the one frame averages is the sequence of winners that the sequential analysis routine in Pfeifer's technique requires. This list of winners was stored on disk, and a Unix "C" program was written to implement the sequential analysis and to tabulate the correct recognitions.

2.5.2 Experimental Design with Pfeifer's Technique

The experiments with Pfeifer's technique were carefully designed so they could be compared directly with those from Markel's technique. The experiments, shown in table 2.5 are identical to those done with Markel's technique (table 2.2), with the exception that the experiment with the voiced subpopulation and the pooled covariance models was not run with Pfeifer's technique.

Table 2.5: EXPERIMENTS WITH PFEIFER'S TECHNIQUE

1. Voiced Speech Subpopulation with Individual Covariance Matrices
2. Single vowels with Individual Covariance Matrices
3. Multiple vowels with Individual Covariance Matrices
4. Multiple vowels with Pooled Covariance Matrices

The experiments with Pfeifer's technique were actually run at the same time as Markel's technique, on the same data base. This was done by running Markel's technique with 1, 25, 50, 100, 200, 400, and 800 frame averaging. Then, the list of winners for the one frame averaging experiments was stored on disk. Next, the winners files were processed by the sequential analysis routine to identify the winning speaker for each recognition trial. The sequential analysis routine finds the model with the most winners over a particular block of data (one block corresponds to one recognition trial). This process is referred to as majority voting. The majority voting was done over various block sizes. The blocks were set up the same as in Markel's experiments for recognition trials of 40 ,20, 10, 5, 2.5, 1.25, and .05 seconds. The blocks contained 800, 400 ,200, 100, 50, 25, and 1 voiced frames respectively. The actual speech used in each block was chosen to be identical to the speech used in the corresponding recognition trial with Markel's technique to permit easy comparison of the techniques.

The orocedure was reoeated for each of the experiments in table 2.5. The experimental results for Pfeifer's technique are presented in the next section.

2.6 EXPERIMENTAL RESULTS OF PFEIFER'S TECHNIQUE

The four experiments listed in table 2.4 were conducted using Pfeifer's technique. The same six models used for Markel's technique (table 2.3) were used in these experiments. As before, the models from session five are of most interest in this program since models will be generated from speech recorded on the same day as the unknowns.

The original testing of this technique by Pfeifer was done using several minutes of speech for both the models and the unknowns. The results obtained indicate that the technique does not perform as well when only 10-20 seconds are used for the models and the unknowns.

Figure 2.11 depicts the performance of four implementations of Pfeifer's technique using 10 second models. The relative performance of the subpopulations is the same as for Markel's technique(figure 2.3). The voiced subpopulation performs approximately 15% better than the multi-vowel subpopulation, which is in turn approximately 20-30% better than the single vowels. As with Markel's results, the recognition rate for the single vowel subpopulation is severely hurt by the small number of frames in the models.

Figure 2.12 shows the results for the experiments with Pfeifer's technique with the 20 second models. The relative performance of the various subpopulations is again very similar to that of Markel's. The voiced subpopulation and the multi-vowel subpopulation exhibit approximately the same performance for 10, 20 and 40 second unknowns.

FOUR IMPLEMENTATIONS OF PFEIFER'S TECHNIQUE

10 SECOND MODELS

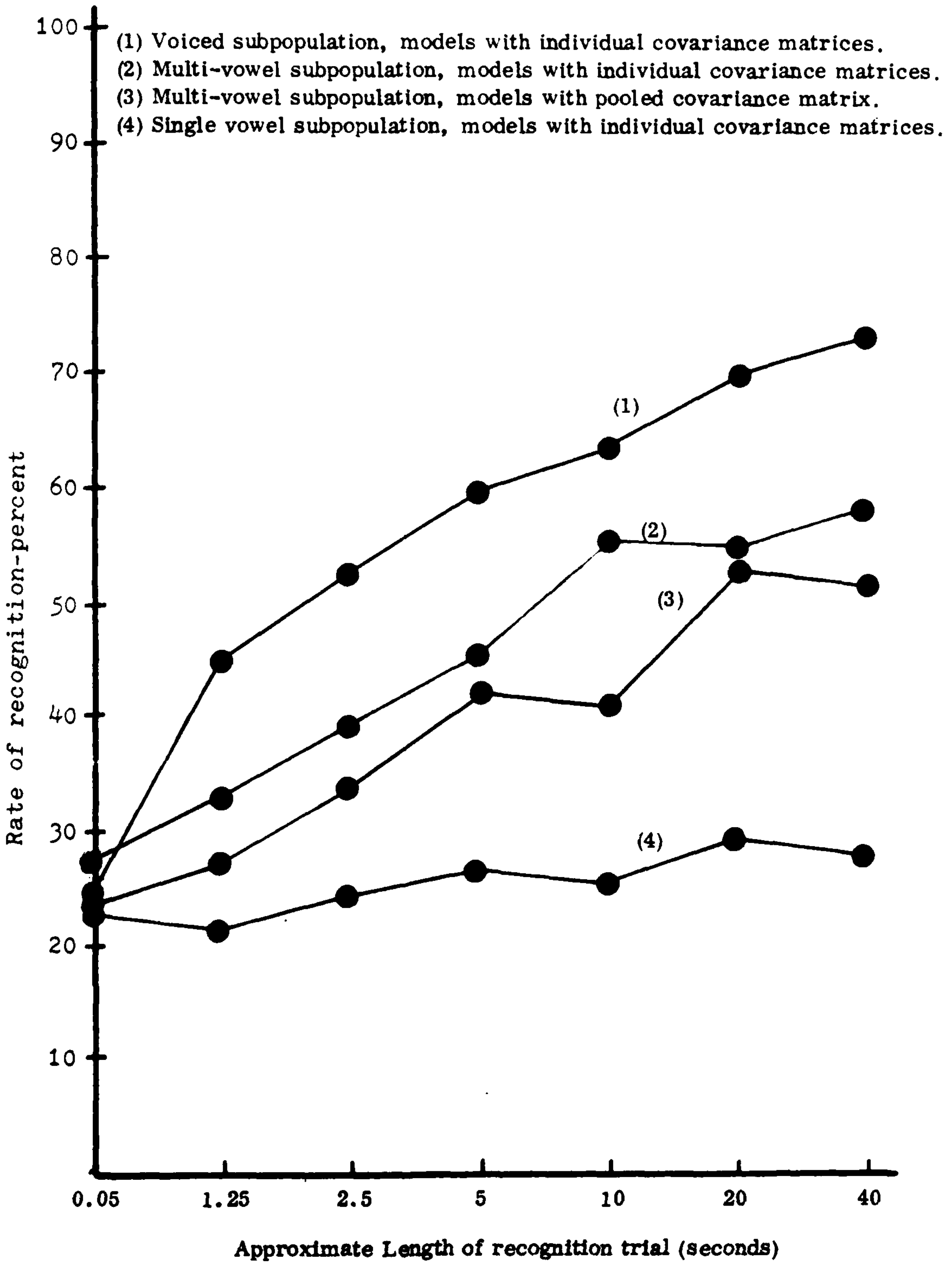
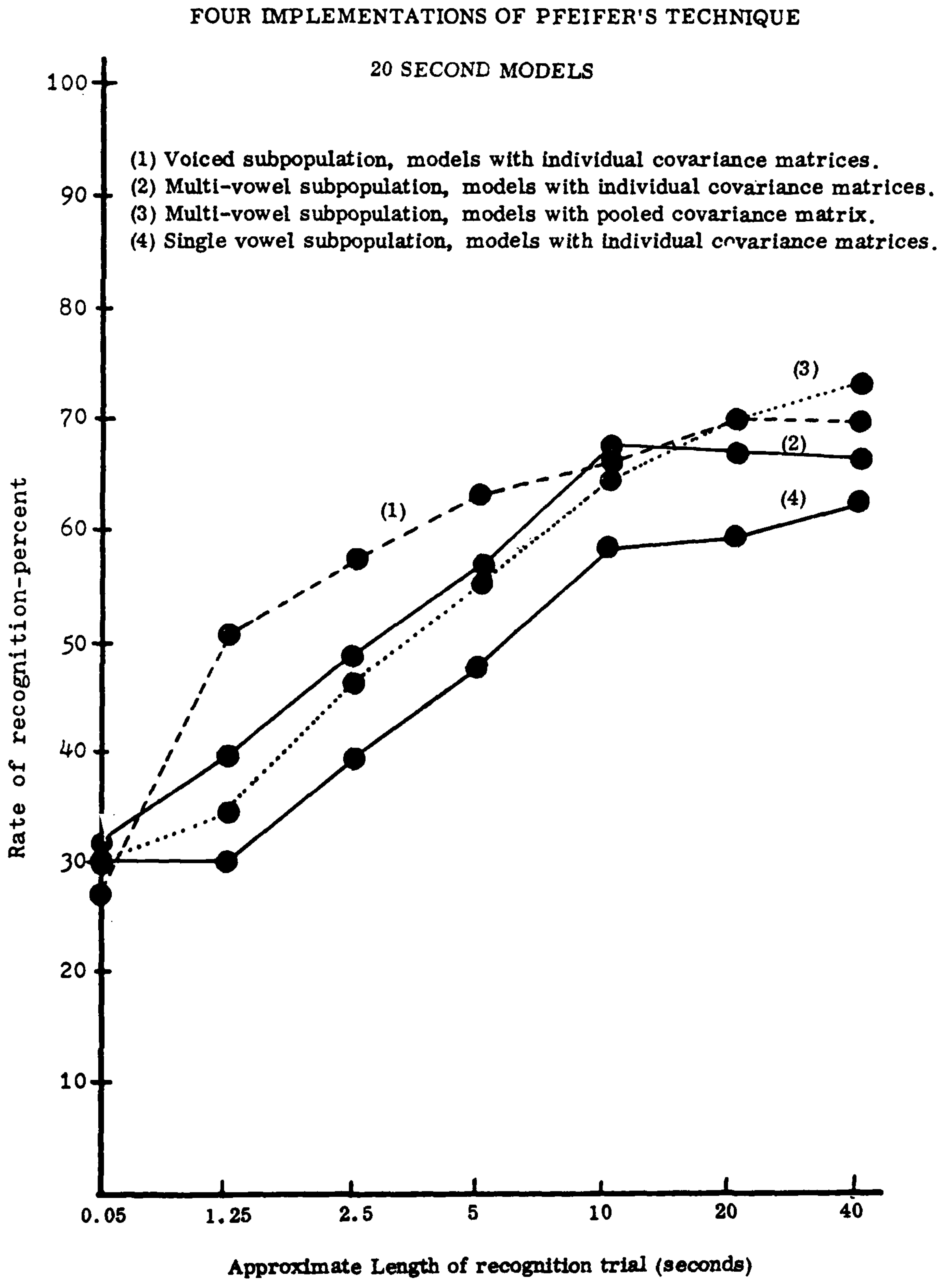


FIGURE 2.11



FOUR IMPLEMENTATIONS OF PFEIFER'S TECHNIQUE

20 SECOND MODELS

FIGURE 2.12

2.7 COMPARISON of MARKEL'S and PFEIFER'S TECHNIQUE

When short speech segments are used for the models and the unknowns, the overall performance of Pfeifer's technique is not as good as that of Markel's. Figure 2.13 is a comparison of the two techniques for the voiced subpopulation. Curves 1 and 3 are for the 10 second models with Markel's and Pfeifers techniques respectively. Curves 2 and 4 are with the 20 second models. For both model lengths, Markel's technique is approximately 20% better than Pfeifer's.

2.8 COMBINATIONS OF MARKEL'S and PFEIFER'S TECHNIQUE

As discussed earlier, Markel's technique uses averaged frames as recognition parameters, and does not use sequential analysis. Pfeifer's technique, on the other hand does not average frames, and uses sequential analysis. The results of the experiments in sections 2.3.3 and 2.4.3 indicate that Markel's technique (averaging frames) is superior to Pfeifer's technique (no averaging) under the test conditions. However, there is no reason why the two techniques cannot be combined. Therefore the performance of a combination of the two technique was tested as part of this contract.

Figure 2.14 shows a combination of Markel's and Pfeifer's techniques. The combined system was implemented by averaging blocks of frames to form recognition features, and then performing sequential analysis on the sequence of winners produce by the distance metric.

Two experiments were done with the combination system; one using 10 second unknowns and the other using 20 second unknowns. The voiced speech subpopulation and the individual covariance models were used for both experiments.

COMPARISON OF MARKEL'S AND PFEIFER'S TECHNIQUES

10 and 20 SECOND MODELS, INDIVIDUAL COVARIANCE MATRICES

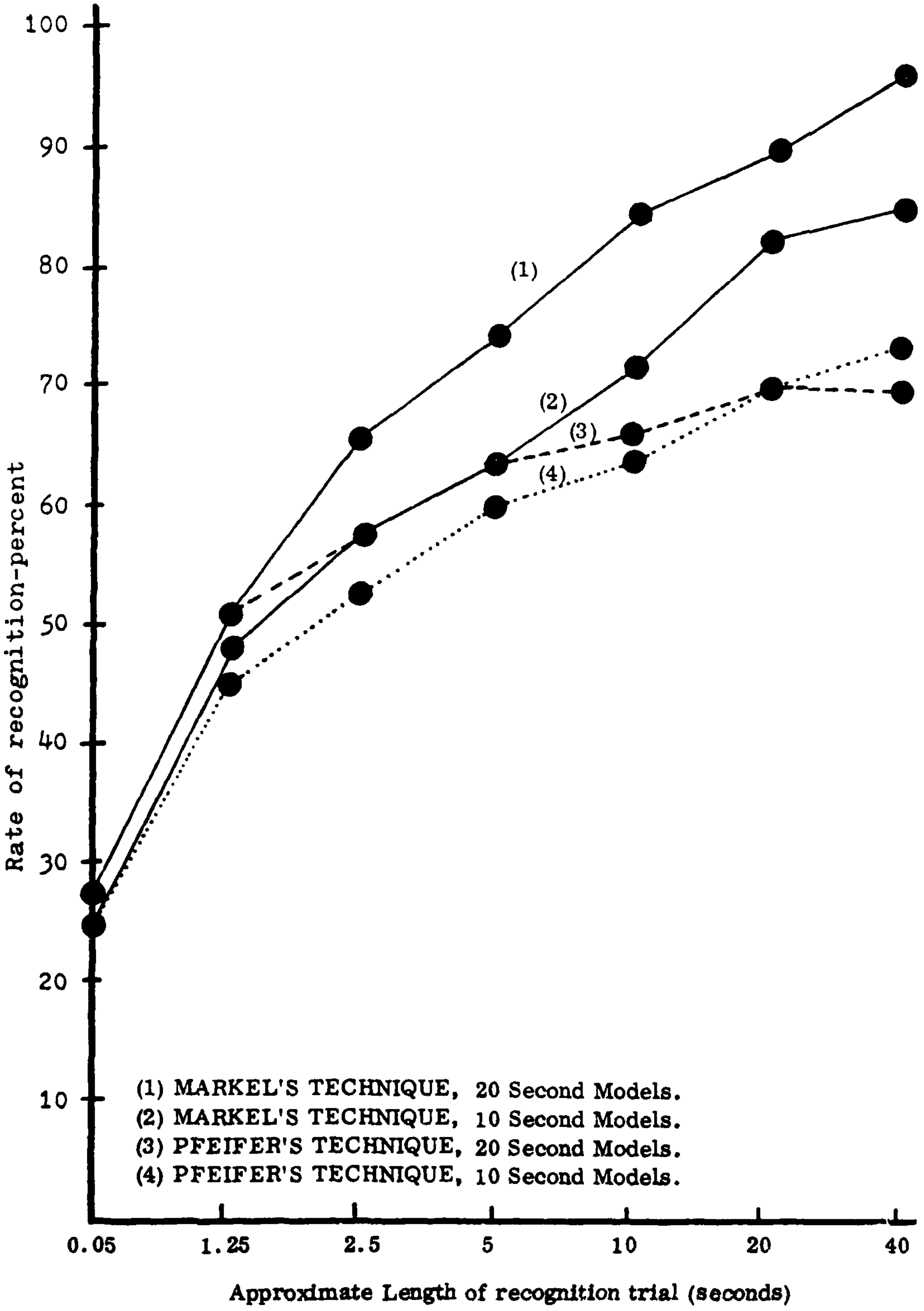


FIGURE 2.13

A COMBINATION OF MARKEL'S AND PFEIFER'S SPEAKER AUTHENTICATION TECHNIQUES

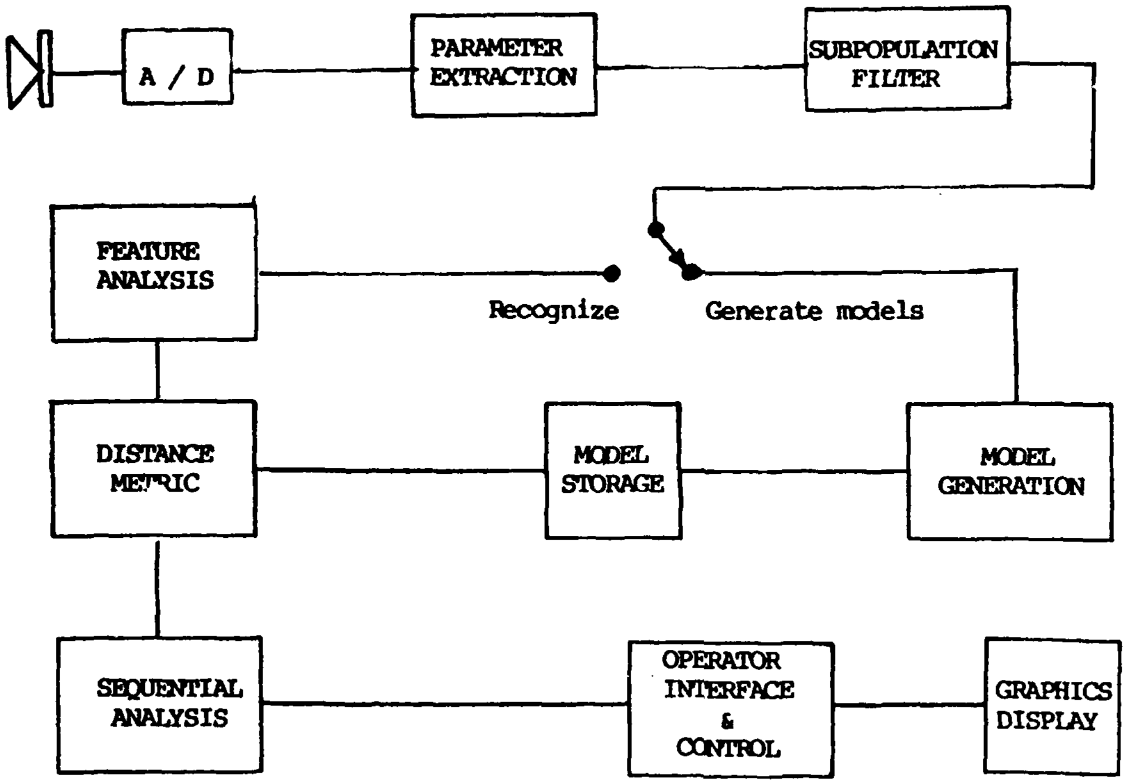


FIGURE 2.14

The first experiment was conducted with 10 second unknowns, and involved processing the data in the following five ways.

1. Blocks of one frame were averaged before the distance metric, and sequential analysis was done on 200 blocks. This is Pfeifer's Technique.

2. Blocks of 25 frames were averaged before the distance metric, and sequential analysis was done on eight blocks.

3. Blocks of 50 frames were averaged before the distance metric, and sequential analysis was done on four blocks.

4. Blocks of 100 frames were averaged before the distance metric, and sequential analysis was done on two blocks.

5. Blocks of 200 frames were averaged before the distance metric, and sequential analsis was done on one block. This is Markel's technique.

The results of the experiment for 10 second unknowns are shown in figure 2.15. The results generally indicate that the more averaging that is done prior to the distance metric, the better the system performs. The best performance was obtained for 200 frame averages, and majority voting on one block (Markel's technique).

The second experiment used 20 second unknowns. For this experiment, the data was processed six ways, but always with 20 seconds of speech per recognition trial:

1. Blocks of one frame were averaged before the distance metric, and sequential analsis was done on 400 blocks. This is Pfeifer's technique

2. Blocks of 25 frames were averaged before the distance metric, and sequential analsis was done on 16 blocks.

COMBINATIONS OF MARKEL'S AND PFEIFER'S TECHNIQUES

200 FRAME (10 SECONDS) RECOGNITION TRIALS

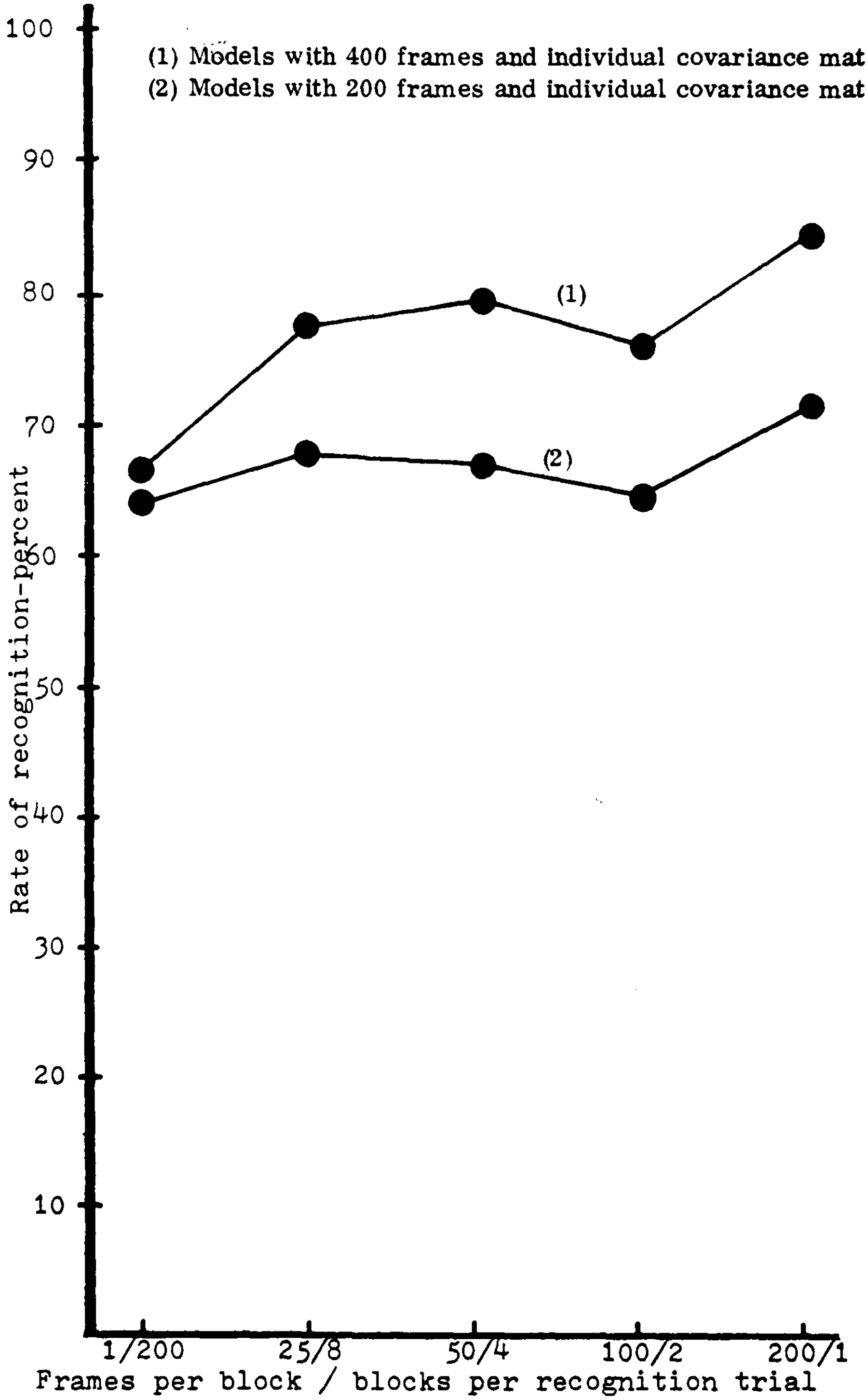


FIGURE 2.15

3. Blocks of 50 frames were averaged before the distance metric, and sequential analsis was done on eight blocks.

4. Blocks of 100 frames were averaged before the distance metric, and sequential analsis was done on four blocks.

5. Blocks of 200 frames were averaged before the distance metric, and sequential analsis was done on two blocks.

6. Blocks of 400 frames were averaged before the distance metric, and sequential analsis was done on one block. This is Markel's technique.

The results of the experiment with 20 second unknowns are shown in figure 2.16. The same general pattern exists for the 20 second experiment as for the 10 second experiment. Again, the best performance is obtained for Markel's technique without sequential analysis.

2.9 SUMMARY OF THE ALGORITHM SELECTION STUDY

The major result of the algorithm selection study is that Markel's technique has been shown to perform with accuracies in excess of 95% for 17 talkers when used with the voiced speech subpopulation, and models generated from 20 seconds of speech. The accuracy is above 85% when 10 second models are used. The accuracy of Pfeifer's technique is consistently 10% to 15% less than that of Markel's technique when 10 and 20 second models are used. In addition, Markel's technique was shown to perform better than hybrid systems that combine the frame averaging of Markel' technique with the majority voting of Pfeifer's technique.

COMBINATIONS OF MARKEL'S AND PFEIFER'S TECHNIQUES

200 FRAME (20 SECONDS) RECOGNITION TRIALS

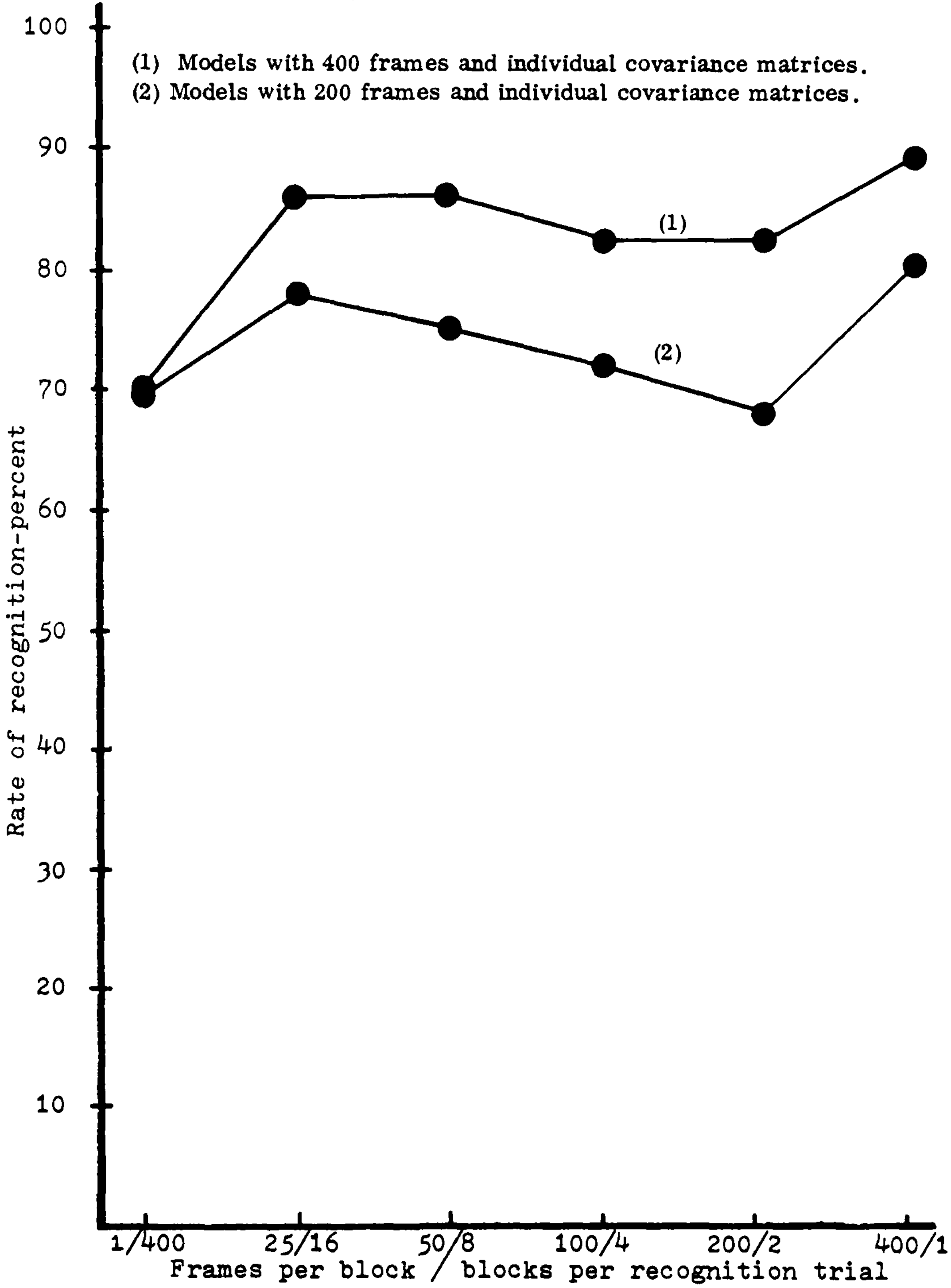


FIGURE 2.16

Other findings of the study are list below:

1. The voiced speech subpopulation yielded consistently better results than either the single vowel or multi-vowel subpopulation. This result is most likely due to the limited number of frames available for model generation when the vowel subpopulations are used. If more speech were available for generating models, the vowel subpopulation becomes more attractive. However, for this contract, the amount of speech for model generation is extremely limited, and therefore the voiced subpopulation was chosen for implementation in the demonstration system. There is also considerable computational savings in using the voiced subpopulation rather than the vowel subpopulation.

2. The choice of a pooled covariance matrix or the individual covariance matrices does not have a significant impact on the recognition performance when the models are generated from 10-20 seconds of speech, and the unknowns contain up to 40 seconds of speech. The value of pooled covariances lies in the limited amount of space that is required for storage. A mean vector must be stored for each model, but only one covariance matrix is stored. In a dedicated system where memory may be limited, this saved space may be significant.

3. The use of individual frames as input to the the model generator was shown to produce better recognition results than the use of averaged frames in the model generation. This is most likely due to the improved estimate of the covariance that is obtained when a larger number of input frames are available.

Based on the results of the algorithm selection study, it was determined that a demonstration system sould be implemented using Markel's technique that could obtain high recognition accuracies when used with short speech segments for both the models and the unknowns. The implementation and testing of this speaker authentication demonstration system is discussed in chapter 3.

CHAPTER 3: LABORATORY DEMONSTRATION SYSTEM

3.1 INTRODUCTION

A primary goal of this study was to develop a speaker recognition demonstration system. The realtime capability of the ITT speaker recognition test bed provided the foundation for the required demonstration system, and therefore ITT was able to develop a realtime laboratory demonstation system. Under this contract an improved operator interface was added to the existing realtime recognition system. This interface is engineered so that an untrained operator can generate and document new models and perform recognition in realtime with minimal instruction. In addition, other added features provided the operator with the capability to generate graphic hard copy for each recognition trial, archive and retrieve models, and display pertinent information on each model in the system.

The realtime demonstration system was implemented with the hardware shown in figure 3.1. The system uses a PDP-11/60 for the operator interface, control, and display formating. An ITT developed high speed signal processor, the Quintrell RAM, is used for the computationaly intensive calculations required for realtime recognition. The remainder of this chapter discusses the details of the realtime demonstration system implementation, and the system performance.

3.2 SPEAKER RECOGNITION ALGORITHM

The algorithm selected for implementation in the demonstration system is an extension of the technique reported by Markel (section 2.4). As discussed in section 2.6, this algorithm performed better in all tests than the technique reported by Pfeifer. A diagram detailing the functional blocks of the algorithm is shown in figure 3.2. This section discusses the function of each block.

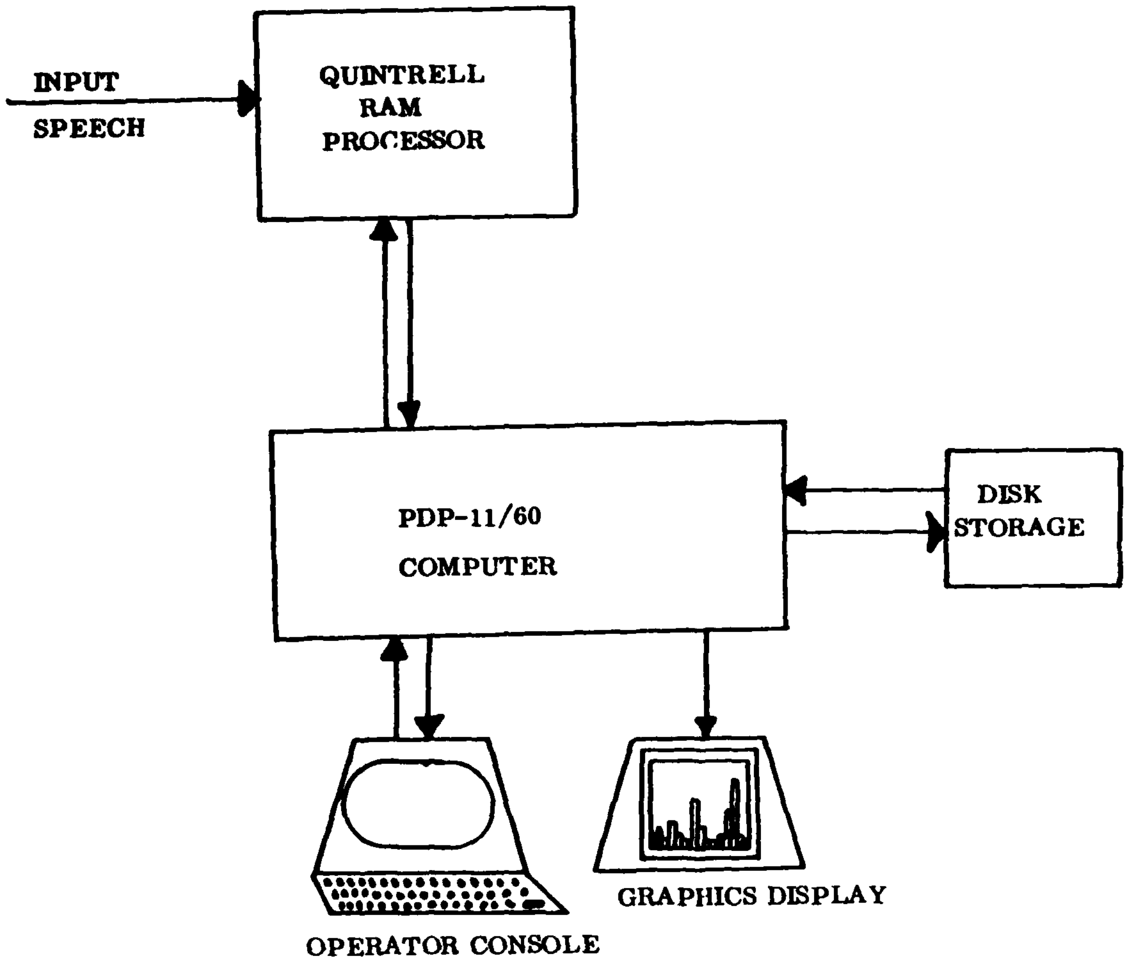


Figure 3.1: REALTIME LABORATORY SPEAKER RECOGNITION SYSTEM

MARKEL'S SPEAKER AUTHENTICATION TECHNIQUE

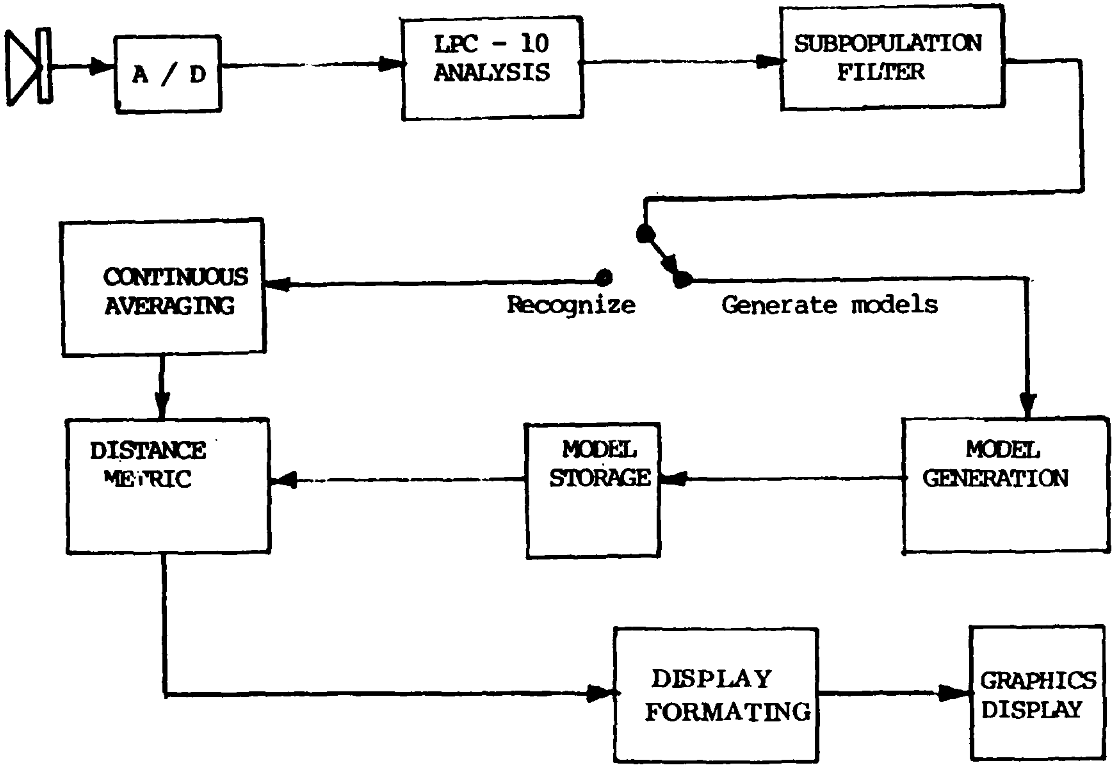


FIGURE 3.2

After the analog signal is digitized using a 12 bit A to D converter, an LPC-10 analysis is performed. The LPC-10 reflection coefficients were chosen as the speech analysis parameters for this system because of previous work at ITTDCD. An earlier study indicated that reflection coefficients performed superior to other LPC derived parameters, and were also superior to spectral and cepstral parameters.

The next function in the algorithm is the subpopulation filter. The subpopulation chosen for the demonstration system is that of all voiced speech. The other subpopulation that was considered is the vowel subpopulation. During the algorithm selection phase of the contract, these two subpopulations were evaluated, and the performance of the all voiced speech subpopulation was superior to the vowel subpopulation in all tests.

As indicated in figure 3.2 the remainder of the algorithm is split into two modes, one for recognition, and one for model generation. Since reference models are required before performing recognition, the model generation mode is discussed first.

The reference models used in the demonstration system consist of the mean vector and the individual covariance matrix for the reflection coefficients. The actual amount of speech used for the generation of the reference models is determined by the operator. However, the amount of speech must be large enough so that the resulting covariance matrix can be inverted. For a 10 X 10 matrix, 100 voiced input frames (approximately 3-5 seconds) are adequate to insure invertability. After the reference models are generated they are stored for comparison with the unknown speech during the distance calculation of the recognition mode.

In the recognition mode the output of the subpopulation filter is subjected to a continuous or running average calculation. The running averaging technique was chosen to incorporate the advantages of the sequential analysis decision process used in Pfeifer's original technique. Because of the interactive nature of the system, it is advantageous to present the operator with partial results as the recognition proceeds. By use of the running average (calculating the average at each point in time as opposed to averaging over a fixed block length), the recognition time is not limited to a fixed interval. Instead, the recognition process can continue until all the input speech has been processed, or until the operator determines that no further data is required to identify the speaker.

The next functional block in figure 3.2 is the distance metric. The "distance" between the average of the voiced reflection coefficients and the set of stored reference models is computed and used to derive a similarity score between the unknown talker and each reference speaker in the system. The distance metric used in the demonstration system is the Mahlanobis metric (Eq 2.1) which is a Euclidean metric weighted by the inverse covariance matrix of the reflection coefficients.

The distances calculated by the distance metric are further processed by the display formating function. Two display modes are available. The first displays a similarity score between the unknown talker and the models based on the inverse distance. The similarity score is defined in equation 3.1 below.

$$\text{Similarity Score} = \text{Const} / \text{Distance} \qquad \text{Eq. 3.1}$$
$$\text{for Distance} > \text{Const}/100,$$

and

$$\text{Similarity Score} = 100$$
$$\text{for Distance} < \text{Const}/100$$

where Const is a scale factor determined experimentally.

This function produces high similarity scores for reference models with a small distance to the unknown, and small scores for these with large distances. The similarity score is always less than or equal to 100.

The similarity score can be plotted for the operator on the graphics display as shown in figure 3.3. The similarity score for each model is displayed as a bar above the corresponding model name. In addition, the confidence score (defined below) for the top three models is displayed

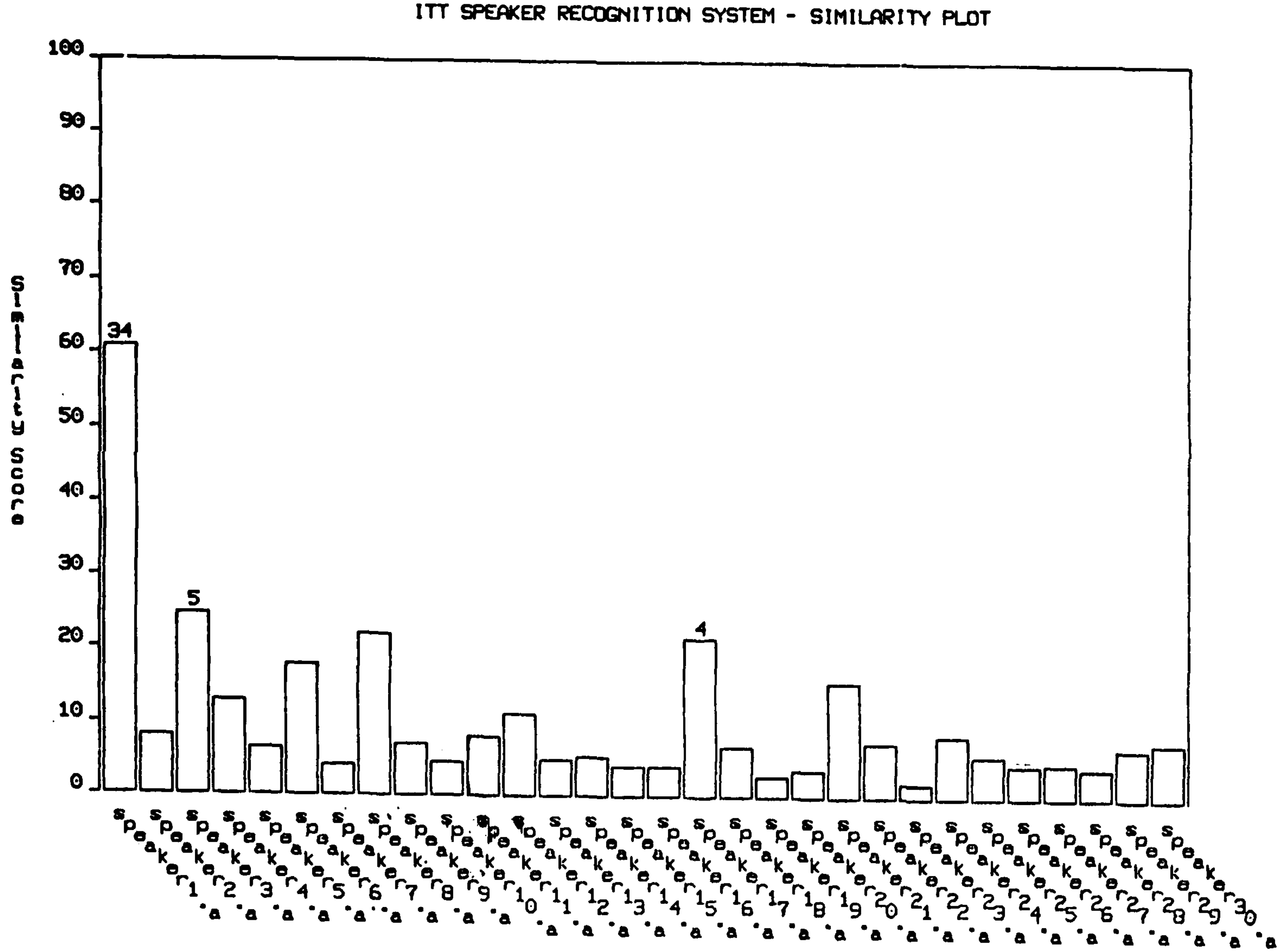


Figure 3.3 Test using 14 seconds of speech from speaker1

above the corresponding bars. The similarity score and the confidence score are continuously computed and used to update the display whenever new input speech is available. The number of frames used in the current recognition trial is continuously updated and displayed on the operator console.

The second display mode is a confidence measure display. This is a attempt to display for each reference model a value between 0 and 100 that represents the confidence that the unknown talker corresponds to that reference model. The confidence measure for the i^{th} talker is defined in equation 3.2 below.

Eq. 3.2

$$C_i = S_i \cdot \frac{F \cdot S_i}{\{ \underset{m=1}{\overset{N}{\text{SUM}}} [S_m]^2 \}^{1/2}}$$

where

F = Frame Count/500 for Frame Count < 500,

$F = 1$ otherwise,

S_i = Similarity Score for the i^{th} talker,

and

N = the number of models.

The confidence measure uses the product of three parameters to estimate the confidence: the similarity score, the number of frames processed at the current time (Frame Count), and the ratio of the model's score to the total RMS score for all models. The similarity score is a measure of the goodness of the match between the current averaged reflection coefficients and the model. The "number of frames" parameter is scaled so that it increases linearly from zero to one as the number of frames used for the recognition increases from zero to five-hundred(approximately 20 seconds of speech). This parameter is used to indicate that the confidence in a recognition increases with longer unknown speech samples. The final parameter, model score over total RMS score for all models, weights the confidence bv the a factor that indicates how large the model's score is compared to the scores for all the other models. The RMS measure of total score is used to weight the model scores closest to the maximum model score more heavilv than the scores of the non-contending models. Fiqure 3.4 presents the confidence displav for the same data shown in figure 3.3. The confidence factor for each model is displayed as a bar above the corresponding model name. As with the similarity score plot, the confidence scores are continuously updated and displayed.

3.3 DEMONSTRATION SYSTEM IMPLEMENTATION

The speaker recognition laboratory demonstration system is implemented using a PDP-11/60 as the system controller and an ITT developed, high speed signal processor, the Quintrell RAM, for the computationaly intensive realtime processing. A block diagram of the system is shown in figure 3.5. All operator interaction with the system is done through the PDP-11. The control program in the PDP then directs the Quintrell to perform the appropriate tasks. In addition, the PDP-11 is used for such functions as display formating, plotting and hard copy. The Quintrell is used for the processing that must be done in realtime, such as the LPC-10 analysis and the continuous averaging. A brief description of the Quintrell RAM is given in table 3.1 below.

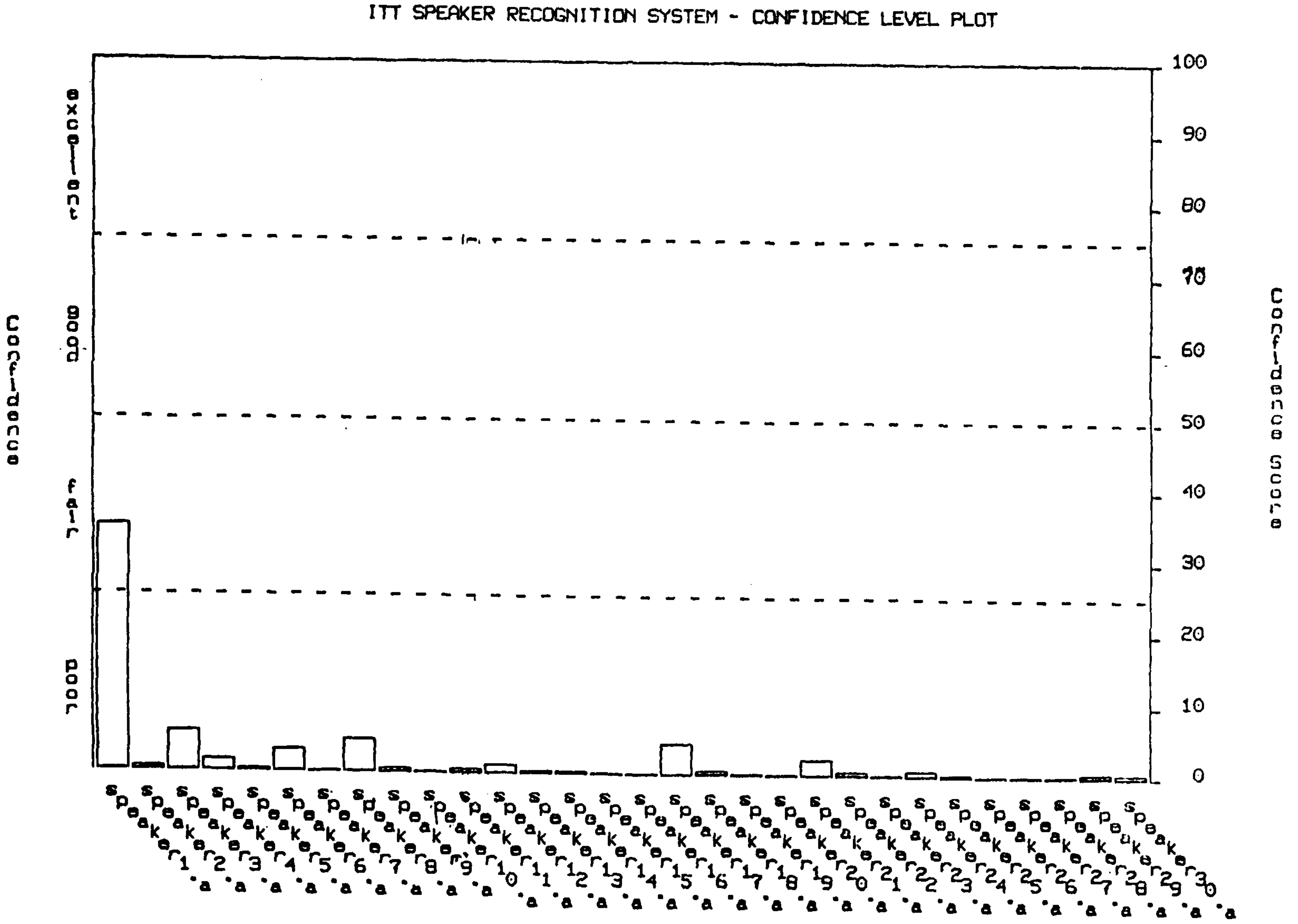


Figure 3.4 Test using 14 seconds of speech from speaker1

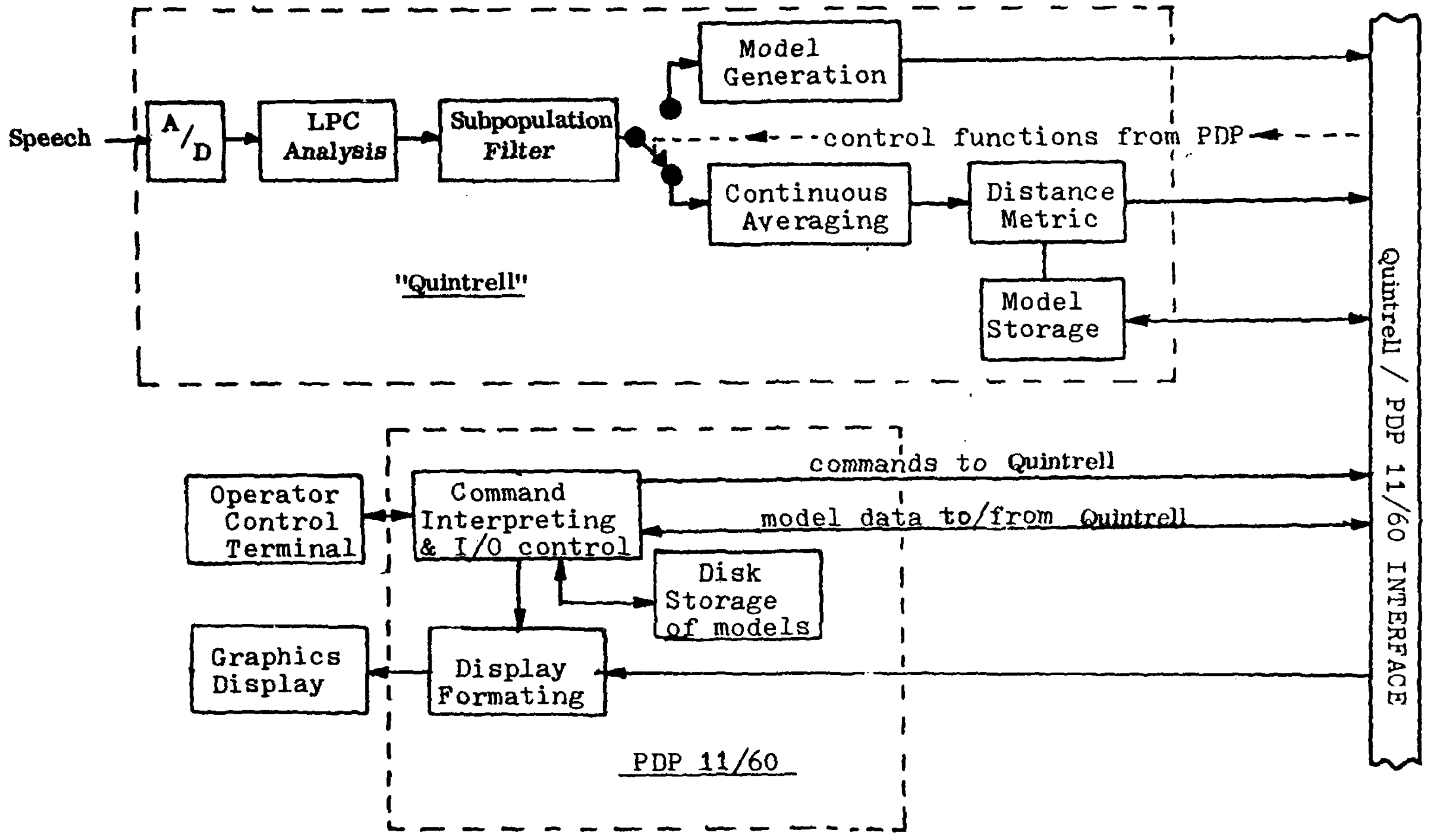


Figure 3.5 LABORATORY DEMONSTRATION SYSTEM BLOCK DIAGRAM

Table 3.1 QUINTRELL PROCESSOR RAM MODEL SPECIFICATIONS

Data Memory (16 bit words)		12K
Program Memory (16 bit words)		12K
Microprogram Memory (52 bit words)		1K
Micro Cycle (nanoseconds)		225
Data Memory Access Cycle (nanoseconds)		450
Multiply & Accumulate 32 bit product (nanoseconds)		900
Data Processor	AMD	2901
Address Processor	AMD	2901

3.3.1 Operator Interface / PDP-11 Software

The operator interface was designed to be extremely easy to use so that an untrained operator could use the system with little or no instruction. The system can be operated in any of four modes; the command mode, the recognition mode, the model generation mode, and the archive mode. The operator is prompted by the system in each mode with all the valid options available in the current mode. A list of the commands available to the operator for control of the laboratory demonstration system is shown in table 3.2. The four modes and the associated commands are discussed in the following paragraphs.

COMMAND MODE: When the demonstration system is started, the system is initially in the command mode. The user is prompted with the menu of available operations shown. When the operator enters the letter corresponding to a command, followed by a carriage return, the PDP-11 will execute the corresponding routine. If an invalid command is entered, the command mode menu will be redisplayed. If a command to change modes ('a','g', or 'r') is entered, the menu of available operations for the new mode will be displayed. The following paragraphs describe the functions in the command mode.

List Current Models: To list the current models (the models available for recognition) the operator enters an "l". The names of all the current models are then displayed on the operator console.

Display Speaker Data: In the command mode, the system will respond to

A. COMMAND MODE INSTRUCTIONS

a	- Enter archive mode
g	- Enter model generation mode
h	- Plot hardcopy of the current display
l	- List current models
l.	- List speaker data for current models
L	- Load models into RAM
m	- Retrieve models from RAM (diagnostic tool only)
p	- Print speaker data for current models
r	- Enter recognition mode

B. ARCHIVE MODE INSTRUCTIONS

g	- Get an archived model an add it to the current model list
h	- Make hard copy of the speaker data for the archived models
l	- List the archived models
l.	- Display the speaker data for the archived models
p	- Put a current model into the archive model list
"cr"	- Return to the command mode

C. RECOGNITION MODE INSTRUCTIONS

c	- Display confidence score plot
d	- Display similarity score plot
h	- Make hard copy of current display
s	- Stop recognition (return to command mode)
"cr"	- Clear recognition buffers

D. MODEL GENERATION MODE INSTRUCTIONS

"cr"	- Stop model generation (enter speaker data)

Table 3.2: OPERATOR INTERFACE COMMAND STRUCTURE

a "1." by displaying all information contained in the model files about each speaker. This information is originally entered by the operator at the time the model is generated (see "MODEL GENERATION MODE" below).

Print Speaker Data: The same information that is displayed on the operator console for the DISPLAY SPEAKER DATA command described above, is printed on the line printer in response to a "p" in the command mode. A typical output is shown in figure 3.6.

Make Hard Copy: The operator can make a hard copy of the current graphics display by entering an "h". This can also be done while in the archive or recognition mode.

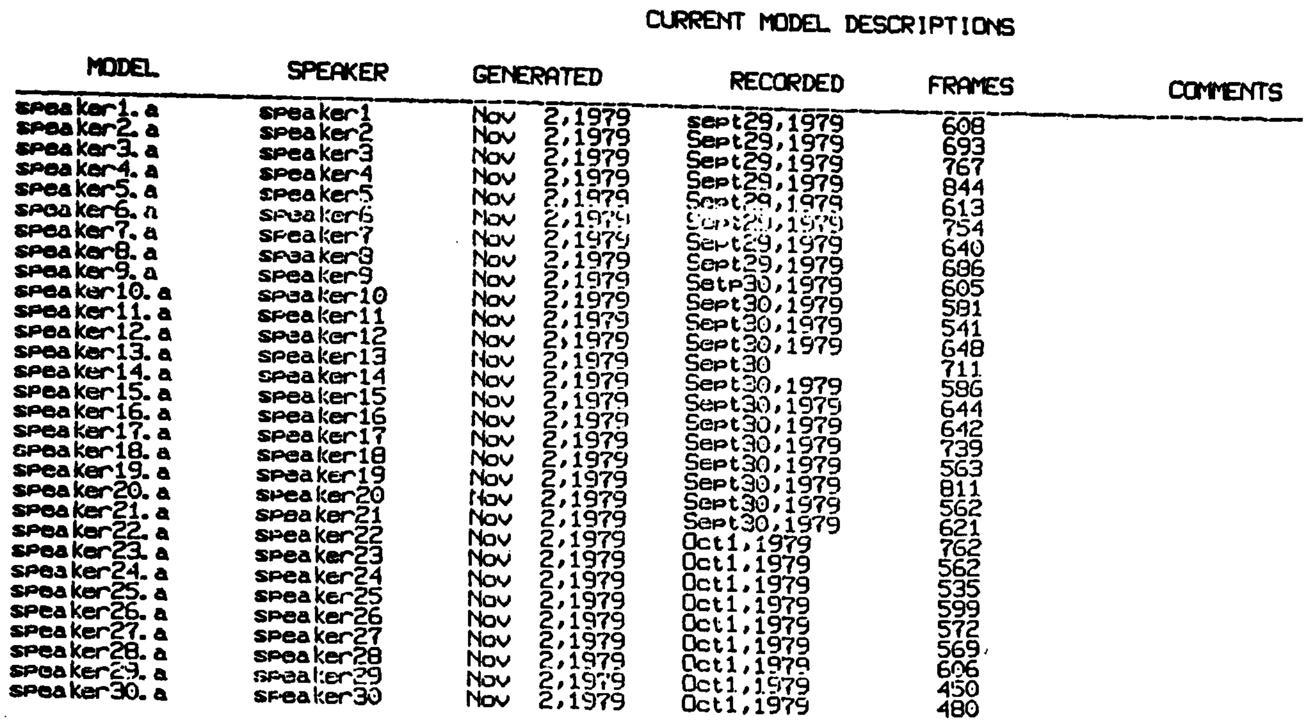
Loading Models: Models are loaded into the Quintrell from the PDP-11 in response to an "L" in the command mode. All the models in the current model list are transfered to the Quintrell. The name of each model is displayed on the operator console, along with the number of models loaded. If the number of models in the current model list exceeds 30, only the first 30 are loaded.

Enter Archive Mode: The letter "a" is typed in the command mode to enter the archive mode. The archive mode is described below.

Enter Recognition Mode: The letter "r" is typed in the command mode to enter the recognition mode. The recognition mode is described below.

Enter Model Generation Mode: The letter "g" is typed in the command mode to enter the model generation mode. The model generation mode is described below.

ARCHIVE MODE: The system as currently implemented divides the models into two lists, the current model list, and the archived model list. The current model list contains all the models that can be used for recognition at a given time. The archived model list provides storage for any models



```
                              CURRENT MODEL DESCRIPTIONS
   MODEL          SPEAKER      GENERATED        RECORDED      FRAMES         COMMENTS
--------------------------------------------------------------------------------------
speaker1.a      speaker1      Nov  2,1979   sept29,1979      608
speaker2.a      speaker2      Nov  2,1979   Sept29,1979      693
speaker3.a      speaker3      Nov  2,1979   Sept29,1979      767
speaker4.a      speaker4      Nov  2,1979   Sept29,1979      844
speaker5.a      speaker5      Nov  2,1979   Sept29,1979      613
speaker6.a      speaker6      Nov  2,1979   Sept29,1979      754
speaker7.a      speaker7      Nov  2,1979   Sept29,1979      640
speaker8.a      speaker8      Nov  2,1979   Sept29,1979      686
speaker9.a      speaker9      Nov  2,1979   Sept29,1979      605
speaker10.a     speaker10     Nov  2,1979   SetP30,1979      581
speaker11.a     speaker11     Nov  2,1979   Sept30,1979      541
speaker12.a     speaker12     Nov  2,1979   Sept30,1979      648
speaker13.a     speaker13     Nov  2,1979   Sept30           711
speaker14.a     speaker14     Nov  2,1979   Sept30,1979      586
speaker15.a     speaker15     Nov  2,1979   Sept30,1979      644
speaker16.a     speaker16     Nov  2,1979   Sept30,1979      642
speaker17.a     speaker17     Nov  2,1979   Sept30,1979      739
speaker18.a     speaker18     Nov  2,1979   Sept30,1979      563
speaker19.a     speaker19     Nov  2,1979   Sept30,1979      811
speaker20.a     speaker20     Nov  2,1979   Sept30,1979      562
speaker21.a     speaker21     Nov  2,1979   Sept30,1979      621
speaker22.a     speaker22     Nov  2,1979   Oct1,1979        762
speaker23.a     speaker23     Nov  2,1979   Oct1,1979        562
speaker24.a     speaker24     Nov  2,1979   Oct1,1979        535
speaker25.a     speaker25     Nov  2,1979   Oct1,1979        599
speaker26.a     speaker26     Nov  2,1979   Oct1,1979        572
speaker27.a     speaker27     Nov  2,1979   Oct1,1979        569
speaker28.a     speaker28     Nov  2,1979   Oct1,1979        606
speaker29.a     speaker29     Nov  2,1979   Oct1,1979        450
speaker30.a     speaker30     Nov  2,1979   Oct1,1979        480
```

Figure 3.6 Example of Speaker data print out

that are not required for the current recognition task. Models may be moved from one list to the other by entering the archive mode. To enter the archive mode, the operator types an "a" in the command mode, and the system responds by displaying the commands available. These commands are listed in table 3.2B.

List Archived Models: In the archive mode, the operator may obtain a listing of the archived models currently in the system by entering a "l".

Display Speaker Data: In the archive mode, the system responds to an "l." by displaying the speaker data that was entered when the model was generated (see model generation mode below).

Print Speaker Data: The same speaker data that is displayed by the "l." command can be obtained in hard copy by entering an "h" in the archive mode.

Get Archived Model: To retrieve an archived model and place it in the current model list, the operator enters a "g". The system then asks for the model name to be entered on the operator console. If the name is correctly entered, the model will be moved to the current model list. If not, the system will respond with "can not move file ....." and prompt the operator to enter the command again.

Archive Model: To put model in the current model list into the archive list, the operator enters a "p". The system then asks for the model name, and proceeds as in the "g" command outlined above.

Return to Command Mode: To return to the command mode from the archive mode, the operator enters a carriage return.

RECOGNITION MODE The recognition mode is entered when the operator types an "r" in the command mode. If no models have been loaded into the Quintrell in the current recognition session, the system responds with "No Models Loaded" and returns to the command mode. If the models have been loaded, the system enters the recognition mode and displays the available commands. These commands are listed in table 3.2.C and are discussed below.

Clear Recognition Buffers: A carriage return ("cr") in the recognition mode causes the system to clear the Quintrell buffers that store statistics for the current unknown speech. These buffers should be cleared by the operator any time a new recognition trial is begun or when a change in talker is suspected. The number of frames used in each recognition trial is continuously updated on the operator console, and this frame count is set to zero whenever the recognition buffers are cleared.

Display Similarity Score Plot: As described above, the system calculates two scores, the Similarity Score, and the Confidence Score. The graphics display default condition is the Similarity Score plot as shown in figure 3.3. The similarity score plot can also be selected by typing a "d" in the recognition mode.

Display Confidence Plot: The second display available in the recognition mode is the Confidence Plot as shown in figure 3.4. The Confidence Plot is selected by typing "c" in the recognition mode.

Make Hard Copy: A hard copy of the current display (either similarity or confidence plot) can be obtained by entering an "h" while operating in the recognition mode.

Stop Recognition: To exit from the recognition mode and return to the command mode, the operator enters an "s".

MODEL GENERATION MODE: The model generation mode is entered in response to the 'g' (generate model) command in the "command mode". The operator console will display:

Model being generated, hit 'cr' to stop >

FRAME COUNT ______

The PDP-11 then sends a command to the Quintrell to start model generation. The PDP-11 continually reads the number of voice frames used in the model and displays it as the frame count on the operator console. The Quintrell continues to update the model statistics and return the current voiced frame count until the operator enters a carriage return. The system then prompts the operator to supply the necessary information to document the model. This includes the speaker name, the date of the recording, and any comments the operator has about the speaker or the model. The system produces a model name for each model that is generated. The model name is the speaker name followed by a ".a" or ".b" etc up to ".z". The system will check to see how many models exist for the speaker, and append the next available letter to form the model name. With this naming convention, every speaker can have up to 26 models in the system. The capability to have more than one model for a given speaker allows the operator to generate models under various noise and channel conditions to improve recognition performance. As soon as the model data has been entered, the system returns to the command mode.

3.3.2 Quintrell RAM Software

A stated earlier, the PDP-11 is used only for operator interface functions and display formating. All of the required signal processing for the speaker recognition system is performed in the Quintrell RAM. The Quintrell can be programed at both the micro-instruction level, and at a macro level. For the speaker recognition system, the machine is programmed entirely at the macro level, using a standard assembly language instruction set.

The Quintrell portion of the speaker recognition system can operate in two modes as directed by the PDP-11. The first is the "recognition mode" where the input speech is processed, compared to the models stored in the Quintrell, and the distances returned to the PDP-11. The second mode is the "model generation mode" where the input speech is analyzed, and the parameters required to generate a model (the means and the covariance matrix for the reflection coefficients) are accumulated. This model generation data can then be transmitted to the PDP-11 on command.

All of the processes in the Quintrell are controlled by a program refered to as the RAM process scheduler. The scheduler receives commands from the PDP-11 to execute various processes. The scheduler can be directed by the PDP-11 to:

load models from the PDP-11,
enter/exit recognition mode,
reset recognition mode (clear accumulated statistics),
enter/exit model generation mode,
transmit the accumulated model statistics to the PDP-11,
return models to the PDP-11 (diagnostic use only).

The Quintrell programs for the demonstration system operate on two levels. The time critical portions of the code that must be executed every analysis frame, such as the LPC analysis and continuous averaging, run as foreground processes. The remainder of the programs, including the RAM process scheduler, are run as background processes whenever the foreground process is idle. The next two sections discuss in detail the programs for the recognition mode and the model generation mode in the Quintrell.

3.3.2.1 Recognition Mode Processing in the Quintrell

The flow diagram for the Quintrell recognition mode processing is shown in figure 3.7. When the Quintrell process scheduler receives the command to enter the recognition mode, two actions are initiated. First, the interrupts are enabled to allow the LPC-10 analysis to be executed as a

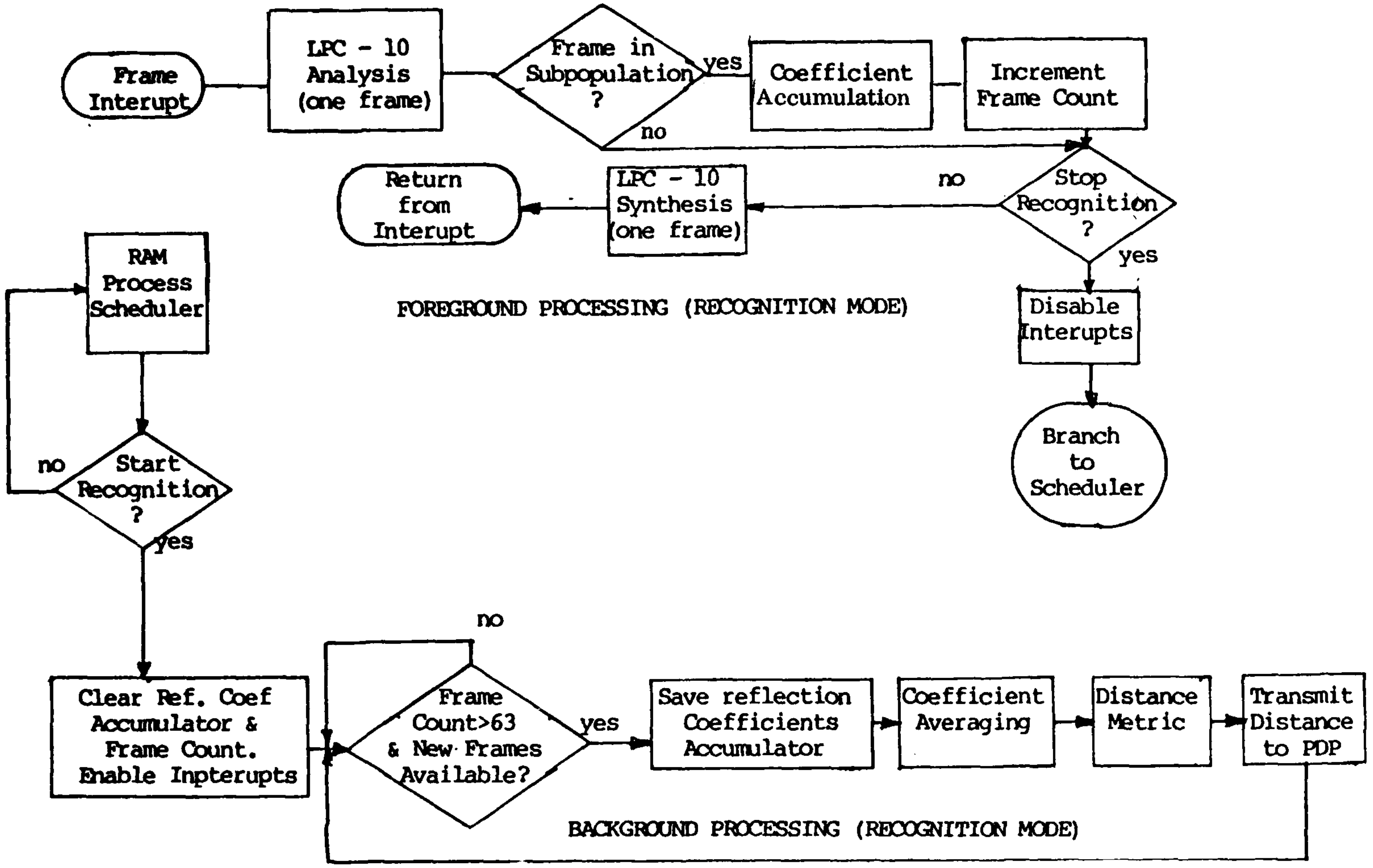

Frame Interupt
LPC - 10 Analysis (one frame)
Frame in Subpopulation ?
yes
no
Coefficient Accumulation
Increment Frame Count
Stop Recognition ?
no
yes
LPC - 10 Synthesis (one frame)
Return from Interupt
Disable Interupts
Branch to Scheduler
FOREGROUND PROCESSING (RECOGNITION MODE)
RAM Process Scheduler
Start Recognition ?
no
yes
Clear Ref. Coef Accumulator & Frame Count. Enable Inpterupts
Frame Count>63 & New Frames Available?
no
yes
Save reflection Coefficients Accumulator
Coefficient Averaging
Distance Metric
Transmit Distance to PDP
BACKGROUND PROCESSING (RECOGNITION MODE)

Figure 3.7 RAM RECOGNITION MODE PROCESSING FLOW

foreground process. Second, the frame monitoring program is begun as a background process. The operation of the various programs during recognition is described in the following paragraphs.

LPC-10 ANALYSIS: The LPC-10 analysis is performed using a modified Atal algorithm developed under a previous contract. The analysis period is 22.5 ms. The algorithm is pitch synchronous in that the analysis windows for sequential frames are separated by a integral multiple of the pitch period during voiced segments. The program runs as a foreground process, and is initiated by an interrupt that is generated by the Quintrell every frame. Ten LPC reflection coefficients, pitch, and power are calculated using the current digitized speech from the Quintrell's A/D converter. The input speech is bandlimited from 300 to 3600 Hz. The reflection coefficients are saved in the Quintrell's memory for further processing.

SUBPOPULATION FILTERING: The second foreground process to run each analysis frame is the subpopulation filter. As discussed in section 3.2, the all voiced speech subpopulation is used in the demonstration system. The voicing decision is made as part of the pitch tracking algorithm in the LPC-10 analysis routines. All unvoiced frames are marked by setting the pitch parameter equal to zero.

Originally, the subpopulation filter for voiced speech simply retained those frames with non-zero pitch. However, when the system was tested using the same input speech over and over, the results of the recognition process showed considerable variation. In particular, the number of frames identified as voiced frames varied by as much as 20%. This problem was investigated by computing histograms of pitch and power for each frame passed by the subpopulation filter. By looking at the distributions of the pitch for several recognition trials with the same input speech, it was determined that large numbers of very low power frames were sometimes labeled as voiced, and sometimes as unvoiced. The problem was corrected by using a power threshold in the subpopulation filter, so that low power frames are discarded independent of whether they are voiced or unvoiced. The performance of the recognition system was improved with the addition of this threshold, and the results were much more consistent from one trial to the next.

COEFFICIENT ACCUMULATION: The next foreground process to be executed is the coefficient accumulation routine. This program is used to store all the information required to calculate the mean reflection coefficient vector for the unknown talker at the present point in time. The accumulation program is called every analysis period when a voiced frame is detected. Each input vector is added, coefficient by coefficient, to a vector accumulator, and the frame count is incremented by one. The input reflection coefficients and the frame count register are 16 bit signed integers, and the accumulated sums are stored as 32 bit double precision integers. As a result, the continuous averaging can accumulate 16,383 voiced frames, which corresponds to approximately 12 minutes of speech. Since recognition can be accomplished in approximately 10 to 20 seconds, 12 minutes was considered more than adequate for all applications envisioned. The results in the vector accumulator are available for use by the coefficient averaging program operating in the background.

COMMAND CHECKING: After the completion of the coefficient accumulation routine, a program is executed to test the PDP-11 interface for new commands. The commands that are allowed at this point are "clear" or "stop recognition". If a clear command is received, the frame count is reset to zero, and the coefficient accumulator is cleared. The stop recognition command turns off the foreground process by disabling the interrupts, and returns control to the RAM process scheduler.

LPC-10 SYNTHESIS: If a stop recognition command is not received, the recognition mode processing continues. The next foreground program to be run is the LPC-10 synthesis. This program is not necessary for the operation of the speaker recognition system, but it is useful to be able to hear the synthesized speech to verify the correct operation of the analysis program, and to adjust the analog input level to the A/D converter.

The synthesis program is the last foreground routine run during each analysis frame. After the synthesis routine has completed, the background level programs resume execution until an interrupt signals the beginning of the next analysis frame period, and control is returned to the foreground routines. The background processes are discussed in the following paragraphs in the order they are executed.

FRAME MONITORING: The first background process is the frame monitoring routine. It determines whether more than 64 frames have been accumulated, and whether new frames have been accumulated since the last time the coefficient averaging program was executed. If both these conditions are satisfied, the programs are executed to calculate the coefficient averages and the distance metric, and the distances are then transmitted to the PDP-11. While these routines may require more than one frame time to complete execution if the number of models in the system is large, no timing problem occurs since the foreground processes will continue to update the coefficient accumulator. The only effect is that the display will not be updated on every frame. This is not a limitation, however, since the operator can not react to information that is presented faster than about once a second.

COEFFICIENT AVERAGING: The coefficient averaging program is executed whenever the conditions required by the frame monitor routines are satisfied. The function of this program is to calculate the average reflection coefficient vector at the current point in time. The first step is to make a copy of the current accumulator buffer and frame count. This step is taken to prevent the values from being changed by the foreground processes during the remainder of the background calculations. The coefficient averages are then calculated by dividing the accumulated coefficients sums by the frame count. The resulting average coefficient vector is then passed to the distance routine.

DISTANCE METRIC CALCULATION: The distance routing is next called to evaluate the Mahlanobis distance metric for each model in the system. The Mahlanobis distance metric is defined as:

$$D = (\overline{F} - \overline{M})\ [W]^{-1}\ (\overline{F} - \overline{M}) \qquad \text{Eq. 3.3}$$

where $\overline{F}$ is the average coefficient vector,
$\overline{M}$ is the mean vector from a model,
and $[W]^{-1}$ is the inverse covariance matrix from a model.

The matrix multiplies required to compute the metric are performed as a series of dot products. The dot product is a standard macro instruction in the Quintrell, and produces a 32 bit result. The scaling for the distance metric is important to avoid loss of precision in the results. The scaling used in the Quintrell implementation of the metric calculation is as follows:

1. The model mean vector, $\overline{M}$, and the input feature vector, $\overline{F}$ are scaled so a reflection coefficient of one is represented as $2^{14} - 1$.

2. The model inverse covariance matrix is scaled so the maximum element is equal to $2^{15} - 1$.

3. The difference vector, $(\overline{F} - \overline{M})$, is then scaled to $2^{12} - 1$.

4. The elements resulting from the first matrix multiply are divided by 2^{16}, and therefore are less than $10 * 2^{15} * 2^{12} * 2^{-16} < 2^{15}$. Thus no overflow is possible.

5. The result of the final matrix multiply is limited to $2^{15} * 2^{12} * 10 < 2^{31}$. This result is converted to a normalized floating point number with a 16 bit mantissa and 16 bit exponent.

The distance metric routine returns the distance from the unknown feature vector to each model in the system. The resulting distances are then transmitted to the PDP-11.

DISTANCE TRANSMISSION: The last background routine is the data transmission program. This routine sends the distance for each model to the PDP-11 for display. The data is transmitted in 64 character bursts and then the process waits for an acknowledge from the PDP. When the acknowledge is received, the next 64 characters are sent, and process is repeated until all the data is transmitted. After the data has been sent to the PDP-11, the frame monitoring routine is called, and all the background routines are executed again. This sequence continues until a stop recognition command is received, and control is returned to the RAM process scheduler.

3.3.2.2 Model Generation Mode Processing in the Quintrell RAM

When the Quintrell process scheduler receives the command to enter the model generation mode, two actions similar to the recognition mode are initiated. First, the interrupts are enabled to allow the LPC-10 analysis to be executed as a foreground process. Second, the frame count transmit program is begun as a background process. Figure 3.8 shows the flow diagram for model generation. The operation of the various programs during the model generation mode are described in the following paragraphs.

LPC-10 ANALYSIS: The LPC-10 analysis program is executed as the first foreground process in response to the frame interrupt. This is the same routine that is called in the recognition mode.

SUBPOPULATION FILTERING: The subpopulation filter is executed next exactly the same as in the recognition mode.

COEFFICIENT ACCUMULATION: As in the recognition mode, the next foreground process is the coefficient accumulation routine, which calculates the running sum of the reflection coefficients.

CROSS PRODUCT ACCUMULATION: The only difference in the foreground processing between the recognition mode and the model generation mode is the execution of one additional routine in the modeling mode. The cross product accumulation routine calculates the running sum of the cross products of the reflection coefficients. The double precision cross products are accumulated in a triple precision accumulator for the duration of the model generation. The cross product accumulator can accumulate 16 thousand voiced frames, so that models can be generated from up to

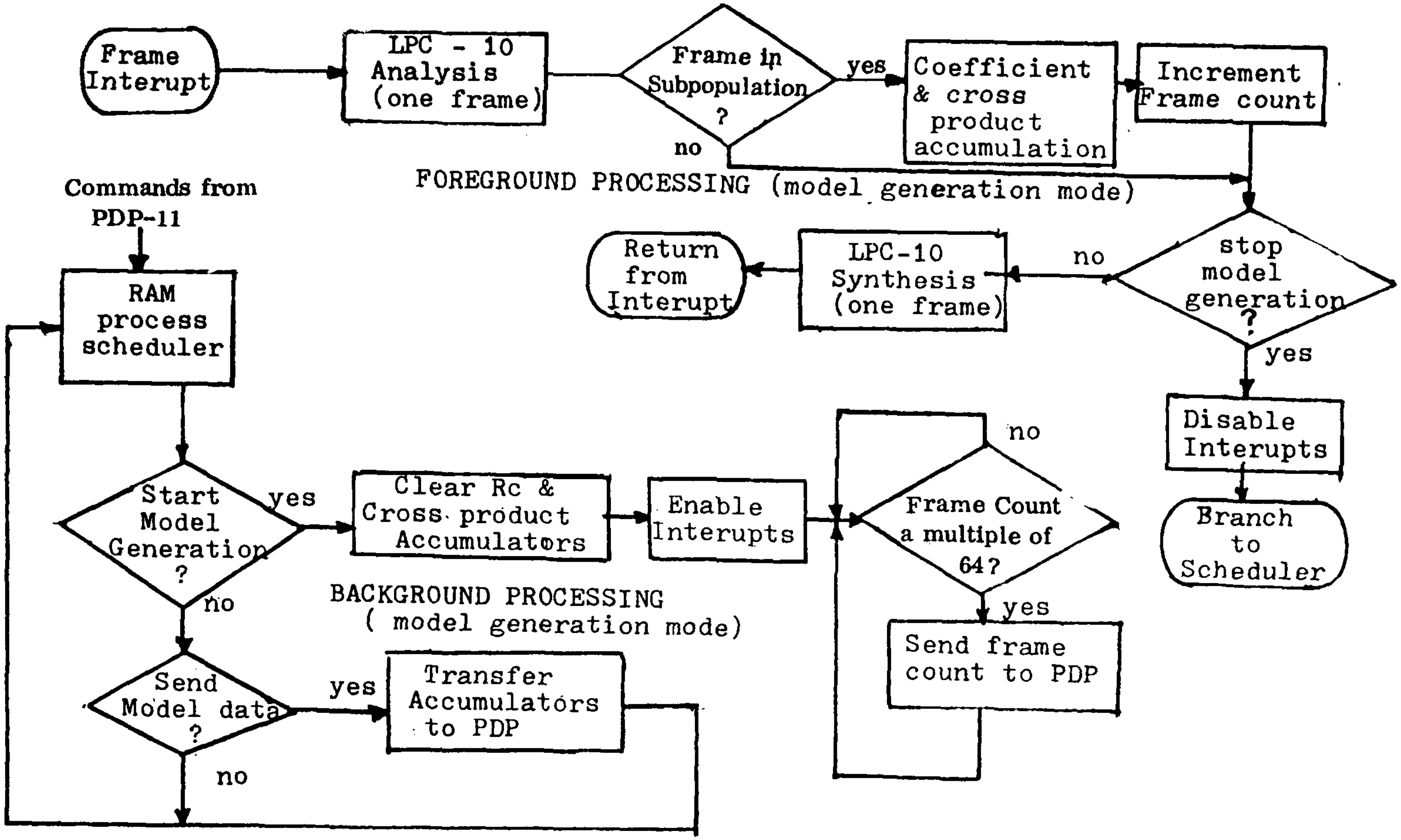


Figure 3.8 "RAM" MODEL GENERATION MODE PROCESSING FLOW

approximately 12 minutes of speech. The majority of the models of interest in this study are limited to approximately 20 seconds.

COMMAND CHECKING: As in the recognition mode, the PDP-11 interface is checked for new commands. If a "stop model generation" command is received, the interrupts are disabled, and control returns to the RAM process scheduler.

LPC-10 SYNTHESIS: If a stop command is not received, the LPC-10 synthesis routine in called. As in the recognition mode, this program is not required, but is useful to verify proper operation of the analysis routines.

FRAME TRANSMITTING: The only background process that is run during the model generation is the frame transmitting program. This routine simply transmits the frame count to the PDP-11 so that the number of frames in the model can be displayed to the operator. The frame count is only transmitted in multiples of 64 so that the amount of data sent to the PDP can be kept to a minimum.

3.3.2.3 Other Quintrell RAM Programs.

In addition to recognition and model generation, there are several other programs that can be executed under the control of the RAM process scheduler. These programs are discussed briefly in the following paragraphs.

LOAD MODELS: The Quintrell can load reference models (mean vectors and covariance matrices for each speaker) from the PDP-11. The mean vectors are scaled so the magnitude of the largest mean for each speaker is between 8192 and 16383. This assumes that when the mean is subtracted from a similarly scaled reflection coefficient in the distance metric, the result will not overflow 16 bits. The inverse covariance matrices are scaled so the largest element in each matrix is exactly 215-1.

TRANSMIT ACCUMULATED MODEL STATISTICS: After the model generation mode has been run, the Quintrell can be directed to send the accumulated model statistics to the PDP. The statistics transmitted are the accumulated sum for each of the ten reflection coefficients(ten double precision integers),the accumulated sum for each of the 55 unique reflection coefficient cross product terms (55 triple precision integers), and the number of frames accumulated in generating the model (one single precision integer).

RETURN MODELS TO PDP: The Quintrell may also be directed to return the models stored in the Quintrell to the PDP. This is a diagnostic tool used to verify the proper operation of the Quintrell/PDP interface. The program was retained in the final version as an aid in trouble shooting.

3.4 PERFORMANCE OF THE DEMONSTRATION SYSTEM

Limited testing of the speaker recognition system was performed under the contract. The testing included the use of ten and twenty seconds of speech for model generation, and ten and twenty seconds of speech for unknowns.

3.4.1 TV DATA BASE

An analog data base was generated by recording speakers from broadcast television programs such as news reports, interviews, and talk shows. Approximately one minute of speech was obtained from 30 different male speakers. The first 20 seconds of speech from each talker was used a the reference set to generate models, and the second 20 seconds was used as the unknown. The differences in the channel characteristics for the various speakers should not be a problem since the audio bandwidth of the television broadcasts is large compared to the 300- 3600 Hz analysis bandwidth used by the demonstration system.

3.4.3 RECOGNITION RESULTS USING THE DEMONSTRATION SYSTEM

The demonstration system was tested using two sets of speaker models. The first model set was generated using 20 seconds of speech from each of the 30 talkers, and the second using 10 seconds of speech. Two recognition trials were run for each model set. The first trial used 10 seconds of speech for each unknown, and the second used 20 seconds of speech. The results of the four tests are shown in table 3.3 below.

Table 3.3 Demonstration System Recognition Results

	10 second models	20 second models
10 second unknowns	93% (28 correct out of 30)	90% (27 correct out of 30)
20 second unknowns	100% (30 correct out of 30)	97% (29 correct out of 30)

The results are very encouraging since the requirement of 95% recognition on 90% of the talkers was met for all four conditions. This requirement means that for 27 talkers (90% of 30), the recognition rate must exceed 95%. It is somewhat surprising that the 10 second models performed better than the 20 second models, however, it must be remembered that this was an extremely small test. Only one recognition trial was run for each speaker. Further testing is required to adequately estimate the system performance.

To indicate the types of results that are obtained when running the speaker recognition system on a 30 talker data base, the individual results for the system test using 10 second models and 20 second unknowns are given in the Appendix. The Appendix contains both the similarity score plot and the confidence measure plot for each of the thirty recognition trials. Figure 3.9 shows a typical recognition trial where speaker 4 was

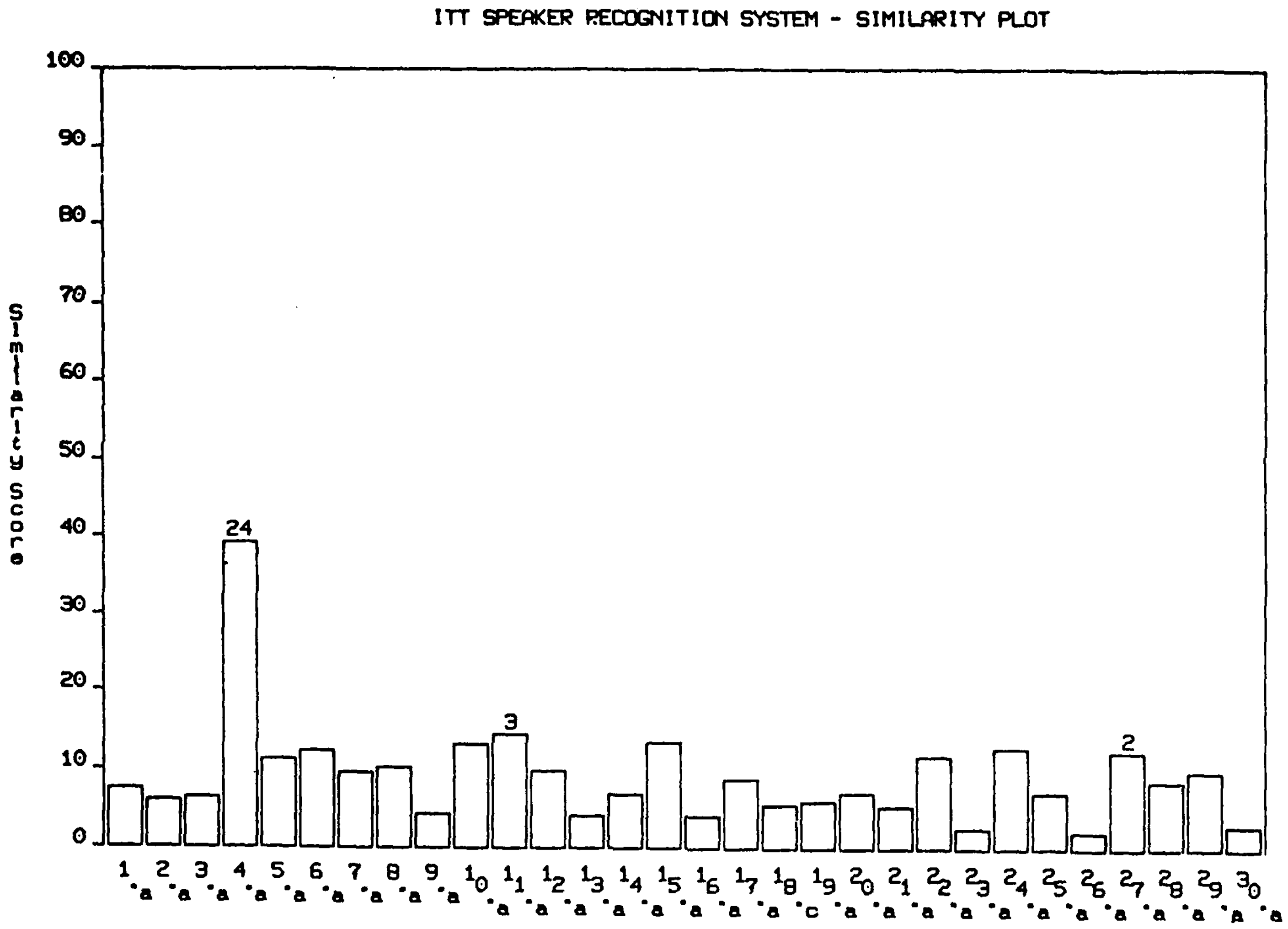


Figure 3.9 Test using 20 seconds of speech from speaker4

correctly identified. Figure 3.10 is the similarity score plot for 10 seconds of speech for speaker 1, one of three speakers improperly identified during the ten second tests. It is important to note that even though the correct talker was not identified, no one speaker is clearly better than another, and speaker 1 is one of the top three contenders.

Another interesting result is that during all of the tests, most of the recognition errors that did occur, were made on speakers whose recording levels where much lower than the other speakers. It is expected that the addition of some type of automatic gain control at the input to the Quintrell's analog to digital converter could improve the system performance on these talkers.

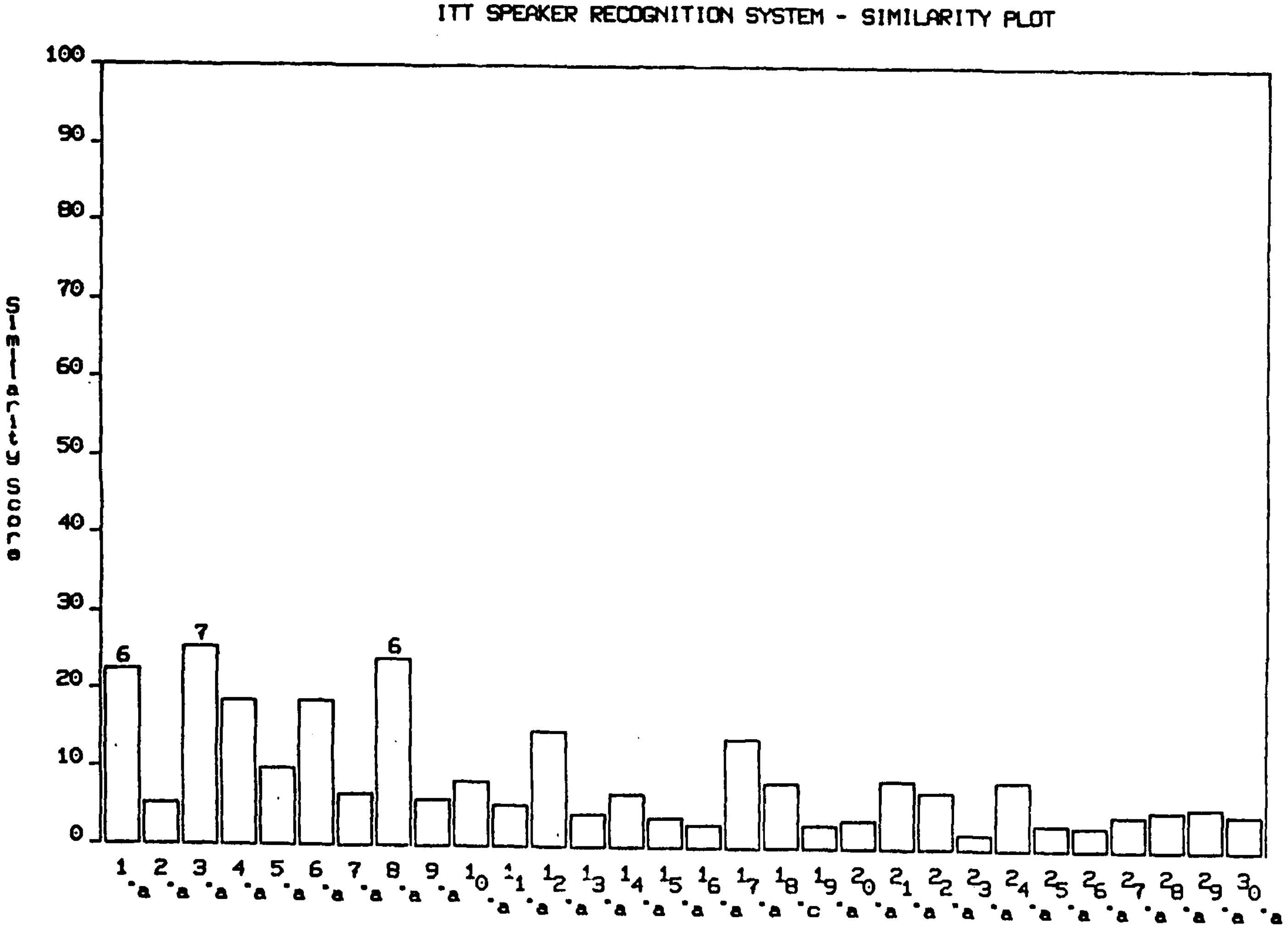


Figure 3.10 Test using 10 seconds of speech from speaker1

CHAPTER 4. RECOMMENDATIONS

The results of this first attempt at demonstrating a realtime speaker recognition capability are extremely promising. The algorithms and the technology necessary to implement a highly accurate realtime speaker recognition system are available. There are, however, several areas where further study would be beneficial. In addition, the demonstration system needs further, expanded testing. Additional study is recommended as follows:

1. Communication channel effects such as distortion and noise should be investigated. The limited testing so far used only white noise, and did not address problems such as channel distortion and colored noise.

2. As part of the noise investigation, ways to improve robustness of LPC analysis under the expected noisy conditions should be determined. Several techniques such as spectral subtraction and Wiener filtering currently exist which can significantly improve the quality of LPC analysis of noisy speech. The area to be investigated would be the effect of these techniques on speaker recognition performance.

3. Improvements to the subpopulation decision should be tested. As part of this contract, the recognition accuracies were improved by including not only a voicing decision, but a power threshold as well. The inclusion of an absolute power threshold has made the system sensitive to the input speech power level. This power threshold should be refined to make it proportional to the input power. In addition, an automatic gain control should be tested at the front end of the system to further reduce the effects of power level throughout the system.

4. Effects of model aging and techniques for optimal model generation should be studied further. For the realtime demonstration, models were generated using speech from the same time period as the unknowns. During the algorithm study phase of the contract, it was shown that models generated with data separated in time from the unknown speech by one week performed significantly poorer (10-15%) than the models generated with speech from the same time period as the unknown. Previous work at ITTDCD has demonstrated that if current data is not available for model generation, the best performance is obtained by using speech segments recorded over a long period of time rather than a single speech segment recorded all at once. Further study should be conducted to determine how older speech samples should be incorporated into the models to improve performance.

5. Human factors improvements should be investigated. The system must be demonstrated to potential users to obtain their inputs as to how an operator can best interact with the speaker recognition system. The current operator interface is a first attempt at producing a useful interface, and changes may be required before the interface becomes optimum.

6. Tests should be made to determine the effects of increasing and decreasing the number of current speaker models that are compared with the unknown, on system recognition accuracy. It would also be desirable to investigate the performance of the system on unknown speech data of less than 10 seconds.

7. New methods for computing and displaying the confidence and similarity scores should be investigated. These methods should focus on providing a meaningful and clear display of these scores. The confidence score in particular should probably include a factor which reflects the amount of speech data used in model generation.

8. In tests where the unknown speakcer is incorrectly identified, it would be desirable to know whether the correct speaker's model was among the top two or three choices. This would be a further indication of the robustness of the algorithm. An effective automatic change-in-speaker detector should be developed.

9. Tests should be performed to determine the effectiveness of the speaker authentication system on languages other than American-English.

10. Algorithms should be developed which can incorporate information such as signal strength and directivity to aid in making the speaker authentication decision.

ITTDCD studies separate from this contract indicate that a practical realtime speaker recognition system can be built using off the shelf microprocessors augmented with a fast multiplier in an architecture optimized to the speaker recognition problem. This development can be done after completion of the study tasks outlined above, or in parallel with them.

REFERENCES

1. Markel J.D., Oshika B.T., and Gray A.H. Jr., "Long-Term Feature Averaging for Speaker Recognition" IEEE Trans. on Acoustics, Speech, and Signal Processing Vol ASSP-25, No 4, pp. 330-337. Aug 1977.

2. Pfeifer Larry L., "Feature Analysis for Speaker Identification" RADC-TR-77-277, Final Technical Report for Rome Air Development Center, August 1977, A044311.

APPENDIX

REALTIME RECOGNITION EXPERIMENTS

Realtime Recognition Experiments using 10 second models and 20 second unknowns

AD-A085 845   ITT DEFENSE COMMUNICATIONS DIV SAN DIEGO CA   F/G 9/4
SPEAKER AUTHENTICATION OPERATIONAL TEST AND EVALUATION.(U)
APR 80   E H WRENCH   F30602-78-C-0324
UNCLASSIFIED   RADC-TR-80-64   NL

2 OF 2
AD
A085845

END
DATE
FILMED
8-80
DTIC

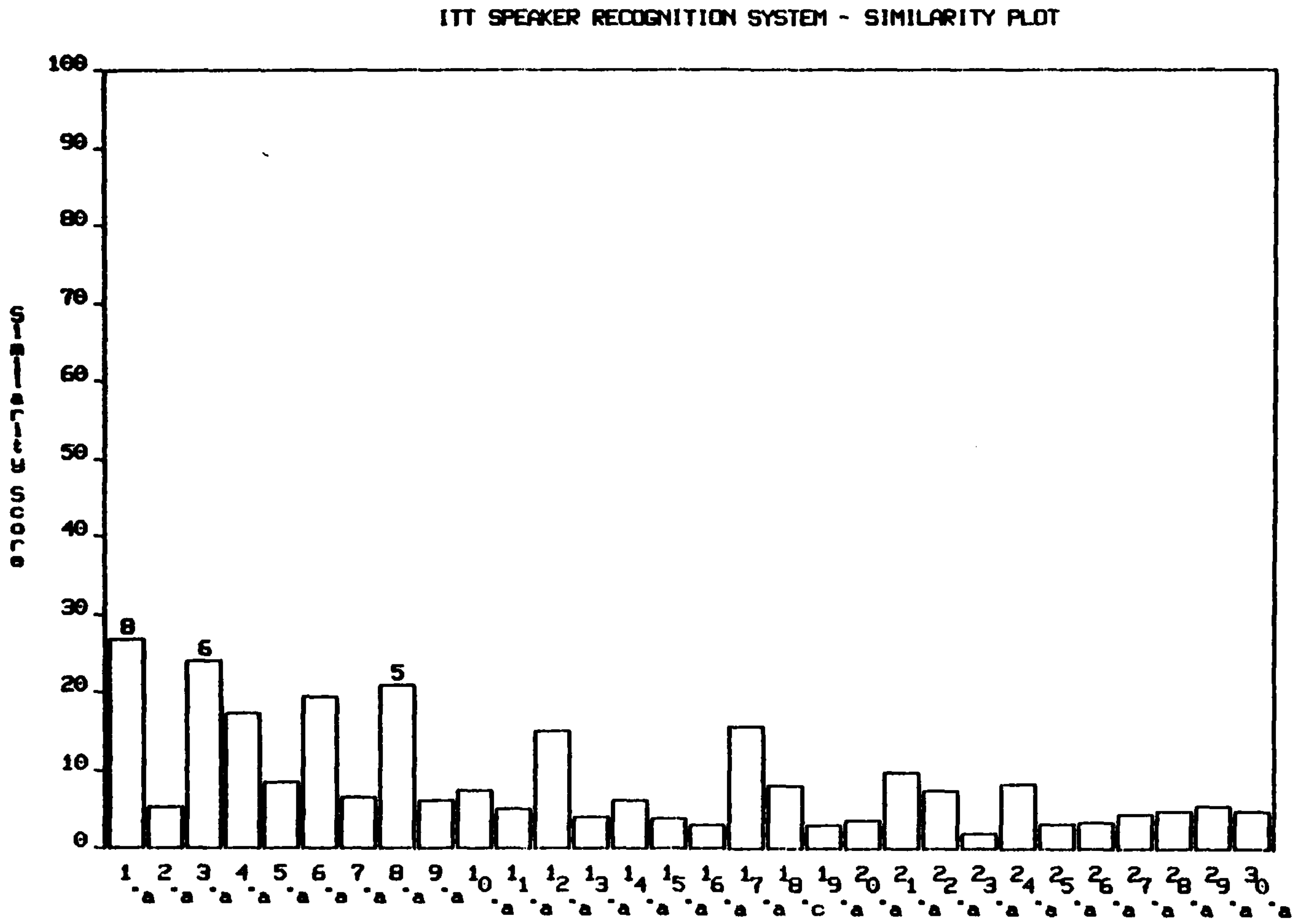


Figure 1.1: Test using 20 seconds of speech from speaker1

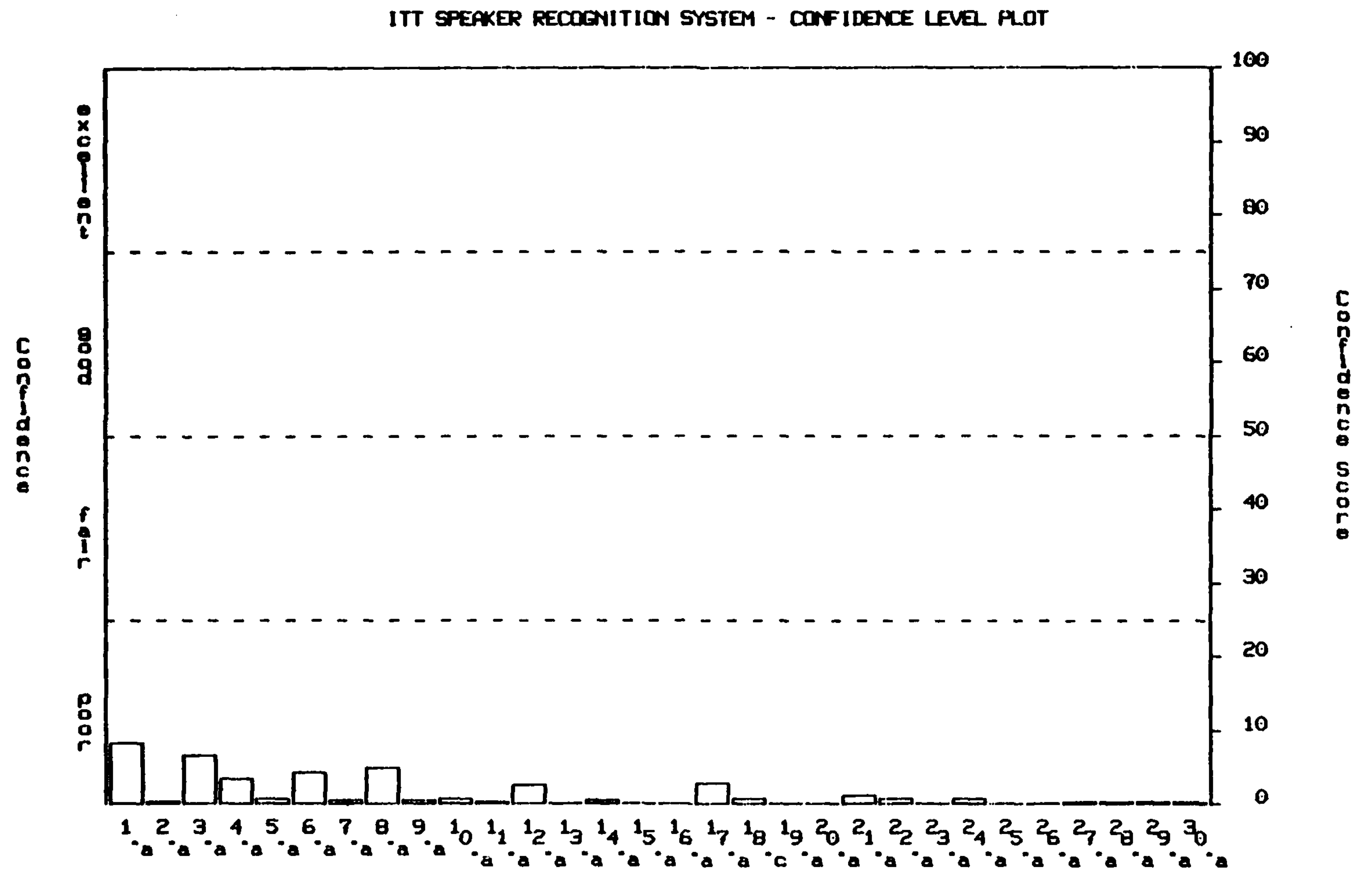


Figure I.2: Test using 20 seconds of speech from speaker1

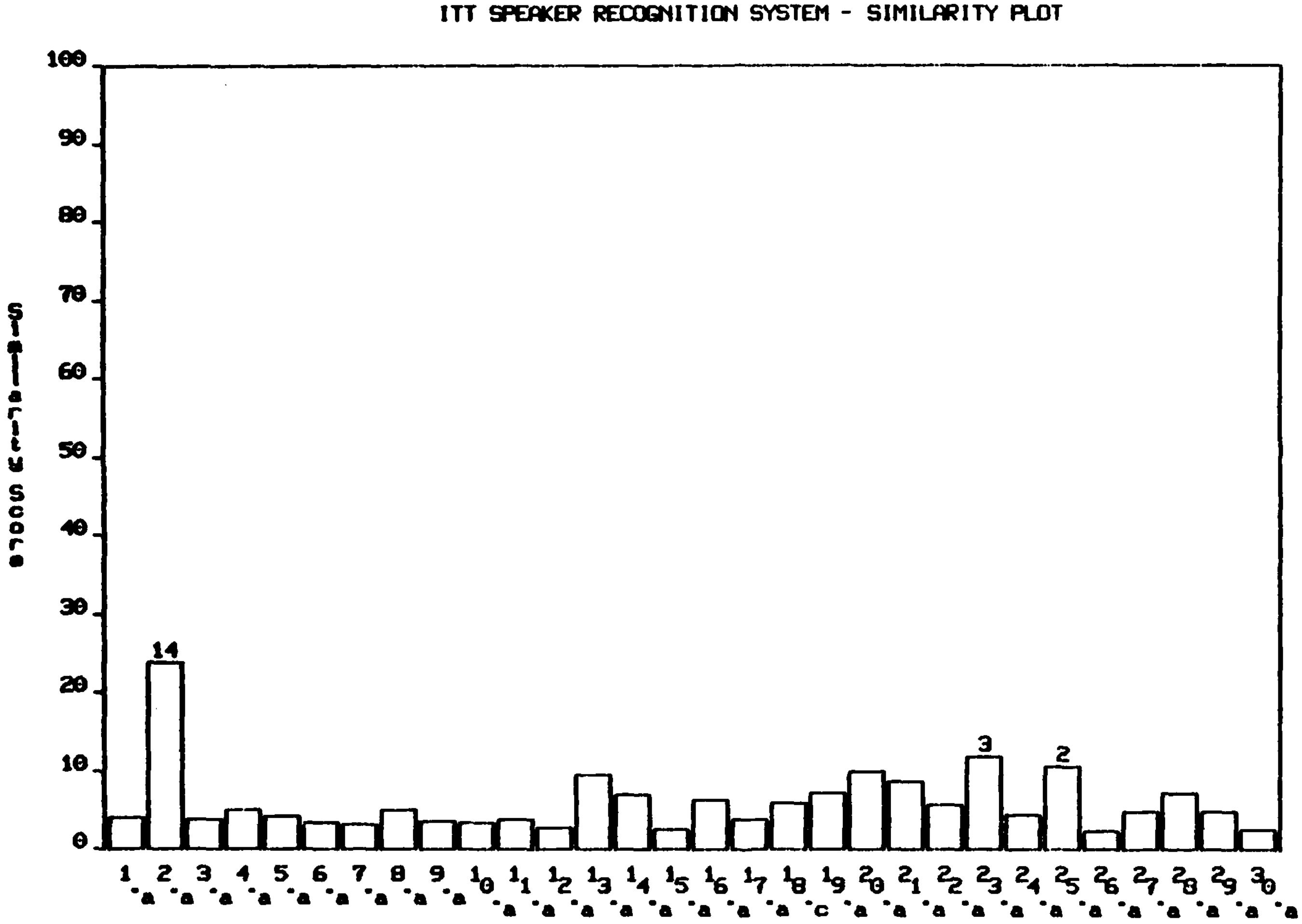


Figure I.3: Test using 20 seconds of speech from speaker2

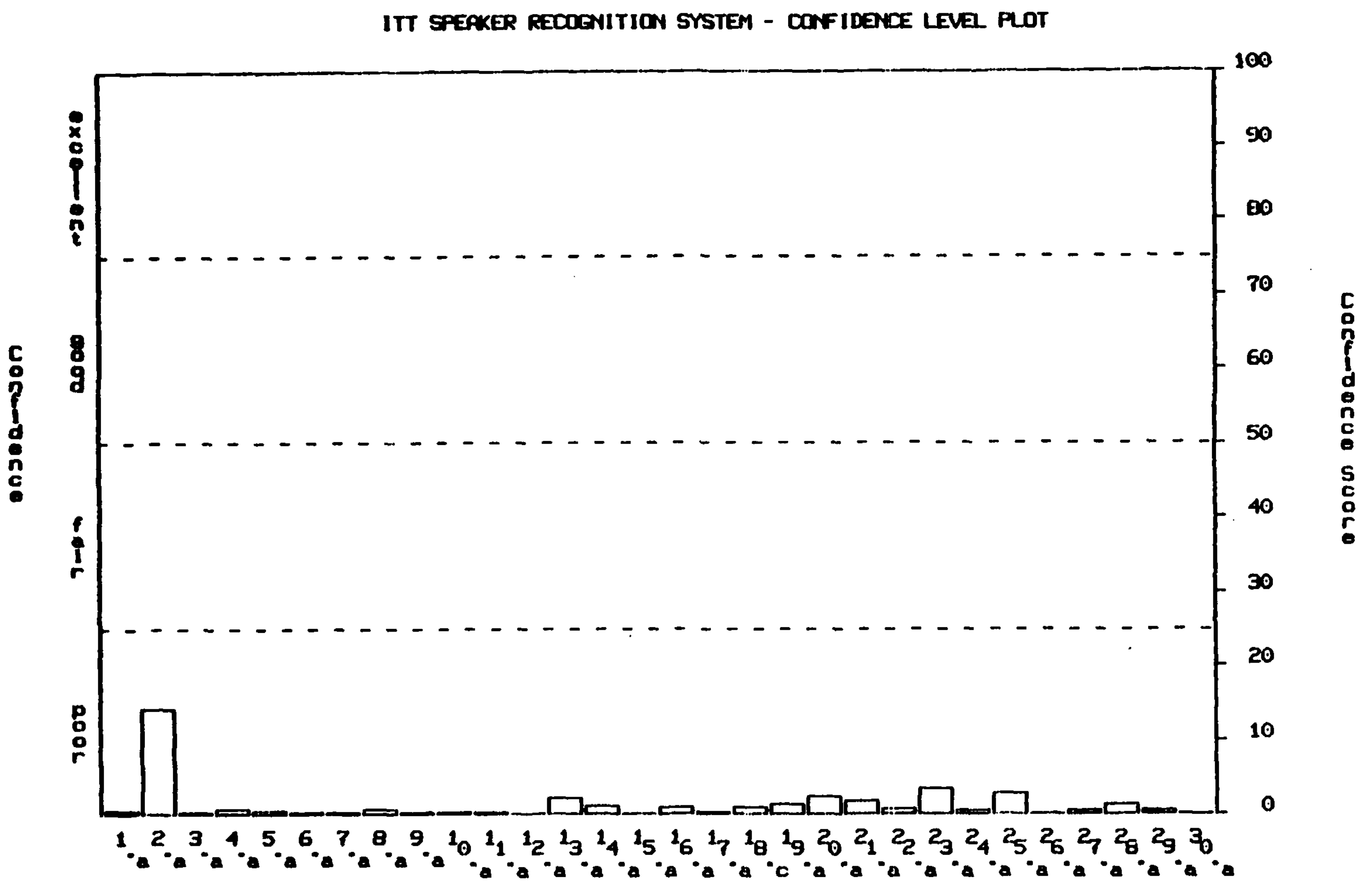


Figure 1.4: Test using 20 seconds of speech from speaker2

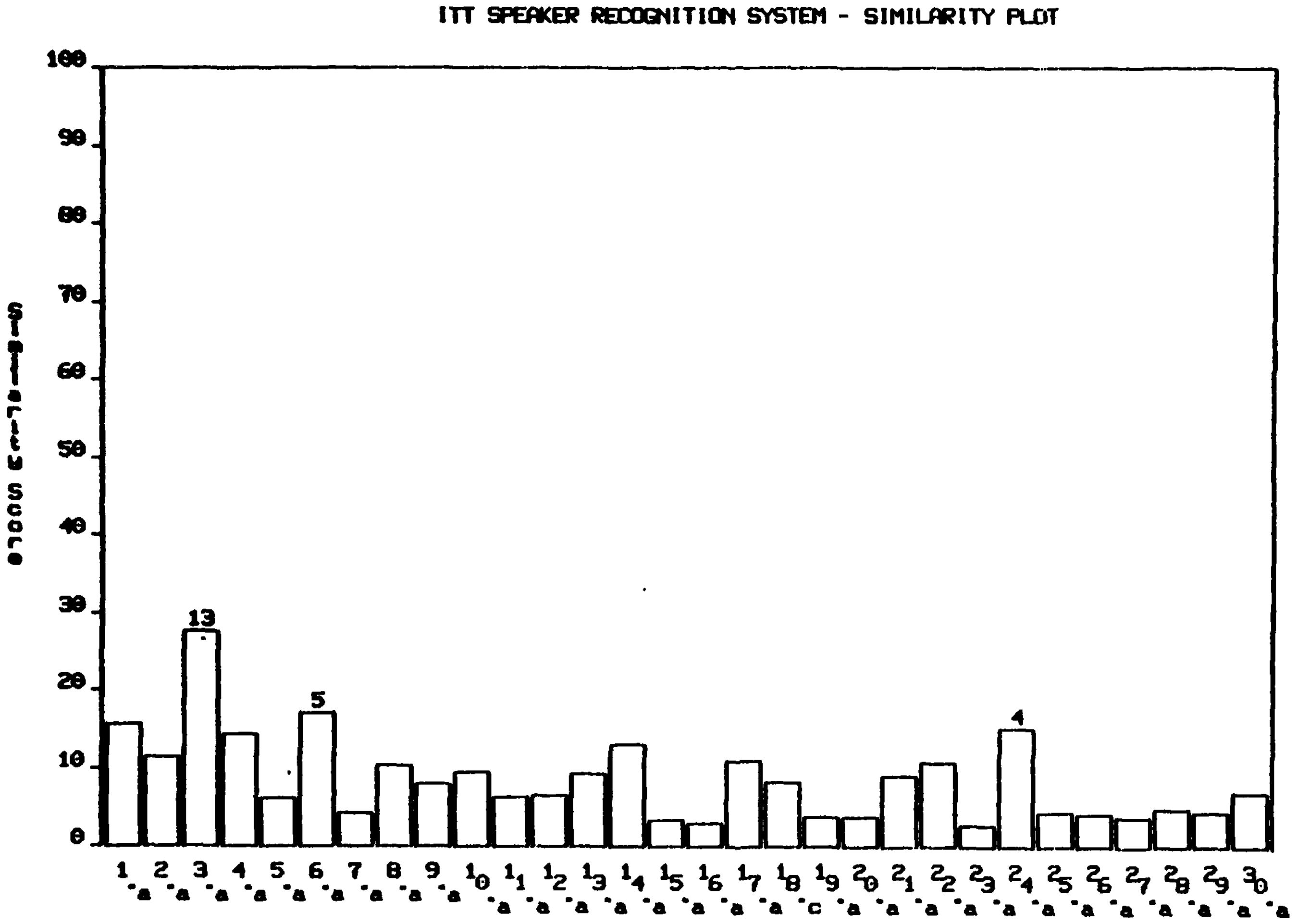


Figure I.5: Test using 20 seconds of speech from speaker3

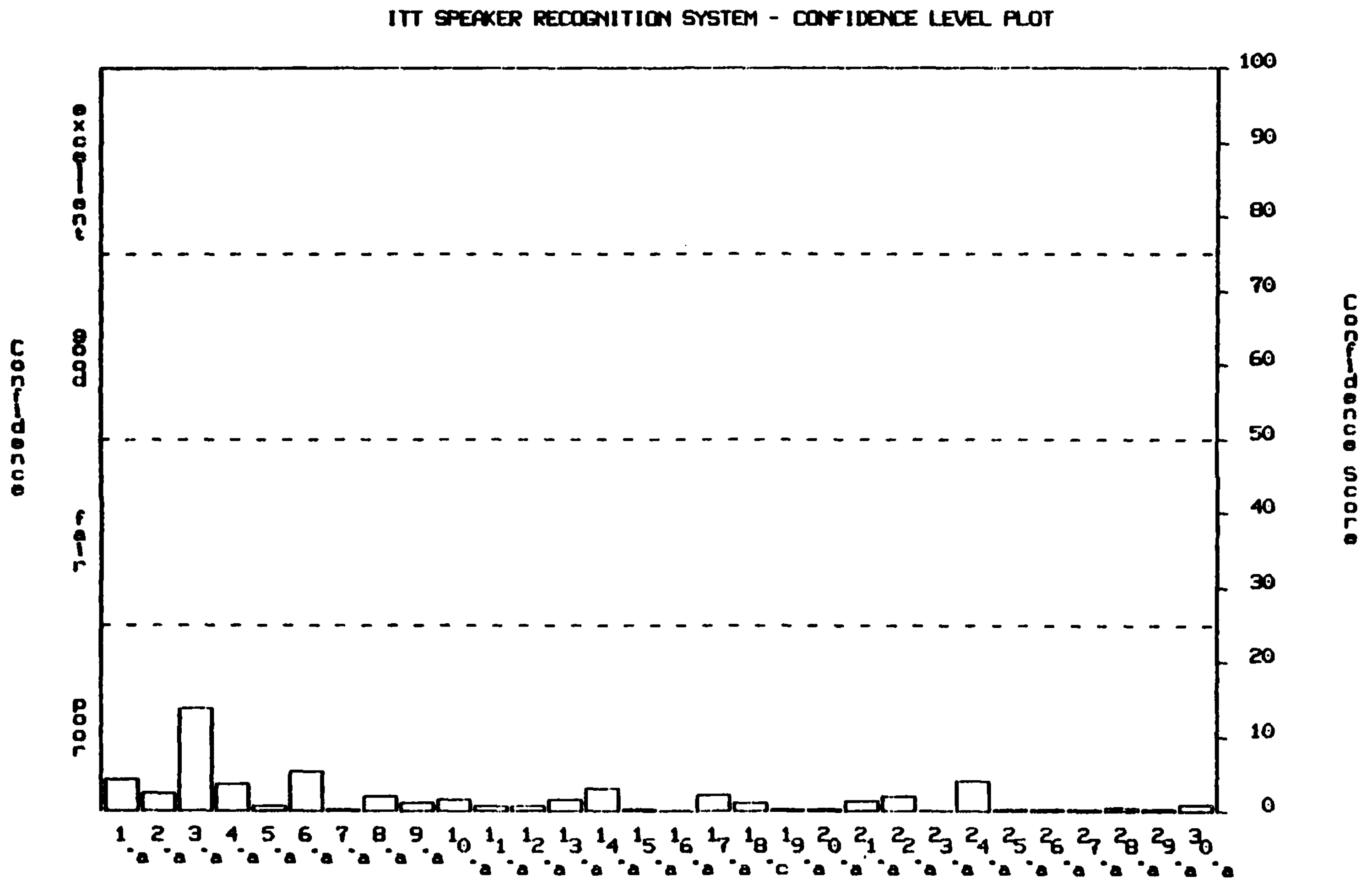


Figure I.6: Test using 20 seconds of speech from speaker3

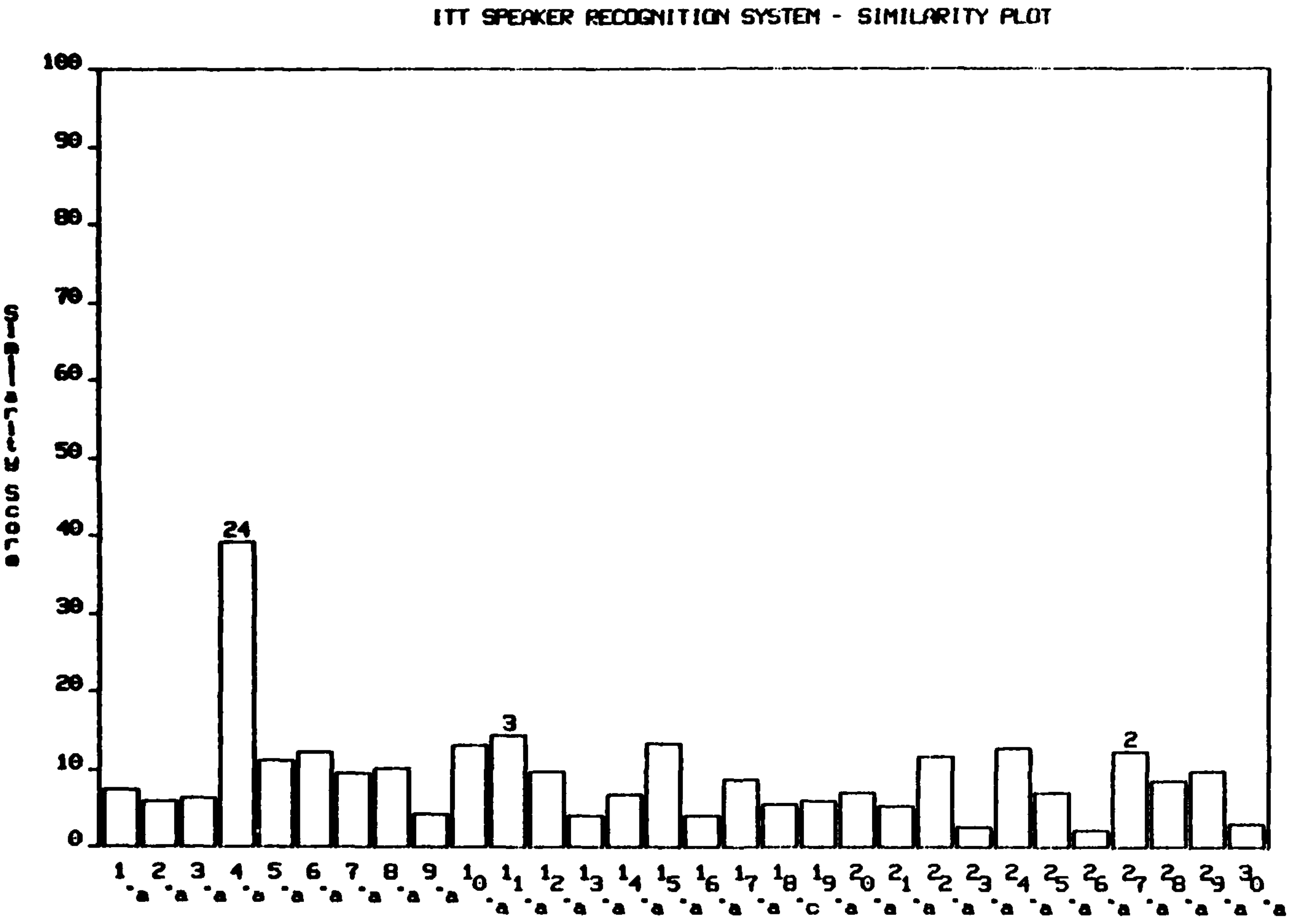


Figure 1.7: Test using 20 seconds of speech from speaker4

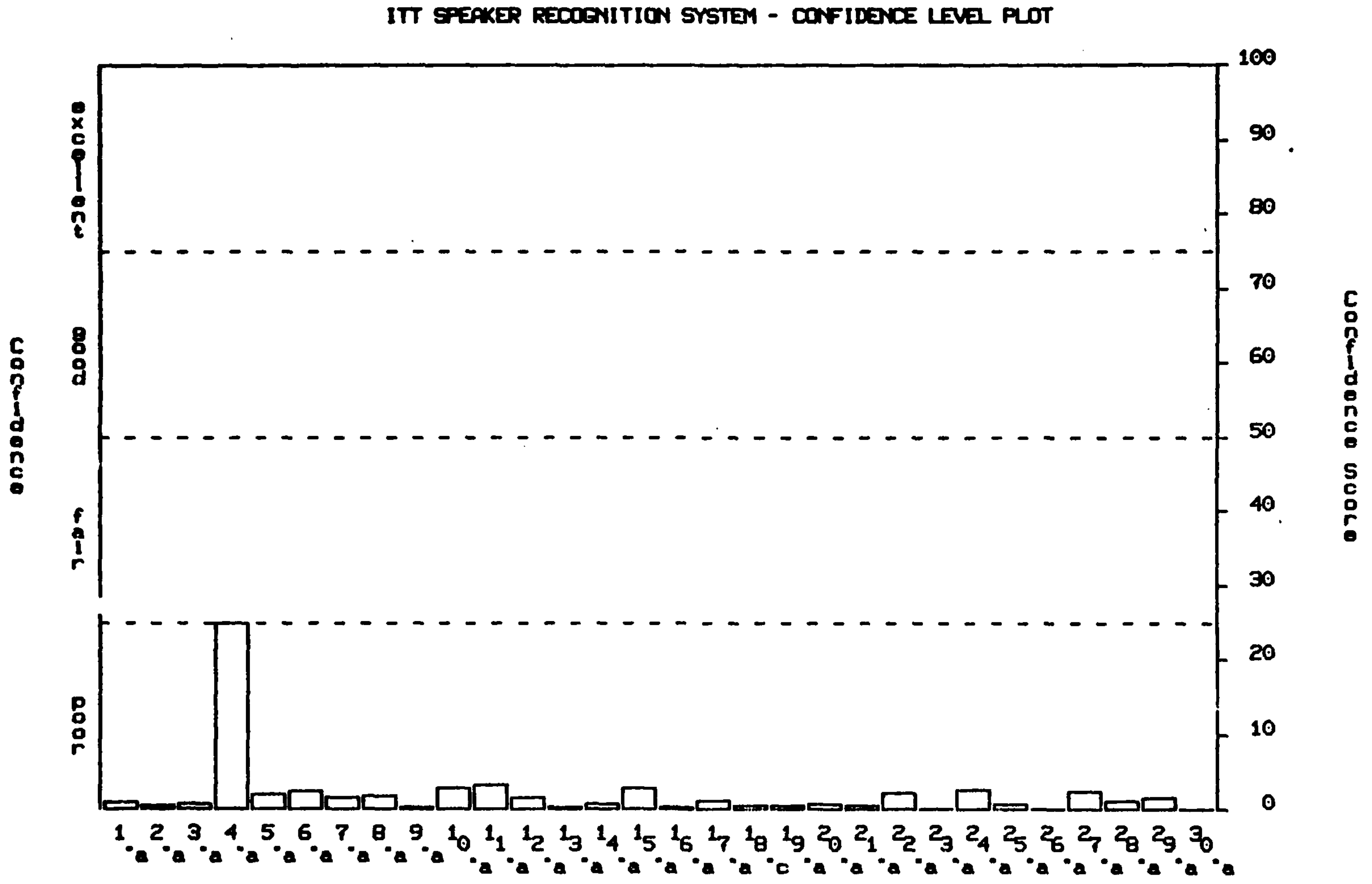


Figure I.8: Test using 20 seconds of speech from speaker4

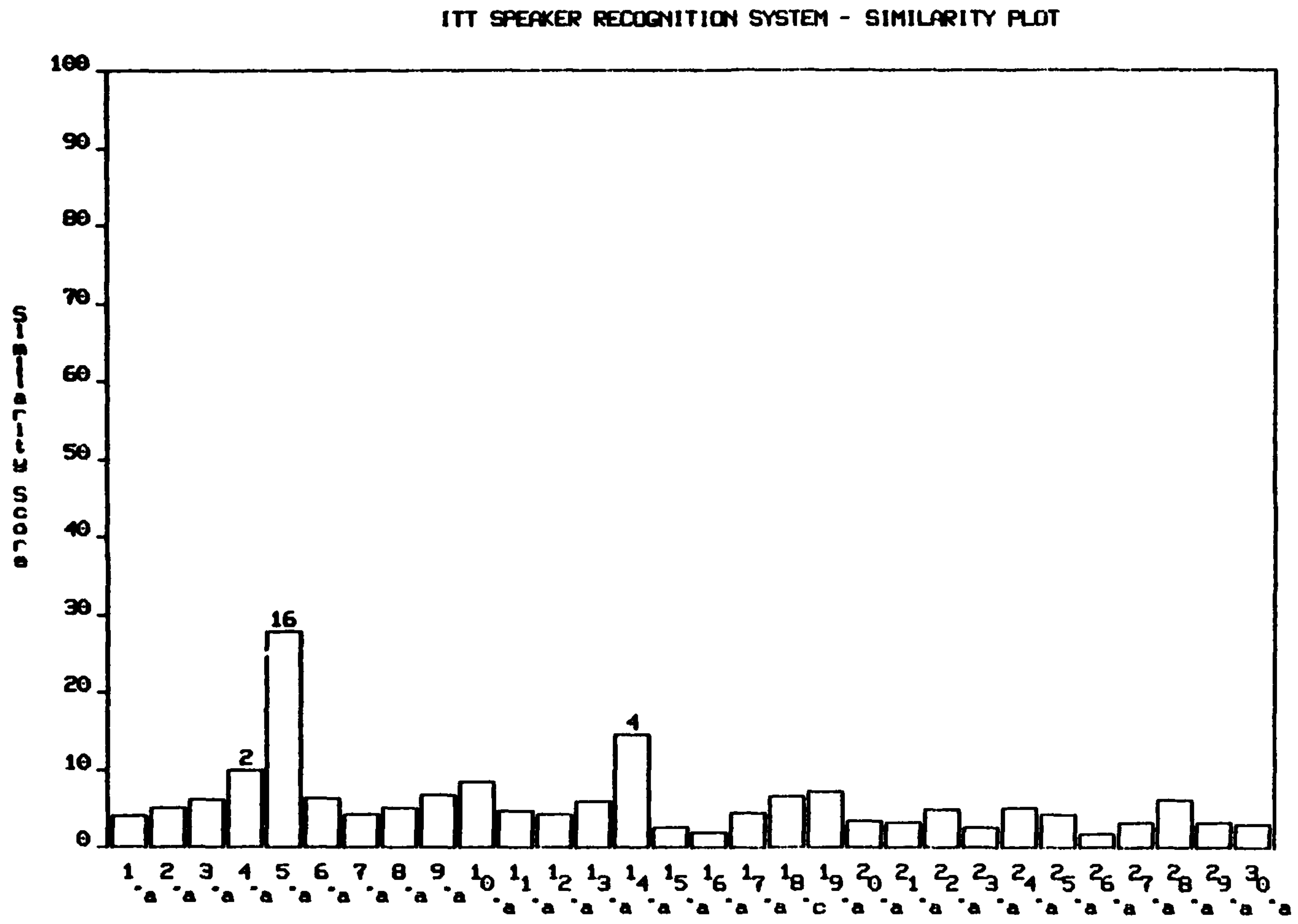


Figure I.9: Test using 20 seconds of speech from speaker5

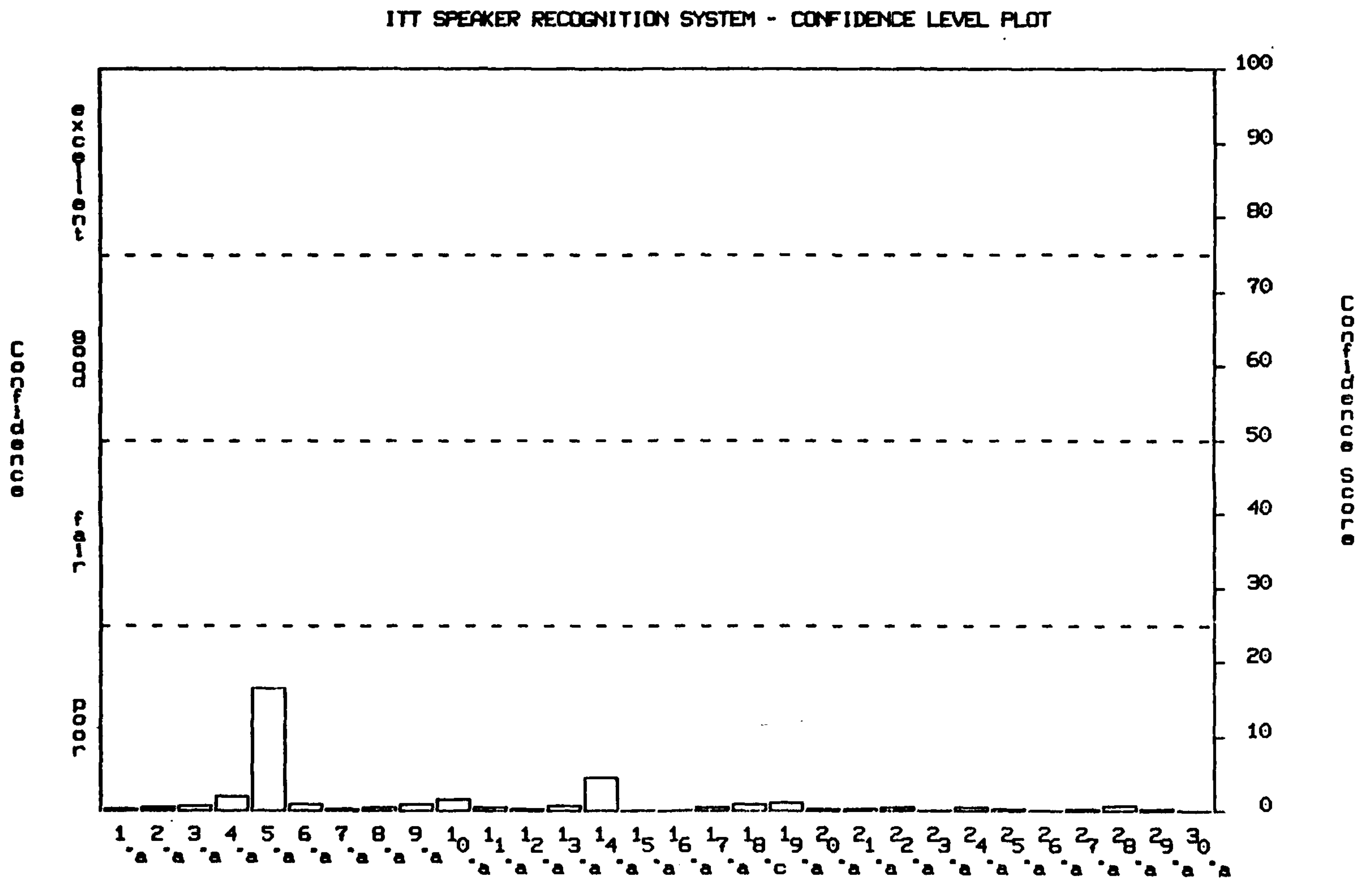


Figure I.10: Test using 20 seconds of speech from speaker5

ITT SPEAKER RECOGNITION SYSTEM - SIMILARITY PLOT

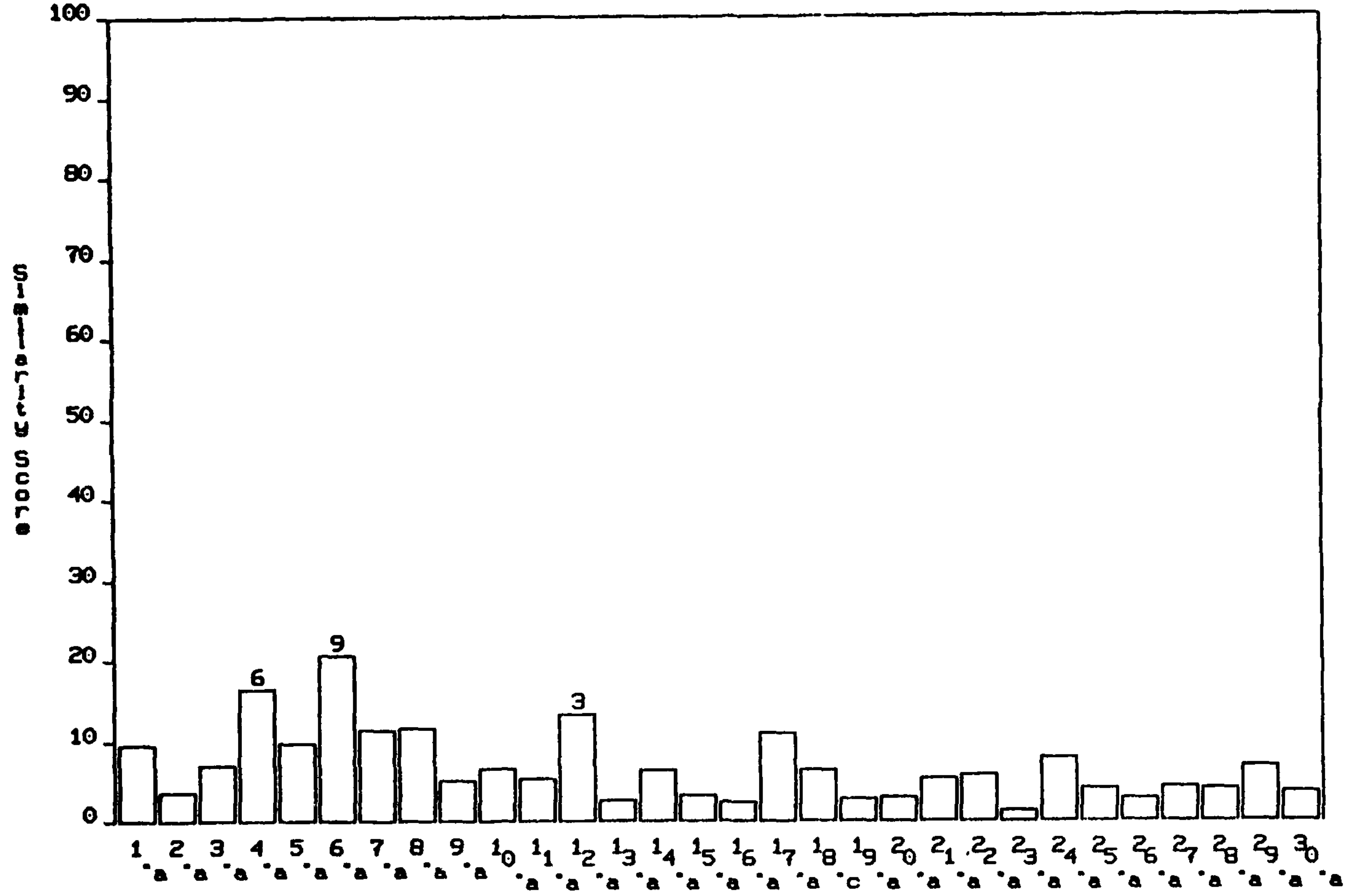


Figure I.11: Test using 20 seconds of speech from speaker6

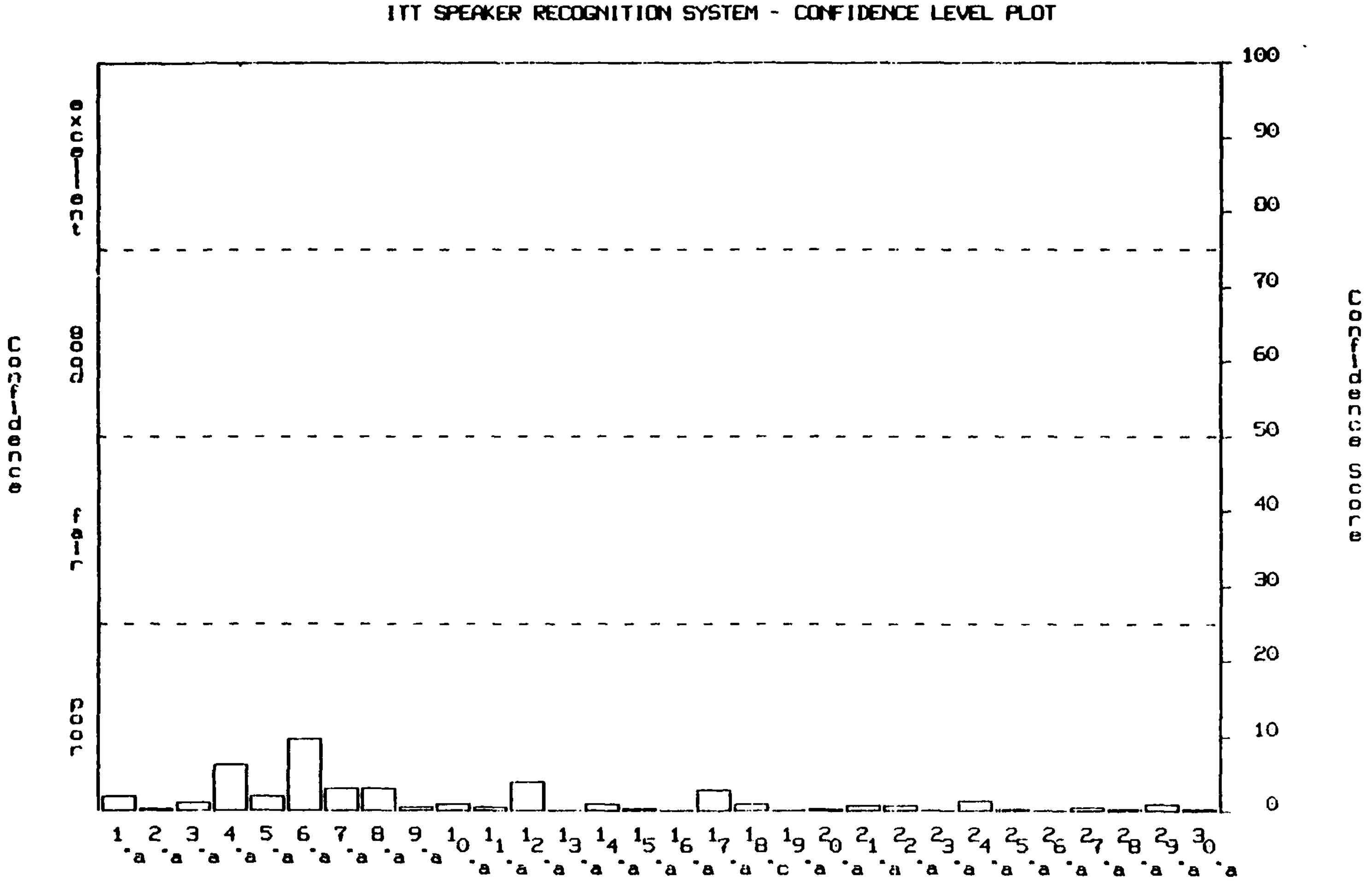


Figure I.12: Test using 20 seconds of speech from speaker6

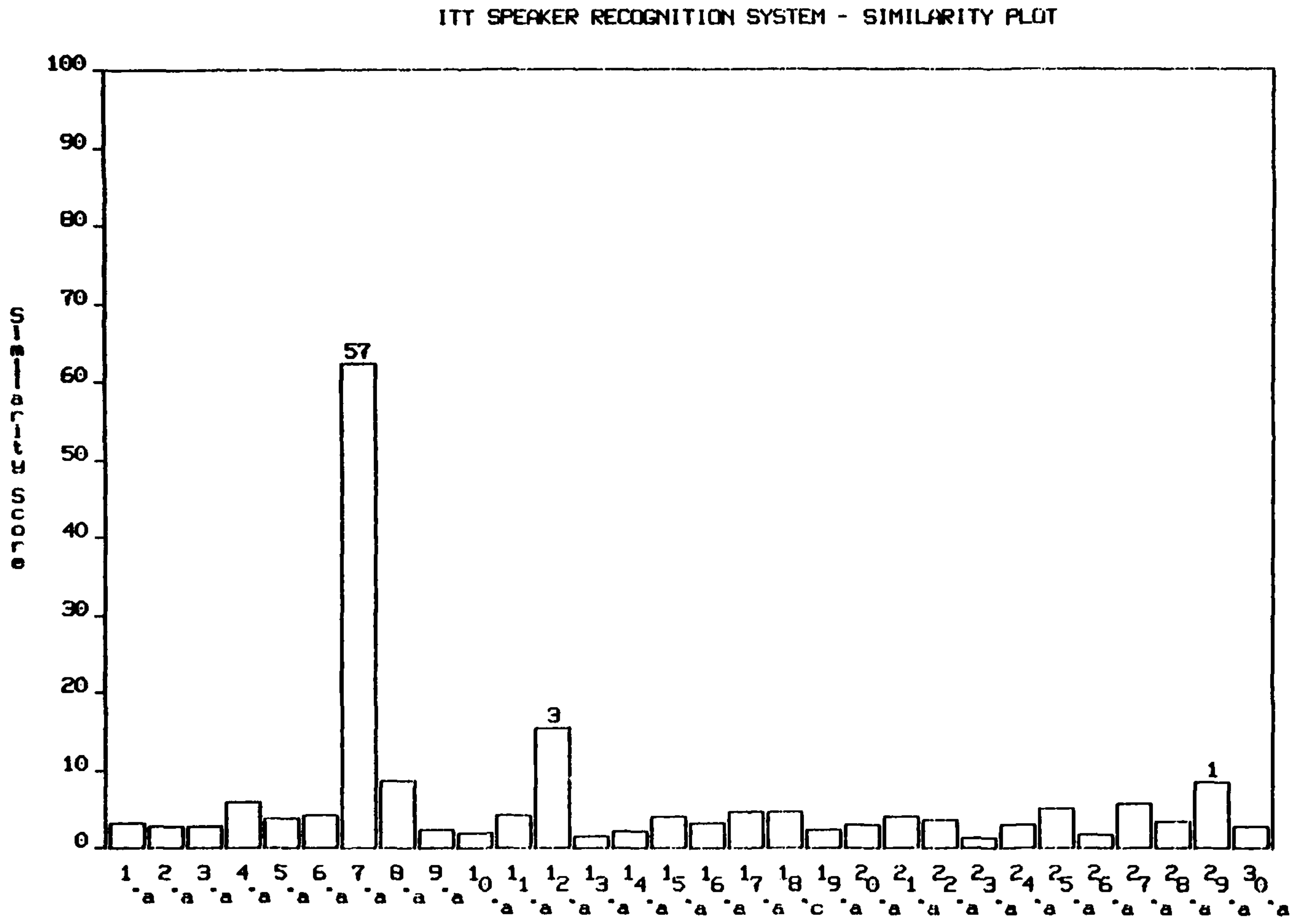


Figure I.13: Test using 20 seconds of speech from speaker7

ITT SPEAKER RECOGNITION SYSTEM - CONFIDENCE LEVEL PLOT

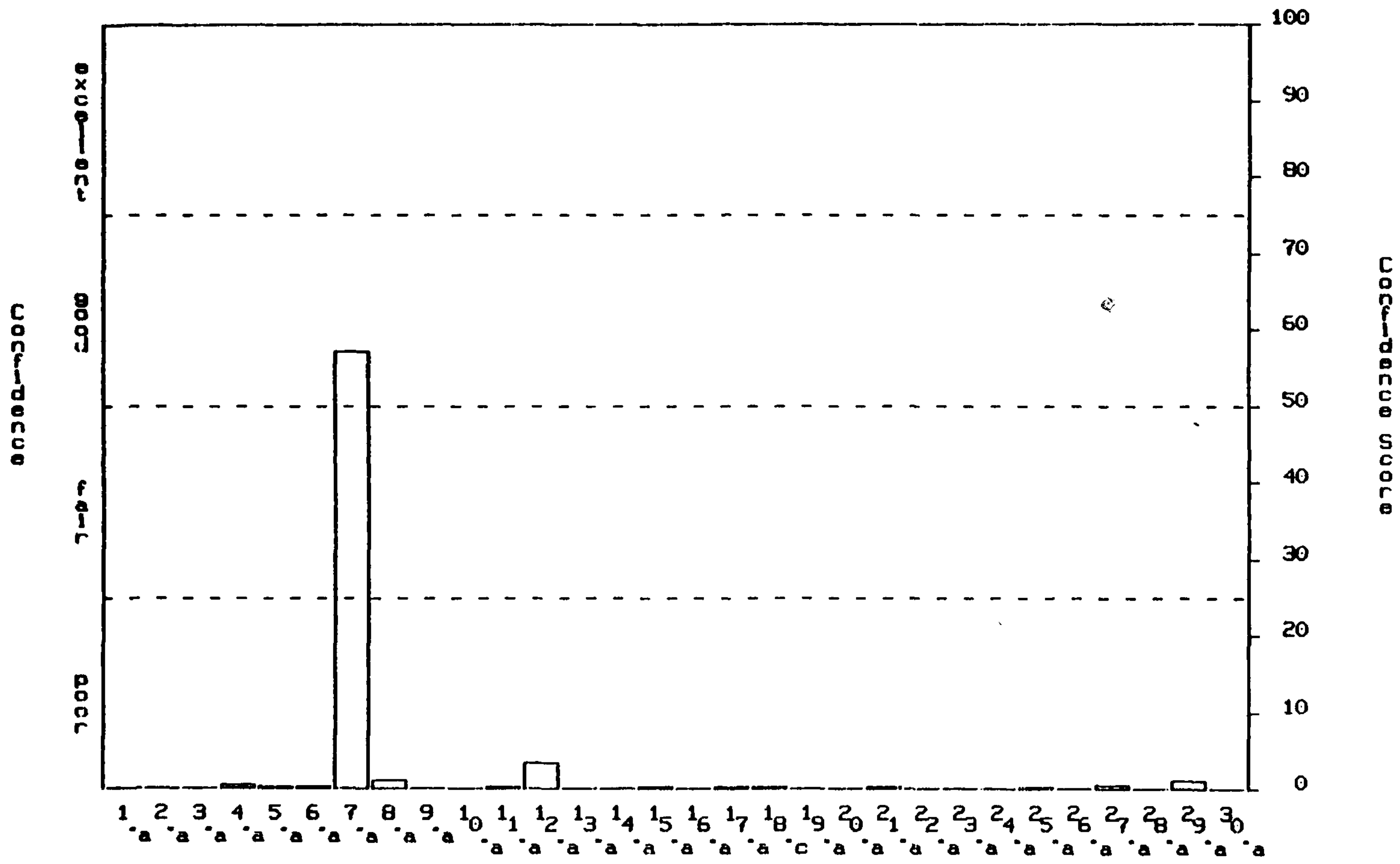


Figure I.14: Test using 20 seconds of speech from speaker7

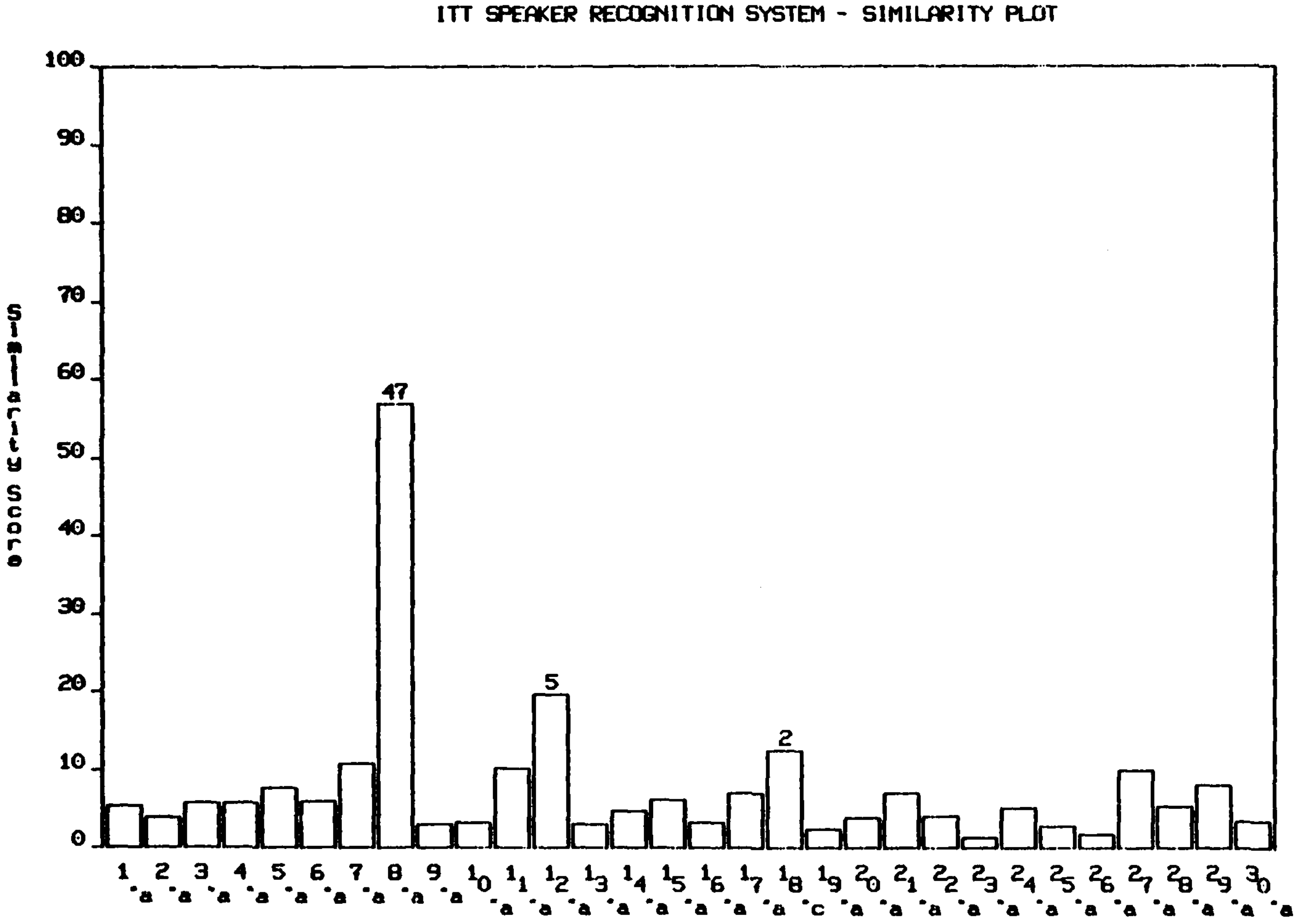


Figure 1.15: Test using 20 seconds of speech from speaker8

ITT SPEAKER RECOGNITION SYSTEM - CONFIDENCE LEVEL PLOT

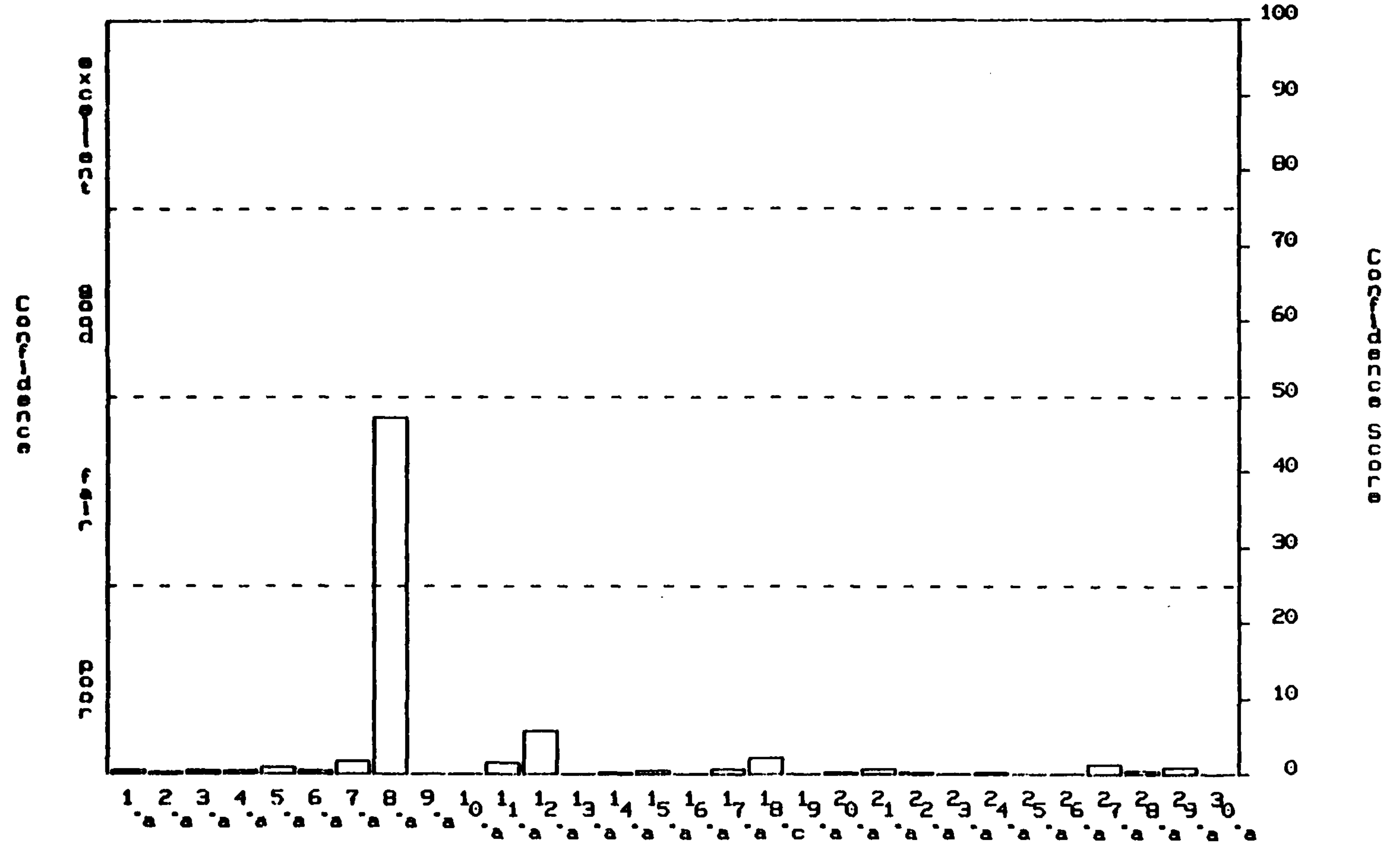


Figure 1.16: Test using 20 seconds of speech from speaker8

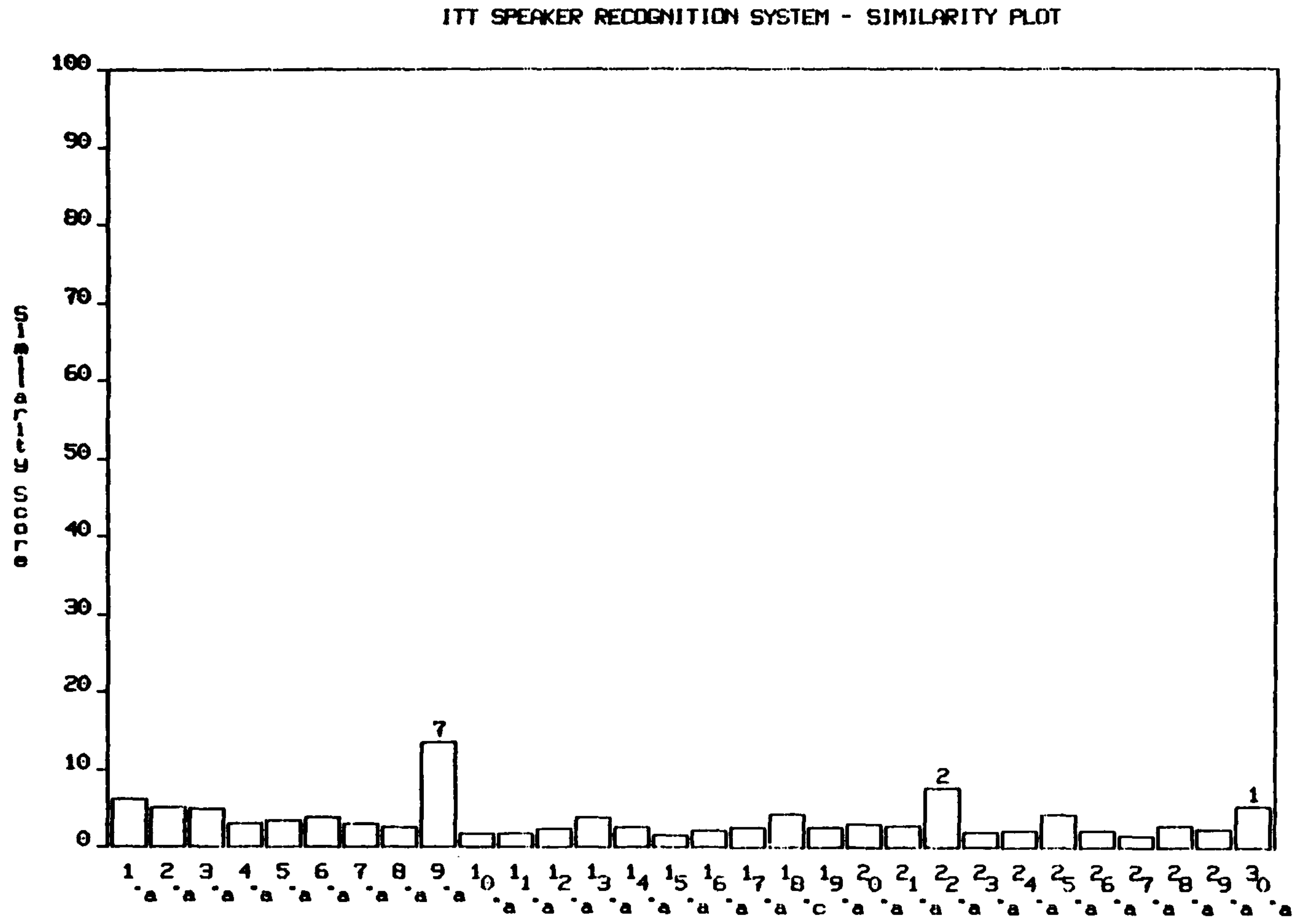


Figure I.17: Test using 20 seconds of speech from speaker9

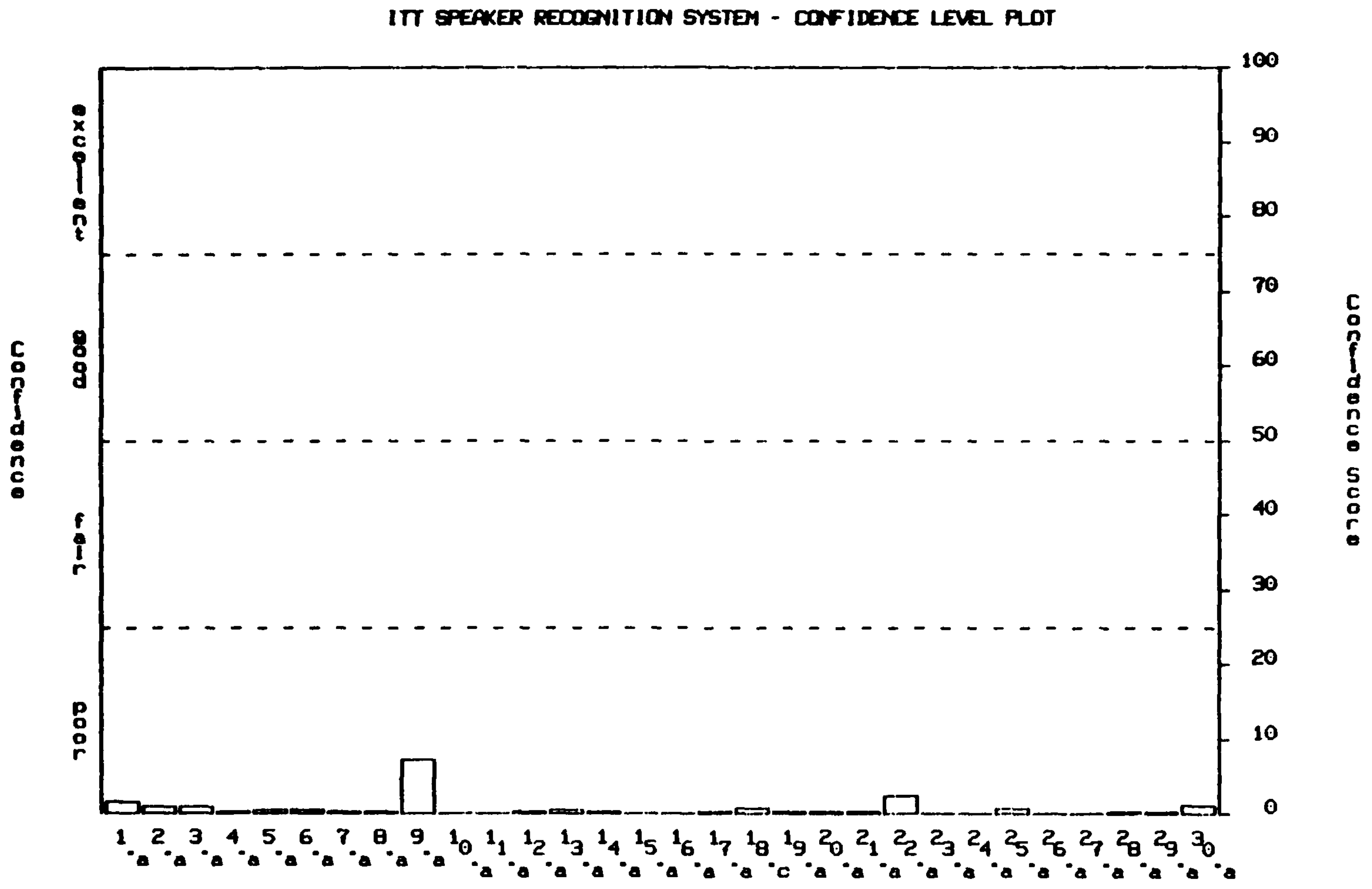


Figure I.18: Test using 20 seconds of speech from speaker9

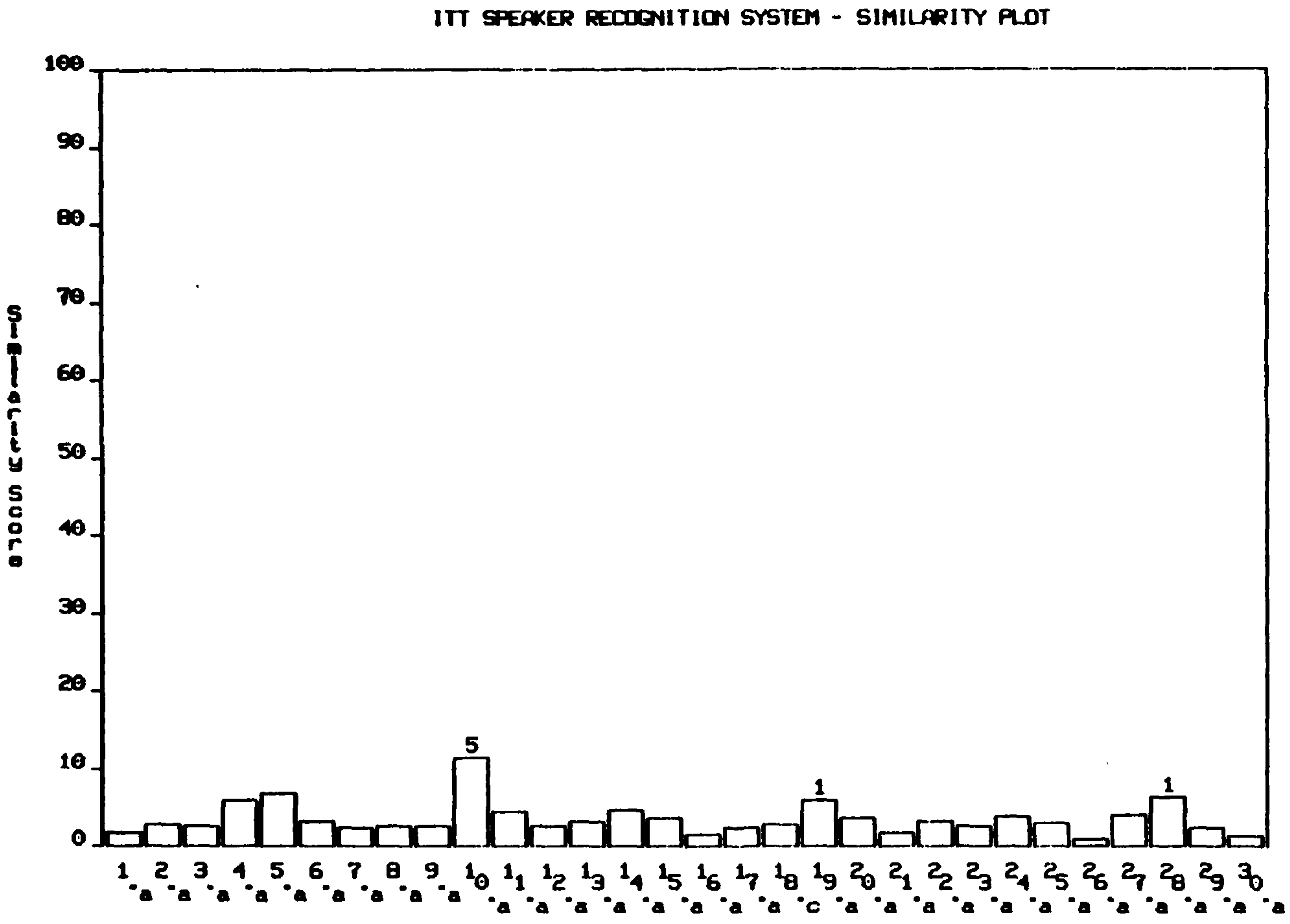


Figure I.19: Test using 20 seconds of speech from speaker10

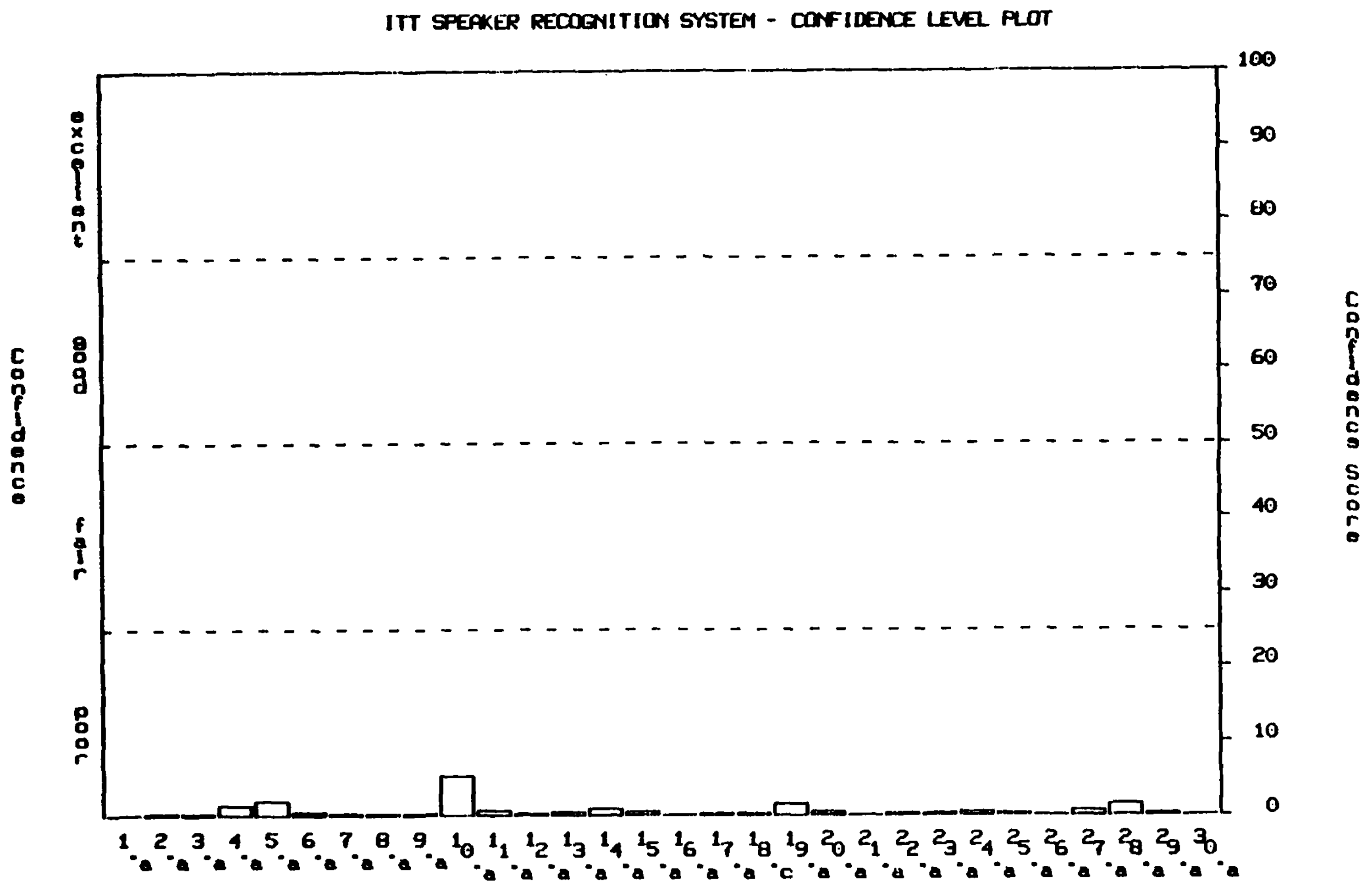


Figure I.20: Test using 20 seconds of speech from speaker10

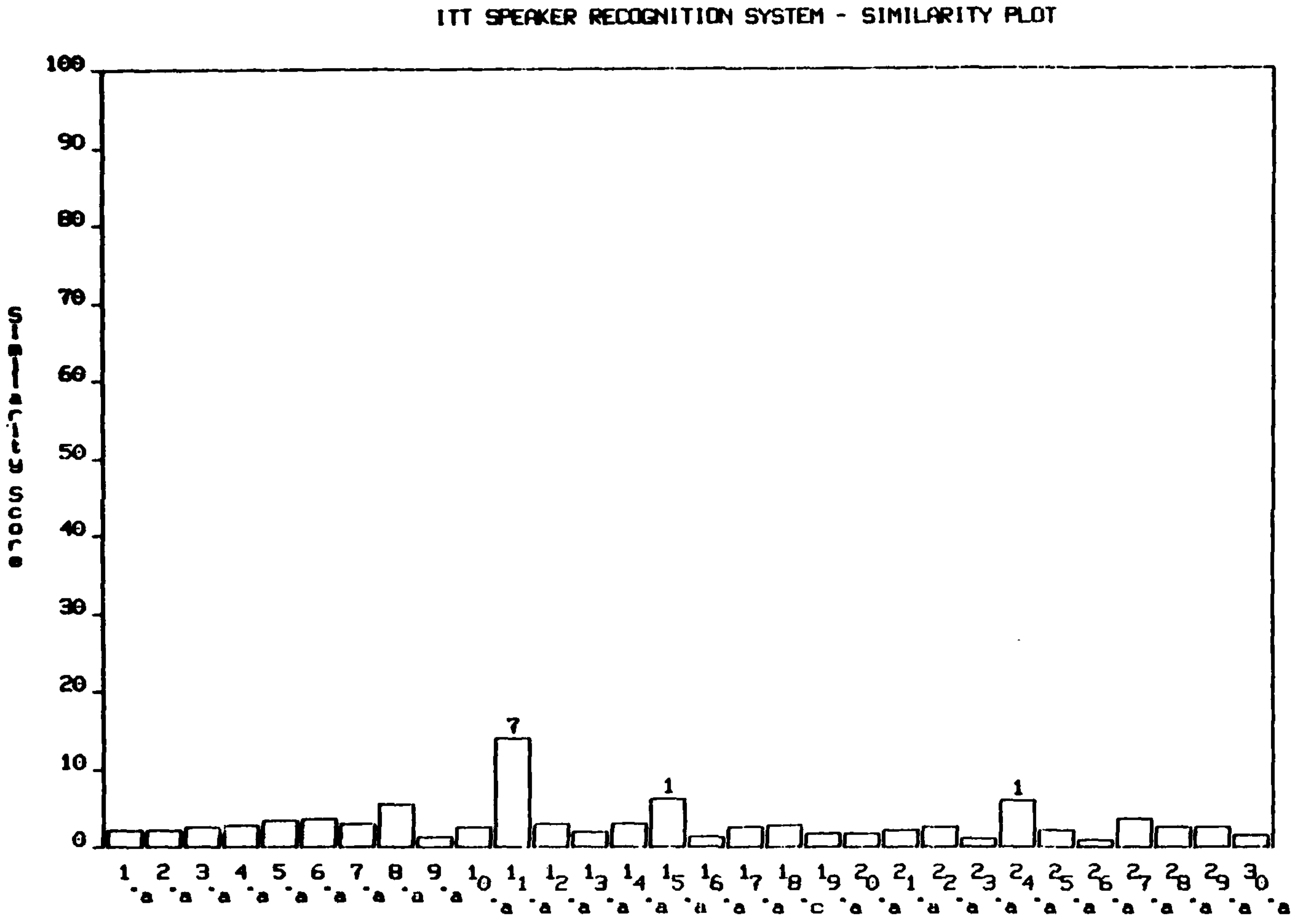


Figure I.21: Test using 20 seconds of speech from speaker11

ITT SPEAKER RECOGNITION SYSTEM - CONFIDENCE LEVEL PLOT

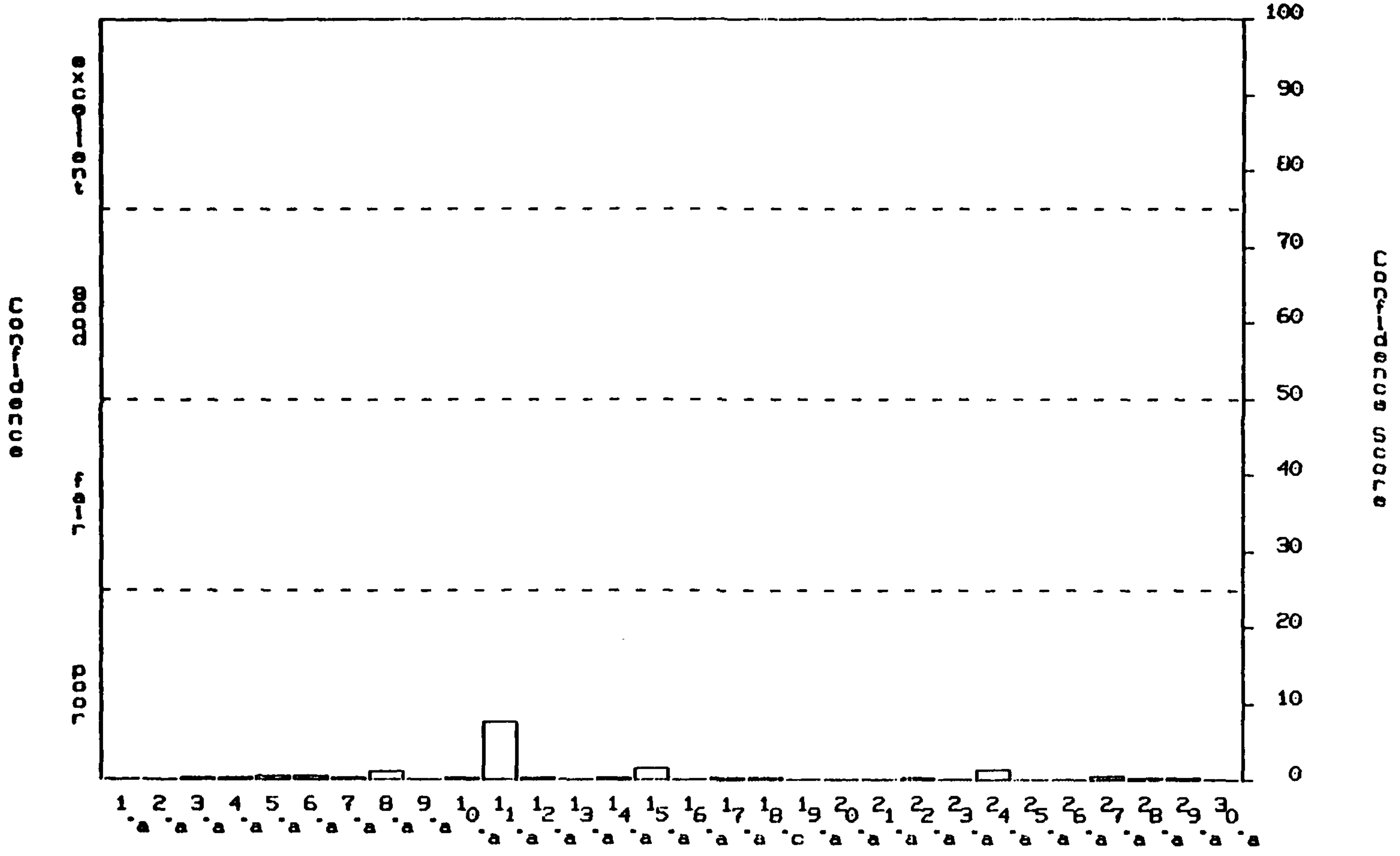


Figure 1.22: Test using 20 seconds of speech from speaker11

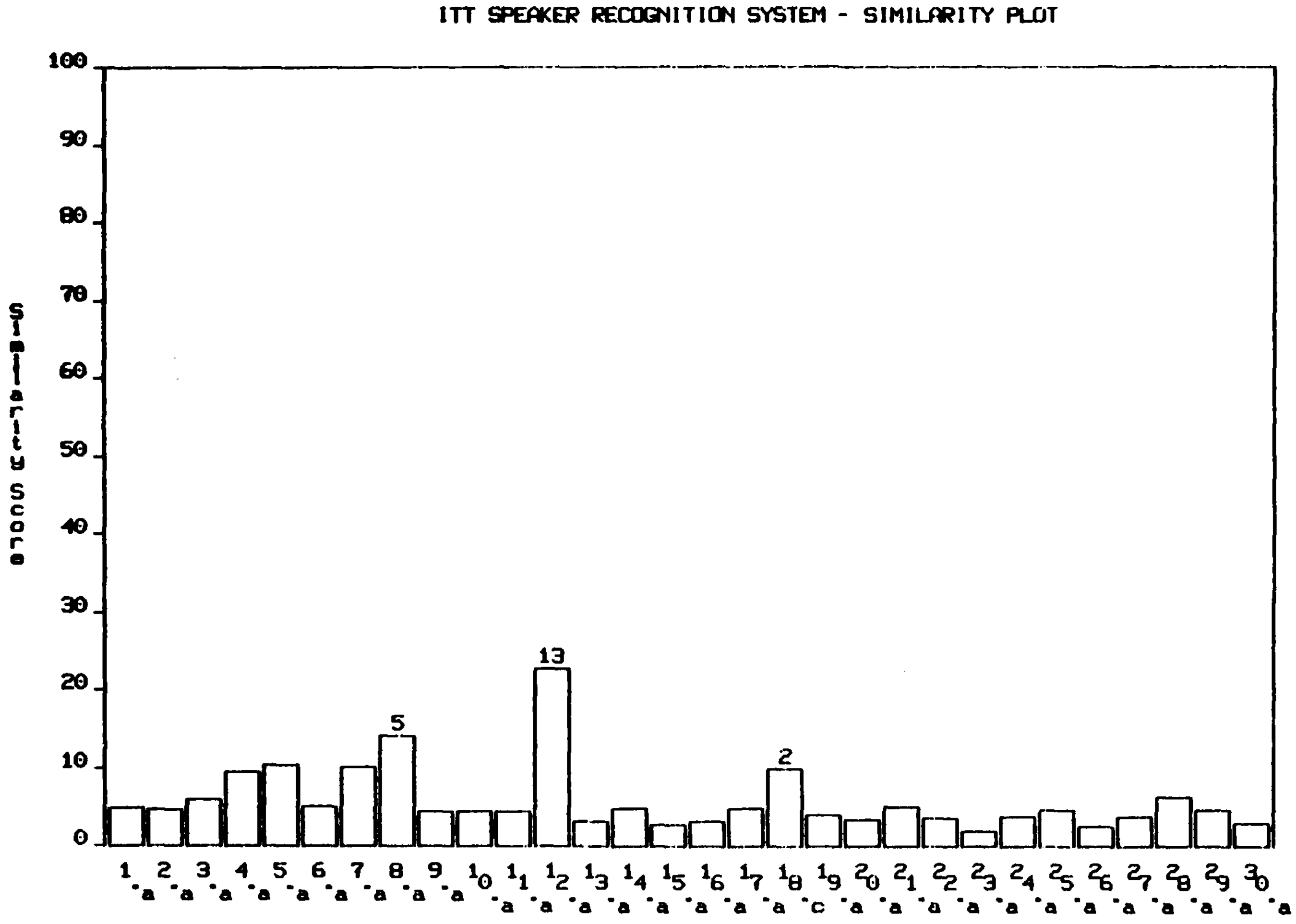


Figure I.23: Test using 20 seconds of speech from speaker12

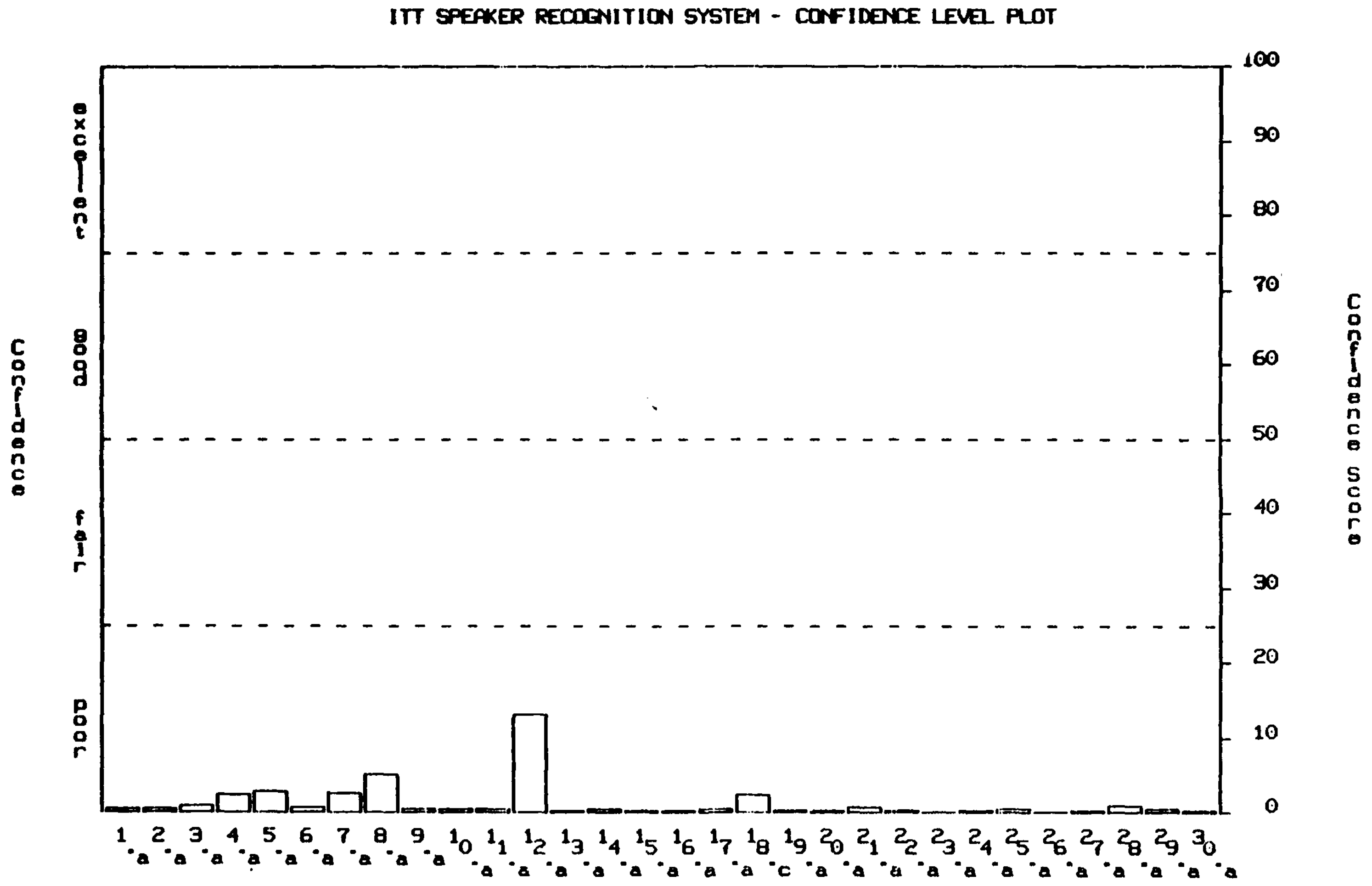


Figure I.24: Test using 20 seconds of speech from speaker12

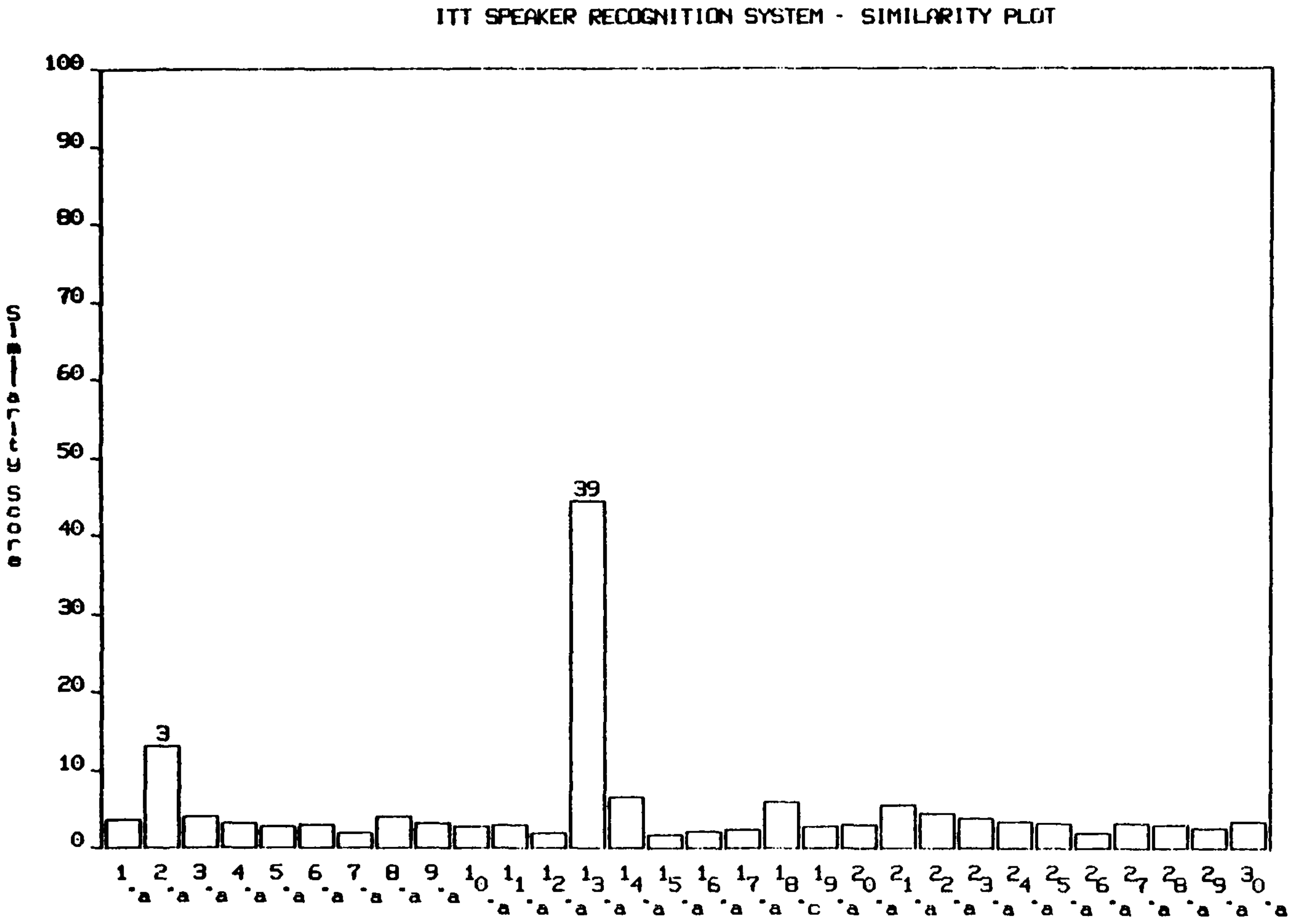


Figure I.25: Test using 20 seconds of speech from speaker13

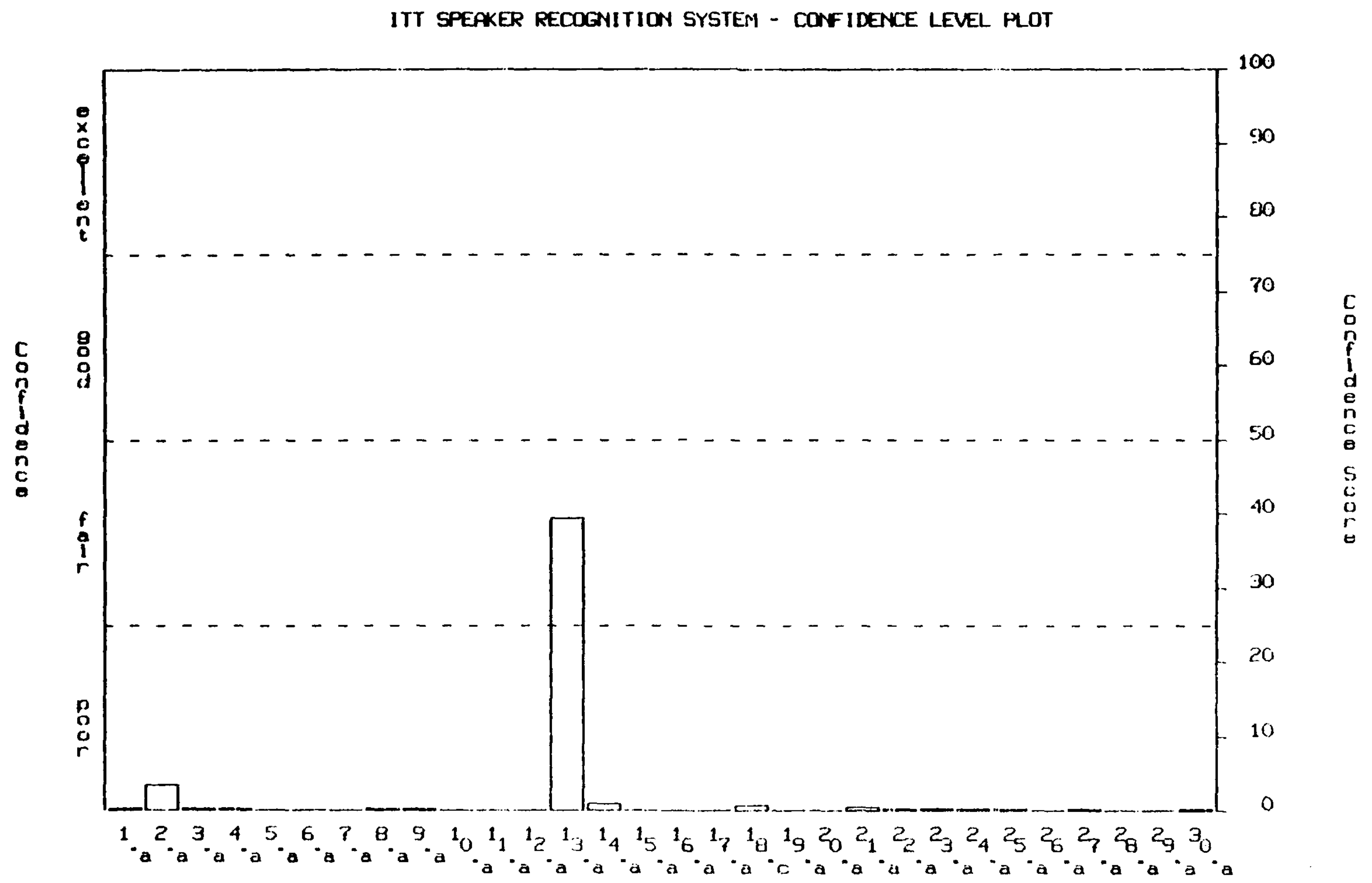


Figure I.26: Test using 20 seconds of speech from speaker13

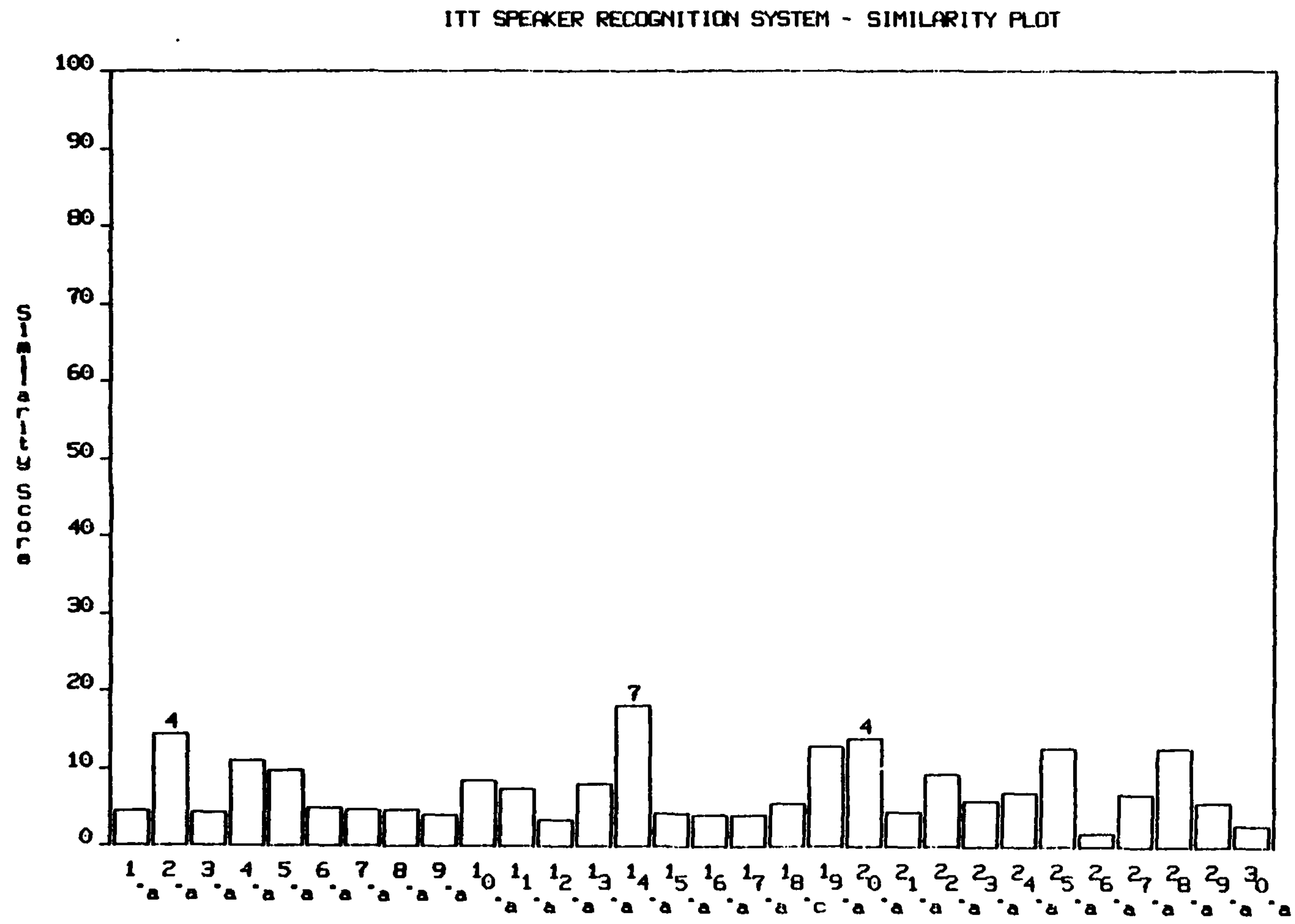


Figure I.27: Test using 20 seconds of speech from speaker14

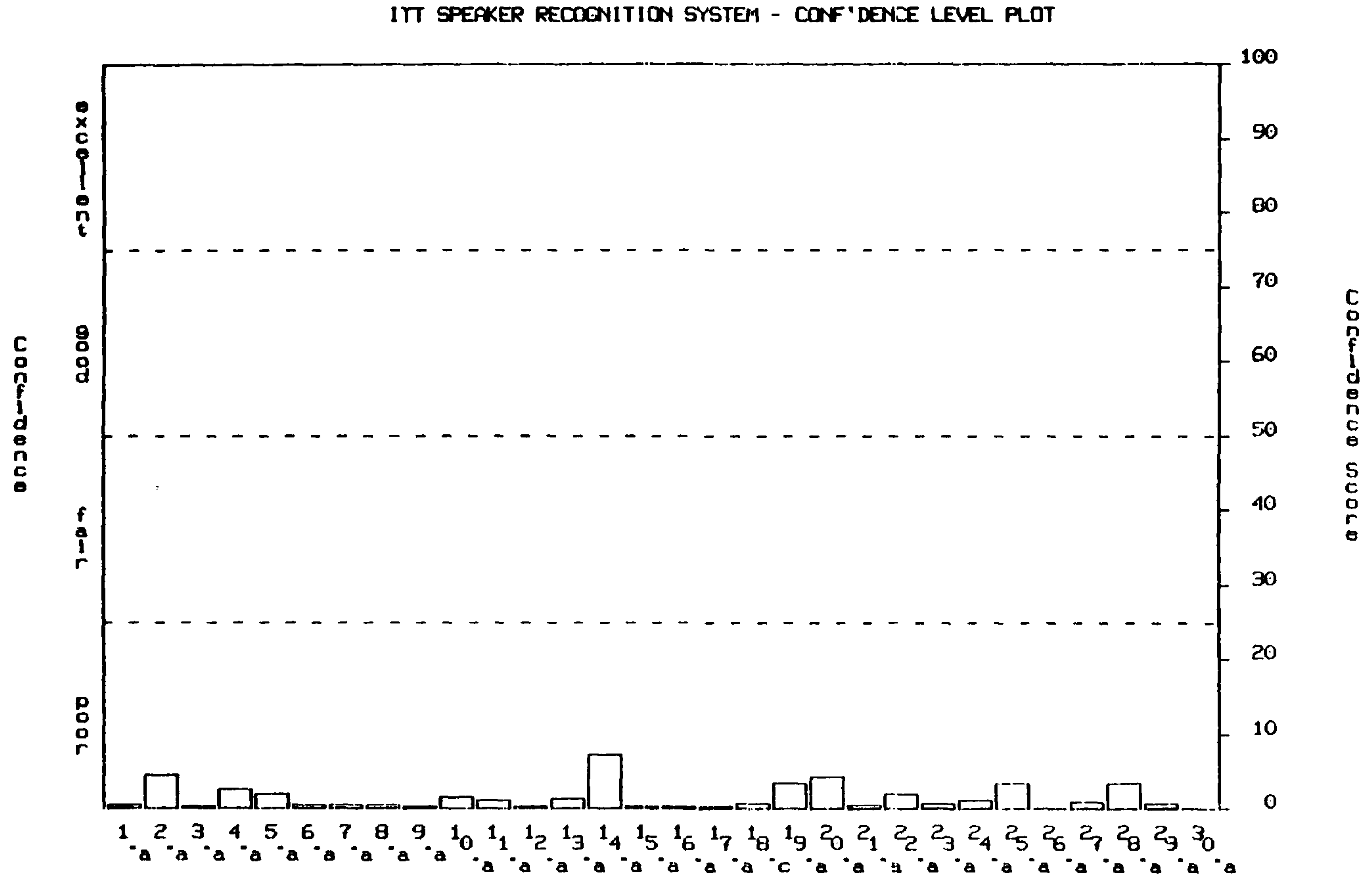


Figure I.28: Test using 20 seconds of speech from speaker14

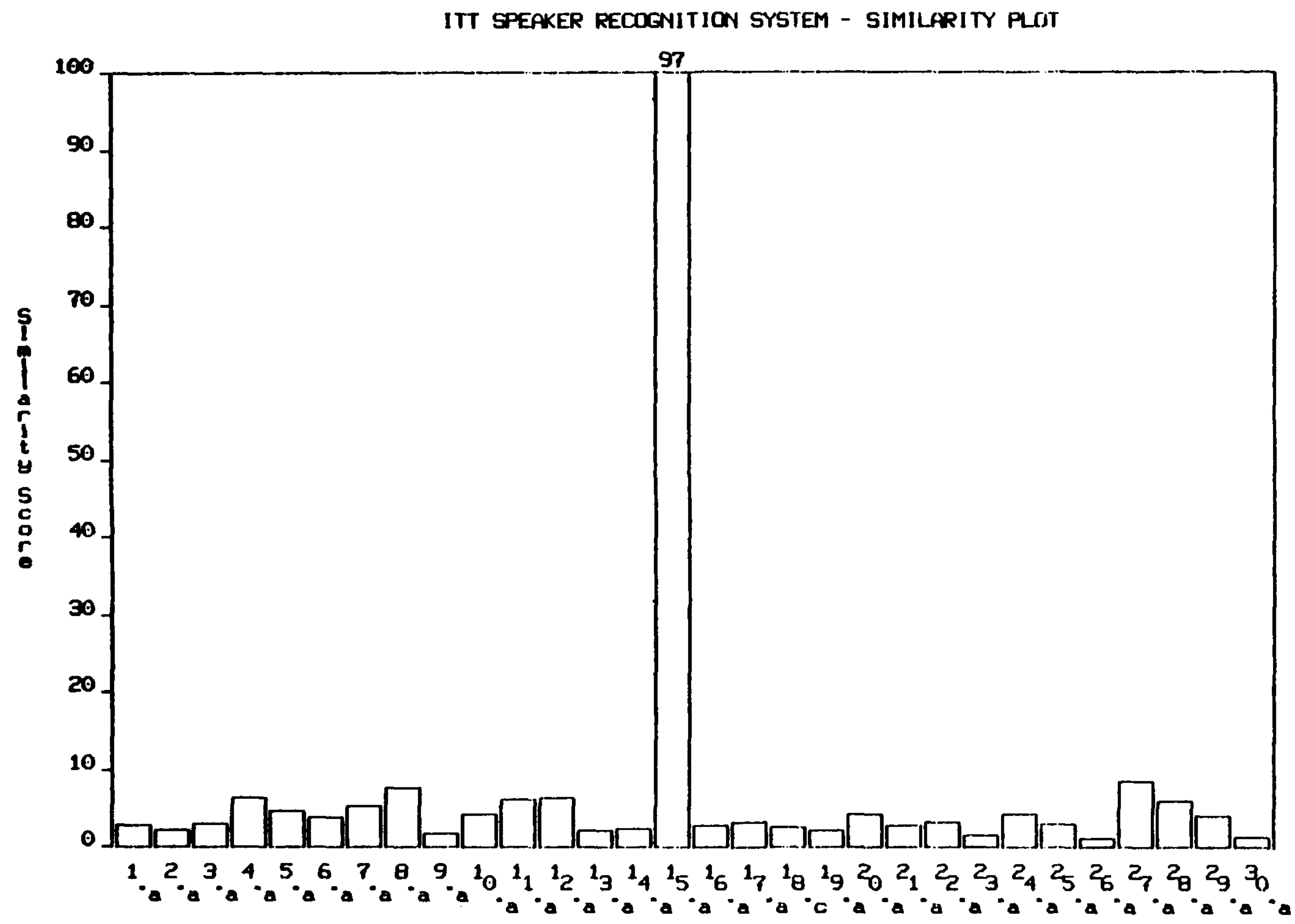


Figure 1.29: Test using 20 seconds of speech from speaker15

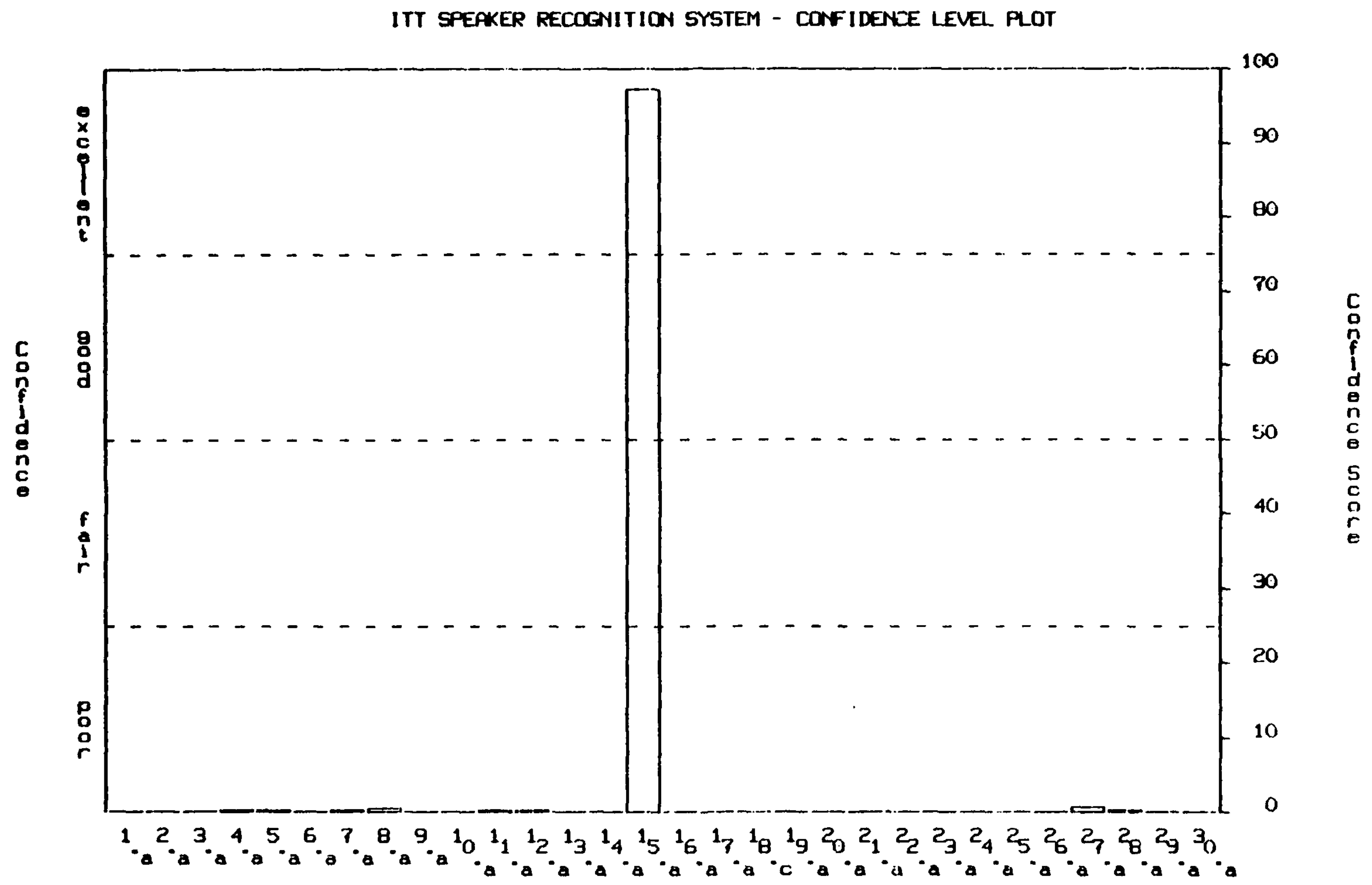


Figure I.30: Test using 20 seconds of speech from speaker15

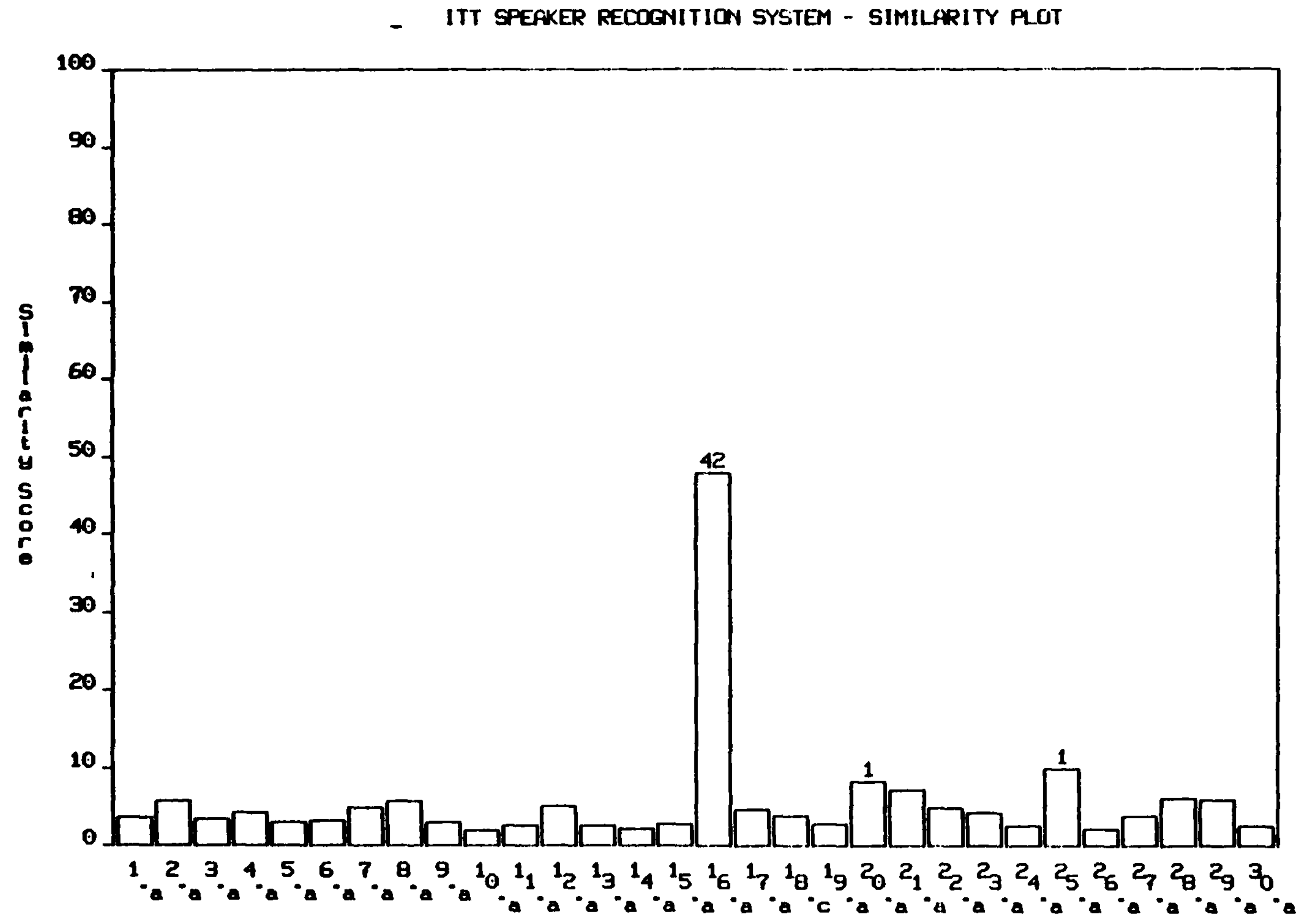


Figure I.31: Test using 20 seconds of speech from speaker16

ITT SPEAKER RECOGNITION SYSTEM - CONFIDENCE LEVEL PLOT

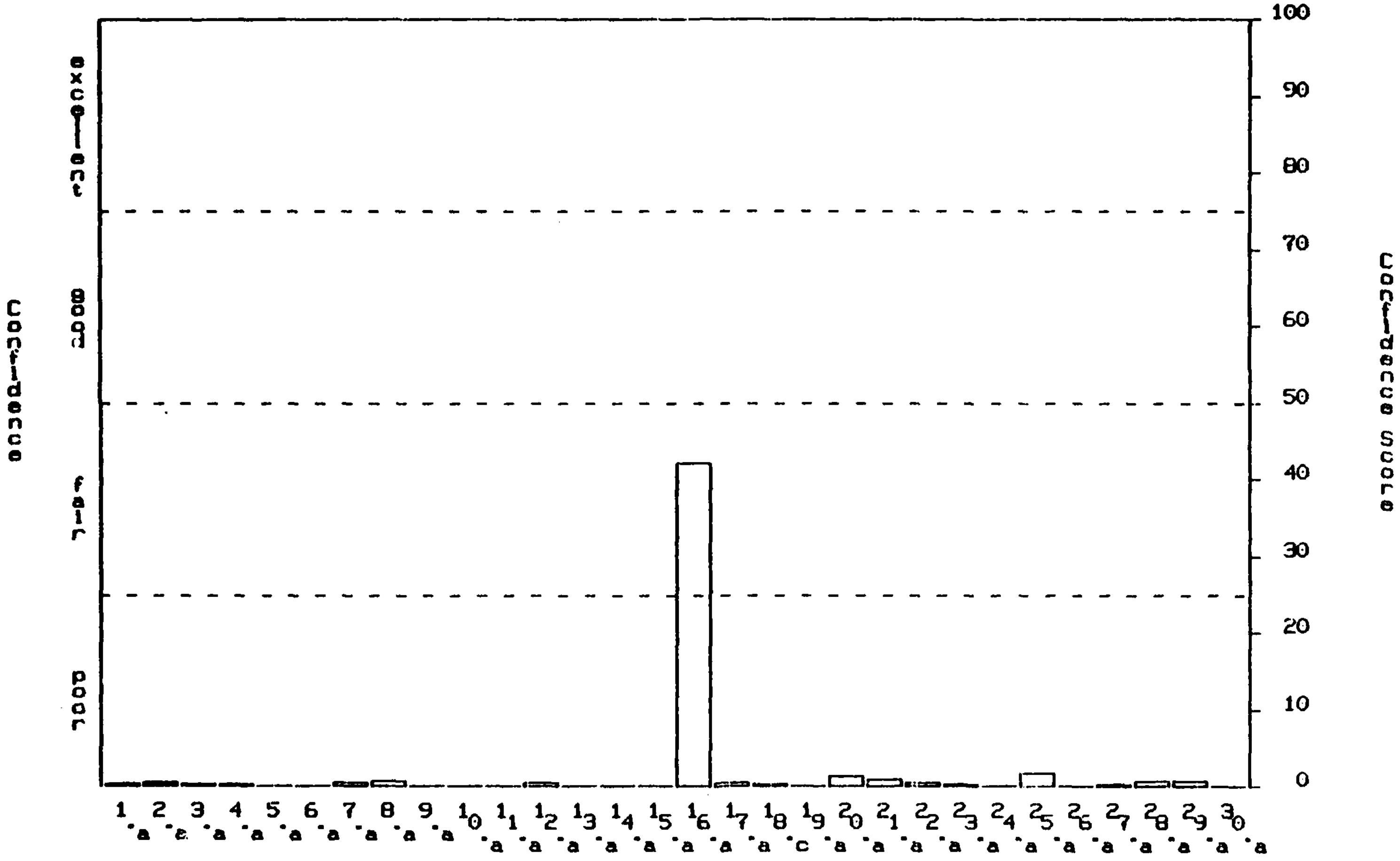


Figure I.32: Test using 20 seconds of speech from speaker16

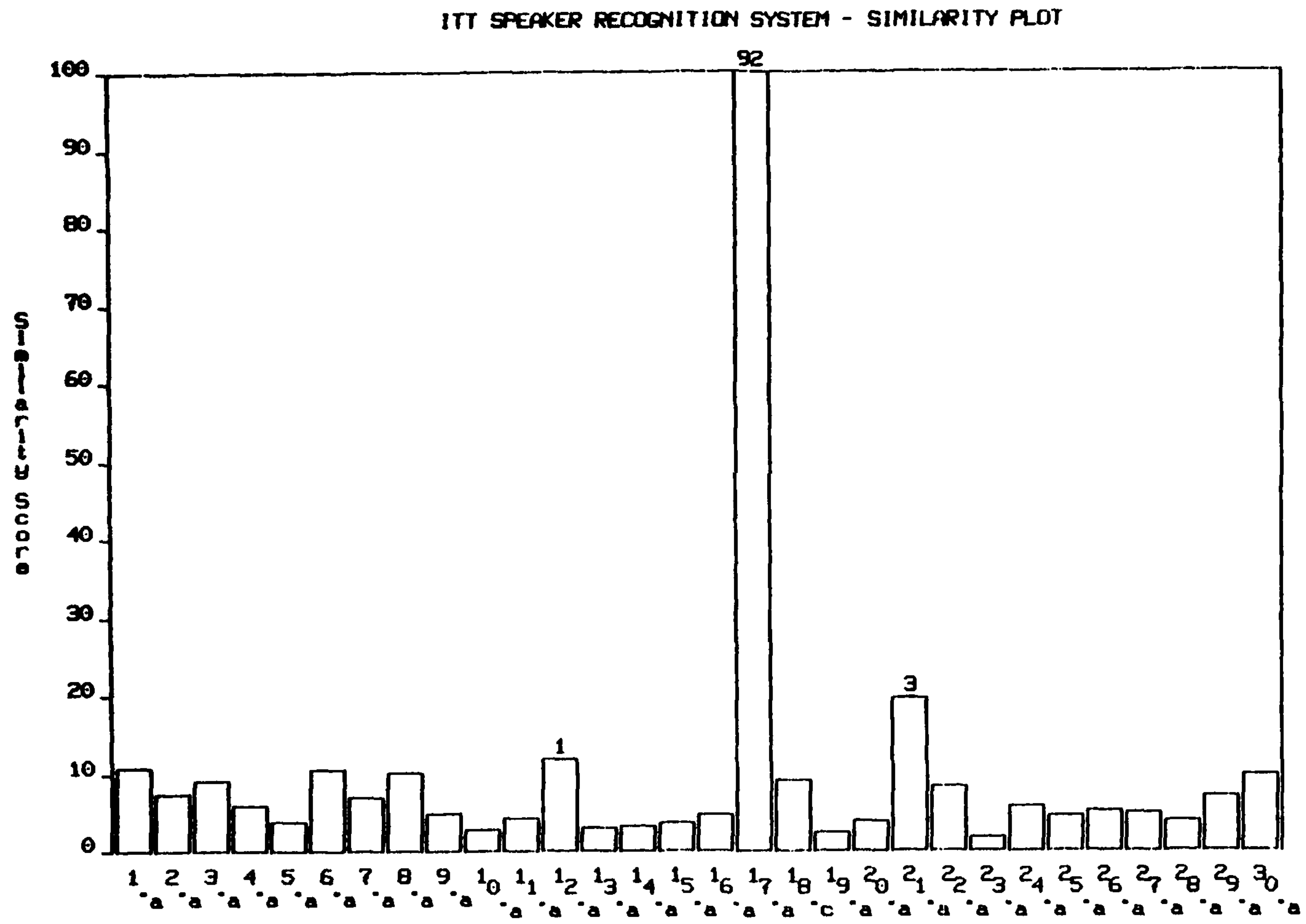


Figure I.33: Test using 20 seconds of speech from speaker17

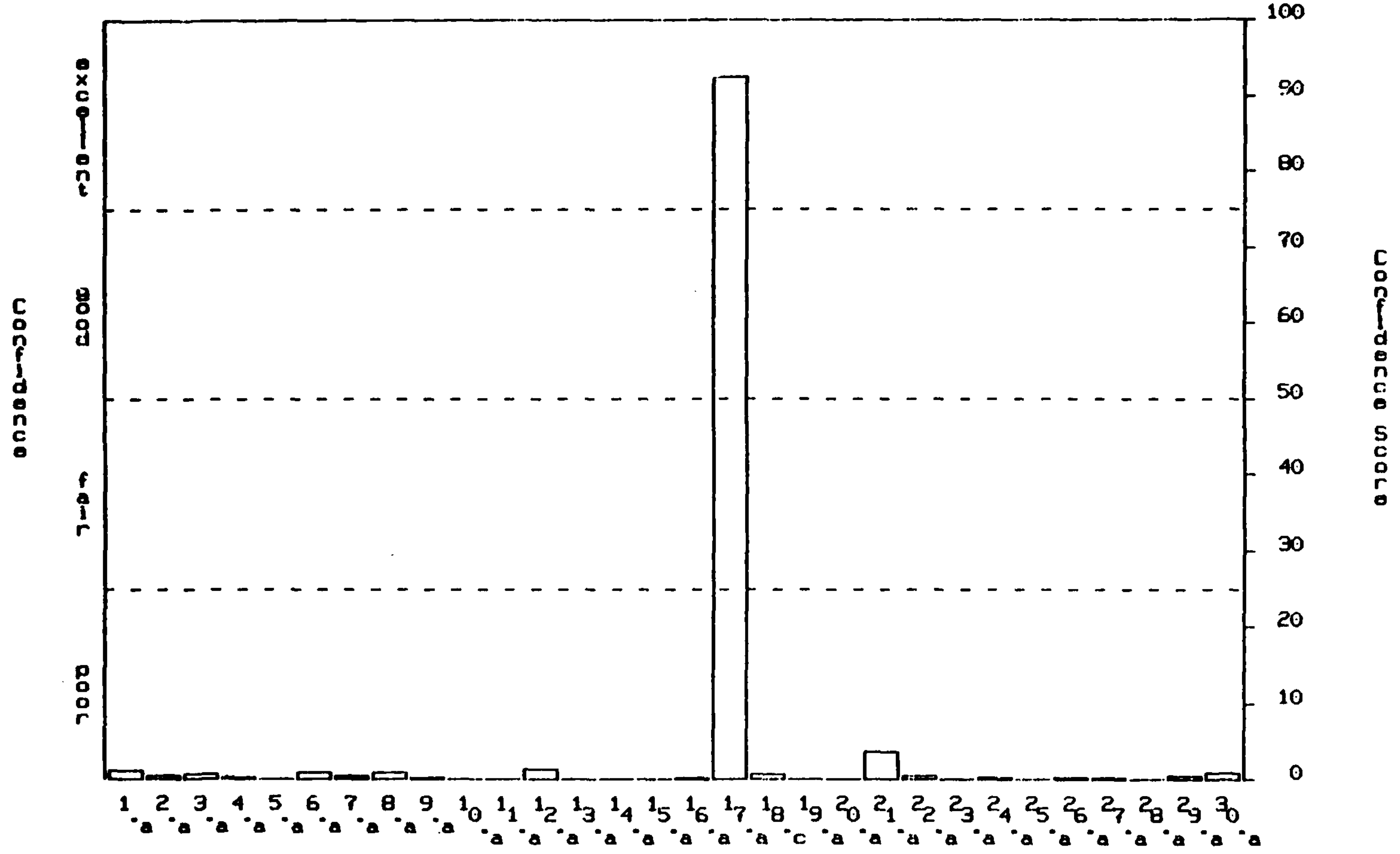

ITT SPEAKER RECOGNITION SYSTEM - CONFIDENCE LEVEL PLOT

Figure I.34: Test using 20 seconds of speech from speaker17

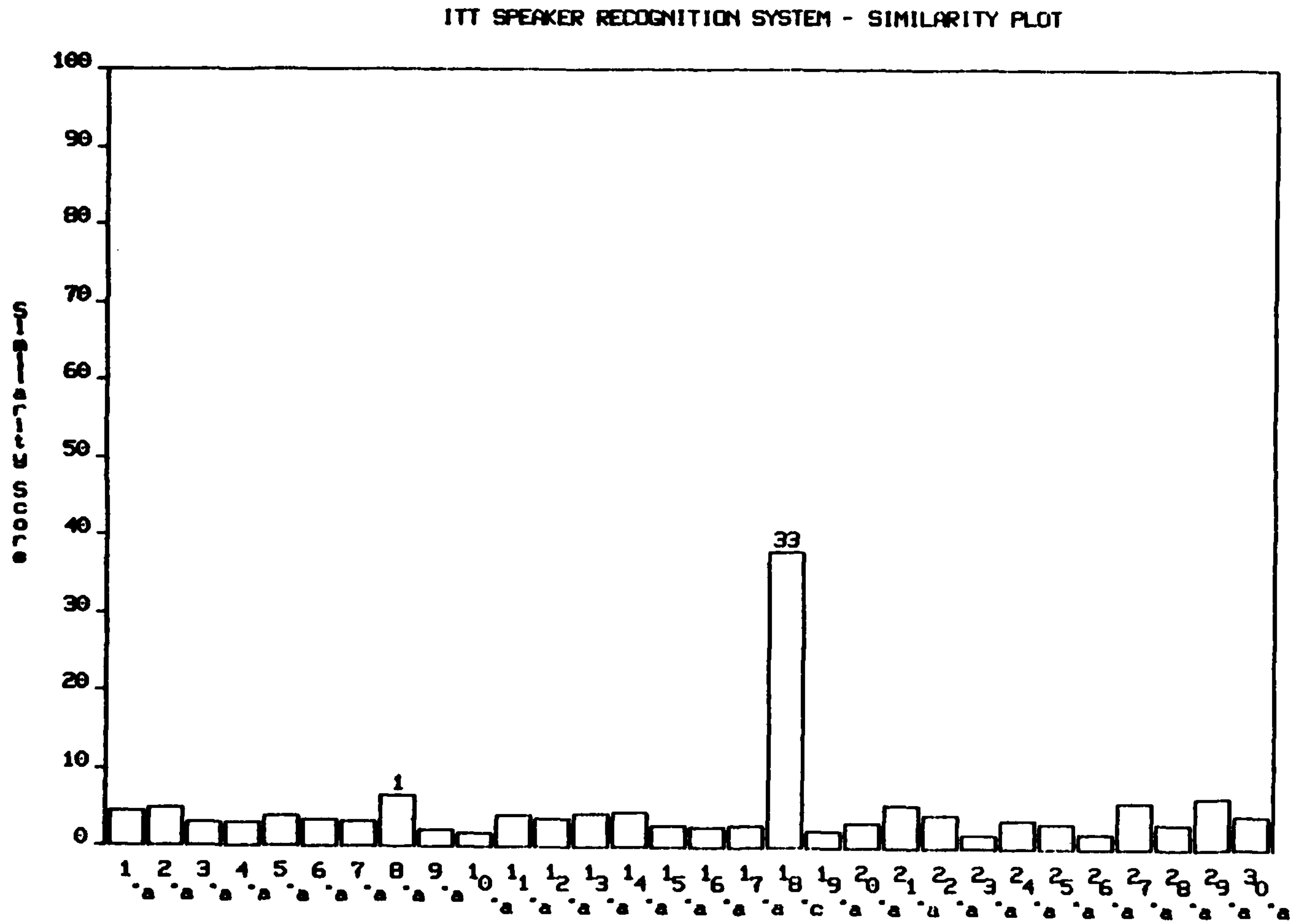


Figure I.35: Test using 20 seconds of speech from speaker18

ITT SPEAKER RECOGNITION SYSTEM - CONFIDENCE LEVEL PLOT

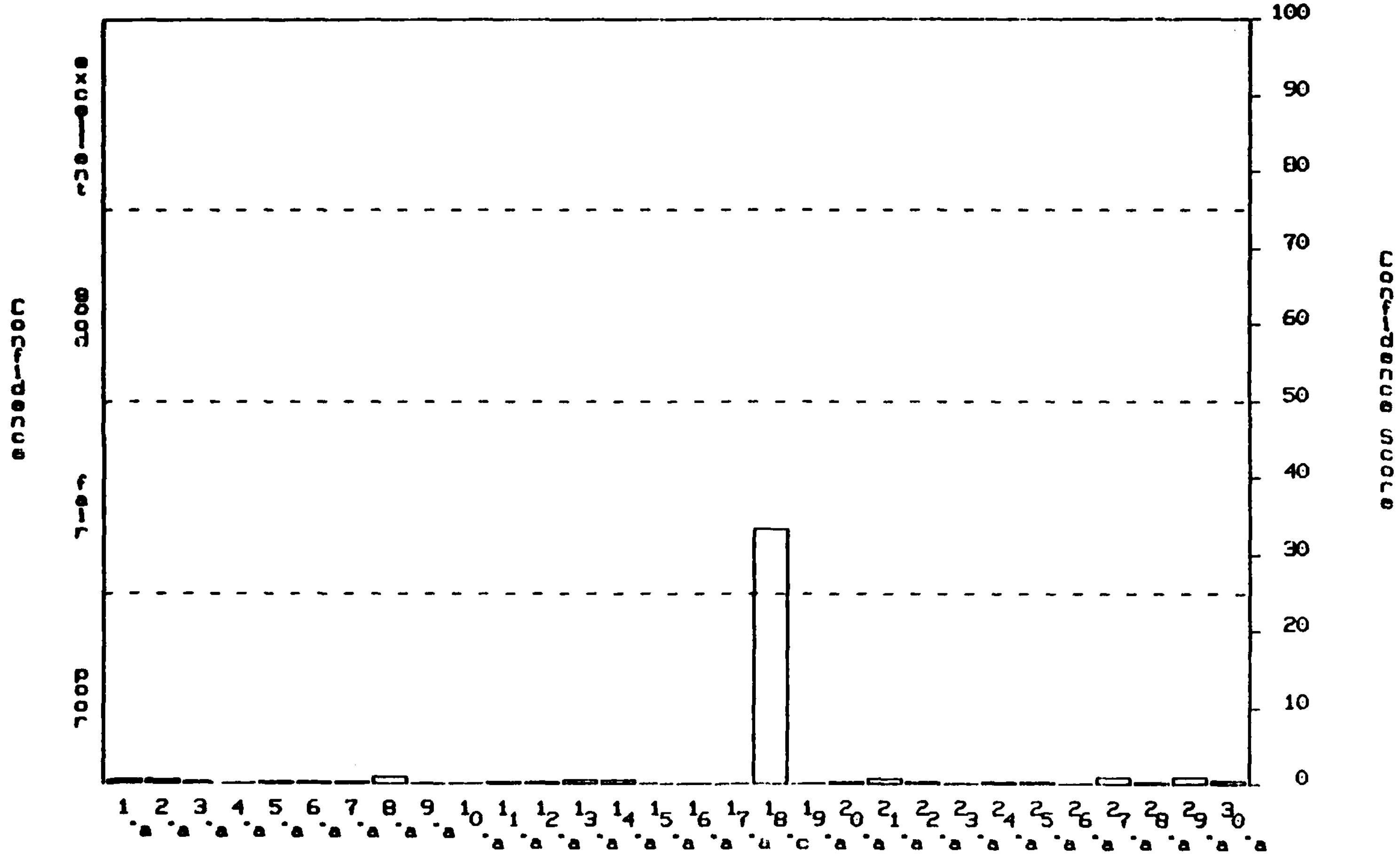


Figure 1.36: Test using 20 seconds of speech from speaker18

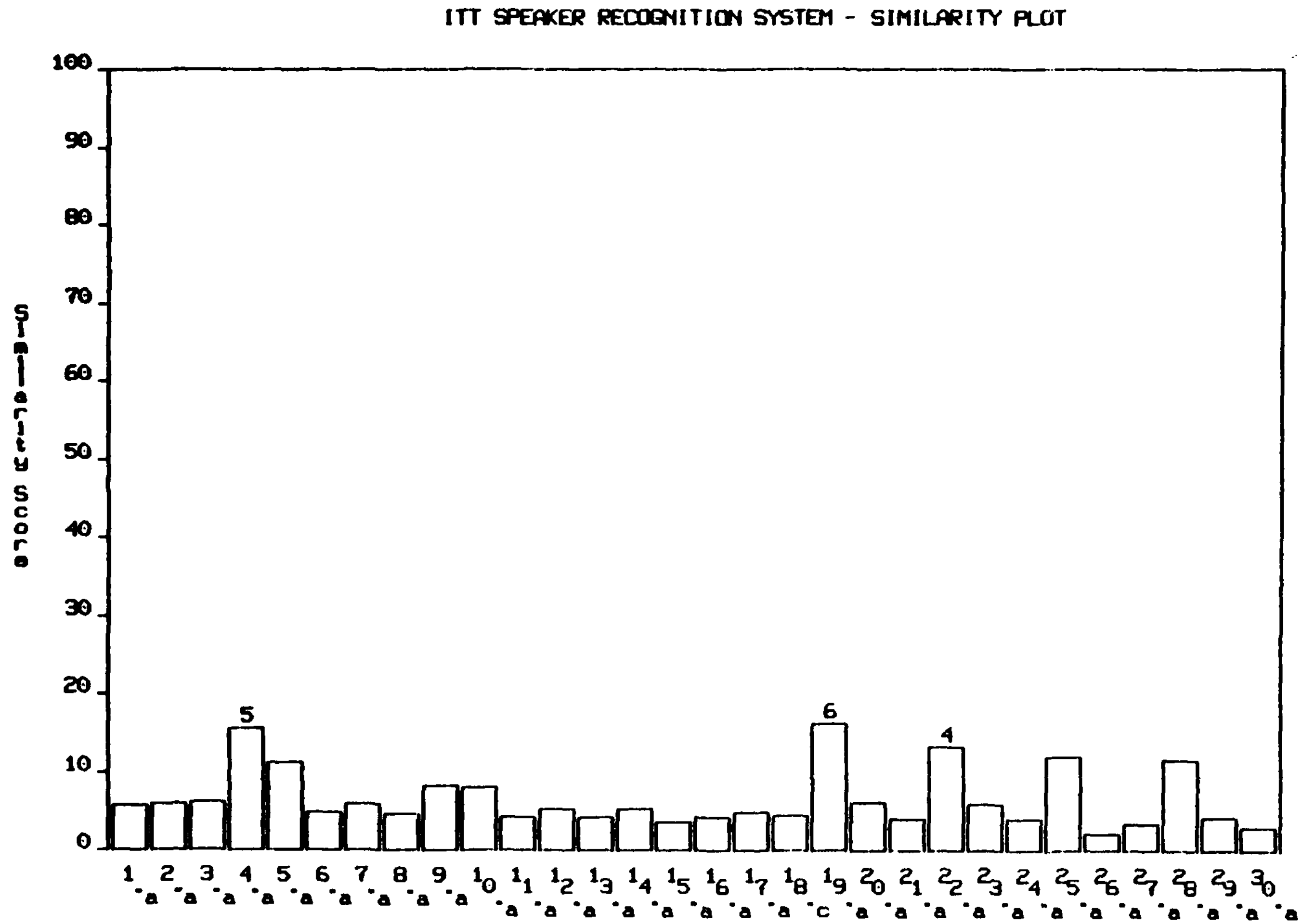


Figure I.37: Test using 20 seconds of speech from speaker19

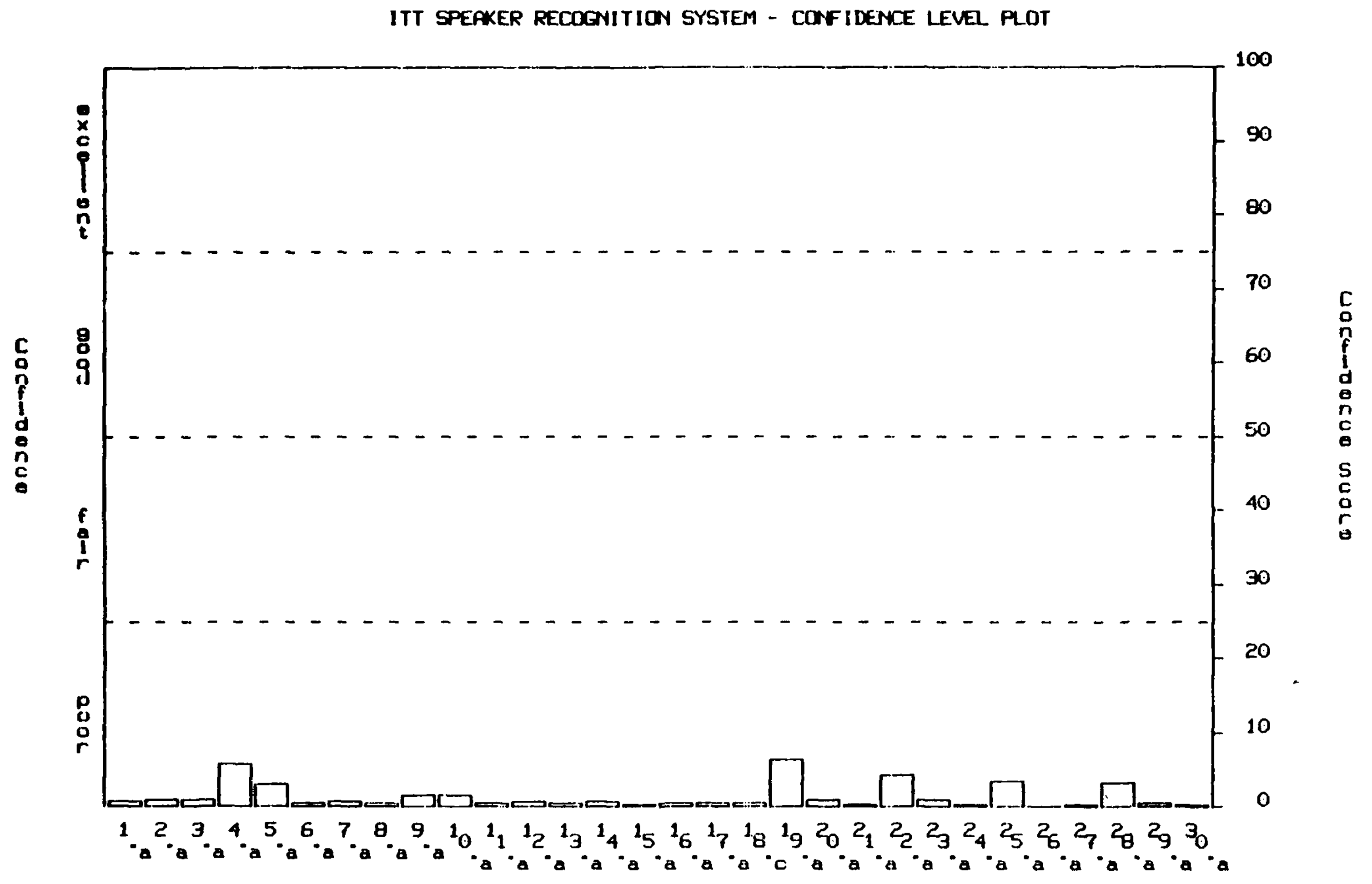


Figure I.38: Test using 20 seconds of speech from speaker19

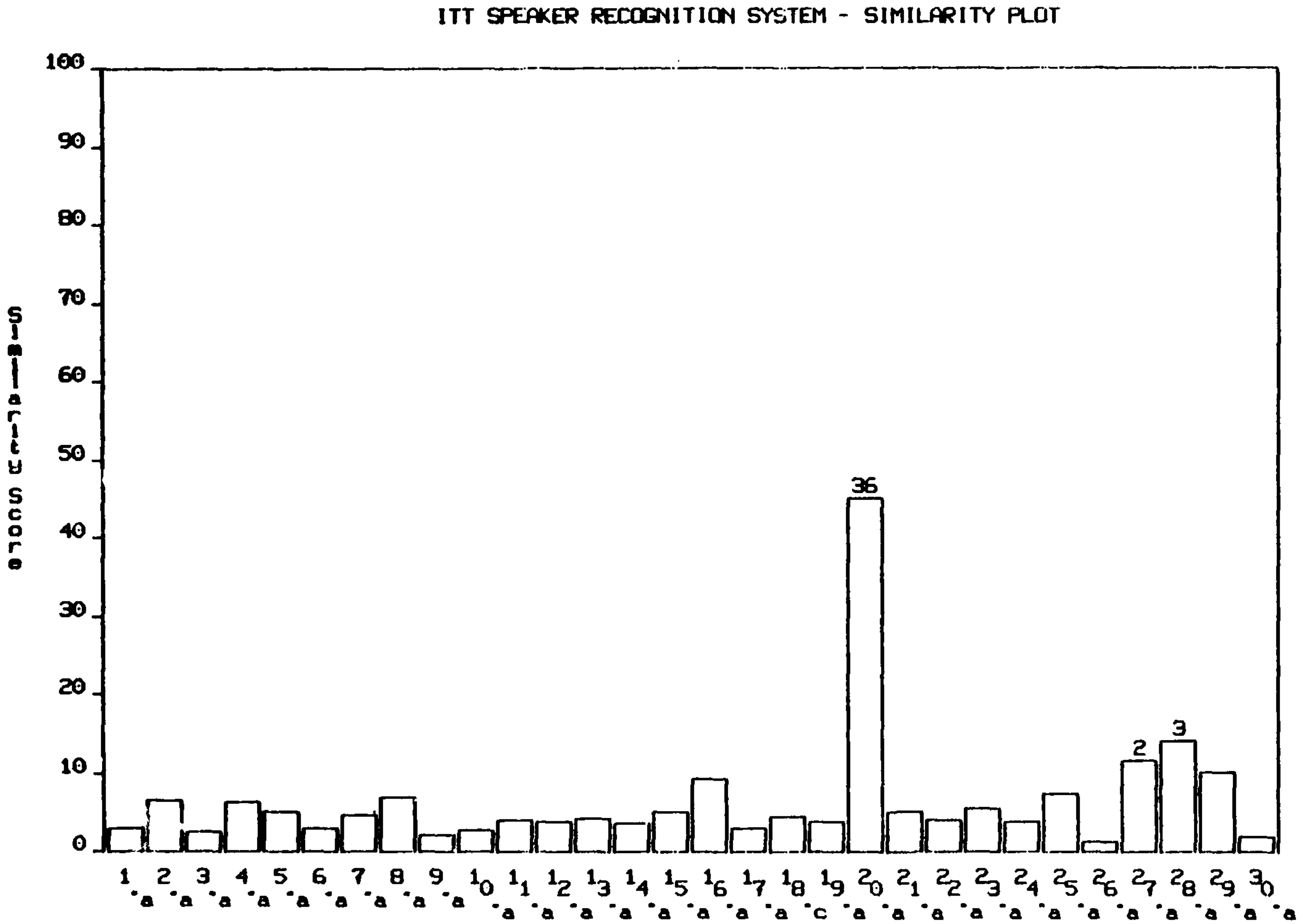


Figure I.39: Test using 20 seconds of speech from speaker20

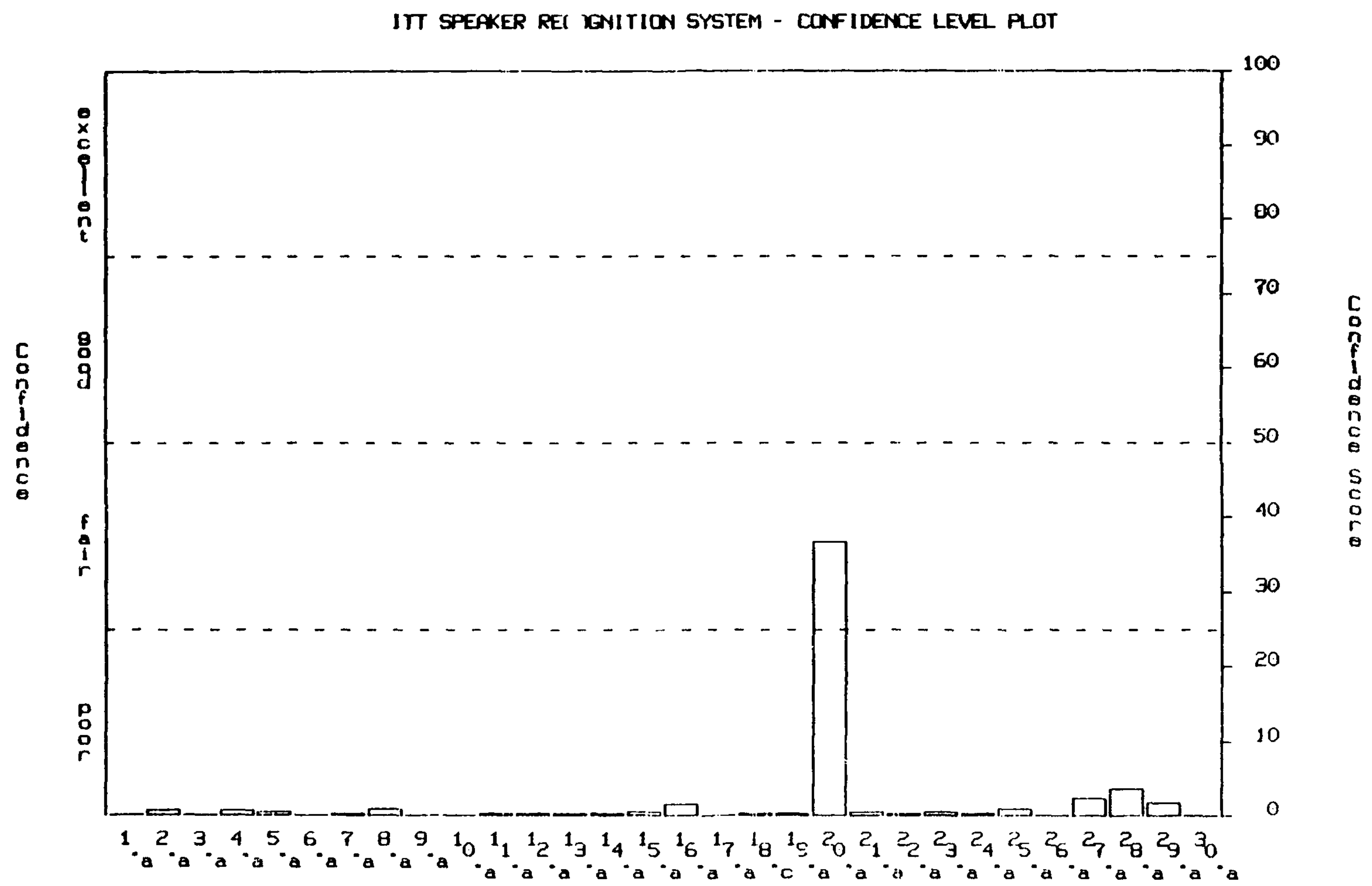


Figure I.40:   Test using 20 seconds of speech from speaker20

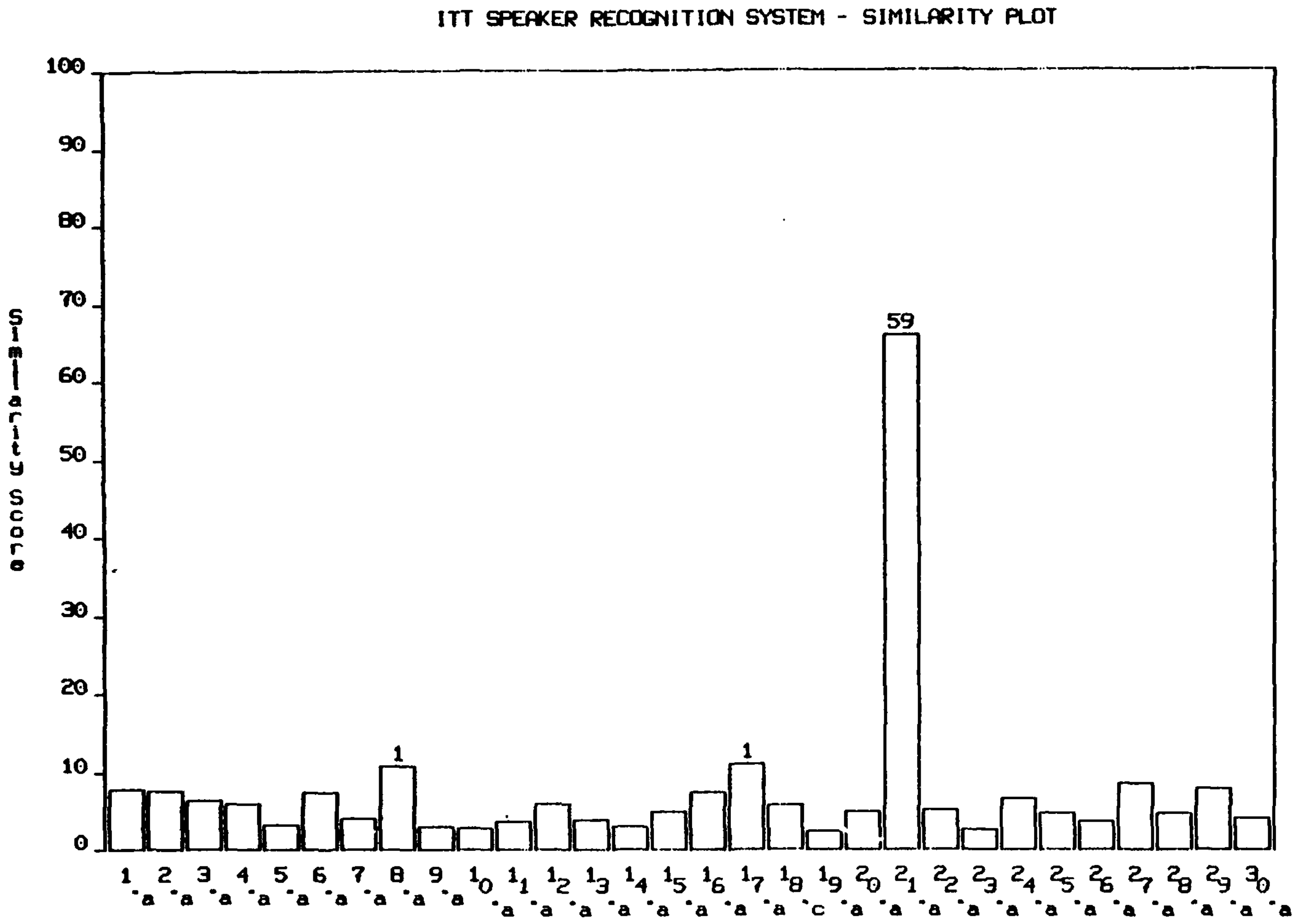


Figure I.41: Test using 20 seconds of speech from speaker21

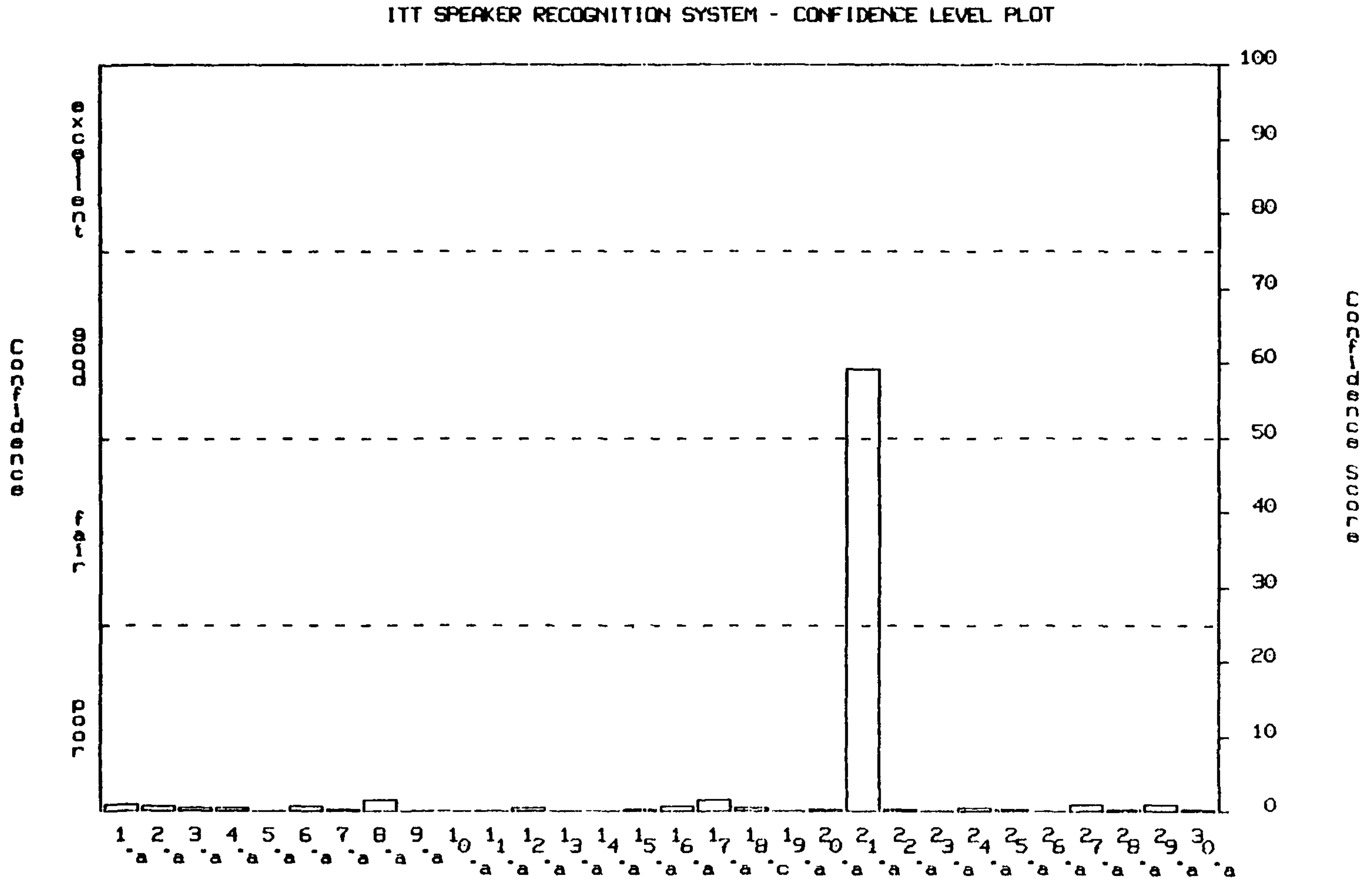


Figure I.42: Test using 20 seconds of speech from speaker21

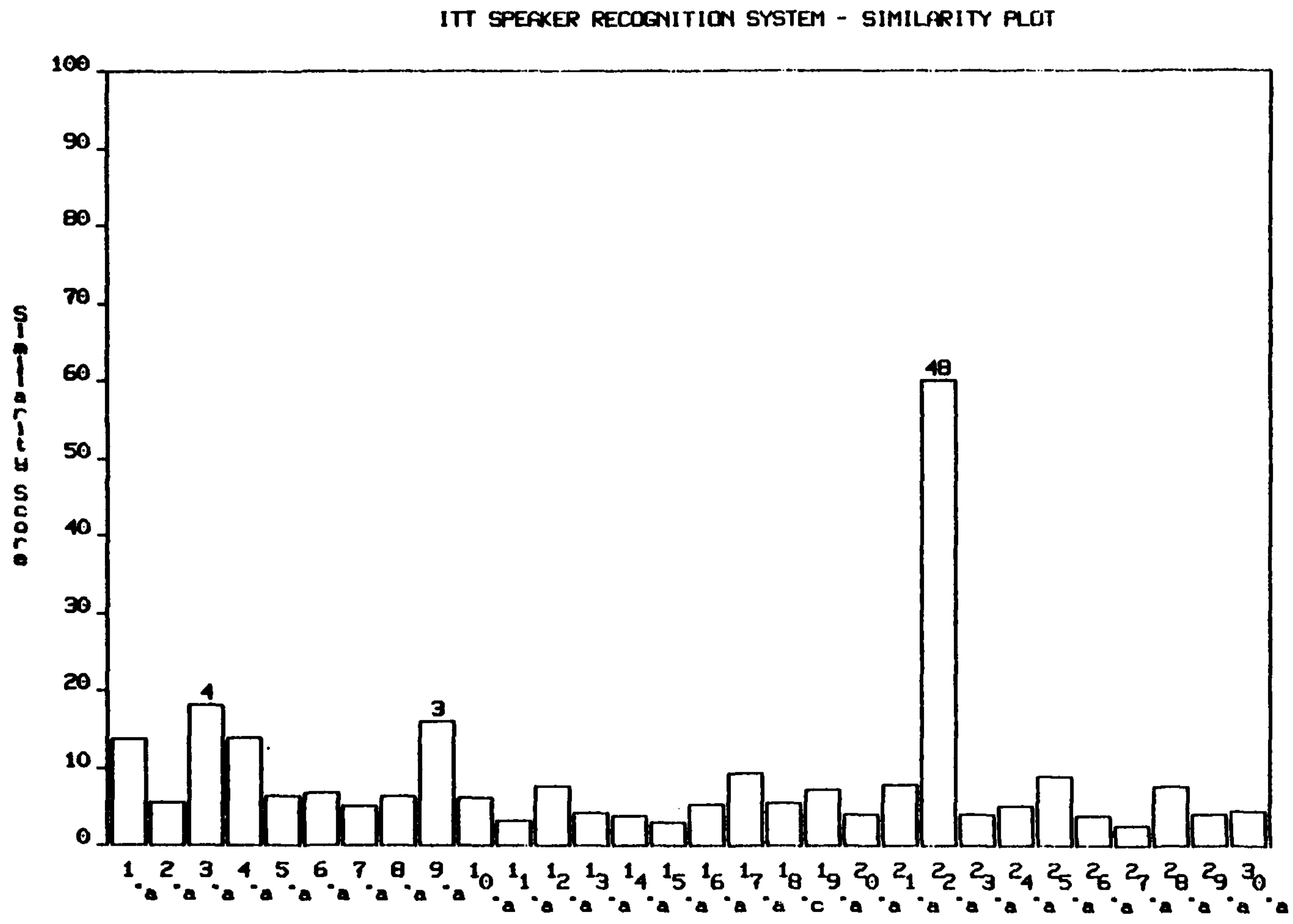


Figure I.43: Test using 20 seconds of speech from speaker22

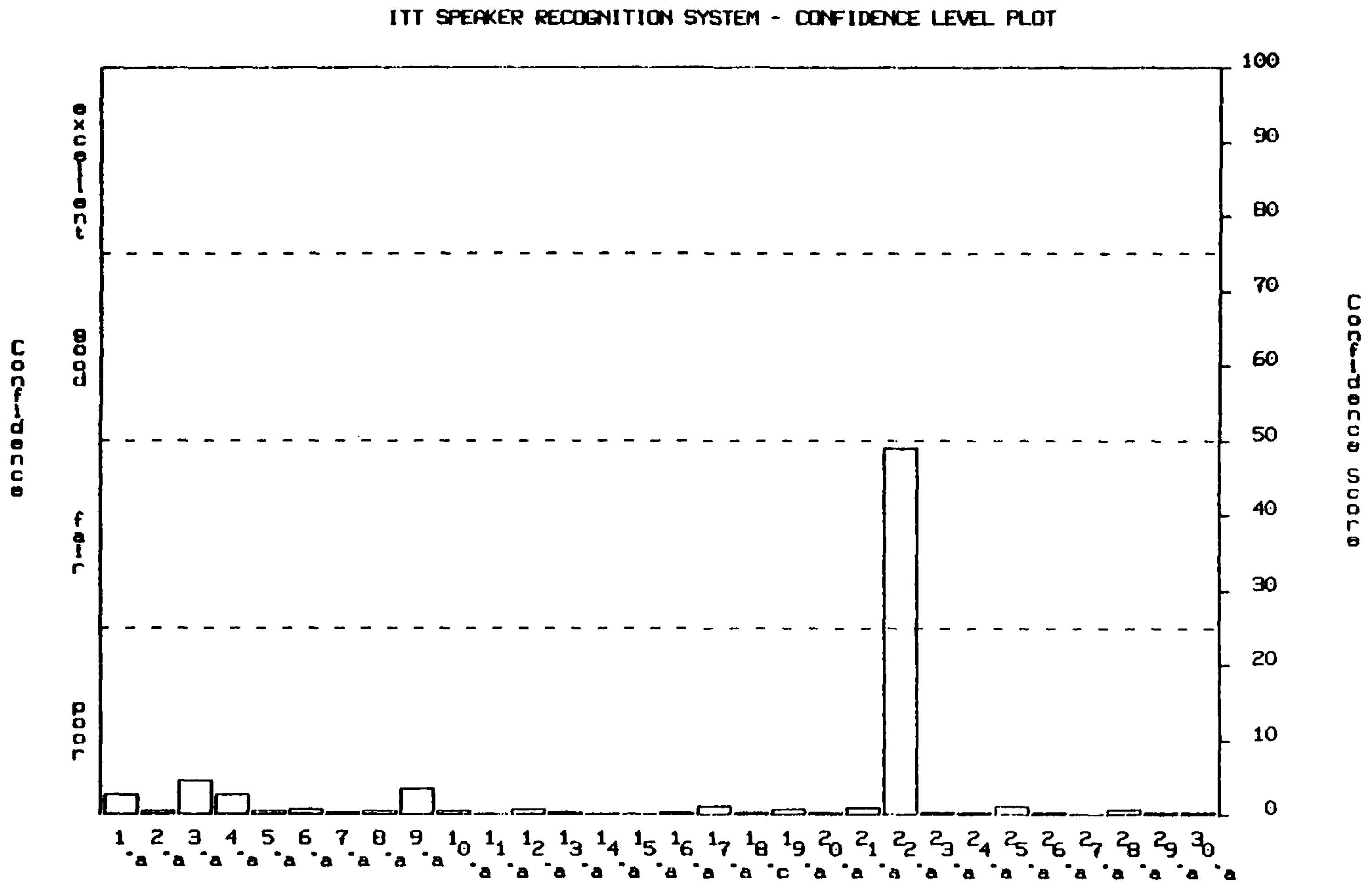


Figure I.44: Test using 20 seconds of speech from speaker22

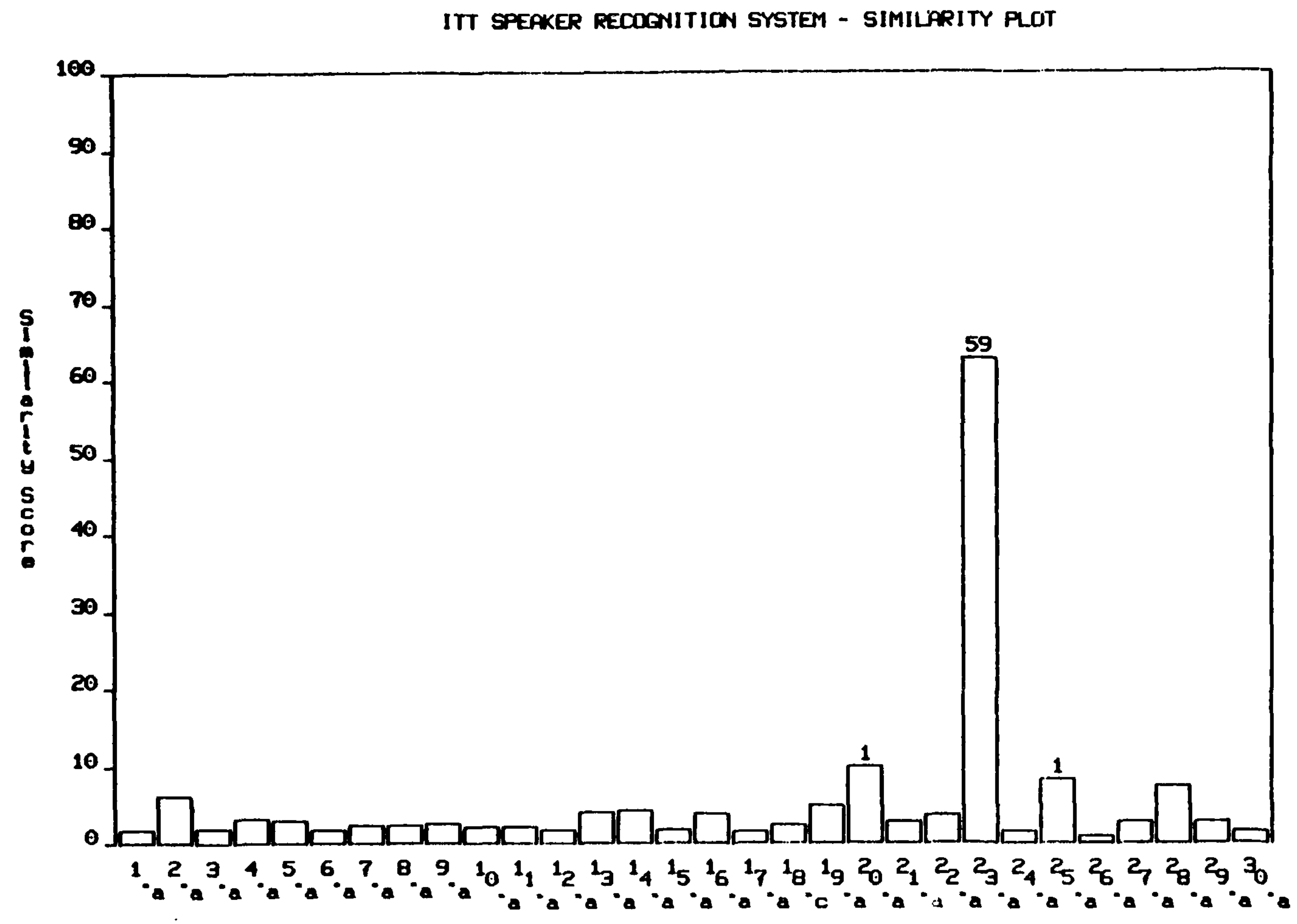


Figure I.45: Test using 20 seconds of speech from speaker23

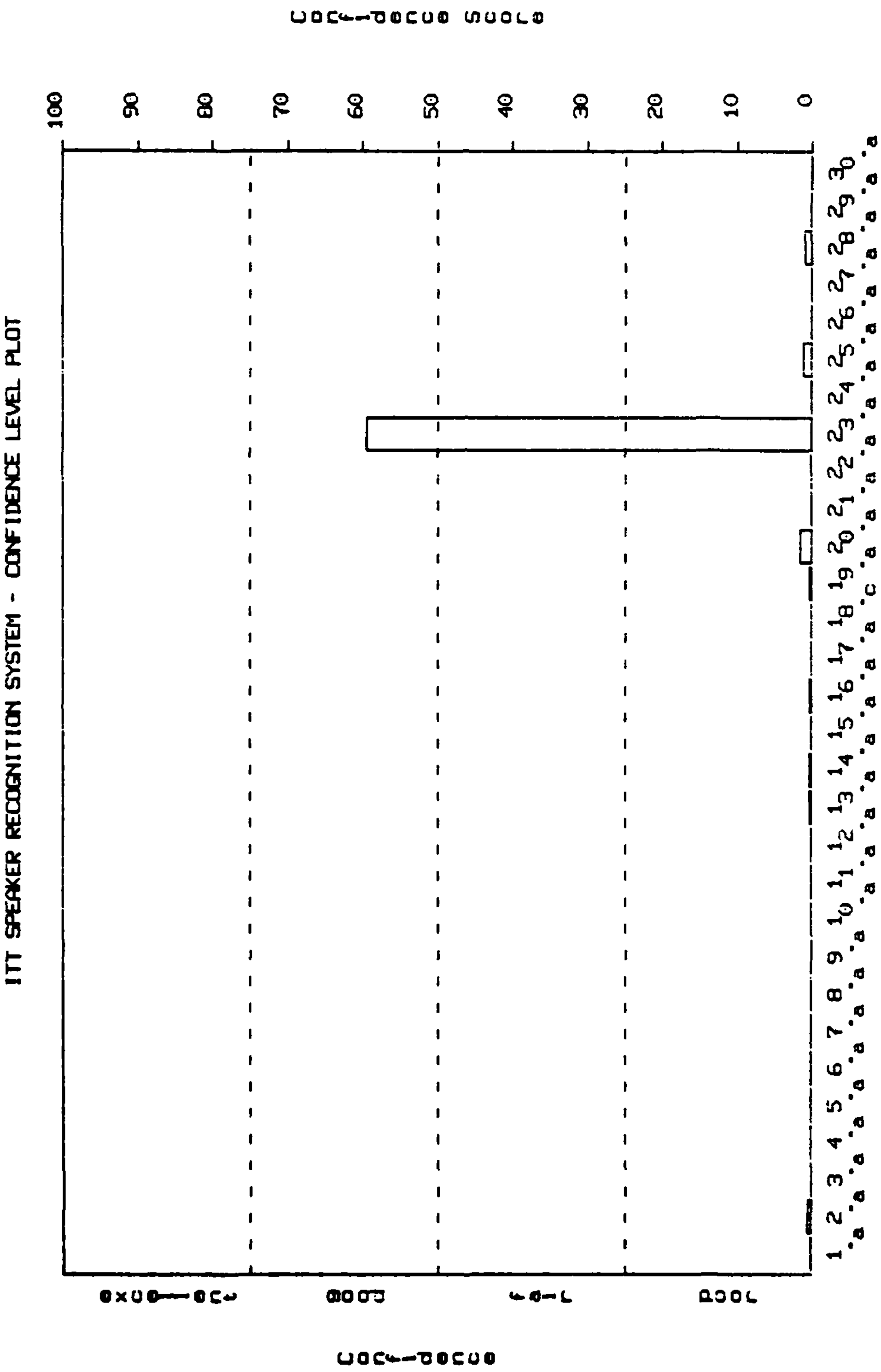


Figure I.46: Test using 20 seconds of speech from speaker23

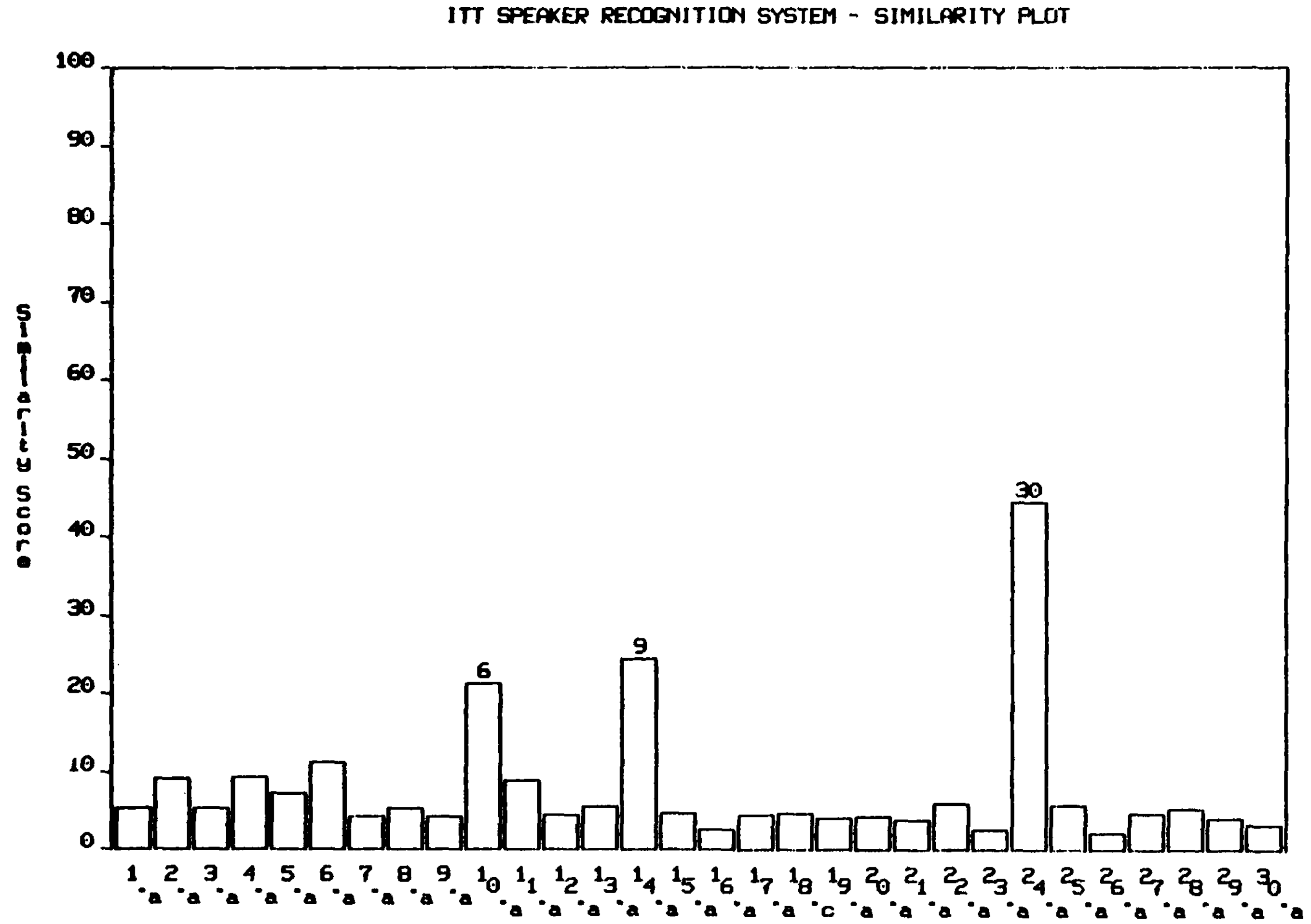


Figure I.47: Test using 20 seconds of speech from speaker24

ITT SPEAKER RECOGNITION SYSTEM - CONFIDENCE LEVEL PLOT

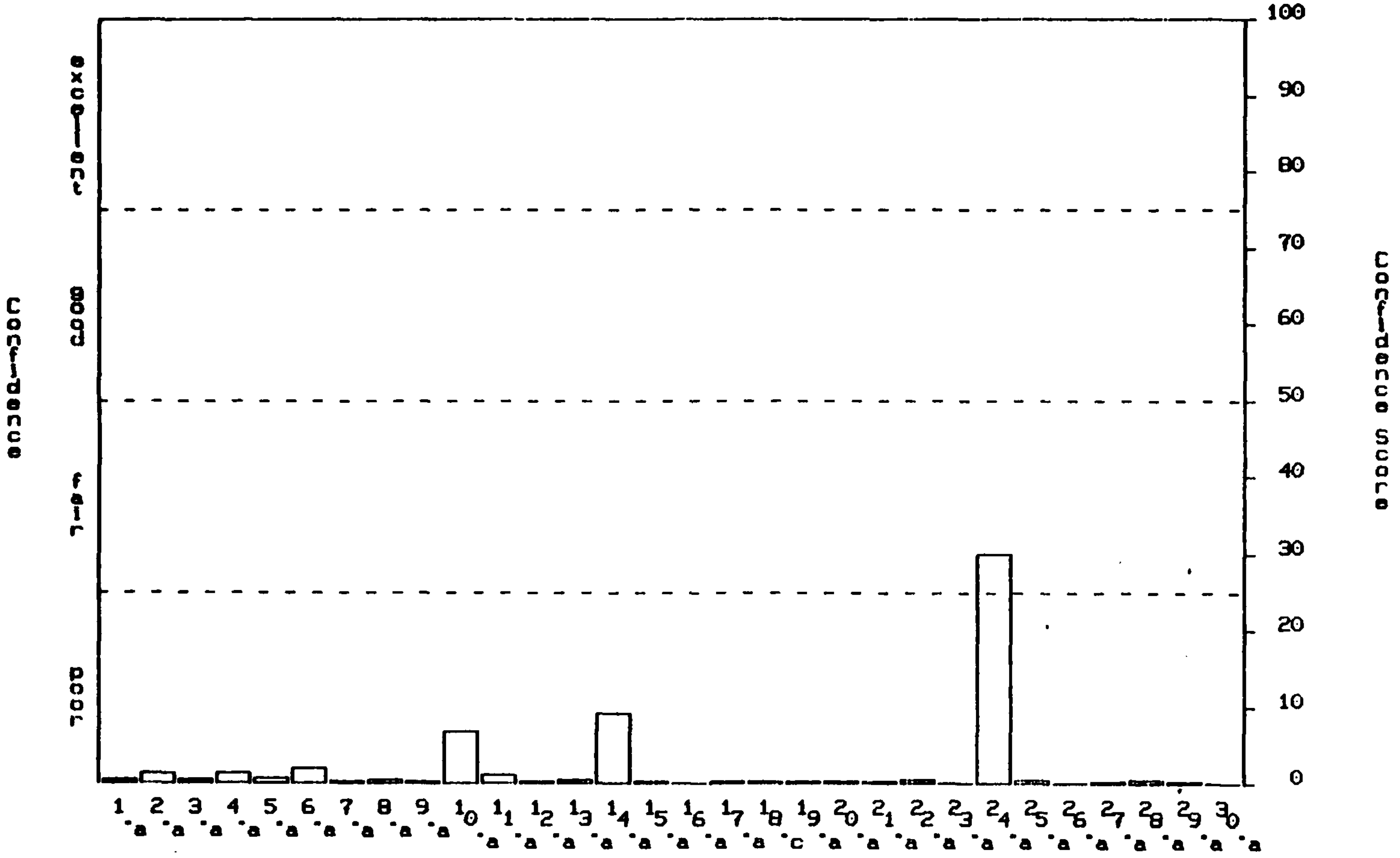


Figure I.48: Test using 20 seconds of speech from spaaker24

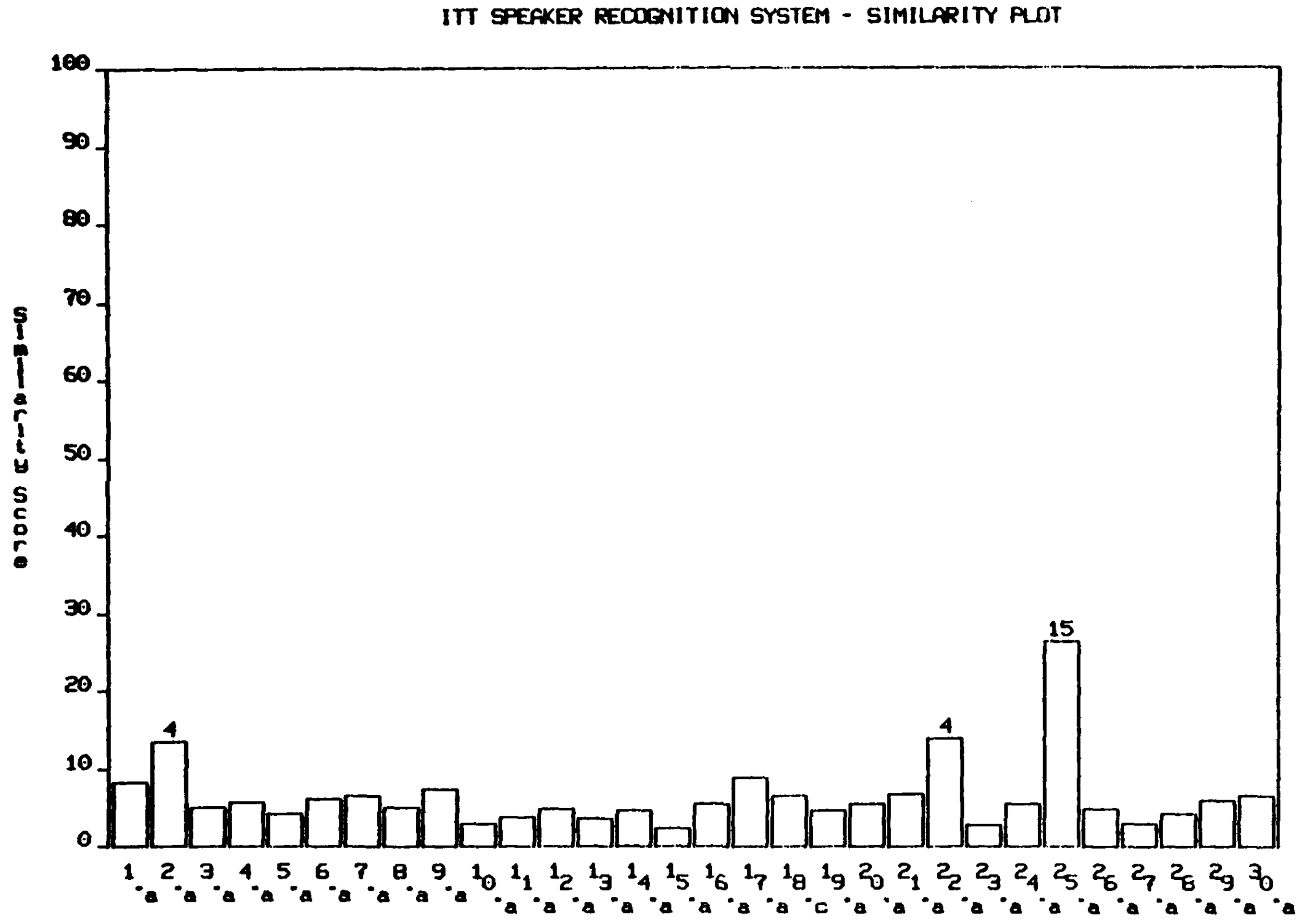


Figure I.49: Test using 20 seconds of speech from speaker-25

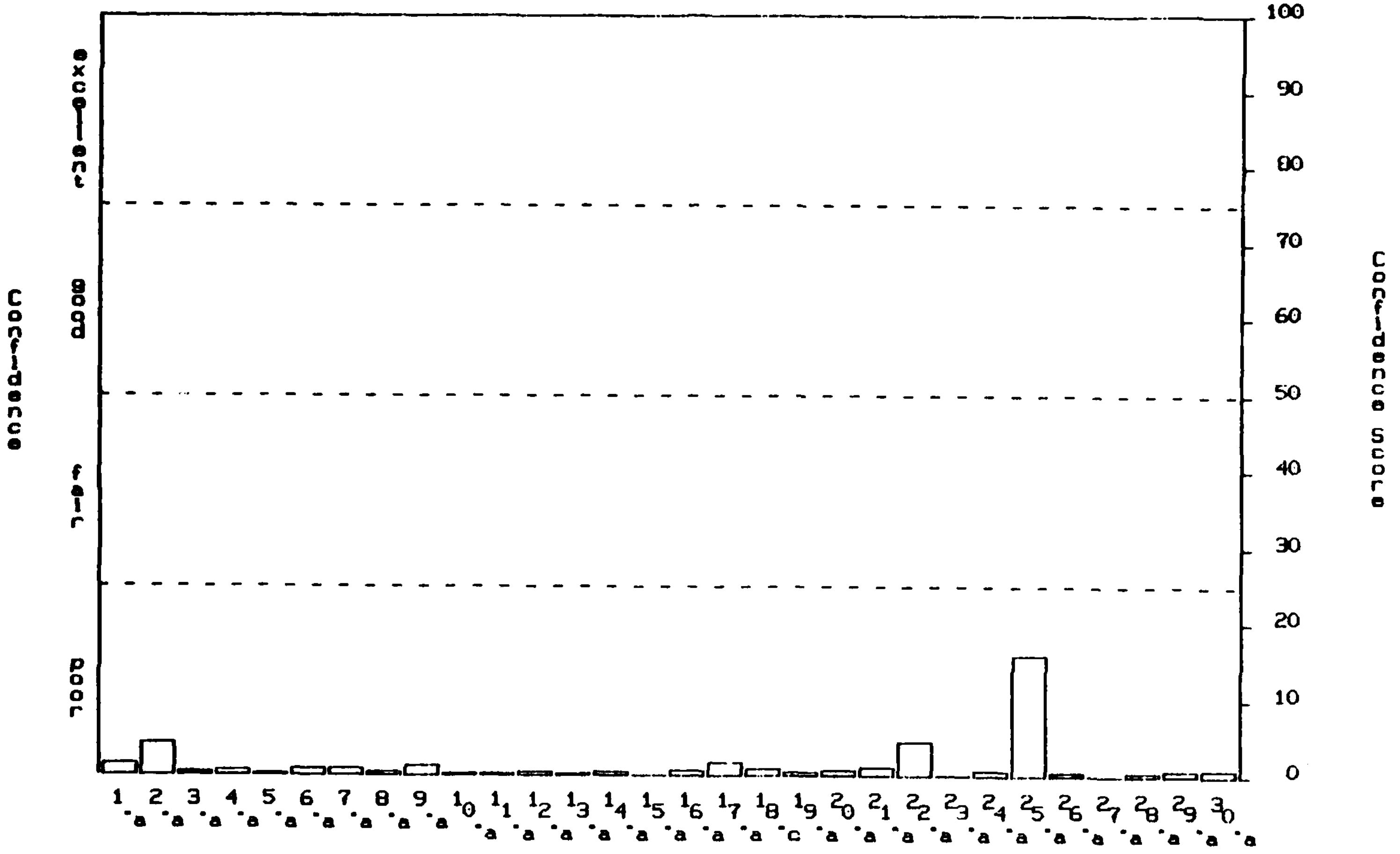

ITT SPEAKER RECOGNITION SYSTEM - CONFIDENCE LEVEL PLOT

Figure I.50: Test using 20 seconds of speech from speaker25

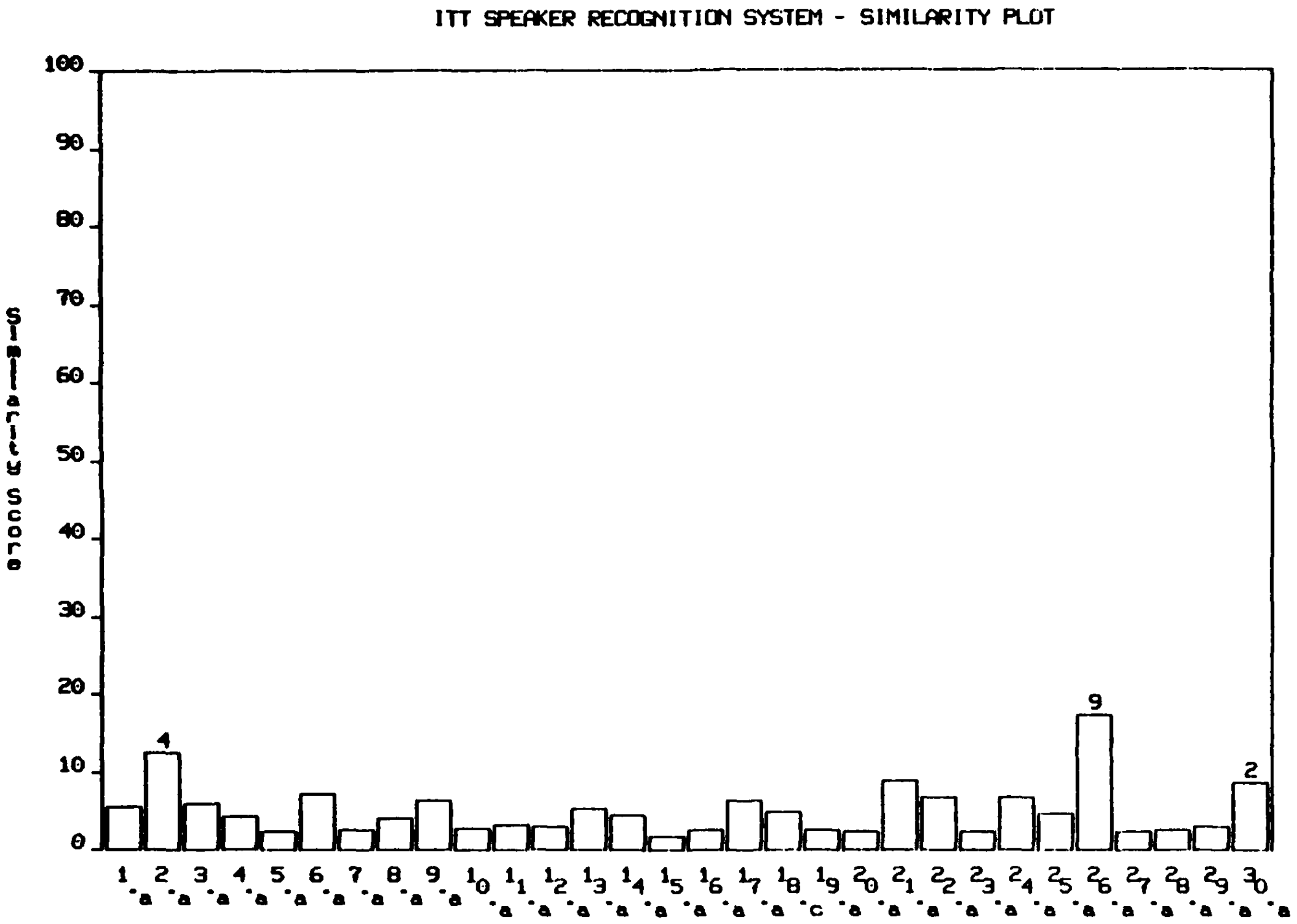

ITT SPEAKER RECOGNITION SYSTEM - SIMILARITY PLOT

Figure I.51: Test using 20 seconds of speech from speaker26

ITT SPEAKER RECOGNITION SYSTEM - CONFIDENCE LEVEL PLOT

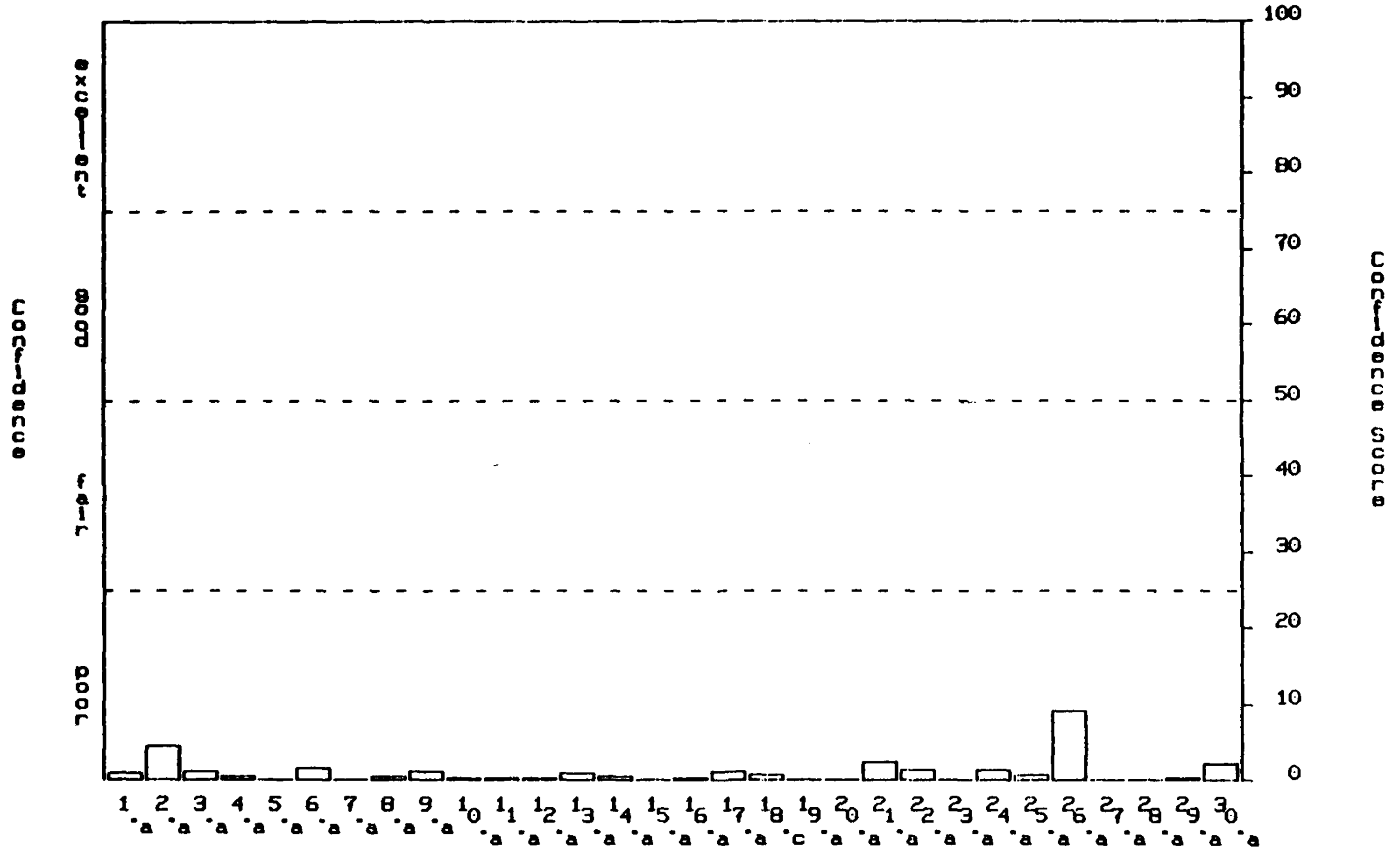


Figure I.52: Test using 20 seconds of speech from spsaker26

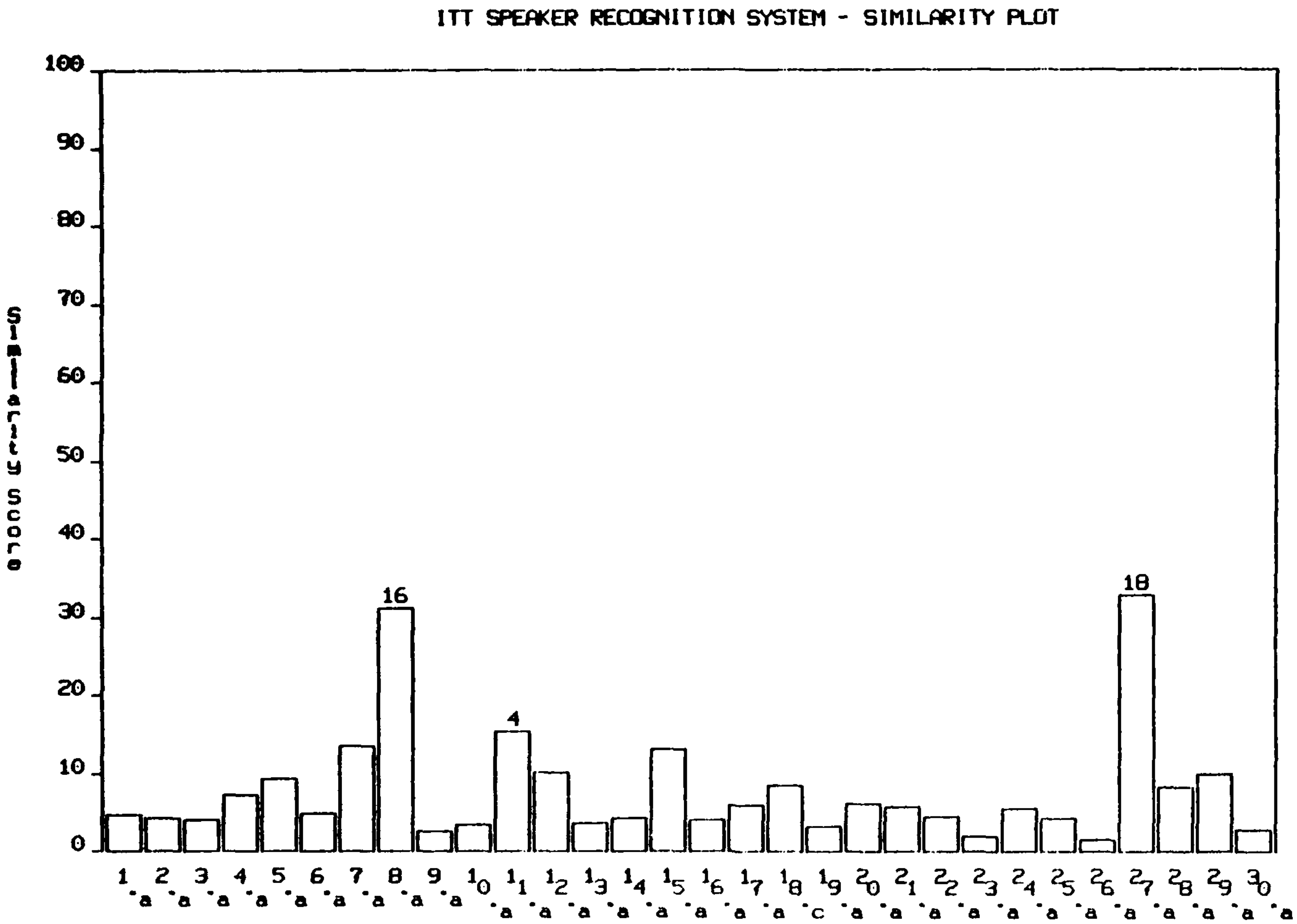


Figure I.53: Test using 20 seconds of speech from speaker27

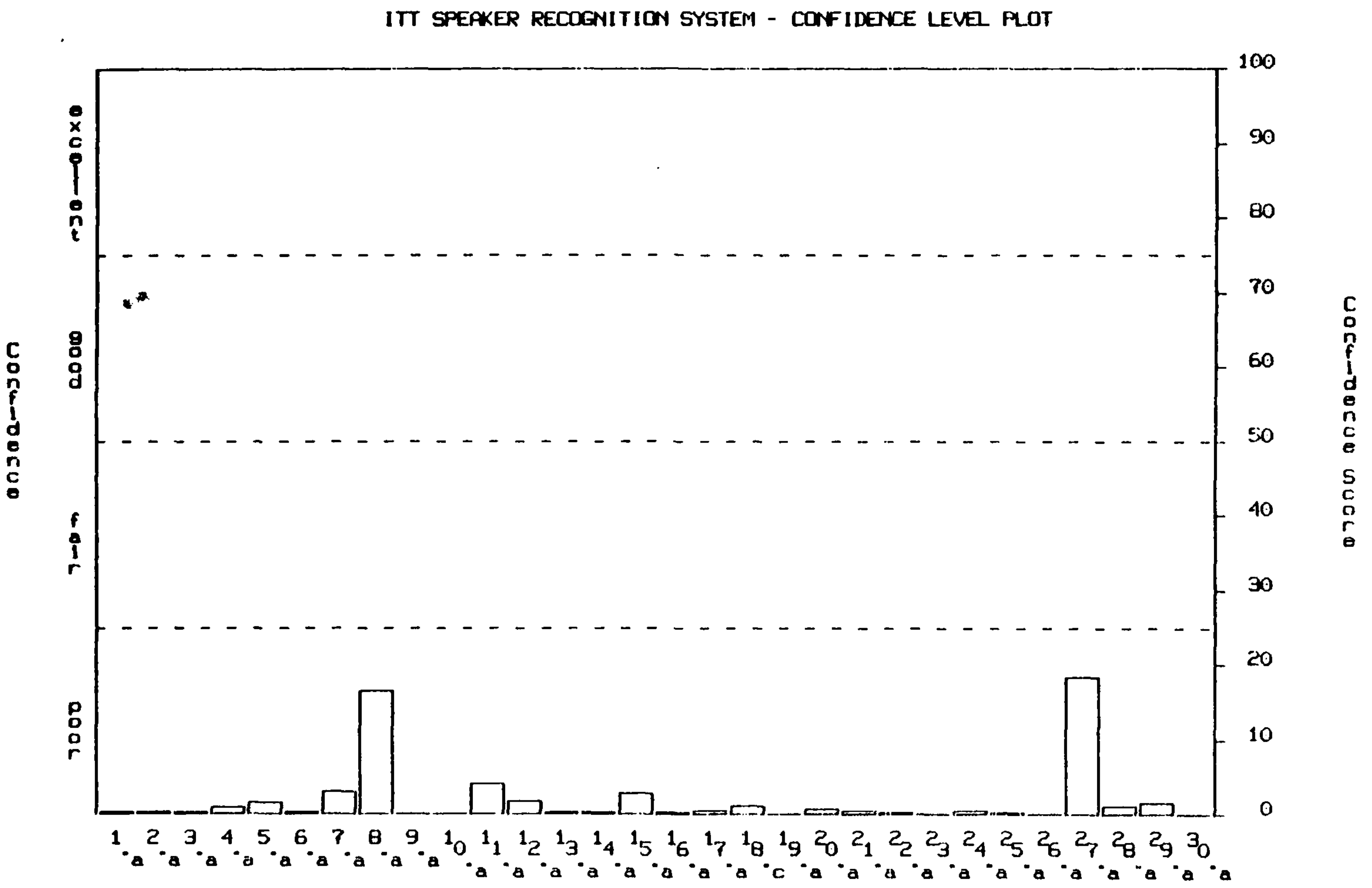


Figure I.54: Test using 20 seconds of speech from speaker27

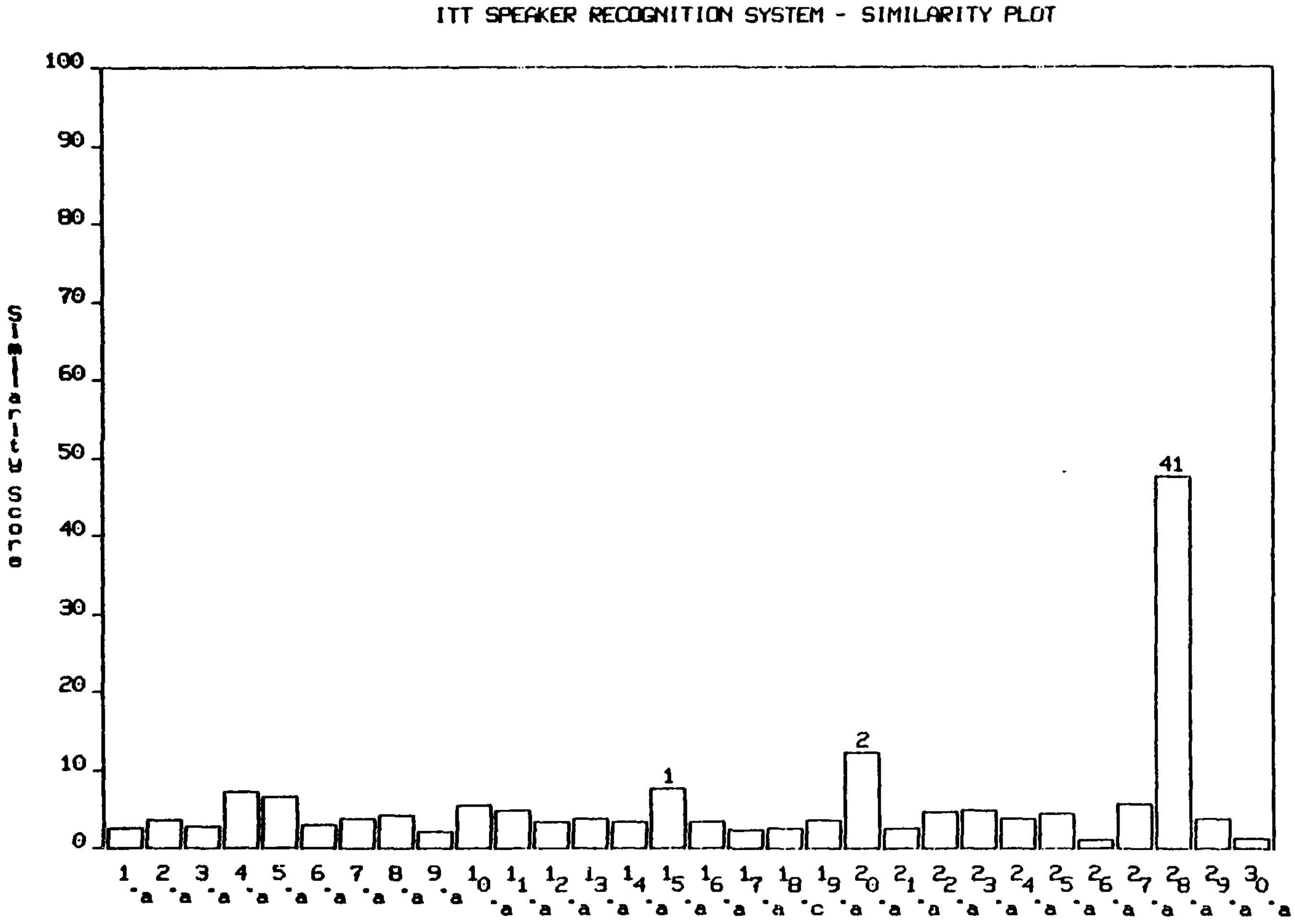


Figure I.55: Test using 20 seconds of speech from speaker28

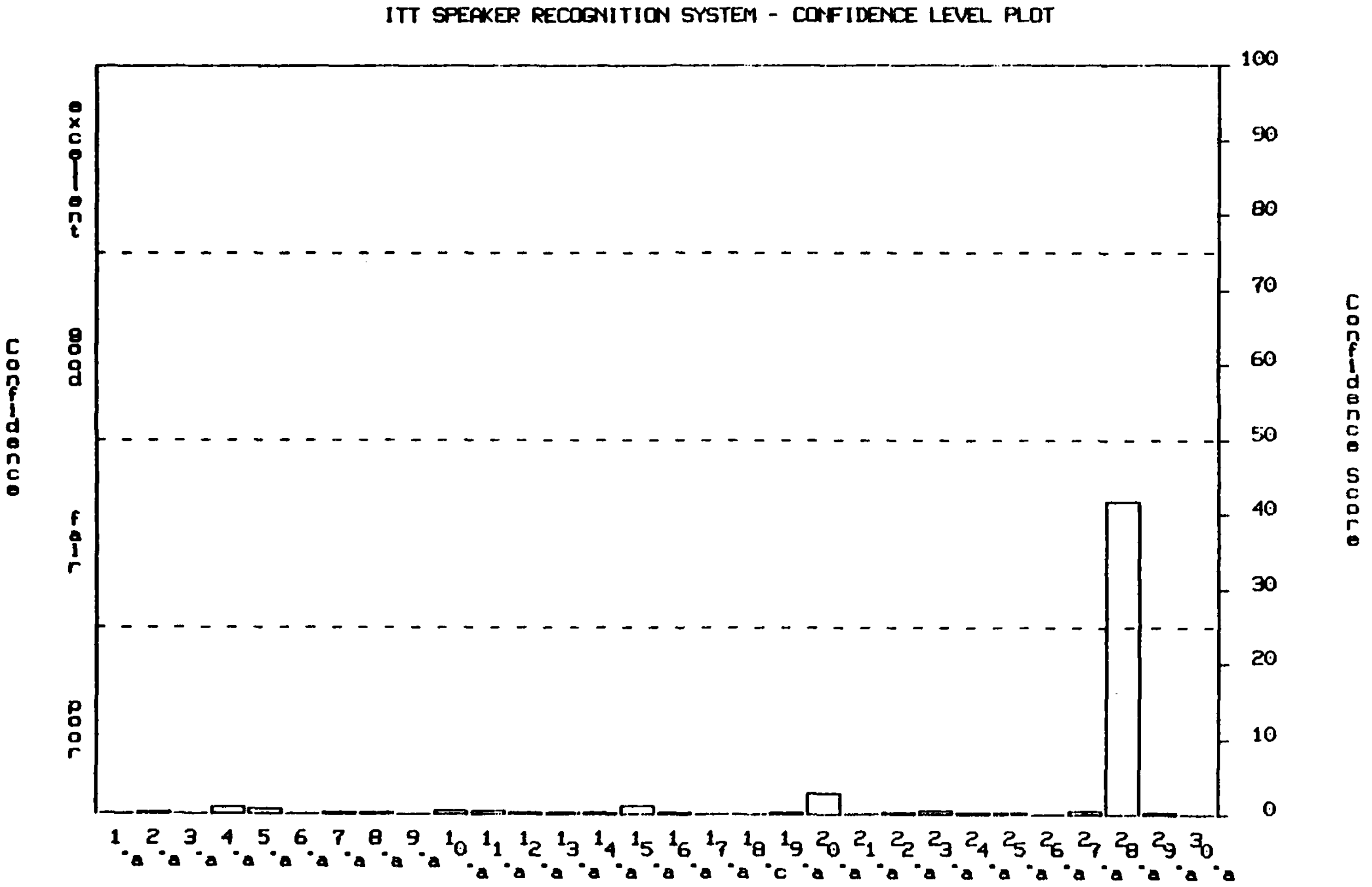


Figure I.56: Test using 20 seconds of speech from speaker28

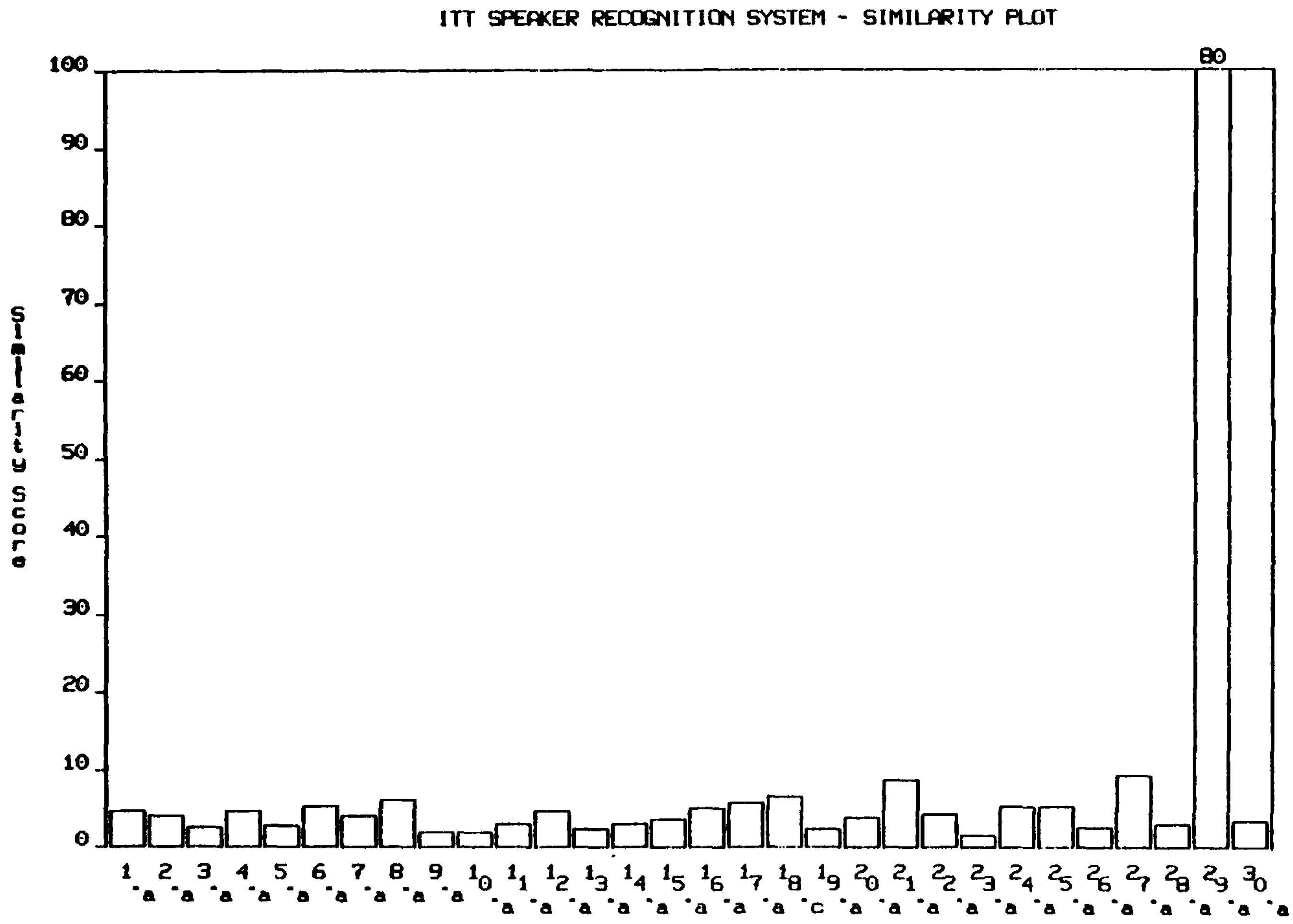


Figure I.57: Test using 20 seconds of speech from speaker29

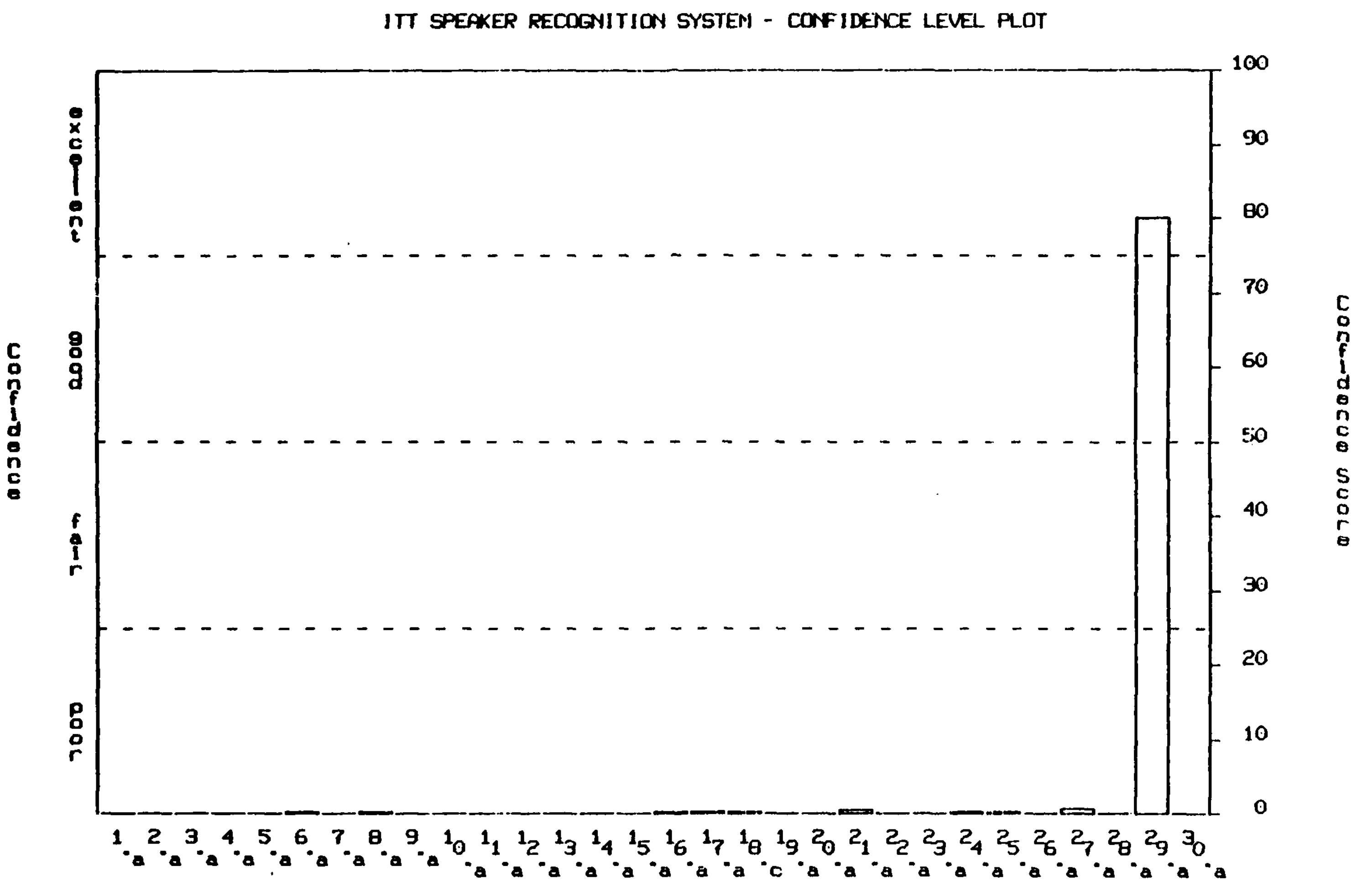


Figure I.58: Test using 20 seconds of speech from spsaker29

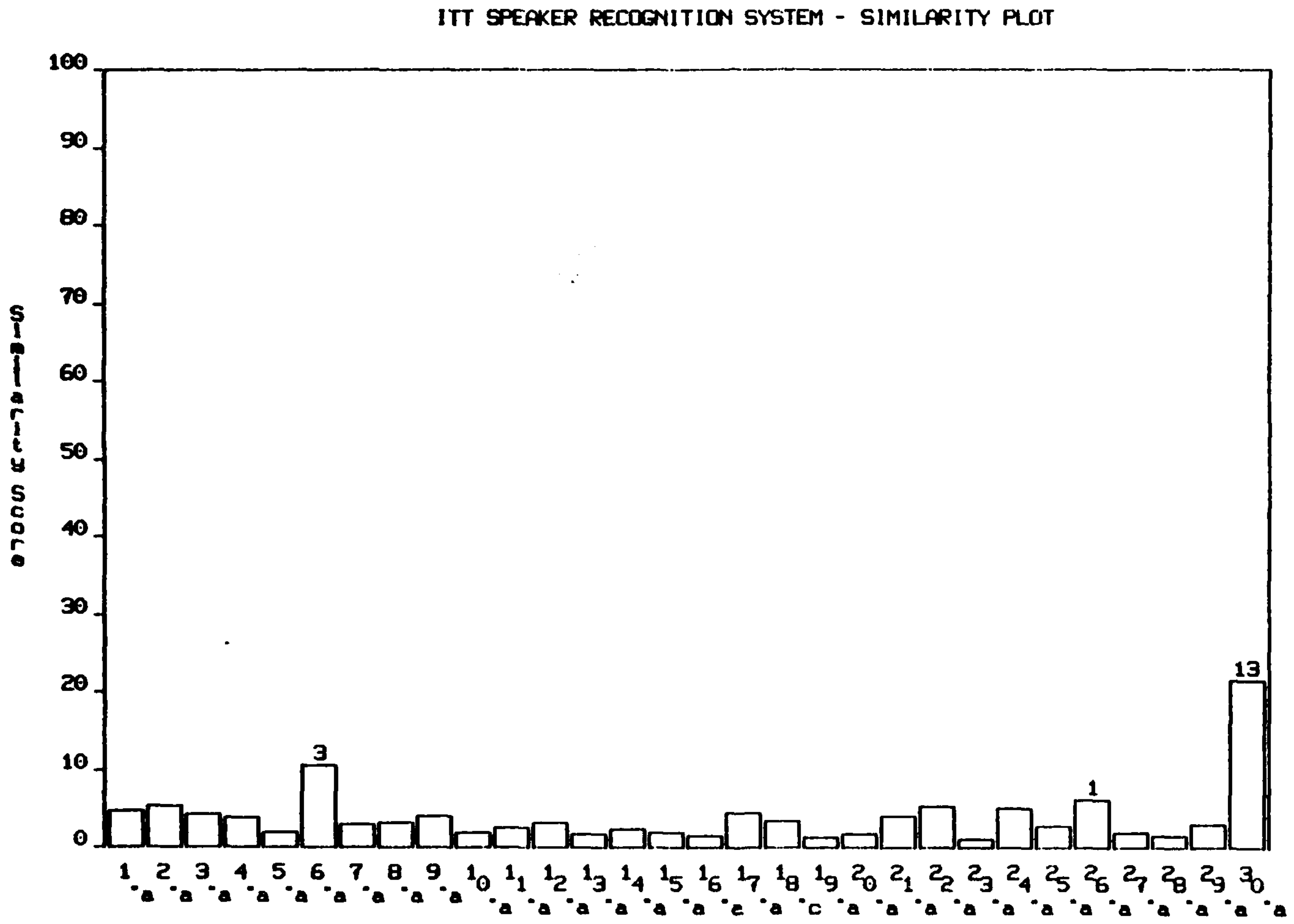


Figure I.59: Test using 20 seconds of speech from speaker30

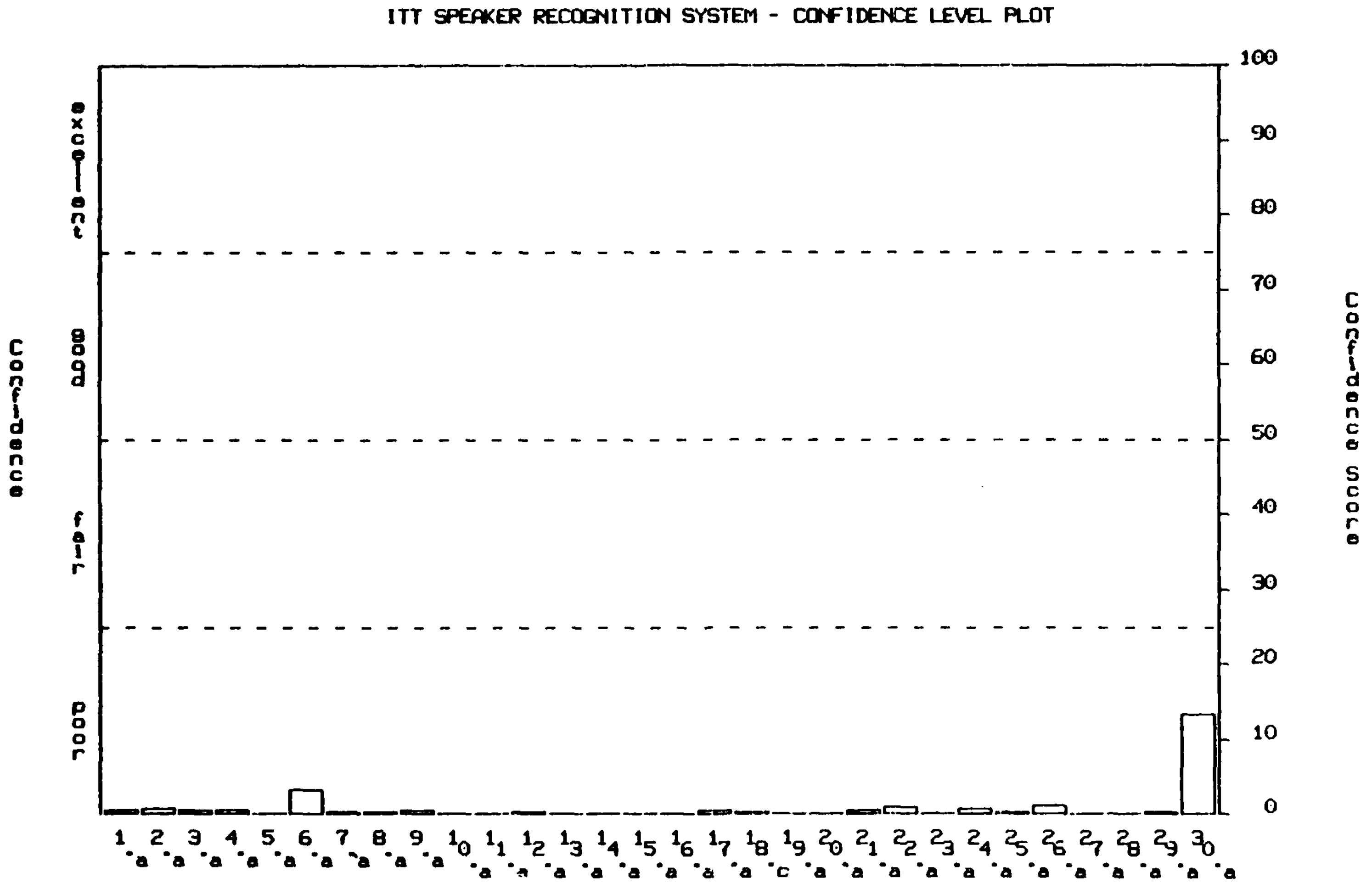


Figure I.60: Test using 20 seconds of speech from speaker 30

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.