TR-I-0242

Speech Recognition Expert System

A Study on Knowledge and Neural Networks Intragration

小森 康弘
Yasuhiro KOMORI

1992.2.18

内容梗概

筆者が、１９８８年9月１日から１９９２年2月２９日まで、ＡＴＲ自動翻訳電話研究所にて行った研究の最終報告書である。本報告は「エキスパートシステム」と「ニューラル・ネットワーク」に基づく音声認識に関する研究であり、これらの音声認識技術を統合することの有効性を論じ、各々の音声認識技術の改良について述べる。

Achievements

< *Journal* >

1. ”スペクトログラム・リーディング知識を用いた音韻セグメンテーション・エキスパートシステム”
   畑崎香一郎, 小森康弘, 川端豪, 鹿野清宏
   信学論 D-II, J73-D-II, No.1, pp.1-9, 1990-01.
2. ”Phoneme Segmentation Expert System Using Spectrogram Reading Knowledge”
   Kaichiro Hatazaki, Yasuhiro Komori, Takeshi Kawabata, Kiyohiro Shikano
   Systems and Computers in Japan,Systems and Computers in Japan, Scripta Technica Journals in Electronics, Computers & Systems Sciences, Vol. 21, No. 12, pp.90-100, 1990-12.
3. ”スペクトログラム・リーディング知識とニューラル・ネットワークを用いた音韻認識エキスパートシステム”
   小森康弘, 畑崎香一郎, 川端豪, 鹿野清宏
   信学論 D-II, J73-D-II, No.1, pp.10-19, 1990-01.
4. ”Phoneme Recognition Expert System Using Spectrogram Reading Knowledge and Neural Networks”
   Yasuhiro Komori, Kaichiro Hatazaki, Takeshi Kawabata, Kiyohiro Shikano
   Systems and Computers in Japan,Systems and Computers in Japan, Scripta Technica Journals in Electronics, Computers & Systems Sciences, Vol. 21, No. 12, pp.101-111 1990-12.
5. ”An Integration of Knowledge and Neural Networks toward a Phoneme Typewriter without a Language Model”
   Yasuhiro Komori, Kaichiro Hatazaki
   Trans. IEICE (信学論) E (in D-II), pp. 1797-1805, 1991-07.

< *International Conference* >

1. "Phoneme Segmentation Using Spectrogram Reading Knowledge"
   Kaichiro Hatazaki, Yasuhiro Komori, Takeshi Kawabata, Kiyohiro Shikano
   IEEE, ICASSP89, 1989-05.

2. "Phoneme Recognition Expert System Using Spectrogram Reading Knowledge and Neural Networks"
   Yasuhiro Komori, Kaichiro Hatazaki, Takaharu Tanaka, Takeshi Kawabata, Kiyohiro Shikano
   Proc. Eurospeech89, 1989-09.

3. "Combining Phoneme Identification Neural Networks into an Expert System using Spectrogram Reading Knowledge"
   Yasuhiro Komori, Kaichiro Hatazaki, Takaharu Tanaka, Takeshi Kawabata
   IEEE, ICASSP90, 1990-04.

4. "Robustness of a Feature Based Phoneme Segmentation System to Speaker Independent and Continuous Speech"
   Yasuhiro Komori, Kaichiro Hatazaki
   Proc. ICASSP91, 1991-05.

5. "Time-State Neural Networks (TSNN) for Phoneme Identification by Considering Temporal Structure of Phonemic Features"
   Yasuhiro Komori
   Proc. ICASSP91, 1991-05.

6. "An Integration of Knowledge and Neural Networks toward a Phoneme Typewriter without a Language Model"
   Yasuhiro Komori, Kaichiro Hatazaki
   Proc. Eurospeech91, 1991-09.

7. "Neural Fuzzy Training Approach for Continuous Speech Recognition Improvement"
   Yasuhiro Komori
   Proc. ICASSP92, 1992-03. (to appear)

8. "A Segment-based Speaker Adaptation Neural Network Applied to Continuous Speech Recognition"
   Keiji Fukuzawa, Yasuhiro Komori, Hidefumi Sawai, Masahide Sugiyama
   Proc. ICASSP92, 1992-03. (to appear)

< Conference & Workshop >

1. "スペクトログラム・リーディング知識に基づく
音声認識エキスパートシステムの構築"
小森康弘, 畑崎香一郎, 田中孝明, 川端豪
音学講論, 春季研究発表会, 3-6-11, 1989-03.

2. "スペクトログラム・リーディング知識による
音韻セグメンテーションの評価"
畑崎香一郎, 小森康弘
音学講論, 春季研究発表会, 2-P-2, 1989-03.

3. "スペクトログラム・リーディング知識とニューラル・ネットワーク
を用いた音韻認識エキスパートシステム"
小森康弘, 畑崎香一郎, 田中孝明, 川端豪, 鹿野清宏
信学技報, SP89-33, 1989-06.

4. "音韻認識エキスパートシステム"
鹿野清宏, 畑崎香一郎, 小森康弘
電子情報通信学会, 秋季全国大会, パネル討論, PD-1-4, 1989-09.

5. "スペクトログラム・リーディング知識に基づく
音韻認識エキスパートシステムにおける音韻識別
ニューラル・ネットワークの融合法の検討"
小森康弘, 畑崎香一郎, 田中孝明, 川端豪, 鹿野清宏
音学講論, 秋季研究発表会, 3-1-14, 1989-10.

6. "音韻認識エキスパートシステムにおける知識と TDNN の融合法"
小森康弘, 畑崎香一郎, 川端豪, 鹿野清宏
信学技報, SP89-84, 1989-12.

7. "音韻認識エキスパートシステムにおける母音認識
- TDNN による母音スポッティング -"
小森康弘, 畑崎香一郎, 川端豪, 鹿野清宏
音学講論, 春季研究発表会, 2-P-18, 1990-03.

8. "時間構造を考慮したニューラル・ネットワークによる音韻認識"
小森康弘, 南泰浩, 鹿野清宏
音学講論, 春季研究発表会, 2-P-19, 1990-03.

9. "音素識別ニューラルネットにおけるファジー学習法"
小森康弘, A.H.Waibel, 嵯峨山茂樹
音学講論, 春季研究発表会, 1-5-15, 1991-03.

10. "特徴ベースによる音素セグメンテーションのロバスト性"
小森康弘, 畑崎香一郎

音学講論, 春季研究発表会, 2-5-17, 1991-03.

11. ”HMMとスペクトログラム・リーディング知識に基づく
ハイブリッド音素セグメンテーションシステムの構想”
藤原紳吾, 小森康弘, 杉山雅英
音学講論, 春季研究発表会, 2-5-16, 1991-03.

12. ”ニューラル・ファジー学習法によるTDNN-LR連続音声認識システムの性能向上”
小森康弘, A.H.Waibel, 嵯峨山茂樹
信学技報, SP91-24, 1991-06.

13. ”A Hybrid Labeling System Using HMM and Spectrogram
Reading Knowledge”
Shingo Fujiwara, Yasuhiro Komori, Masahide Sugiyama
Korea-Japan Joint Symposium on Acoustics, 1991-07.

14. ”ニューラル・ファジー学習法の連続音声認識における効果”
小森康弘, 福沢圭二, 杉山雅英, A.H.Waibel, 嵯峨山茂樹
音学講論, 秋季研究発表会, 2-5-11, 1991-10.

15. ”セグメント話者適応ニューラル・ネットワークを用いた
文節音声認識”
福沢圭二, 小森康弘, 沢井秀文, 杉山雅英
音学講論, 秋季研究発表会, 3-5-10, 1991-10.

16. ”HMMとスペクトログラム・リーディング知識に基づく
ハイブリッド音素セグメンテーションシステム”
藤原紳吾, 岩橋直人, 小森康弘, 杉山雅英
音学講論, 秋季研究発表会, 2-5-20, 1991-10.

17. ”自動セグメンテーションによる音声合成単位の作成”
岩橋直人, 藤原信吾, 小森康弘, 杉山雅英, 匂坂芳典
音学講論, 秋季研究発表会, 1-6-21, 1991-10.

18. ”セグメントベース話者適応ニューラル・ネットワークと
ＴＤＮＮ－ＬＲを用いた文節音声認識”,
福沢圭二, 小森康弘, 沢井秀文, 杉山雅英,
信学技報, SP91-105, 1992-01.

19. ”ＴＤＮＮ－ＬＲ連続音声認識における不特定話者ＴＤＮＮと
話者適応ニューラル・ネットワークの性能比較”
福沢圭二, 小森康弘, 杉山雅英
音学講論, 春季研究発表会, 1992-03. (to appear)

Contents

Chapter 1

INTRODUCTION

1.1 Speech Communication

Speech has been the most natural and easiest way to exchange information between humans in the long history of the human race. This is because humans are able to converse with each other simply by using their own faculties. A human, of course, is able to exchange information by other means such as letters and gestures. However, humans usually use speech for exchanging information because it is facile and because it has many other advantages such as real-time response, no need for special training, individuality and conveying emotion information.

Before computers were developed, human only had to exchange information with other humans. The computer, though originally developed as a calculation machine, has passed through many levels or technological innovation. Computers now cope with a great amount of information which requests human-machine communication. In practice, the exchange of information between humans and machines is ever increasing.

Considering the demand for communicating with machines and the advantages of speech, it is very natural to make use of speech for human-machine communication. The study of human-machine speech communication is an on-going task, and speech recognition and speech synthesis are its basic elements.

This report concerns a study of a speech recognition expert system that integrates human knowledge and neural networks.

1.2 Background

In this section, the history of the study of speech recognition is described along with a background of the study of neural networks and the knowledge based approach [Nakata 77] [Nakata 78] [Niimi 79] [Furui 85] [Nakagawa 88] [Waibel 90].

1.2.1 Speech Recognition

The study of speech began in about 1940 with the development of the vocoder by Dudley [Dudley 40] and the sound spectrogram by Potter [Potter 47]. The first speech recognizer, whose speech recognition task was digits [Davis 52], was proposed in 1952 by Davis. In the 1960's, fundamental studies in speech processing were achieved by Franagan [Franagan 55] and Fant [Fant 60]. Around that time, research aiming at a phoneme typewriter were proposed by Olson [Olson 56] and Sakai [Sakai 63], although these studies only showed the difficulties of automatic speech recognition.

In Japan, the "Maximum Likelihood Spectrum Distance" and "Linear Predictive Coding (LPC) Analysis" were proposed by Saito and Itakura, [Itakura 69] [Itakura 71] and were dramatic developments in speech analysis. The LPC analysis has had a great effect on the speech signal processing field up to the present time. Furthermore, Sakoe formulated the problem of non-linear time-warping of speech using "Dynamic Programming (DP or DTW: Dynamic Time Warping)" [Sakoe 71]. This method also spurred the study of speech recognition based on template-matching. Influenced by these developments, many speech recognizers were realized, albeit with limitations such as speaker dependency, isolated word utterance input, limited vocabulary, and so on.

At almost the same time in the U.S.A, research into utilizing natural language as a human-machine interface was proposed by Woods and Winograd [Woods 70] [Winograd 72]. They demonstrated effective and sophisticated natural language communication by implementing "Question-and-Answer" systems. Such research in natural language processing showed the possibility of realizing a human-machine communication system with a very limited domain such as a Question-and-Answer system.

These developments in the study of speech processing and of natural language processing influenced the ARPA (Advance Research Projects Agency) established in 1971 [Klatt 77]. The aim of the ARPA project was the development of a speech understanding system that integrated speech recognition and natural language processing, whose vocabulary size was around 1,000 words. Two famous systems were developed at Carnegie Mellon University: the "HearSay II" system [Lesser 75] and "HARPY" system [Lowerre 76]. The HearSay II system is wellknown because it proposed a new architecture called the "blackboard model". The blackboard model is able to cope with variable type knowledge and variable hierarchical knowledge. Furthermore, the invention of the blackboard model was the dawn of "Artificial Intelligence" (AI) study, the beginning of expert system development. HARPY is wellknown because it gave the best speech understanding performance of 97% in the ARPA project, by combining the advantages of the "DRAGON" system [Baker 75] and the HearSay II system.

This ARPA project also had considerable influence on the research in Japan and Europe. In Japan, Kyoto Institute of Technology's "SPOKEN-BASIC" [Niimi 77], Kyoto University's "LITHAN" [Nakagawa 76], NTT's "Voice Q-A system" [Shikano 81], Waseda University's "WABOT II" [Kobayashi 85], and in Europe, CNET's "KEAL" system [Mercier 77] are some of the better known results.

The ARPA project showed the advantage of integrating natural language as a constraint into the speech recognition process. However, it also proved the necessity of more accurate and detailed acoustic models for speech recognition.

At the same time, computer progress has been incredible. Computer hardware has become faster, smaller and cheaper and memories have expanded geometrically. This progress greatly speed up data handling. On the software side, many programming languages were invented not only for fast calculation, but also to facilitate complex system implementation such as knowledge based systems in AI.

The progress in computers also brought break-throughs in the study of speech recognition, making possible several new approaches, such as "Hidden Markov Models (HMM)" [Rabiner 86], "Neural Networks" [Rumelhart 86] [Lippmann 87] and "Expert Systems" [Buchanan 85]. These approaches are the newest techniques in the study of speech recognition and are break-throughs from the DTW template-matching speech recognition approach.

Considerable progress has been made using these techniques and many speech recognition systems have been developed as a result. "SUMMIT" [Zue 90] and "SPREX" [Mizoguchi 87] based on "Expert Systems", "TANGORA" [Jelinek 85], "SPHINX" [Lee 89], "BIBLOS" [Chow 87] and "ATR-HMM-LR" [Hanazawa 90] based on "Hidden Markov Models." As for "Neural Networks" research into "Time-Delay Neural Networks" (TDNN) [Waibel 89], "Dynamic Neural Networks" (DNN) [Sakoe 89] and "Neural Prediction Models" (NPM) [Iso 90], is now progressing. Each of the aforementioned systems has attained very impressive accuracy and most have overcome some recognition constraints such as speaker dependency, utterance style, vocabulary size, etc.

At present. many speech recognition studies are concentrating on improving one of these techniques or on integrating certain of these techniques to improve the overall speech recognition performance.

Among these studies, HMM appeared to be a good approach to continuous speech recognition. The main advantages of HMM for speech recognition are as follows:

- Invention of a strong training algorithm, Baum-Welch algorithm [Baum 70] or EM algorithm [Dempster 77].
- Probabilistic representation in each acoustic model.
- Capability of dealing with non-linear time warping.
- Facile integration into language models by concatenating phoneme HMMs.

On the other hand, speech recognition based on neural networks and expert systems appeared to have some difficulties when applied to continuous speech recognition. In the neural network approach, although the performance of the phoneme classification is greater than that of HMM [Waibel 89], it has various problems such as mis-activation in the untrained regions for the input utterance. Moreover, in the approach of classification-type neural networks, the problem of normalizing time warping of speech features is serious. In the approach of expert systems research has proved the effectiveness of using human expert knowledge. However, on the other hand, it also proved the difficulties of full formulating human knowledge into explicit rules and the difficulties of full automatic acoustic feature extraction.

1.2.2 Neural Networks

The study of neural networks applied to pattern recognition was in fashion in the 1950's and 60's. However, after 1969, the theoretical limitation of neural networks for pattern recognition was shown by Minsky [Minsky 69], and studies in this field have faded away. The main reason for this was that a good training algorithm for multiple layer neural networks could not be developed at that time.

Recently, there were break-throughs in the study of neural networks: one was the development of the back-propagation algorithm for multiple layer neural network training, and the other was the incredible speed-up of computers. These break-throughs made it possible to use a great amount of data to train multiple layer neural networks which were thought to be impossible or very difficult to train.

This back-propagation algorithm is very easily realize on computers. Thus, it spread to many research fields and many neural network application studies are now on-going, such as speech processing, image processing, language processing, system control and so on. In particular, joined with study of the Massively Parallel Distribution Processing in parallel computer science, the study of neural networks has become fashionable again.

Since the back-propagation algorithm was developed, many neural network applications into speech recognition have been proposed. Several neural network approaches in speech recognition, such as the "Neural Prediction Models" (NPM)

[Iso 90], "Dynamic Neural Networks" (DNN) [Sakoe 89] and "Time-Delay Neural Networks" (TDNN) [Waibel 89], have shown its effectiveness.

NPM and DNN are able to deal with the time warping of speech features. TDNN has a capability for time-shift tolerance. Considering the type of neural networks, the NMP can be classified as a non-linear mapping model, while DNN and TDNN can be considered as neural network classifiers. NPM and DNN are proposed for a word speech recognizer and TDNN is proposed as a phoneme classifier. TDNN showed incredibly high performance on phoneme identification, compared with the HMM identification results obtained using the same data [Waibel 89].

1.2.3 Expert Systems

The origin of the knowledge based approach derived from the blackboard model of the HearSay II system. This approach directly deals with human knowledge. Knowledge engineering is a discipline that seeks to understand the human knowledge, especially the knowledge of human experts using an engineering technique. In practice, this discipline aims at the construction of an expert system which is able to automatically solve problems the human expert is able to solve. In this study, knowledge is the heuristic and experimental knowledge which was difficult to cope with in the previous studies. This heuristic knowledge is described as a production rule and is used in the production system to solve the problem by hypothesizing and evaluating the evidence while properly considering the constraints. These knowledge based systems are developed through a recursive or iterative trial and error rule-retraining as shown in Figure 1-1.

This knowledge engineering technique is adopted in speech recognition because a human expert has some sort of knowledge for speech recognition. There are two major advantages to using a knowledge based approach for speech recognition:

(1) Knowledge should be explicitly described as production rules which help the researcher make sure and to order one's knowledge.
(2) Decision path, the way of recognition, can be easily obtained by back-tracking the production rules and makes it easy to modify rules for system improvement.

These are the main advantages that can be obtained from the knowledge based approach compared to the conventional approaches.

The most famous knowledge for speech recognition is "spectrogram reading knowledge." Spectrogram reading is a technique to identify a phoneme category with its boundaries on a speech spectrogram using the visual acoustic phonetic features. A spectrogram reader obtains experiential knowledge of acoustic phonetic features through spectrogram reading, which makes it possible to recognize phonemes in a continuous speech spectrogram with high accuracy (over 80%) [Zue 79], performing phoneme segmentation and phoneme identification simultaneously using spectrogram reading knowledge. For these reasons, several knowledge based speech recognizers have been developed [Zue 86] [Carbonell 86] [Mizoguchi 87] [Stern 86] [Connolly 86]. These previous research studies proved the effectiveness of spectrogram reading knowledge for phoneme identification.

1.3 Purpose

This report has two purposes:

(1) To simulate spectrogram reading behavior by an expert system.
(2) To construct a speech recognizer by integrating human knowledge and neural networks without a language model.

The first purpose is advancing the early study of Hatazaki, which is a feature based phoneme recognition expert system [Hatazaki 87] [Hatazaki 88]. Hatazaki's study has various benefits compared with the previous study of speech recognition expert systems because it simulates the human expert spectrogram reading process. This means that the system uses not only static human knowledge but also dynamic knowledge, i.e. strategies of a human expert.

Since Zue showed the effectiveness of utilizing spectrogram reading knowledge for speech recognition, as previously described, several knowledge based speech recognition systems have been developed [Zue 86] [Carbonell 86] [Mizoguchi 87] [Stern 86] [Connolly 86]. Most of these systems are basically separable into two parts as shown in Figure 1-2:

(1) Acoustic feature extraction part.
(2) Verification and recognition part.

In the acoustic feature part, acoustic analysis is performed, then features are extracted into explicit fact representations. In the recognition part, these extracted features are evaluated by the production rules which represent for phoneme identification knowledge.

However, in this structure, the system is not able to represent human knowledge fully. The human knowledge for reading spectrograms consists not only of the facts of acoustic evidence for phoneme recognition but also the strategies, in other words the ways to manage the acoustic evidence for phoneme determination. Thus, to realize an adequate knowledge based speech recognizer, the system should utilize not only the static knowledge but also the dynamic knowledge of the human expert.

Various kinds of acoustic features have to be extracted from a spectrogram for phoneme identification. Moreover, they are complex and fuzzy. It is also very difficult to automatically extract these acoustic features from a spectrogram. Thus, in most conventional speech recognition systems, very limited acoustic features are used. They are extracted and symbolized by pre-processing and/or by hand-labeling. Extracting acoustic features by hand from a spectrogram remains a major problem in fully automatic speech recognition. When extracting acoustic features by pre-processing, there are also problems:

- The system is not able to extract precise acoustic features according to the phoneme context, because of the lack of knowledge concerning phoneme variations and coarticulation effects.
- The system is not able to pre-process all acoustic feature extraction which appear on a speech spectrogram. There are various kinds of acoustic features which are global and rough, or local and precise.
- The system is not able to manage the usage of acoustic features because the usage differs considerably according to phoneme context.

Thus, the separation of feature extraction and verification/recognition makes it difficult to extract and to control all the necessary acoustic features.

In Hatazaki's study, these problems were overcome by the use of dynamic human expert knowledge as strategies for phoneme recognition. Acoustic feature extraction was performed under the demand of the strategy by considering phoneme contexts, which made it possible to extract precise and various acoustic features. In Hatazaki's system, the feature extraction and phoneme verification are elegantly integrated and very well-formed in controlling acoustic features.

However, there remain other problems in feature based expert system approaches. In general, acoustic features which are useful for phoneme identification, such as distinctive features between /m/ and /n/ or /b/ and /d/, etc., are especially difficult to extract automatically. Even a human expert is not able to find these features. Moreover, spectrogram reading knowledge such as pattern matching, which is not a small part of the whole human knowledge, is hard to describe as explicit rules. Thus, it is very difficult to fully formalize phoneme recognition knowledge, and to extract acoustic features automatically.

Considering these difficulties, in the approach of this report, only explicit knowledge is described into rules, e.g. strategies, phoneme boundary detection, rough phoneme class features, and so on. Knowledge, which is difficult to describe as rules, e.g. pattern matching-type knowledge, is implicitly represented inside the neural networks.

The second purpose of this report is to construct a speech recognizer with no language model. Most speech recognition systems or speech understanding systems previously proposed are designed to choose the best word sequence from the word dictionary, under some constraint of a language model and limited domain knowledge. However, when humans converse with each other, many new and non-entry words arise, even if the domain is limited. A recent approach that recognizes such non-entry words [Asidi 90], only gets the rough category of the meaning for the input speech using the constraints of a task-dependent language model.

However, in conversation, it is sometimes necessary to get the exact sequence of the input speech. For example, to spell out a name such as "KOMORI", K-O-M-O-R-I. There is some challenging research aiming toward a phonetic typewriter proposed by Kawabata [Kawabata 91] and Kohonene [Kohonene 88]. Kawabata's study is based on the "ATR-HMM-LR" speech recognizer [Hanazawa 90] while its grammar is modified as a syllabic trigram model. The disadvantage of this system is that the performance strongly depends on the syllabic trigram. In this sense, this system has some kind of language model because the syllabic trigram is trained using a language database, and phonemes are not recognized only from the acoustic information. Kohonene's approach is a combination of the neural network approach and knowledge based approach. The neural network is used to produce the frame-by-frame phoneme identification results and the knowledge is the phonotactics constraint in the language model. In Kohonene's approach, the acoustic information is fully analyzed in a neural network, and is not a feature based approach.

To realize a phoneme recognition system aiming at phoneme typewriter without a language model, powerful methods for phoneme segmentation and phoneme identification are indispensable and the architecture of the system should be constructed in a full bottom-up style. Many bottom-up style speech recognizers have been proposed in recent research. These recognizers consist of some sort of phoneme segmentation and phoneme identification. Although, in their segmentation part, they did not find the exact phoneme boundary, they obtained every possible acoustic boundary or performed very rough segmentation. As for phoneme identification, a high performance phoneme identifier, such as neural networks, was not developed until recent.

From this point of view, the boundary obtained by spectrogram reading knowledge is every bit as accurate as that of a human expert [Hatazaki 90] and the TDNN is one of the most powerful phoneme identifiers available [Waibel 89]. Thus, this combination is one of the most promising ways to realize speech recognition without a language model. The system proposed in this report is realized as a sophisticated integration of knowledge and TDNNs.

1.4 Contents

This report proposes a phoneme recognition expert system which integrates knowledge and neural networks, aiming at a speech recognizer without a language model by simulating the spectrogram reading behavior of a human expert.

This report consists of four major parts:

(1) Introduction, Chapter 1.

(2) Recognition system and evaluation, Chapter 2 to Chapter 7.
(3) System expansion to continuous speech, Chapter 8 to Chapter 10.
(4) Conclusions, Chapter 11.

Chapter 1: The introduction begins with speech as a human-machine communication, then describes the background and the purpose of this study along with recent studies of speech recognition and finally the contents of this report.

Chapter 2: An example of spectrogram reading behavior is shown. Then, a knowledge representation that simulates the human expert behavior is described. The framework of the expert system, spectrogram reading knowledge for explicit knowledge, non-deterministic strategy, representation of uncertainty using certainty factor and fuzziness, on-demand top-down control feature extraction under phoneme context constraints, and neural networks representing implicit knowledge are described.

Chapter 3: The architecture of a speech recognition expert system without a language model is proposed along with its hardware configuration. The system is realized as an integration of human knowledge and neural networks. The system mainly consists of two parts: consonant recognition and vowel recognition.

Chapter 4: The consonant recognition part is described in this chapter. Consonant recognition consists of two main parts:

(1) Feature based phoneme segmentation.

(2) Neural network based phoneme identification.

The details of each part are presented and the experimental result tested on an ATR database [Takeda 88] is discussed.

Chapter 5: In this chapter, five mechanisms for integrating knowledge and neural networks are studied to enhance their respective advantages. Consonant recognition experiments are carried out, and show that the closer integration of knowledge and neural networks improves not only identification performance but also segmentation accuracy, effectively reducing insertion errors.

Chapter 6: The details of the vowel recognition part are described. Vowel recognition utilizes phoneme-spotting neural networks for vowel detection and knowledge for verifying its category and boundaries. The effectiveness of this approach is shown through a vowel detection experiment.

Chapter 7: The evaluation of the overall expert system is performed using the best integration of knowledge and neural networks proposed in this report. A phoneme recognition experiment is shown for all Japanese phonemes on 2,620 isolated words in the ATR database.

Chapter 8: The robustness of a feature based segmentation against speaker independency and utterance styles (speaking rate) is shown through experimental results. The added and modified knowledge for system expansion is also discussed.

Chapter 9: This chapter focuses on neural network structure to improve the phoneme identification performance of continuous speech and reports a new structure for phoneme identification neural networks, "Time-State Neural Networks" (TSNN). Phonemes in Japanese have certain rough temporal structures of phonemic features which do not greatly change even when the utterance is an isolated word or continuous speech. The proposed TSNN is able to deal with the temporal structure of phonemic features, which is helpful for identifying phonemes. Several types of TSNNs are described along with their phoneme identification performance, tested on Japanese phonemes /b,d,g,m,n,N/, taken from isolated words, phrase and sentence utterances.

Chapter 10: This chapter focuses on neural network training to improve continuous speech recognition. A new training method for phoneme identification neural networks, called "Neural Fuzzy Training" method, is proposed. The general idea is described and the experimental results of phoneme identification are presented. Moreover, continuous speech recognition experiments using the TDNN-LR speech recognizer [Sawai 91] are performed. Dramatic improvements of the proposed Neural Fuzzy Training method compared with the conventional training method are shown.

Chapter 11: Finally, this chapter summarizes the study of this report and discusses further studies.

Appendixes, References, Index are appended at the end of this report.

1.5 Figures & Tables

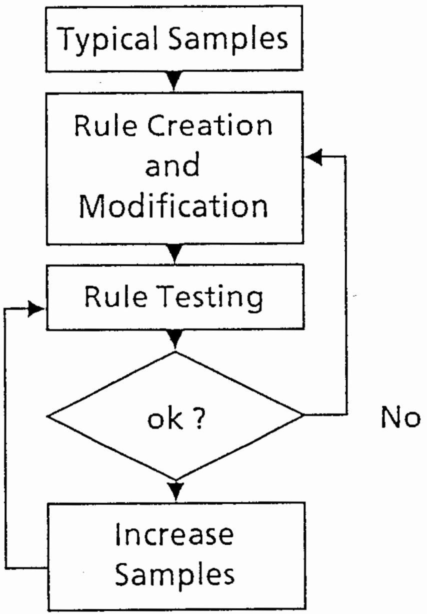


Figure 1-1: Rule Creation and Modification

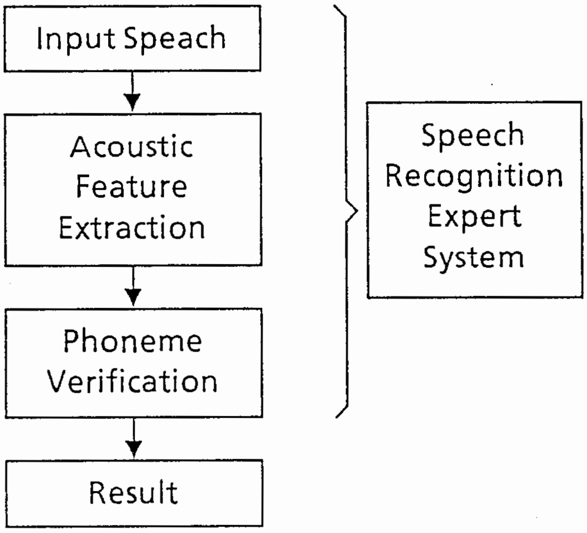


Figure 1-2: Rule Based Systems

Chapter 2

KNOWLEDGE REPRESENTATION

2.1 Introduction

In this chapter, an example of the spectrogram reading process performed by a human expert is described in order to show the kinds of human knowledge and how they are used. This knowledge is required to be incorporated into the system naturally to implement a high performance knowledge based speech recognizer.

Secondly, this chapter presents knowledge representation in order to simulate the spectrogram reading behavior. The knowledge and strategy used by a human expert in spectrogram reading strongly depend on phoneme context; moreover, it is fuzzy. And knowledge consists of various kinds of precise and local, or rough and global acoustic phonetic features. To simulate a spectrogram reading process and to describe complex human knowledge easily and naturally on a computer, a well-formed framework is indispensable. The knowledge representations that are incorporated in the system are listed below:

- Expert system.
- Spectrogram reading knowledge.
- Non-deterministic contextual strategy.
- Representation of uncertainty.
- Representation of fuzziness.
- On-demand top-down control acoustic feature extraction.
- Time-Delay Neural Networks.

2.2 Spectrogram Reading Process

Figure 2-1 shows an example of a spectrogram uttered by a male speaker. The utterance /sukunakutomo/ is visualized as a two-dimensional pattern. The horizontal axis indicates the time scale and the vertical axis indicates the frequency scale. Shading indicates the power inside a certain region of the time and frequency domain. Spectrogram reading is a technique for identifying phoneme categories with their boundaries by using these visual acoustic phonetic features on a speech spectrogram. The spectrogram reading process for this example takes place as follows:

1. Rough segmentation is easily performed. In this case, the spectrogram is separated into 11 regions. 1) From 350ms to 500ms, 2) 500ms to 570ms, 3) 570ms to 640ms, 4) 640ms to 670ms, 5) 670ms to 760ms, 6) 760ms to 830ms, 7) 830ms to 890ms, 8) 890ms to 950ms, 9) 950ms to 1,040ms, 10) 1,040ms to 1,090ms, 11) 1,090ms to 1,200ms.
2. These regions are roughly classified into 3 regions.

   silence region: 2), 6) and 8) with no power over the entire frequency range (0-6,000Hz).

   unvoiced region: 1) and 7) with no power in the low frequency range (0-500Hz).

   voiced region: 3), 4), 5), 9), 10) and 11) using the power in the low frequency range (0-500Hz).
3. Region 1) has considerable power in the high frequency range (4,000-6,000Hz), and no power in the low frequency range (0-500Hz). The duration is long. Thus, the unvoiced-stop for /ch/ or /ts/, the unvoiced-fricative for /s/ or /sh/ or the phoneme /h/ are hypothesized as phoneme candidates.
4. The phoneme contexts are hypothesized at the same time and acoustic evidence is evaluated. The acoustic features for the phoneme contexts for region 1) are silence for both left and right. The silence of the left context derives from the location at word initial position. The right silence indicates the possibility of the following vowel devocalization for it is uttered between unvoiced consonants. The left boundary is not so sharp and the strong high frequency power exists above 4,000Hz. Thus, the first candidate for region 1) is the phoneme /s/. Phonemes /h/, /ch/, /ts/ will be the next candidates. The right phoneme boundary is detected at the point of increase and the left phoneme boundary at the point of decrease of the high frequency power.
5. Region 2) is silence, which may be the unvoiced-stop closure. There is something in the high frequency range, which seems to be a double burst. This is one indication of the phoneme /k/. The aspiration is not short enough to suggest /p/ or /t/, and not long enough to suggest /ch/ or /ts/. Thus, the first candidate will be the phoneme /k/. The right boundary will be obtained at the start point of voicing where the low frequency power increases.

6. From steps 4 and 5, the hypothesis of vowel devocalization is in accord with the knowledge "vowel between unvoiced consonants happens to be devocalized".
7. Region 3) is a vowel-like pattern, and using the pattern matching knowledge, the first candidate is hypothesized as the phoneme /u/. Other evidence can be found such as very low frequency power exists, which indicates that it is not the phoneme /a/ or /o/. There is a strong power around 1,200Hz which indicates this is not the phoneme /i/.
8. Region 4) is detected from a sharp spectral gap, which is one evidence of a nasal. The duration is not short and there is a strong low frequency power which is other evidence of a nasal. Thus, the phoneme candidates are /m/ or /n/. To distinguish /m/ and /n/, the formant movement (almost invisible in this case) of the preceding and the following vowel should be captured. The left- and the right boundaries are obtained by the spectral gap.
9. Region 5) is another vowel-like pattern, and using the pattern matching knowledge, the first candidate is hypothesized as the phoneme /a/.
10. Inside the region 6), 7) and 8), there are two silence closures and a fricative-like pattern in the middle. Here, two concatenated unvoiced-stops with a devocalized vowel are hypothesized. At 825ms, a burst is observed. The duration of region 7) is not particularly long. The left boundary of region 7) is not sharp. Region 8) is a complete silence with a sharp boundary on the right side, which is a burst. The aspiration after the burst is very short. From this evidence, the left phoneme candidate is /k/ and the right phoneme candidates are /p/ or /t/. To distinguish /p/ and /t/, the formant movement to the following vowel is important. The left boundary of the first phoneme is detected by the low frequency power decreasing point where silence begins. The boundary of the two phonemes is detected at the point where high frequency power decreases around 870ms. The right boundary of the second phoneme is detected at the low frequency power increasing point.
11. Region 9) is again a vowel-like pattern, and using the pattern matching knowledge, the first candidate is hypothesized as the phoneme /o/. The second formant of this vowel goes up into the preceding closure, which raises the possibility that preceding phoneme is a /t/ rather than a /k/.
12. Region 10) is not short and has a very low frequency power and does not have a high frequency power. This indicates the possibility of a nasal. In this case, the first- and the second formant of the preceding and following vowel go down into region 10). Thus, the first candidate for this region is hypothesized as phoneme /m/. The left and the right boundaries are obtained by the edges of the spectrum.
13. Finally, region 11) is a vowel-like pattern, and using the pattern matching knowledge, the first candidate is hypothesized as phoneme /o/. The pattern of regions 9) and 10) are similar which indicates that these two phonemes are the same vowel. The right boundary is obtained at the point where the following silence begins.

2.3 Expert System

Since the knowledge based approach [Zue 86] was proposed for speech recognition, several speech recognizers have been developed [Carbonell 86] [Mizoguchi 87] [Stern 86] [Connolly 86]. However, most of these conventional systems adopted separate structures for 1) the acoustic feature extraction part and 2) the phoneme recognition part, utilizing only static human knowledge for phoneme identification. Knowledge was represented using frameworks of simple if-then rules and certainty factors. Moreover, very limited acoustic features were used, because of the difficulties of automatic extraction. These acoustic features were extracted and symbolized by pre-processing to be executed by the rules for phoneme recognition.

However, as shown in the aforementioned spectrogram reading process of a human expert, human knowledge consists of not only static knowledge but also dynamic knowledge. A human expert spectrogram reader recognizes phonemes by simultaneously performing phoneme segmentation and phoneme identification using his/her dynamic knowledge in combination with static knowledge. Dynamic knowledge is the strategy for phoneme recognition which is performed by hypothesizing phoneme contexts and by extracting appropriate acoustic features, while static knowledge is the verification of the acoustic evidence.

To cope with this knowledge, a good framework is required. In the proposed expert system, an assumption-based inference is incorporated to describe phoneme contextual knowledge and to realize a contextual non-deterministic strategy. Also, certainty factors and the idea of fuzzy sets, are adopted to represent the uncertain and fuzzy knowledge. Acoustic phonetic features are automatically extracted (on-demand top-down control feature extraction), using appropriate methods and parameters according to the phoneme contexts when the features are referred by the rules. And knowledge, which is difficult to explicitly describe, is represented by neural networks. These techniques make it possible to incorporate human expert knowledge into a system easily and naturally.

2.4 Spectrogram Reading Knowledge

As already mentioned, a human expert simultaneously performs phoneme segmentation which determines phoneme positions in speech as well as phoneme identification while reading a spectrogram. Not only for identification but also for segmentation, a human expert has his/her knowledge concerned with the acoustic phonetic features and coarticulation, and uses this knowledge for segmentation by extracting acoustic features through his/her strategy according to phoneme contexts. Thus, a human expert is able to obtain highly accurate phoneme boundaries, regardless of acoustic variations in the phoneme caused by coarticulation.

The acoustic features which are used for phoneme segmentation are more facile than those of phoneme identification, not only in extracting acoustic features but also in describing in explicit rules.

In this system, the spectrogram reading knowledge, both dynamic and static, is mainly focused on the phoneme segmentation purpose rather than phoneme identification, by simulating the human spectrogram reading process. The segmentation process detects phonemes on a speech spectrogram and determines their left and right boundaries along with their phoneme classes using human expert knowledge and strategy described in rules. Some knowledge for phoneme identification is also incorporated into the system, such as formant frequency range of vowels, however this knowledge is used as additional information and not as the main information for phoneme identification.

2.5 Non-Deterministic Strategy

Phonemes in continuous speech have a number of acoustic variations caused by the effect of coarticulation from the preceding and/or from the following phonemes. For this reason, a human expert hypothesizes various phoneme contexts and acoustic variations of phonemes and evaluates these hypotheses by verifying acoustic evidence when reading a spectrogram. To obtain more reliable phonemes, appropriate strategies have to be selected, and suitable acoustic features have to be extracted from a spectrogram according to the phoneme contextual hypotheses.

Through an assumption-based inference technique, the expert system is able to deal with phoneme contextual knowledge and variations, and also is able to realize a non-deterministic contextual strategy. Phoneme contextual knowledge is described as rules under the conditions of each phoneme context, and is only applied within the hypothesized phoneme context. When several kinds of phoneme contexts may be hypothesized, phoneme detection is performed under each condition of each phoneme contextual hypothesis in parallel, independently. ART's [ART 87] ATMS (assumption-based truth maintenance system) [de Kleer 86] manages the consistency of these hypotheses, by a prohibition of combining contradictory hypotheses.

Each hypothesis is evaluated as correct or incorrect. When the hypothesis is correct, a sequence of certain rules under the hypothesis is applied without contradiction, to determine a phoneme. On the other hand, when the hypothesis is incorrect, some condition of the rules (in which the phoneme context is described) differs from the actual phoneme context on the spectrogram. In such a case, the phoneme cannot be determined because of the contradictory phoneme context, or else the phoneme will be determined with a low certainty. The phoneme with the highest certainty is selected as the final result among all candidates determined from each hypothesis. The assumption-based inference makes it easy and natural to describe knowledge which is dependent on phoneme contexts and on phoneme variations as rules. This is because the conditions of these rules are phoneme contexts and phoneme variations. Moreover, this assumption-based inference also makes it very easy to describe these conditional rules, because it only has to take account of each phoneme context and each phoneme variation condition in each rule.

2.6 Representation of Uncertainty

When reading a spectrogram, a human expert hypothesizes several phoneme contexts and phoneme variations. Simultaneously, these hypotheses are judged as correct or incorrect, by collecting positive or negative evidence using acoustic features extracted from a spectrogram. However, it cannot be defined clearly that these hypotheses were correct or incorrect, and the correctness of these hypotheses can only be obtained. The existence of acoustic evidence on a spectrogram is also very difficult to clearly identify. Most of the evidence can be characterized as "the acoustic feature can be observed clearly" or as "the acoustic feature can be observed but not clearly". These examples show that the existence of the acoustic evidence should be represented with some sense of certainty. Thus, the certainty of the hypothesis is evaluated by its importance, by its certainty, and by the amount of evidence. In this way, the hypothesized candidates through spectrogram reading are represented with a degree of certainty, which cannot definitely be evaluated as right or wrong.

Generally, when solving a problem of uncertainty, there are relations such as *AND*, *OR* and *COMB* (combination) between various evidence used in the hypotheses. The evidence in an *AND* relation shows a necessary condition, and the evidence in an *OR* relation shows a sufficient condition. Evidence in a *COMB* relation can be independent positive or negative proof [Ishizuka 85].

For instance, "the power between 0Hz to 500Hz is large": This evidence is a positive proof that a phoneme is a vowel, and is also a necessary condition. On the other hand, "a double burst exists": This evidence is a positive proof that a phoneme is a burst, but it is not a necessary condition, for no other phoneme except phoneme /k/ has a double burst. Moreover, in spectrogram reading, this kind of evidence is evaluated independently and regardless of order.

There are some methods to deal with uncertainty like Bayesian probability, "MYCIN" certainty factor [Buchanan 85] system, subjective Bayesian method, probabilistic theory of Demster-Shafer, or fuzzy set theory. Each of these methods has advantages and disadvantages. This system basically adopts the certainty factor calculation model of the MYCIN system, and modifies it to be able to deal with the evidence evaluation, regardless of order. This model is adopted because the MYCIN model is able to deal with unsigned values, is able to calculate the *COMB* relation, and is easy to define the importance and certainty of the evidence intuitionally. Moreover, the calculation of certainty factors is easily understandable.

In the certainty factor calculation model of the MYCIN system, the value of the certainty factor CF lies between $[-1, +1]$. When $CF = -1$, the hypothesis is absolutely negative, when $CF = +1$, the hypothesis is absolutely positive, and when $CF = 0$, it means that the hypothesis is neutral and cannot be defined as right or wrong. When the hypothesis has no evidence, its certainty factor is defined as $CF = 0$. The certainty factor CF_P, where P is a condition of some hypothesis, will be calculated according to the relation between the evidence x and y, with their certainty factors of CF_x and CF_y shown in the following equations.

(1) the relation between the evidence x and y is AND:

$$CF_P = min(CF_x, CF_y)$$

(2) the relation between the evidence x and y is OR:

$$CF_P = max(CF_x, CF_y)$$

(3) the relation between the evidence x and y is $COMB$:

$$CF_P = \begin{cases} CF_x + (1 - CF_x) \cdot CFy & \text{if } CF_x > 0 \text{ and } CF_y > 0 \\ \dfrac{(CF_x + CF_y)}{1 - min(|CF_x|, |CF_y|)} & \text{if } CF_x \leq 0 \text{ or } CF_y \leq 0 \\ CF_x + (1 + CF_x) \cdot CF_y & \text{if } CF_x < 0 \text{ and } CF_y < 0 \end{cases}$$

Though there is no theoretical background in this MYCIN equation of the $COMB$ relation, the certainty factor result from this equation is easy to understand intuitively. This certainty factor calculation model is adopted because any positive and negative certainty factors can be combined in any order. The certainty of the hypothesis from the N evidence can be integrated, in general, by applying these equations of relations one by one.

Each certainty factor calculation equation (and, or, combine) preserves commutativity. This means that if only one kind of relation from these three is used and when integrating more than three bits of evidence, the given result will take the same value regardless of the order of the evidence integration. However, if two or three kinds of relations are used among these three equations for the integration, the result value will change according to the order of the evidence integration. This means that evaluation of the evidence using two or three relations from AND, OR and $COMB$ is not possible, regardless of the order of evidence integration.

To avoid this problem, a three-tuple score representation of certainty factors is proposed for each relation of evidence (AND, OR and $COMB$) during the hypothesis as follows:

$$CF_h = \{CF_{and}, CF_{or}, CF_{comb}\}$$

where CF_{and}, CF_{or}, CF_{comb} are the integrated certainty factor results of each AND, OR and $COMB$ relation. The initial value of the certainty factors is $\{1,-1,0\}$, when there is no evidence at the beginning. Each certainty factor of the evidence is integrated into CF_{and}, CF_{or} and CF_{comb} according to each AND, OR and $COMB$ relation. In addition, some weight is multiplied according to the importance of the evidence in the hypothesis, when its certainty factor is integrated in a $COMB$ relation.

CF_h is accumulated into a scalar score, when all evidence evaluations in all the hypotheses have been completed. Thus, the three-tuple scores can generally be integrated into certainty factors of $COMB$ relations using the $max()$ function for an AND relation and the $min()$ function for an OR relation. This is because an AND relation is a necessary condition and an OR relation is a sufficient condition. Moreover, in this system, evidence of a sufficient condition is treated as more important than the evidence of a necessary condition. This is because, generally in spectrogram reading, the hypothesis, in which the evidence of sufficient condition is observed, is determined to be successful. On the other hand, hypotheses in which the evidence of necessary condition is not observed, are not determined to be unsuccessful. Thus, the CF_h is accumulated as a scalar value in the next equation:

$$CF_h = max\{min(CF_{comb}, CF_{and}), CF_{or}\}$$

In this way, each AND, OR and $COMB$ calculation preserves commutativity during the evaluation of the hypothesis, because every evidence integration for each relation is performed individually. Thus, the final scalar result can be obtained regardless of the order of evidence integration. This means that it is possible to integrate results by evaluating a lot of evidence which has relations of AND, OR and $COMB$, in any order.

2.7 Representation of Fuzziness

The human's knowledge of acoustic features extracted from a spectrogram has a certain fuzziness, in other words, it is qualitative. For instance, "the low frequency power of vowel is strong, but that of voiced-fricative is not so strong". Accordingly, the degree of knowledge is represented in qualitative phrases like "very strong", "strong", "not so strong", "weak" and "very weak", and does not have clear boundaries. Moreover, the mapping from the numerical quantity of an acoustic feature to a qualitative concept used by a human expert, is different in each phoneme context. For instance, -50dB of low frequency power is "slightly weak" for a vowel, but "suitable" for a voiced-fricative, while -20dB is "sufficiently strong" for a vowel, but "too strong" for a voiced-fricative.

To deal with such fuzziness, the theory of fuzzy sets [Zadeh 65] is a good framework. Using the fuzzy membership function of this theory, it is easy to map a qualitative representation into a numerical one. The fuzzy knowledge is represented using this idea in this system.

In a spectrogram reading as described above, the mapping from a qualitative representation such as "strong" and/or "weak" to a physical quantity, is not a simple or a single relation, which means that it is not able to give one single fuzzy membership function to each qualitative representation. Thus, the fuzzy membership function must be defined depending on each extracted acoustic feature and depending on each phoneme context. Figure 2-2 shows an example of a fuzzy membership function of low frequency power (0-500Hz) for voiced-fricatives, which appears in the medial part of the utterance. This function shows how the extracted feature fits into the phoneme contextual hypothesis. In other words, the value obtained by mapping the physical quantity through the fuzzy membership function, represents the certainty of existence for the extracted feature. The dynamic range of this fuzzy membership function is defined between $[-1, +1]$ which lies in the same range as the certainty factor, so as to be directly applied to the calculation model of the certainty factor described in the previous section.

As a result, this makes it easy for a human expert to represent knowledge about a physical quantity for the extracted features from a spectrogram using an intuitional mapping, which also makes it possible to evaluate the certainty of a hypothesis for the extracted feature without using any thresholds.

2.8 Acoustic Feature Extraction

Spectrogram reading uses various kinds of global and local, or rough and precise acoustic features on a spectrogram. A human expert is able to extract such acoustic features under the phoneme contexts, by predicting and focusing on the feature existence on a spectrogram, and by selecting the appropriate method with its thresholds. In the same manner, this system extracts the acoustic features under the phoneme context hypotheses simultaneously, when the rules are executed.

This makes it possible to supply an appropriate method with proper parameters by top-down control, to extract the acoustic features, such as frequency ranges, time ranges, thresholds and smoothing factors. As a result, the various acoustic features used by a human expert can be precisely extracted, easily and accurately.

Here is an example of the feature extraction of a power increasing point. This acoustic feature extraction function (power-increase .....) searches for the power increasing point ?time and obtains its value ?change. This function searches from

?point within ?range toward before (left) or after (right) between ?lowfrq (low frequency) and ?highfrq (high frequency) by using the ?*smoothing* (smoothing rate) and its ?*threshold*.

```
(power-increase  (?time ?change)
        after  ?point  ?range  ?lowfrq  ?highfrq  *?smoothing*  ?*threshold*)
```

As described, every parameter depends upon the feature to be extracted with its phoneme and phoneme context. The following two examples show the difference of feature extraction parameters which closely depend on the phoneme and its context.

(1) Searching power increasing point for the right boundary of phoneme /r/ in the medial part of the utterance.

```
(power-increase
      (?time ?change)  after   ?point  ?lqd-range
      ?lqd-lowfrq  ?lqd-highfrq  *?lqd-smoothing*  ?*lqd-start-threshold*)
```

where ?point is search start point,
?lqd-range = 50ms,
?lqd-lowfrq = 1,000Hz, ?lqd-highfrq = 4,000Hz,
?*lqd-smoothing* = (5 3), almost no-smoothing,
?*lqd-start-threshold* = 0.5

(2) Searching power increasing point for the left boundary of phoneme /s/ in the medial part of the utterance.

```
(power-increase
      (?time ?change)  before  ?point  ?frc-range
      ?frc-lowfrq  ?frc-highfrq  ?*frc-smoothing*  ?*frc-end-threshold*)
```

where ?point is search start point,
?frc-range = 150ms,
?frc-lowfrq = 4,000Hz, ?frc-highfrq = 6,000Hz,
?*frc-smoothing* = (10 9), normal smoothing,
?*frc-start-threshold* = 0.5

Here are the acoustic features which can automatically be extracted in the current system.

(a) Spectral power in certain frequency ranges.
(b) Time when the spectral power increases or decreases across thresholds.
(c) Time and magnitude of spectral power change peaks in certain frequency ranges.
(d) Frequency and magnitude of spectrum peaks.
(e) Cutoff frequency of fricative power.

2.9 Time-Delay Neural Networks

Time-Delay Neural Network, TDNN, is a neural phoneme classifier and consists of a layered feed-forward neural network. Waibel showed the incredible performance of TDNN on phoneme identification, comparing with the HMM identification result obtained on the same data [Waibel 89]. Thus, TDNN is adopted as a pattern matching knowledge for phoneme identification in the expert system.

The following properties are considered in a TDNN architecture, to be useful for speech recognition.

- Multiple layers and sufficient inter-connections between units in each of these layers to ensure that the network has the ability to learn complex non-linear decision surfaces.
- Ability to represent relationships between events in time. These events could be spectral coefficients, but might also be the output of the higher level feature detectors.
- Tolerance in time of the actual features and abstractions learned by the network.
- Small number of weights in the neural network compared to the amount of the training data for better generalization.

In the following, the architecture of TDNN design is described.

The basic unit used in many neural networks computes the sum of the weights its inputs and passes this sum through a non-linear function. In the TDNN, this basic unit is modified by introducing delays D1 through DN as shown in Figure 2-3. The J inputs of such a unit will be multiplied by several weights, one for each delay and one for the undelayed input. For N=2, and J=16, for example, 48 weights will be needed to compute the weighted sum of the 16 inputs, with each input measured at three different points in time. In this way, a TDNN unit has the ability to relate and compare current input with the passed history of event. The sigmoid function was chosen as the non-linear output function F due to its convenient mathematical properties. An example of a four layer TDNN with the overall architecture and its connections for three phoneme identification tasks is shown in Figure 2-4.

In the proposed speech recognition system, a TDNN identifier modularly expanded for 18-consonant identification for Japanese and a TDNN phoneme-spotter for five vowels, one syllabic nasal and two semivowels are adopted as a pattern matching knowledge.

2.10 Figures & Tables

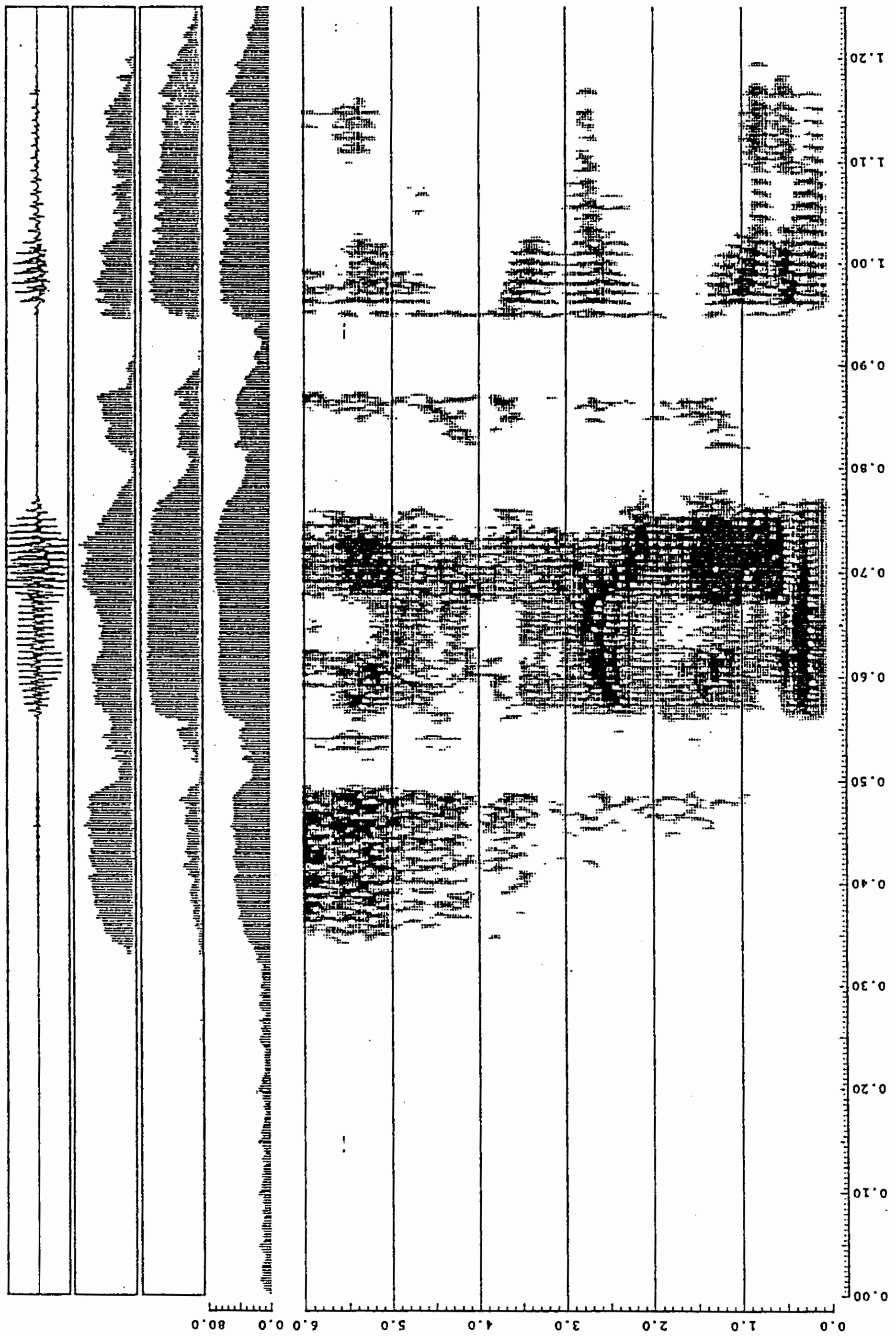


Figure 2-1: Example of Speech Spectrogram (/sukunakutomo/)

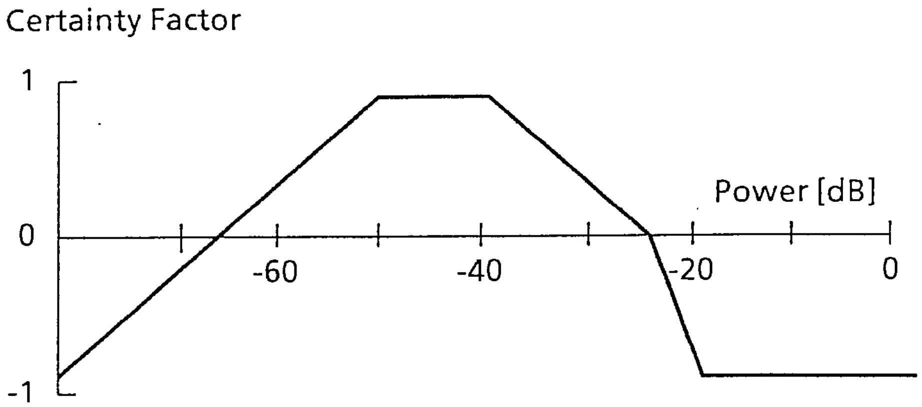


Figure 2-2: Membership Function for 0-500Hz Power of Voiced-fricative

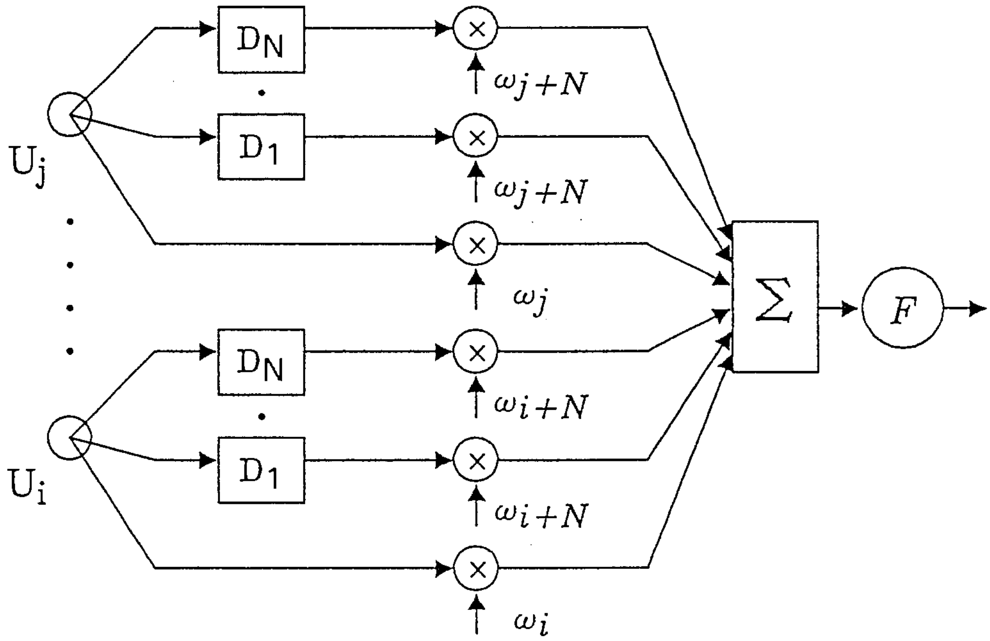


Figure 2-3: Basic Unit of Time-Delay Neural Networks

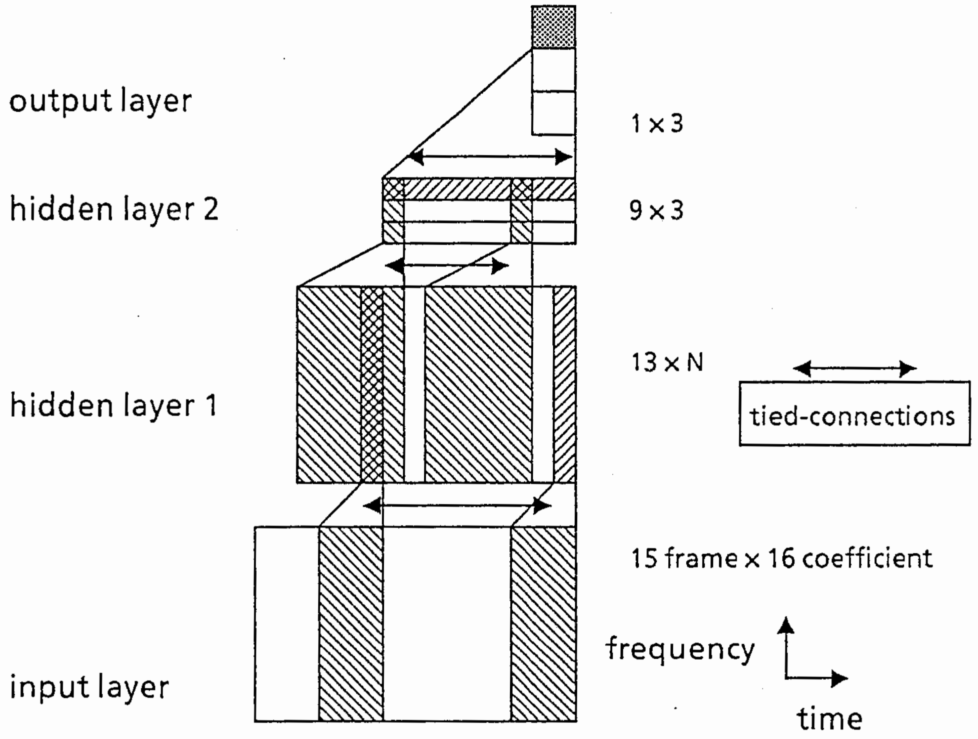


Figure 2-4: Example of Three Class Identification TDNN

Chapter 3

RECOGNITION SYSTEM OUTLINE

3.1 Introduction

In this chapter, the hardware configuration and the system architecture, which realize the proposed phoneme recognition expert system by integrating knowledge and neural networks, are described.

3.2 Hardware

Figure 3-1 shows the hardware configuration of the expert system. The system consists of two workstations, Symbolics and VAX, which are connected by Ethernet for communication.

The system control part and the rule-based part are described by use of ART [ART 87], which is a commercial tool for building expert systems on the Symbolics workstation. The acoustic analysis, feature extraction and neural network perform on a VAX workstation, and these are described in programming language C. According to the requests of the rules on the Symbolics, the VAX replies with the acoustic features and phoneme identification results. The interface program is described in Lisp programming language.

3.3 System Architecture

Figure 3-2 shows the rough architecture of the expert system. The expert system, which recognizes phoneme in continuous speech, reads a spectrogram of an input speech and determine phonemes using the human expert knowledge and strategy by utilizing rules and neural networks. The rules run by hypothesizing and by verifying the possible phoneme candidates and phoneme contexts, as shown in the figure as blocks (□). The lines (——) and arrows (⟶) in the figure show how the hypotheses and verifications perform. The arrow (⟶) indicates the path which gave the final result.

The system mainly consists of three parts:

(1) Consonant recognition.
(2) Vowel recognition.
(3) Phoneme determination.

In the consonant and vowel recognition parts, knowledge and neural networks are integrated so as to improve the overall recognition performance. Finally in the phoneme determination part, the system selects the results of consonant and vowel recognition.

3.3.1 Consonant Recognition

In the consonant recognition part, the knowledge is mainly used for segmentation and the neural network is mainly used for identification. First, the segmentation candidates are obtained using the knowledge. Then, the neural network is closely integrated in order to determine the most likely phoneme category with its boundary.

3.3.2 Vowel Recognition

In the vowel recognition part, the system utilizes a neural network as a phoneme-spotting method for detecting vowel candidates along with their rough locations in the input speech. Then, in combination with rule-based knowledge, the system verifies the vowel categories and determines the vowel boundaries.

3.3.3 Phoneme Determination

In the current system, the consonant recognition result and the vowel recognition result are combined in a simple fashion. The regions where consonant segments are obtained by the expert system are all assumed to be correct, and the other regions where consonant segments are not obtained are assumed to be vowel segments.

3.4 Figures & Tables

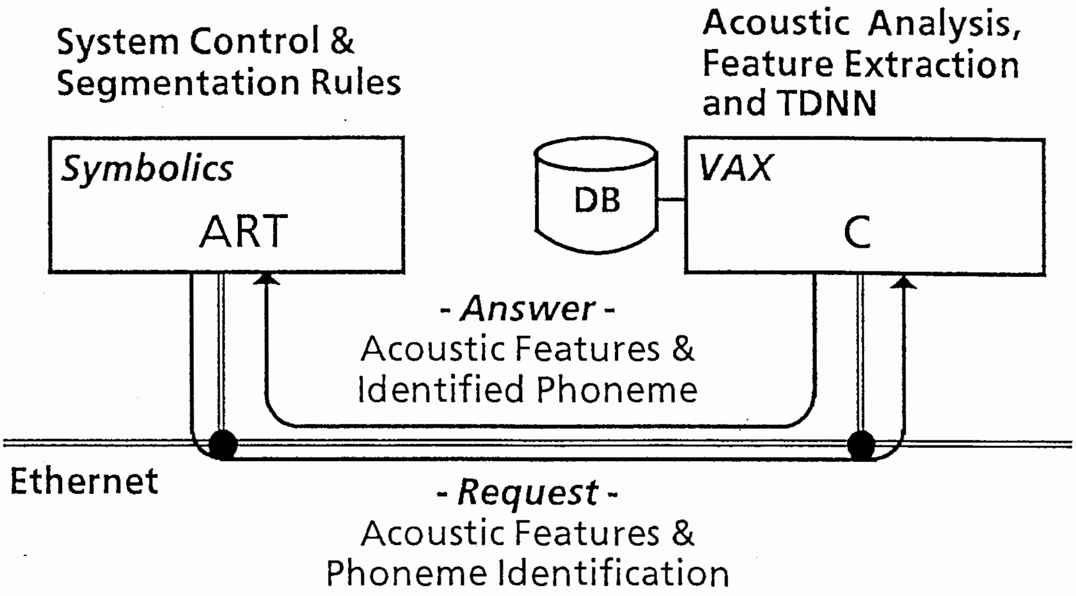


Figure 3-1: Hardware Configration

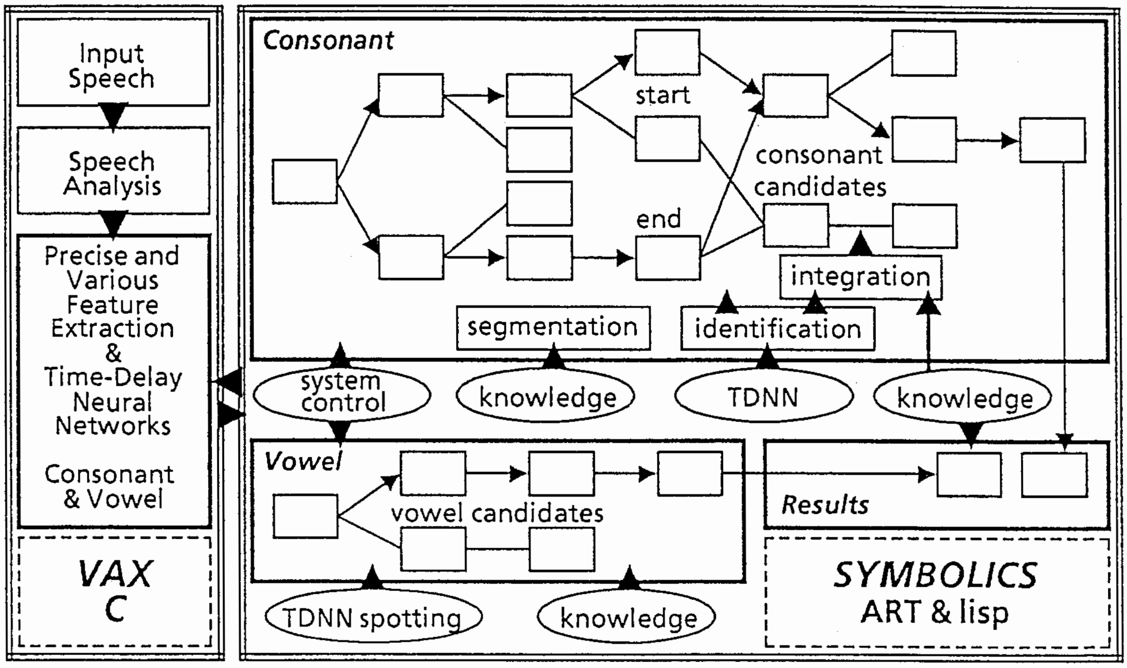


Figure 3-2: System Architecture

Chapter 4

CONSONANT RECOGNITION

4.1 Introduction

In this chapter, the consonant recognition part of the expert system is described. Knowledge in this part mainly consists of two types. 1) human knowledge, both static and dynamic, for phoneme segmentation realized in a rule-based process, and 2) pattern matching phoneme identification knowledge realized in a neural network process. Details of each part and the evaluation of the Japanese 15-consonant recognition is described.

4.2 Consonant Segmentation

Consonant segmentation, which is a rule-based system, is presented in this section. Characteristics, Strategy and Examples are described.

4.2.1 Characteristics

The following characteristics are incorporated into the segmentation part of the system for knowledge representation. Details have been already described in Chapter 2.

- Phoneme boundaries are detected by hypothesizing and by verifying the phoneme contextual acoustic evidence by utilizing the non-deterministic contextual strategy.
- On-demand top-down control feature extraction is performed which makes it possible to extract proper acoustic features using appropriate extraction parameters.

- More reliable phoneme boundaries can be obtained by representing fuzzy human expert knowledge using the theory of fuzzy sets and certainty factors.

4.2.2 Segmentation Strategy

Knowledge of Japanese consonant segmentation is currently incorporated as about 250 rules in the system. The rules are almost described for each phoneme class: 1) unvoiced-stop, 2) unvoiced-fricative, 3) voiced-stop, 4) voiced-fricative, 5) nasal, 6) liquid and 7) glottal. The phoneme segmentation is performed in the following steps as shown in Figure 4-1.

(1) Detecting phoneme candidates.
(2) Hypothesizing phoneme context.
(3) Detecting and evaluating phoneme boundaries.
(4) Selecting more reliable boundaries.

Details of these processes are as follows:

Phoneme Candidate Detection

Phoneme classes and their rough locations are hypothesized as phoneme candidates by referring to global and rough acoustic features, which can be the evidence of the existence of the hypothesized phonemes. At the same time, the certainty factors, computed from this acoustic evidences with some additional evidences, are assigned to the hypotheses. Table 4-1 shows the 7-phoneme class categories in the current system. At this stage, the system uses rough and global acoustic features. Thus, even when the system observes only very slight evidence for a phoneme existence, it tries to hypothesize the existence of the phoneme. As a result, extra phonemes may be hypothesized, in other words more than one phoneme may be hypothesized at the same location in the input speech. These extra phoneme candidates will be rejected by evaluating the suitability of the acoustic evidence to the phoneme contextual hypothesis.

Phoneme Context Hypothesis

The phoneme contexts, which include acoustic variations and their left and right phoneme classes, are hypothesized for each phoneme candidate hypothesized in the phoneme candidate detection stage. In most phonemes, phoneme context is hypothesized as a) silence for the left context and vowel for the right context in the initial part of the utterance, b) vowels for both left and right contexts, in the medial part of the utterance.

Vowel devocalization often occurs when a vowel /i/ or /u/ appears between unvoiced-stops or unvoiced-fricatives. In such cases, vowel devocalization should be assumed. Thus, for a phoneme context hypothesis of an unvoiced-stop which appears in the medial of the utterance, a vowel and a devocalized vowel are hypothesized as its left phoneme contexts, and a vowel, a fricative and another stop as its right phoneme contexts. In addition, the existence of a double burst and the existence of the low frequency power in the aspiration region or in the burst region are also hypothesized for its own acoustic variations.

Boundary Detection and Evaluation

The consonant boundaries are determined in the following steps:

- Boundary Detection
  Phoneme boundary detection, both the start point and the end point, is performed for each phoneme candidate under each phoneme context hypothesis, by referring to local and precise acoustic features. In this way, under the correct hypothesis, correct boundaries can be obtained. Under the wrong hypotheses, some wrong boundaries are obtained and/or are not obtained because of some contradiction in the assumption-based inference. And sometimes, multiple boundaries are obtained under one phoneme context hypothesis.

- Boundary Evaluation
  The certainty of the detected boundary is calculated by integrating the certainty factor of the hypothesis of the phoneme candidate and those of the phoneme contexts. Some hypotheses are evaluated explicitly and others implicitly. Explicit evaluation of a hypothesis is performed by verifying the acoustic evidence directly when the acoustic feature is observed clearly in the phoneme context hypothesis. For instance, in the case of "the right phoneme context is an unvoiced stop", the certainty of this hypothesis will be evaluated by the certainty of the closure existence using its power. On the other hand, in some cases, it is very difficult to evaluate the evidence by direct extraction of the acoustic features. This kind of hypothesis is evaluated implicitly. In such a case, the phoneme boundary is detected without verifying the acoustic evidence in the hypotheses directly, but is assigned a certainty factor which indicates how likely the acoustic measurements are when compared to the conditions around the boundary of the hypothesis. As a result, a phoneme boundary which is obtained under more reliable hypotheses will be assigned a larger certainty factor. For instance, the hypothesis "an unvoiced-stop has no extra low frequency power at the aspiration" is difficult to evaluate directly, because it is not easy to tell the difference between the low frequency power in the aspiration region, from that in the following vowel region.

- Selecting More Reliable Boundaries
  As the result of detecting boundaries, more than one set of left and right

boundaries may be detected for a phoneme, or more than one phoneme may be detected at the same location in the input speech. The more reliable boundaries are selected from all the detected boundaries: By default, the boundaries assigned larger certainty factors are selected. For some phoneme classes, the certainty factors are recalculated by referring to additional acoustic features, for example, phoneme duration after getting the left and right boundaries. Finally, the coarsely classified phonemes and their left and right boundaries are obtained with their certainty factors.

4.2.3 Rule Example

Here, two examples for consonant boundary detection considering their phoneme context are presented.

- **Boundary detection between /h/ and devocalized /u/**
  Figure 4-2 is an example of a spectrogram reading knowledge for phoneme boundary detection: "when the vowel /u/ is devocalized between the phoneme /h/ and an unvoiced-stop, the end point of the phoneme /h/ is located at the boundary between the phoneme /h/ and the closure of the unvoiced-stop". In this system, this kind of phoneme contextual knowledge is described as individual rules. For example, when the condition part of the following rule is sufficient "phoneme candidate /h/ exists, and a silence section exists on its right side", the rule then executes the action, "detect the boundary between phoneme /h/ and silence as an end point of phoneme /h/".

  To detect "the boundary between phoneme /h/ and silence" accurately, the boundary is detected using precise acoustic features according to the phoneme context in the next process;

  (1) First, the system looks for the rough boundary, the rough start point of silence, as a point where the 0-6,000Hz power drops to none.

  (2) Then the system looks for the boundary, hypothesizing that the largest formant of the phoneme /h/ comes into the silence region, using the knowledge that "the phoneme /h/ has the same formant structure of the following vowel". Thus, the system computes the largest formant peak around the start point of silence.

  (3) Next, the system determines that the point where the band frequency power around the largest formant peak ±200Hz drops to none, is the end boundary of phoneme /h/ in this phoneme context.

- **Boundary detection between unvoiced-stop and vowel**
  Figure 4-3 shows a description of a rule to obtain the right boundary of an unvoiced-stop which is followed by a vowel and has no low frequency power in the region of its aspiration. This rule performs as follows:

(1) Rough location where the following vowel starts is found using 0-500Hz power increase.

(2) 0-500Hz power in the vowel region is computed.

(3) The boundary, which is the end point of the unvoiced-stop, is obtained as the time when the 0-500Hz power increases across the threshold of a vowel.

The rule is applied in both cases where the phoneme context hypothesis "a stop having no extra power at the aspiration" is correct and incorrect. When the hypothesis is correct, the correct boundary is detected with a large certainty factor, which is calculated from the power just at the right side of the boundary and from the voiced-onset-time (the duration from the time of burst point to the start time of the following vowel). However, when the hypothesis is incorrect, which means "a low frequency power in the aspiration exists," the boundary is detected at a wrong position (at the start point of the following vowel), where the low frequency power rises. Then, its certainty factor will be calculated from the power of right side of the boundary which will be smaller than the vowel power, and from the voice-onset-time which will be shorter. Thus, the certainty factor of this hypothesis will be smaller than that of the correct hypothesis, which will obtain the correct boundary under a correct phoneme context. As a result, the correct boundary will be selected in the next step.

4.3 Consonant Identification

This section presents a phoneme identification method which applies neural networks to the phoneme segmentation results. The neural network for phoneme identification in this system is the modularly structured Time-Delayed Neural Networks (TDNN) [Sawai 88] which is able to identify Japanese 18 consonants. First, the characteristics and the structure of TDNN are presented.

4.3.1 Characteristics of TDNN

TDNN has the following characteristics:

(1) Easy training using the back-propagation training algorithm.

(2) Easy extension to large phoneme identification tasks by integrating small modules of TDNNs.

(3) High performance phoneme identification using both time and spectrum domain information.

(4) Time-shift tolerance thanks to a time-shifted tied-connected weight architecture.

With these advantages, the neural networks can easily be trained. It can be expected that the system is able to identify a correct phoneme in high performance, even if a slight boundary alignment error occurs in the phoneme segmentation stage.

4.3.2 18-Consonant Identification TDNN

Figure 4-4 shows the TDNN structure used in this system for the identification of Japanese 18 consonants: (/b/,/d/,/g/, /p/,/t/,/k/, /m/,/n/,/N/, /s/,/sh/,/h/,/z/, /ch/,/ts/, /r/,/w/ and /y/).

Structure

This TDNN is made up feed-forward neural networks of four layers. The lowest layer corresponds to spectral input values, the two next layers are hidden layers and the topmost layer, which is the output layer, corresponds to each phoneme output. The hidden layers of this network are modularly constructed from consonant sub-category networks and integrated into one large network.

In particular, the input layer has 15 frames and 16 spectral coefficients (240 units) which make it possible to deal with both the dimensions of time and frequency simultaneously. The first hidden layer has 13 frames and the 2nd hidden layer has 9 frames for the time axis. And in the output layer, there are 18 units which correspond to each of the 18 consonants to be identified.

Connection

The window architecture of the connections between the layers is time-shifted and tied-connected, as shown in Figure 4-4. The connection in the time-shifted window from input layer to hidden layer one is three frames to one frame, and from hidden layer one to hidden layer two is five frames to one frame. The tied-connected link to the output layer has the same weight for each output unit, i.e. the weights of corresponding connections are constrained to be identical by the network training, wherever their positions are shifted frame-by-frame over the time axis. In this way, the network is forced to discover useful acoustic phonetic features in the input regardless of their appearance position within the input window. This is an important property, as it makes the neural network less prone to slight segmentation errors. All weights are adjusted using the back-propagation training procedure. The phoneme corresponding to the highest activated output unit is defined as the classification result.

Training and Identification

In practice, phonemes are culled into data with a length 150ms and analyzed through a 10ms window into 15 frames of input data. For Japanese phoneme training, the phoneme end point of the hand-label is aligned at the 100ms point, from the beginning of the 150ms TDNN input layer. Similarly, in recognition, the end point of the phoneme segmentation result is also adjusted at the same input point of the TDNN input layer. The neural networks were trained using the fast back-propagation training method "Dynet" [Haffner 89]. And the phoneme identification result for the applied segment is determined by the corresponding phoneme of the TDNN output unit which indicates the maximum value.

4.3.3 Knowledge-TDNN Combination

For the baseline system of integrating knowledge and neural networks, the simplest combination of the segmentation part and the identification part is adopted in this chapter. The simplest combination of segmentation and TDNN is shown in Figure 4-5. In this approach, the consonant segmentation result with the largest certainty factor is selected and determined to be the final segmentation result. Then, an 18-consonant identification TDNN (Figure 4-4.) is applied to the segment in order to recognize the exact consonant category.

4.4 Consonant Recognition Experiment

In order to evaluate the consonant recognition process in this expert system, an experimental result is discussed.

4.4.1 Data and Task

The segmentation rules were tuned on an ATR database of phonetically balanced 216 words uttered by one male speaker (MAU). TDNNs were trained on half (even numbered words) of the ATR 5,240 isolated word database [Takeda 88] of the same speaker. Experiments using the proposed system were carried out on the other half (odd numbered words) of the same database. The task given to the expert system was to find the location of consonants in the words and to recognize their categories with their boundaries.

4.4.2 Acoustic Analysis

The acoustic analysis for the input speech is described.

- Input for Segmentation:
  The input speech for phoneme segmentation is sampled at 12kHz and is analyzed by FFT to 64 coefficients of band-powers through a 5ms Hamming window at every 2.5ms shift. Then the spectrogram is smoothed along both the time and frequency axis, and the power is normalized to lie between -20dB and -80dB.

- Input for Identification:
  The input speech for phoneme identification (TDNN) is sampled at 12kHz and is analyzed by FFT through a 21.3ms Hamming window at every 5ms shift. 16 mel-scaled coefficients are computed from the power spectrum to collapse adjacent coefficients in time resulting in an overall 10ms frame rate. The coefficients of each input token are then normalized to lie between −1.0 and +1.0 with the average at 0.0.

4.4.3 Evaluation Criteria

The following are the evaluation criteria for consonant segmentation and consonant recognition.

- Criteria for Segmentation
  The criteria of correct phoneme segmentation, deletion, insertion and substitution are:

  correct: When the boundaries, both the start and the end points of a phoneme detected by the expert system, exist within 50ms of the phoneme boundaries defined by hand-labeling

  deletion: When the boundaries cannot be detected around the correct position where it should be. This case also includes when ether the start or end boundaries detected outside a range 50ms from the hand-labeled boundary.

  insertion: When the phoneme boundaries (both start and end point) are detected by the system where they should not be. This case also includes when either the start or the end boundary is detected out of 50ms range from the hand-labeled boundary. The following are typical examples for insertion errors: "the boundaries of a consonant appeared in a vowel region" or "ether a start or an end boundary is detected out of 50ms range from the hand-labeled boundary."

  substitution: The difference of the number of all consonants and the sum of the corrects and deletions.

- Criteria for Recognition
  When the two following conditions are both sufficient, phoneme recognition is evaluated as correct:

  correct segmentation: Segmentation result is evaluated as correct in the above criteria.

  correct identification: TDNN output unit corresponding to the correct phoneme category obtains the highest activating value and its value is over 0.1.

4.4.4 Result

Table 4-2 shows the experimental result tested on the half of the ATR 5,240 word database not used for TDNN training. Although the neural network is able to identify the 18-consonant categories, the current system is only able to recognize 15-consonant categories, because knowledge for syllabic nasal /N/ and semivowels /y/ and /w/ are not implemented yet. These phonemes are considered to be recognized in a vowel recognition process. Thus, the evaluation of the system is performed on the 15 consonants without these three phonemes.

The segmentation and identification results are described in each column of Table 4-2. **Correct** in the **Segmentation Result** column shows the percentages of the number of phonemes which are evaluated as correct using the criteria described above. **Average Boundary Error** shows the averages of the boundary alignment errors compared with hand-labeled boundaries, and the **Insertion Errors** column shows the rates of extra segments for the number of consonants. **Correct** in the **Identification Result** column, shows the percentages of phonemes both correctly segmented and identified, which indicates the expert system ability. And finally, **In Correct Segment** shows the percentages of correctly identified phonemes for the number in correct segmentation, and **TDNN Ability** shows the percentages of identification tested on phonemes pre-segmented by hand.

The segmentation score was 93.3% with 6.7% deletion error in total, and 5.8ms boundary alignment error on the average. The insertion error rate was 27.8%. TDNN correctly identified 93.0% (**In Correct Segment**) of the phonemes whose segmentation was performed correctly by the system. This score (93.0%) was almost the same score as the 93.3% (**TDNN Ability**) obtained on the hand-labeled pre-segmented phonemes.

Many insertion errors have appeared in the current system and the main reasons for this are:

(1) Most of the current phoneme segmentation rules are described to detect boundaries even if there is a slight possibility of phoneme existence.

(2) Few rules which indicate negative evidence of phoneme existence are incorporated into the system, which is able to reduce the insertion errors.

(3) There are no segmentation rules for vowels, semivowels and syllabic nasals which compete with the consonant segmentation rules; once a consonant segmentation result appears in the vowel regions, it will counted as an insertion error.

Indeed, most of the insertion errors appeared in the regions of vowel and syllabic nasal. If some method for detecting vowels is integrated in this system, the insertion errors will be reduced effectively. Insertion errors caused by voiced-stop and unvoiced-stop segmentation rules mainly appeared at the initial vowel in the utterance. These appear because the acoustic features at the burst point of the initial vowel in an utterance is very similar to the acoustic features of the unvoiced-stop and of the buzz-bar-less voiced-stop. Most of the insertion errors caused by the nasal and the liquid rules occur in the last vowel in the utterance, where the acoustic features, such as spectrum and power, are not stable. These insertion errors appear because the boundary detection rules of these phonemes use very precise spectral features and power changes.

Figure 4-6 shows the distribution of end point alignment errors for the hand-labels and phoneme identification rates for each error location. The horizontal axis shows the alignment error to right or left side compared with the hand-labeled boundaries. More than 90% of the boundaries within the correct segments are detected within the −10ms and +10ms.

From this result, it can be said that the boundaries detected by the system are as accurate as the hand-labeled boundaries, because the hand-labeled ones also have errors averaging less than 8ms [Takeda 88]. When the alignment errors are detected inside the boundary errors of −20ms to +10ms compared with the hand-labels, the phoneme identification rates are almost 90% or more. This performance is as good as the average rate of TDNN ability, and through this result indicated that the time-shift tolerance capability of the TDNN is about 30ms for all consonants, on the average. All these factors indicate the effectiveness of this rule-based phoneme segmentation method and this phoneme identification method based on TDNN.

Figure 4-7 shows the relation between the recognition performance for each phoneme and the segmentation error. The vertical axis indicates the difference between the average phoneme recognition rate (expert system ability) and the original TDNN phoneme identification performance (TDNN ability) obtained by the hand-labeled phoneme identification experiment. The horizontal axis indicates the average segmentation error for each phoneme. The **Average**, × in the figure, shows the average point of the boundary alignment error of 5.3ms for the all phoneme segmentation and -0.9% lower TDNN phoneme identification performance than the original TDNN phoneme identification performance. -0.9% is obtained from "expert system ability" − "TDNN ability" (92.4% − 93.3%).

The unvoiced-fricative /s/,/sh/ and stops /b/,/d/,/t/ were segmented within 5ms of the hand-label accurately and their phoneme identification performance was greater than the average. The tendency of the other phonemes showed that "the larger the boundary error is the lower the identification performance is", except phoneme /p/ and /z/. The reason for the lower performance of phoneme /p/ derives from the insufficient number of the TDNN training data. In this case, even though there is a slight error of the boundary, the performance drops drastically. On the other hand, in the case of phoneme /z/, there are no other phonemes whose feature has both low frequency power and large high frequency power. Thus, the phoneme /z/ is quite different from other phonemes. Thus, even if there is a large segmentation error, high phoneme identification performance on phoneme /z/ can be obtained,

Overall, the expert system correctly recognized 86.8% of the total number of phonemes, both in phoneme segmentation and phoneme identification. Every phoneme recognition rate is over 75%, except for the phoneme /g/. This is primarily because the rules to detect typical /ng/ segments were not described yet, and also because the identification ability of the TDNN for voiced consonants was slightly worse than that for other phonemes. An additional reason for deletion errors in phoneme segmentation is observed. There are several acoustic variations or allophones which appear on the testing database but did not appeared in the 216 word database used for segmentation rule creation.

4.5 Conclusion

This chapter presented the consonant recognition part of the proposed expert system which consists of 1) a rule-based phoneme segmentation based on spectrogram reading knowledge, and 2) phoneme identification based on neural networks adjusted on the resulted segments. It also discussed an experiment result using this system for speaker dependent Japanese consonant recognition.

The consonant recognition part of this expert system has the following characteristics:

(1) Highly accurate phoneme segmentation can be achieved by hypothesizing the coarse classified phoneme and its left and right contexts simultaneously when determining phoneme boundaries.

(2) High performance phoneme identification can be achieved by applying neural networks to the accurate result of phoneme segmentation.

(3) More reliable phoneme recognition results can be obtained because every result and hypothesis for phoneme segmentation and identification are represented with some measure of certainty.

Because of these advantages, the proposed system can achieve a high performance of both phoneme segmentation and identification, which were shown through the experiment. And this method may be one of the most promising ways to build a high performance phoneme recognizer.

The expert system presented in this chapter was realized in a very simple combination of the phoneme segmentation part and the phoneme identification part in which each part performs individually and independently. It is easy to imagine a more sophisticated integration of each part for this expert system, which would improve not only the segmentation accuracy but also the phoneme identification accuracy, and it would definitely improve the phoneme recognition performance itself.

The best way to combine the segmentation and identification methods, so as to make use of their respective merits, should be studied. Phoneme identification could also be improved by applying different kinds of neural networks according to phoneme contexts. In the next chapter, a more sophisticated integration of knowledge and neural networks is proposed and evaluated.

4.6 Figures & Tables

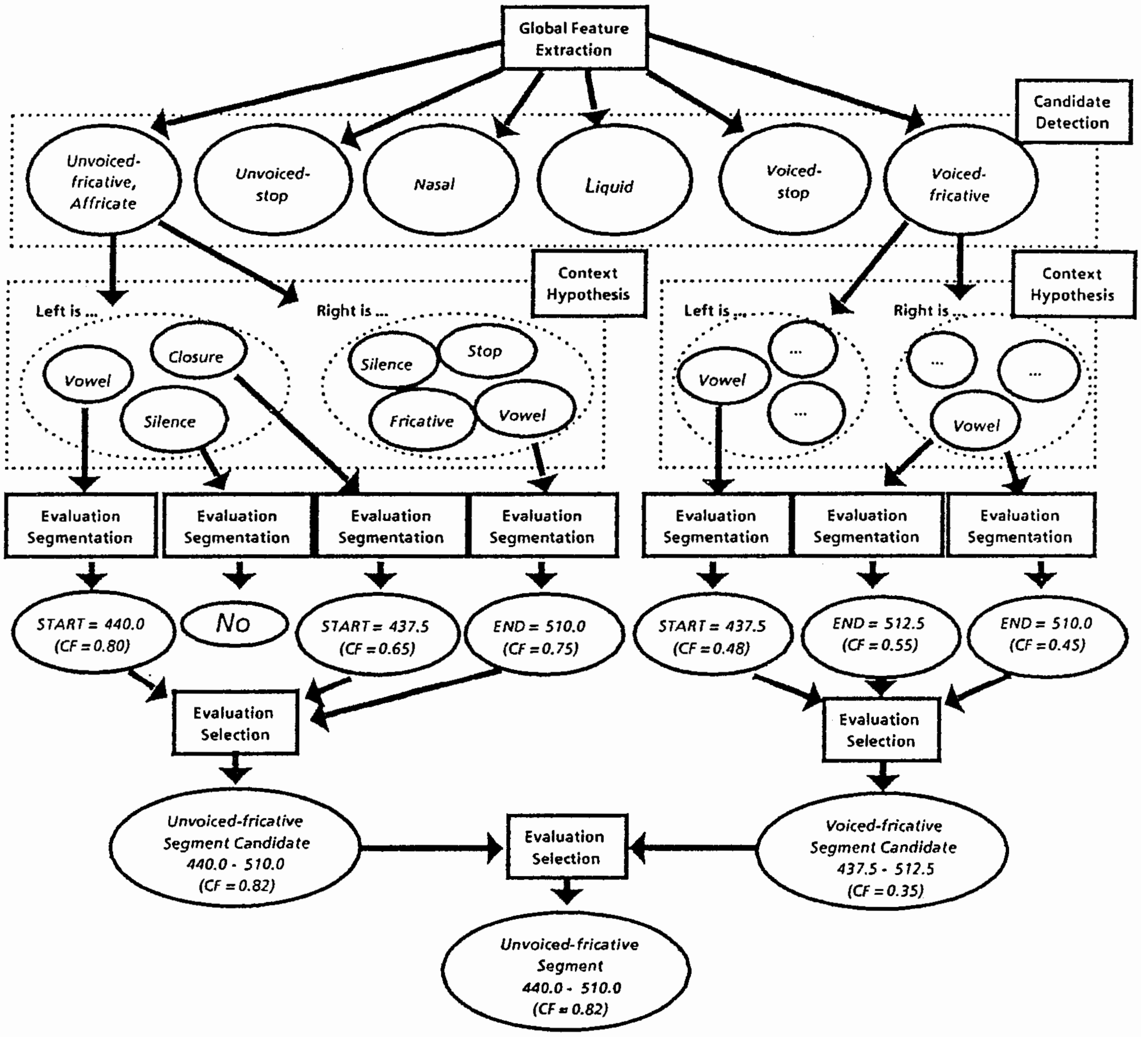

Global Feature Extraction
Candidate Detection
Unvoiced-fricative, Affricate
Unvoiced-stop
Nasal
Liquid
Voiced-stop
Voiced-fricative
Context Hypothesis
Left is ...
Right is ...
Vowel
Closure
Silence
Stop
Fricative
...
Evaluation Segmentation
START = 440.0 (CF = 0.80)
No
START = 437.5 (CF = 0.65)
END = 510.0 (CF = 0.75)
START = 437.5 (CF = 0.48)
END = 512.5 (CF = 0.55)
END = 510.0 (CF = 0.45)
Evaluation Selection
Unvoiced-fricative Segment Candidate 440.0 - 510.0 (CF = 0.82)
Voiced-fricative Segment Candidate 437.5 - 512.5 (CF = 0.35)
Unvoiced-fricative Segment 440.0 - 510.0 (CF = 0.82)

Figure 4-1: Segmentation Strategy

```
(defrule sg-h-right-silence-1
    "Find a right boundary of /h/, which right context is silence."
    (declare (salience ?*right-segmentation*))
    (segment-status ?segment segmentation)
    (category ?segment h)
    (right-context ?segment silence)
?x<-(CF (right-context ?segment silence) ?CFright-silence)
    (not (applied ?segment sg-h-right-silence-1))
    (prop ?segment (candidate-loc ?from ?to))
;;; search segmentation posision of right silence
    (power-end ?power-end&~NONE
        after ?to 150 0 6000 ?*h-right-is-silence-power*)
    (spectrum-peak (?peak-freq&~NONE
        &:(?peak-freq >= 1000)
        &:(?peak-freq <= 6000) ?peak-amp ?peak-Q)
        at =(- ?power-end 10) ?power-end
        ?*spectrum-peak-smoothing-for-h*)
    (not (spectrum-peak (?pk-fq&~NONE&:(?pk-fq >= 1000)
        &:(?pk-fq <= 6000) ?another-peak-amp
        &:(> ?another-peak-amp ?peak-amp) ?pk-Q)
        at =(- ?power-end 10) ?power-end
        ?*spectrum-peak-smoothing-for-h*)
    (power-end ?h-end&~NONE
        &:(< (abs (- ?h-end ?power-end)) 50)
        after =(- ?power-end 50) 100 =(- ?peak-freq 200)
         =(+ ?peak-freq 200) ?*h-right-is-silence-power*)
;;; check charasterictics of right silence
    (power-strength ?h-end =(+ ?h-end 30) 0 6000 ?spw-0-6000)
    (CF (h-power-is-silence-closure ?spw-0-6000)
        ?CFspw-0-6000&:mightbe-valid)
=>
    (retract ?x)
    (assert (applied ?segment sg-h-right-silence-1))
    (assert (prop ?segment (following-silence-start-time ?h-end)))
    (assert (CF (right-context ?segment silence)
         =(CFand (CFcomb ?CFright-silence (CFweight
        ?*evidence* ?CFspw-0-6000)) ?CFspw-0-6000))))
```

Figure 4-2: Rule Example for Glottal /h/.

```
(defrule sg-uvstop-bfr-vowel-3a
        "find a right boundary of an unvoiced stop
        with no  aspiration power in low freq,
        and followed by a vowel."
    (declare (salience ?*left-segmentation*))
    (segment-status ?segment segmentation)
    (CF (category ?segment unvoiced-stop)
            ?CFcategory&:mightbe-valid)
?x <- (CF (right-context ?segment vowel)
            ?CFcontext&:mightbe-valid)
    (CF (has-burst ?segment yes) ?CFburst&:mightbe-valid)
    (prop ?segment (burst ?burst-start ?burst-end ?burst-freq))
    (prop ?segment (has-burst-power-in-low-frequency no))
    (prop ?segment (has-aspiration-power-in-low-frequency no))
    (not (applied ?segment sg-uvstop-bfr-vowel-3a))
    (power-increase (?vowel-region&~NONE ?change)
            after =(- ?burst-start 20) 150
            0 500
            ?*normal-smoothing-size* ?*default-min-change*)
    (power-strength
             =( + ?vowel-region 20) =( + ?vowel-region 40)
            0 500
            ?voicing-power)
    (CF (vowel 0-500-power ?voicing-power) ?CFvowel)
    (power-start ?vowel-start&~NONE
            before =( + ?vowel-region 40) 100
            0 500
             =(- ?voicing-power 10))
    (CF (unvoiced-stop voice-onset-time
             =(- ?vowel-start ?burst-start)) ?CFvot)
=>
    (assert (applied ?segment sg-uvstop-bfr-vowel-3a))
    (retract ?x)
    (assert (prop ?segment (following-vowel-start ?vowel-start)))
    (assert (CF (right-context ?segment vowel)
        =(CFand (CFcomb ?CFcontext
                        (CFweight ?*evidence* ?CFvowel)
                        (CFweight ?*weak-evidence* ?CFvot))
             ?CFvowel
             ?CFvot))))
```

Figure 4-3: Rule Example for Unvoiced-stop.

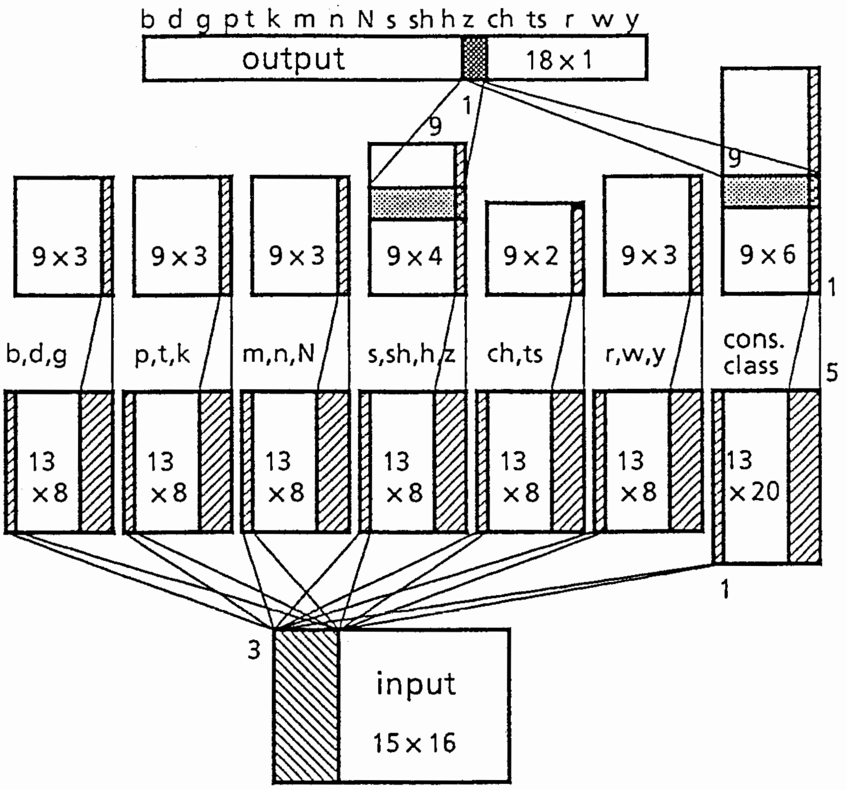


Figure 4-4: 18-Consonant Identification TDNN

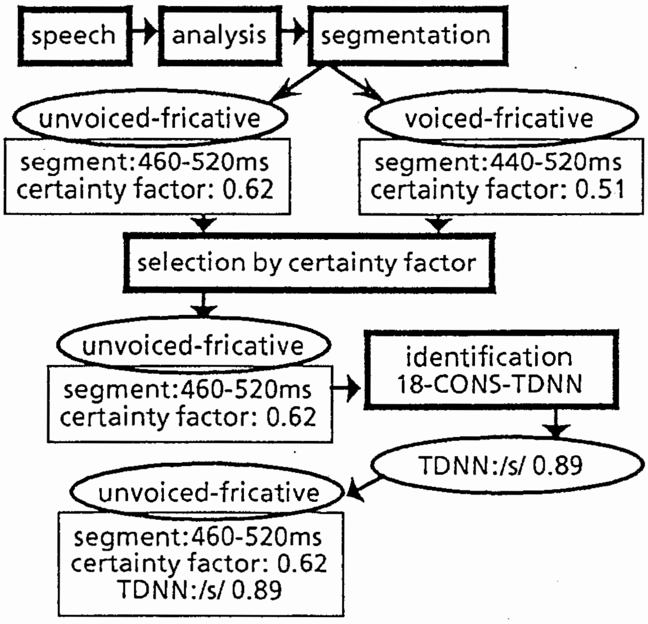


Figure 4-5: Simple Combination of Segmentation and TDNN

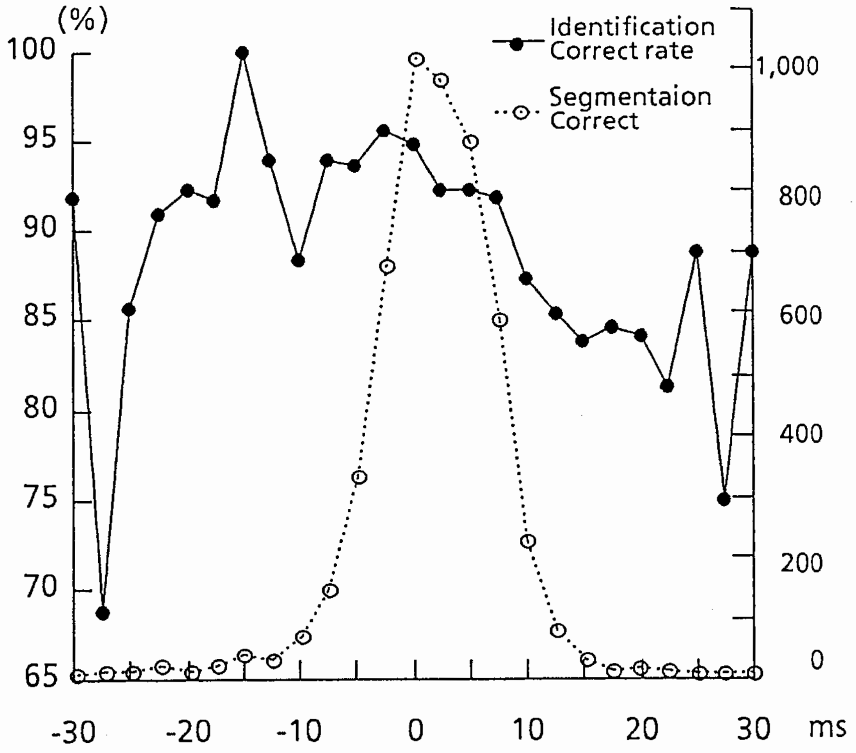


Figure 4-6: Number of End Point Alignment Errors and Phoneme Identification Rate

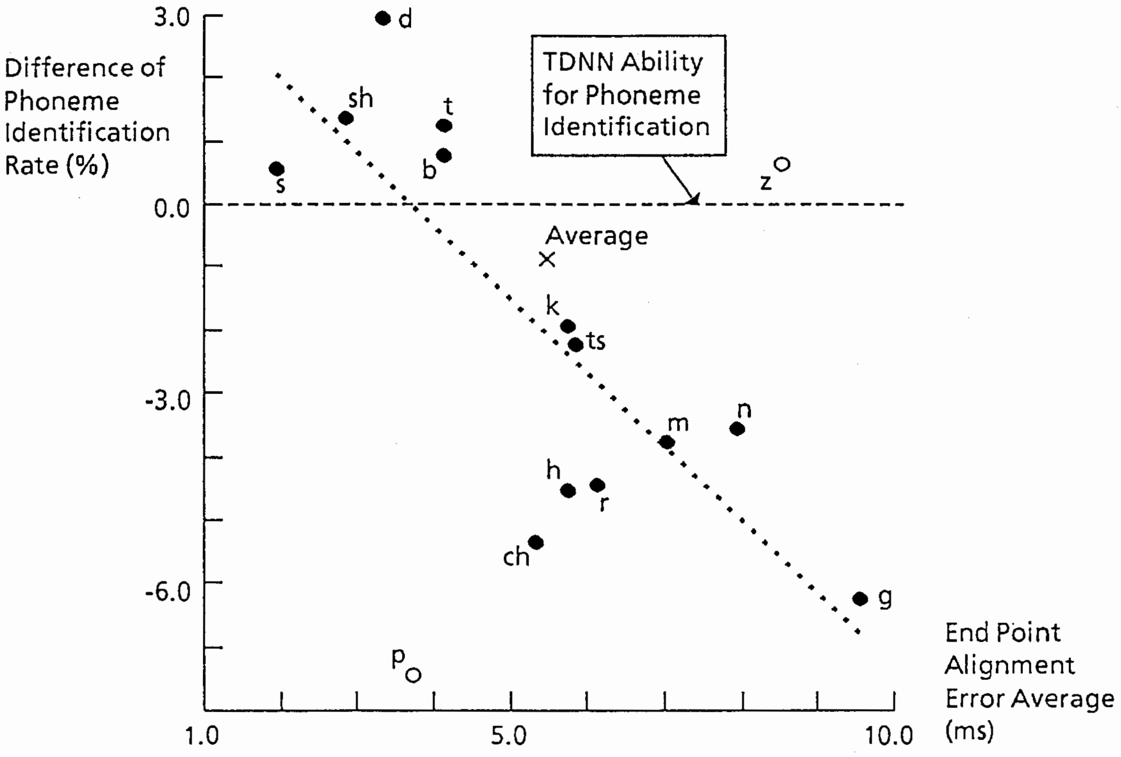


Figure 4-7: Relation between Phoneme Identification Rate and Segmentation Error

Table 4-1: Phoneme Classes and Spectrogram Acoustic Features for Phoneme Candidate Detection

Phoneme Class	Phoneme	Spectrogram Acoustic Features
Unvoiced-stop	p, t, k, ts, ch	Closure & Burst
Unvoiced-fricative	s, sh, h(i)	Large High Frequency Power Indicating Fricative
Voiced-stop	b, d, g	Closure & Burst with Buzzbar, Burst with Weak Buzzbar (Initail Utterance)
Voiced-fricative	z	Large High Frequency Power Indicating Fricative and Weak Low Frequency Power
Glottal	h	Weak Middle and High Frequency Power Indicating Fricative
Nasal	m, n	Large Low Frequency Power and Weak High Frequency Power
Liquid	r	Short Time Power Dip in Middle Frequency

Table 4-2: Phoneme Recognition, Segmentation and Identification Results

Phoneme		Segmentation Result		Insertion Error	Indentification Result		TDNN Ability
Category	Number	Correct [%]	Average Boundary Error [ms]		Correct [%]	In Correct Segment	
p	28	96.4	4.2	17.4	89.3	92.6	100.0
t	461	98.0	4.3		93.9	95.8	94.5
k	1300	97.8	5.8		89.5	91.6	93.5
ch	141	91.5	5.8		75.9	82.9	87.4
ts	220	93.2	5.6		85.5	91.7	93.5
s	572	88.3	3.5	3.5	84.1	95.2	93.5
sh	387	92.0	4.5		91.7	99.7	97.5
h	313	88.8	8.3	7.7	80.5	90.6	94.0
z	315	85.4	9.6	11.1	83.8	98.1	97.5
b	230	98.3	4.7	15.7	93.9	95.6	93.5
d	177	98.3	3.4		93.2	94.8	92.2
g	263	83.7	8.9		70.0	83.6	90.5
m	485	95.3	6.0	94.2	87.0	91.3	93.5
n	273	97.8	5.7		86.1	88.8	89.0
r	760	90.7	6.2	47.4	86.1	94.9	97.5
Total	5925	93.3	5.8	27.8	86.8	93.0	93.3

Chapter 5

INTEGRATING KNOWLEDGE AND NEURAL NETWORKS

5.1 Introduction

This chapter discusses the method of integrating human knowledge and neural networks in the consonant recognition part of this expert system. As previously mentioned, the consonant recognition part is performed in three stages:

(1) Consonant segmentation based on spectrogram reading knowledge.
(2) Consonant identification based on neural networks.
(3) Consonant determination using the results of segmentation and identification stages.

Several mechanisms for integrating phoneme segmentation based on spectrogram reading knowledge and phoneme identification based on neural networks are studied to enhance their respective advantages. Consonant recognition experiments are carried out, and show that the close integration of segmentation and identification improves not only phoneme identification performance but also segmentation accuracy. Furthermore, the proposed integration shows an effective reduction of insertion errors.

5.2 Segmentation and Identification

Details of the phoneme segmentation process and phoneme identification process are already described in Chapter 4.

In the phoneme segmentation process, not only the phoneme boundaries but also the phoneme classes are produced, because the boundaries are obtained under the condition of some assumed phoneme context which includes phoneme class.

In the phoneme identification process, phoneme candidates are produced using the Time-Delay Neural Network. The advantage of the TDNN is a high performance phoneme identification and a time-shift tolerance capability. This capability provides the system with a high phoneme recognition, even if a slight boundary alignment error occurs in the phoneme segmentation stage.

5.3 Integration of Knowledge and TDNN

Several integrating mechanisms of knowledge based segmentation and neural network based identification for the final consonant determination stage are proposed, compared and discussed. Here are the proposed mechanisms:

(1) Simple combination of knowledge and single TDNN (baseline, Chapter 4).
(2) Simple combination of knowledge and selective TDNNs.
(3) Close combination of knowledge and single TDNN.
(4) Close combination of knowledge and selective TDNNs.
(5) Integration of a reject filter.

5.3.1 Simple Combination of Knowledge and Single TDNN

The simple combination of knowledge and single TDNN is shown in Figure 5-1a. In this approach, the consonant segmentation result with the largest certainty factor is selected and determined to be the final segmentation result. Next, an 18-consonant identification TDNN, as shown in Figure 5-2a, is applied to the segment in order to recognize the exact consonant category. Details of this combination are described in Chapter 4.

5.3.2 Simple Combination of Knowledge and Selective TDNNs

Generally, phoneme identification performance of the TDNN is higher when the number of phoneme identification classes is smaller. Thus, if a consonant class is determined with certainty, better identification performance can be obtained by applying a smaller intraclass identification TDNN corresponding to its class. Using this approach, as shown in Figure 5-1b, two separate TDNNs are adopted in order to identify consonants within voiced/unvoiced classes (voiced/unvoiced TDNNs: Figure 5-2b). The appropriate TDNN is chosen according to the voiced/unvoiced class decision, whose result is rarely wrong, obtained in the consonant segmentation stage.

5.3.3 Close Combination of Knowledge and Single TDNN

The rough consonant classification information produced in the phoneme segmentation stage can be utilized in a more sophisticated way in combination with TDNN phoneme identification.

As described in the section on the consonant segmentation part, the system produces not only the phoneme boundaries but also the phoneme classes. This is because the boundaries are obtained under the condition of some assumed phoneme context which includes phoneme class. In the first simplest combination mechanism, this phoneme class information was ignored. The second approach, which is a simple combination of knowledge and selective TDNNs only uses the voiced/unvoiced class decision of the segmentation stage as a pre-process classification. However, to use this sort of information in combination with a TDNN is helpful in improving the overall system performance. The approach proposed here, as shown in Figure 5-1c, is a more sophisticated integration of consonant segmentation knowledge and TDNN identification. The recognition result is determined by considering the suitability of the identified consonant category from the 18-consonant identification TDNN, as shown in Figure 5-2a, with the phoneme class obtained from the consonant segmentation. The final certainty factor of the consonant recognition result, $CF_{rec}()$, is calculated through a suitability function $f()$. The result which obtains the largest certainty factor is determined to be the most reliable recognition result. The integrated knowledge-TDNN certainty factor $CF_{rec}()$, is calculated using the next equation:

$$CF_{rec} = combine(CF_{seg}, CF_{nn})$$

$$CF_{nn} = k \cdot W_{nn} \cdot f(arg(seg), arg(nn))$$

where

CF_{rec}: certainty factor of the final recognition result
CF_{seg}: certainty factor of the segmentation result
CF_{nn}: certainty factor of the identification result
W_{nn}: activating value of the TDNN for the identified consonant
$arg(seg)$: consonant class from segmentation
$arg(nn)$: identified consonant category from the TDNN
k: TDNN reliability (the larger, the more reliable)
$f()$: fitness of consonant for consonant class

```
if (category $\subseteq$ phoneme class)
        then $f()$ returns 1.0;
else if (category $\subseteq$ voiced/unvoiced class)
        then $f()$ returns 0.5;
```

```
else
        $f()$ returns $-1.0$;
combine(): certainty factor calculation model of MYCIN
```

Practically, the suitability function $f()$ is realized as a table of phoneme category, phoneme class and phoneme context. In the current system, the phoneme categories are the 18 consonants; the phoneme classes are voiced-stop, voiced-fricative, unvoiced-stop, unvoiced-fricative, nasal, liquid, glottal (for /h/); the phoneme contexts for this current table are position in the utterance, either initial or medial. The values in the table are between -1.0 and $+1.0$, where $+1.0$ indicates that the results obtained from the knowledge and TDNN have very good positive suitability while -1.0 indicates the contrary. These suitability values have a sense of gradual levels, defined according to the degree of fitness between the phoneme category and the phoneme class.

For example, the phoneme /r/, which appears in the medial of the utterance, has a degree of fitness with phoneme classes as follow: very well with a liquid, quite well with a voiced-stop, slightly with a nasal, rather badly with a voiced-fricative and really badly with the other phoneme classes. And in the current system, the scores in the table for the gradual fitness are fixed as 0.8 for very well, 0.5 for quite well, 0.2 for slightly well, -0.2 for rather badly and -0.8 for really badly.

This integration improves not only the consonant identification performance of this system, but also improves the accuracy of the segmentation. This is because the system chooses the best combination of results obtained by knowledge based segmentation and TDNN based phoneme identification.

5.3.4 Close Combination of Knowledge and Selective TDNNs

The idea of the fourth approach is a combination of the second approach (see 5.3.2) and the third approach (see 5.3.3) applying separate TDNNs for voiced/unvoiced class according to the voiced/unvoiced classification obtained in the segmentation stage, which is shown in Figure 5-1d.

5.3.5 Integration of a Reject Filter

The final approach is shown in Figure 5-1e. In this approach, a reject filter is added in the close combination of knowledge and single TDNN approach. When the result obtained by the segmentation part and the result obtained by the identification part strongly conflict, they will be rejected by this reject filter in order to reduce the number of incorrect insertion errors. In other words, when the phoneme category identified by TDNN conflicts strongly with the phoneme class obtained from the segmentation, the result will be rejected. This mechanism drastically reduces the number of insertion errors. However, this mechanism does not reduce the recognition performance; it only reduces the insertion errors or exchanges the substitution errors for deletion errors. The reason for this is as follows: If the final recognition is correct, the segmentation with phoneme class and the phoneme category is correct. The proposed mechanism does not perform in this condition. If the segmentation is incorrect, this is an insertion error. TDNN applied to this segment may result in some phoneme category. In this case, the phoneme category of TDNN may strongly conflict with the segmentation result and the proposed mechanism performs to reject all these results. Thus, the insertion error will be reduced. If the segmentation is correct but the phoneme category obtained by TDNN is incorrect, this is a substitution error. In this case, the proposed mechanism also performs and if these results strongly conflict, it rejects the results and the substitution error is exchanged for a deletion error.

5.4 Comparison Experiment

Experiments to compare the proposed five mechanisms were carried out using the ATR 5,240 isolated word database. The task given to the expert system was to find the consonants in the words and to recognize their categories with their boundaries.

5.4.1 Experimental Condition

All of the experimental conditions such as database for rule-training, database for TDNN training, database for testing, the task and evaluation criteria, acoustic analysis for segmentation and identification process are exactly the same as those described in Chapter 4 experiment. The neural networks were trained using the fast back-propagation training method "Dynet" [Haffner 89].

5.4.2 Result

Table 5-1 shows the results of consonant recognition experiments using the five proposed mechanisms. The column labeled **Recog.** shows the rate correctly recognized by the expert system for both consonant segmentation and identification. **Ins. Error** shows the insertion error rate. **Seg.** shows the rate correctly segmented. **Boundary Ave. Error** shows the average boundary alignment error of the correct segments for the hand-label. **Ident.** shows the rate correctly identified in the correct segments.

- Simple combination of knowledge and single TDNN ① (baseline).
- Simple combination of knowledge and selective voiced/unvoiced TDNNs ②

- Close combination which considers the suitability of knowledge and single TDNN. k=0.4 ③ (k is the TDNN reliability) and k=0.8 ④ are adopted.
- Close combination of knowledge and selective voiced/unvoiced TDNNs. k=0.4 ⑤ and k=0.8 ⑥ are adopted.
- Integration of reject filter with the close combination of knowledge and single TDNN. k=0.8 ⑦ is adopted.

First, the four mechanisms are compared: ①②③④⑤ and ⑥

Comparing ① and ②, the improvement in columns RECOG. and IDENT. shows the effectiveness of applying smaller consonant identification TDNNs selectively. Comparing ① and ③/ ④ or ② and ⑤/ ⑥, it can be seen that the more sophisticated combination which considers the fitness of the identified consonant category with the consonant class is effective. Almost all results in Table 5-1 improved. In particular, the insertion error rate was effectively reduced.

However, in comparing ③ and ⑤ or ④ and ⑥, no improvement can be seen. This means that the combination mechanism is not adequate for selecting voiced/unvoiced TDNNs. This is because there is no inhibition between voiced and unvoiced consonant classes when using voiced/unvoiced TDNNs selectively. Once a consonant class error occurs in consonant segmentation, an inadequate TDNN is selected and, by combining these results, the certainty factor of the wrong result may be larger than that of the correct one.

The combination mechanism adopted in ④ showed the best total score among the six experiments. The rate, correctly recognized and segmented, shows as good a score as the best in each column. In particular, the insertion error rate is reduced to its lowest value (18.6%), and the average boundary alignment error reaches its minimum value of 5.38ms.

An additional experiment was performed by integrating the proposed reject filter with the best knowledge and TDNN integration mechanism. The function of the suitability between the result obtained form the segmentation part and the identification part are slightly modified to improve the overall system performance. The experimental result ⑦ is shown in Table 5-1. From the result, a drastic reduction in the insertion errors can be seen. Slight recognition improvement can also be observed.

The recognition, segmentation and identification performance of ⑦ for each phoneme is shown in Table 5-2. Correct in the Segmentation Result column shows the percentages of the number of phonemes evaluated as correct using the criteria described above. Average Boundary Error shows the averages of the boundary alignment errors compared with hand-labeled boundaries, and the Insertion Errors column shows the extra segment rate for the number of consonants. Correct in the Identification Result column, shows the percentages of phonemes both correctly segmented and identified, which indicates the expert system ability. And finally, In **Correct Segment** shows the percentages of correctly identified phonemes for the number in correct segmentation, and **TDNN Ability** shows the percentages of identification tested on phonemes pre-segmented by hand.

As a result, a phoneme recognition experiment showed a 89.4% recognition rate for Japanese 18 consonants. The deletion error rate was 5.9%, the substitution error rate 4.7% and the insertion error rate 12.4%, using the best integration mechanism.

5.5 Conclusion

A consonant recognition system, which uses a sophisticated and closer integration of knowledge and TDNN is proposed. The experiments showed that more reliable phoneme recognition results can be obtained by integrating knowledge and TDNN in a more sophisticated manner. Using this approach, the proposed system is able to achieve high phoneme recognition accuracy.

Consonant recognition experiments showed that the closer combination which considers the suitability of knowledge and TDNN improved not only consonant identification but also segmentation accuracy. It also effectively reduced the number of insertion errors. Furthermore, the experiment showed the effectiveness of integrating the proposed reject filter.

5.6 Figures & Tables

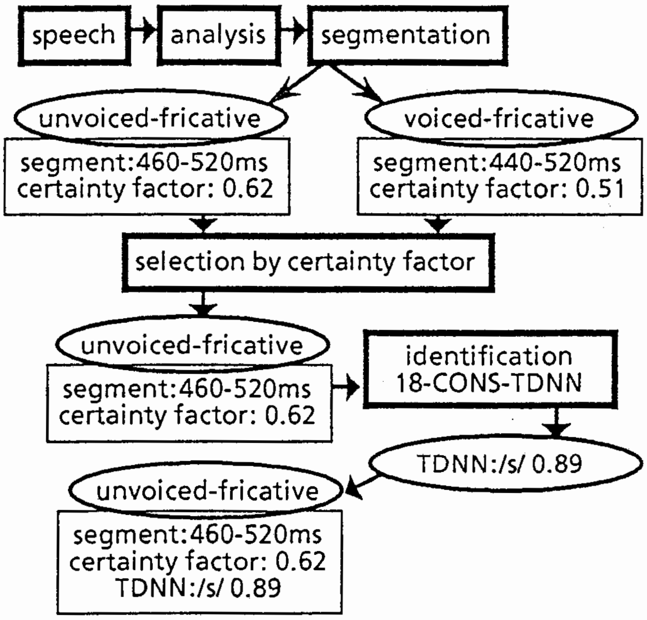


Figure 5-1a: Simple Combination of Knowledge and TDNN

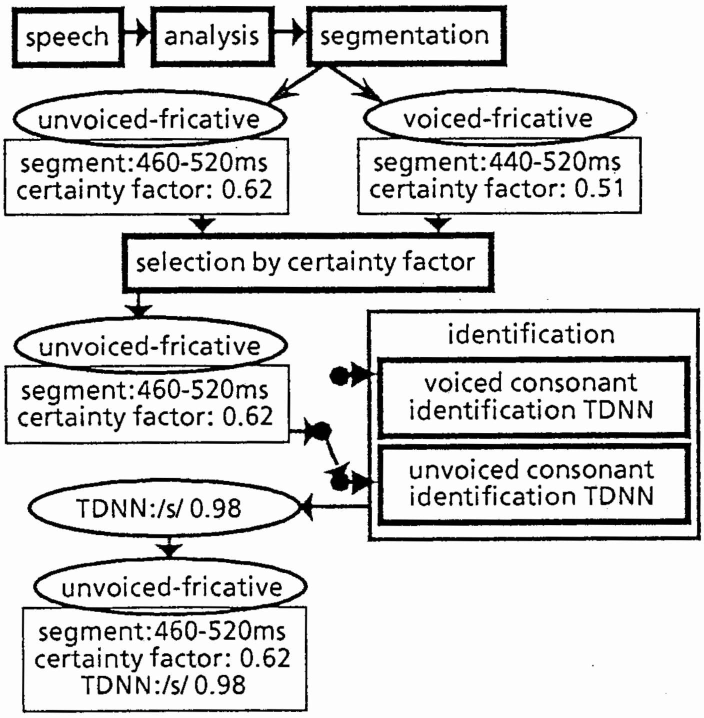


Figure 5-1b: Simple Combination of Knowledge and Selective TDNNs

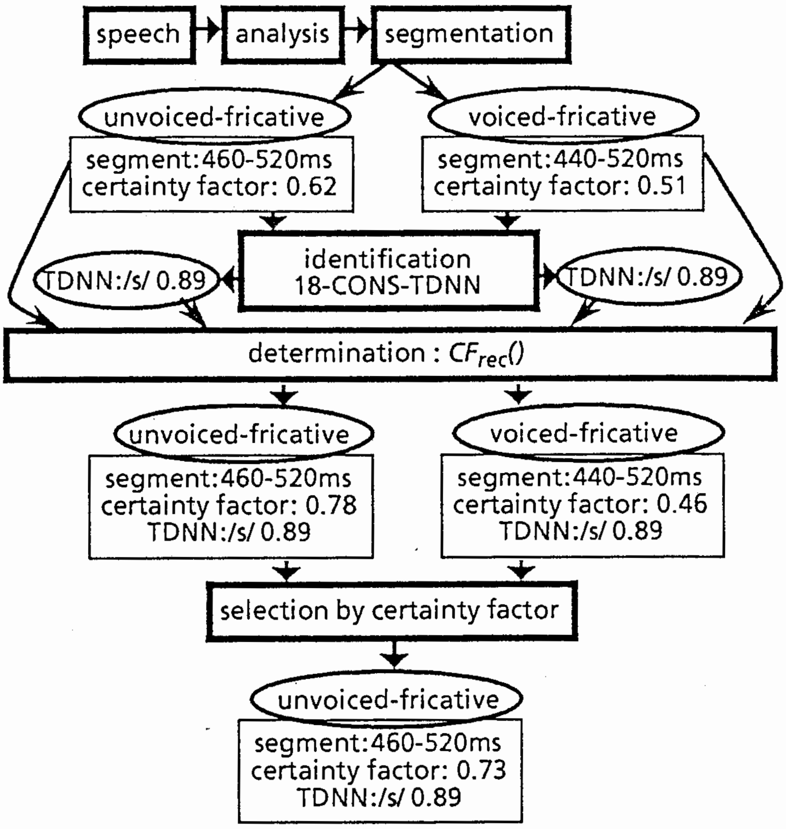


Figure 5-1c: Close Combination of Knwoledge and Single TDNN

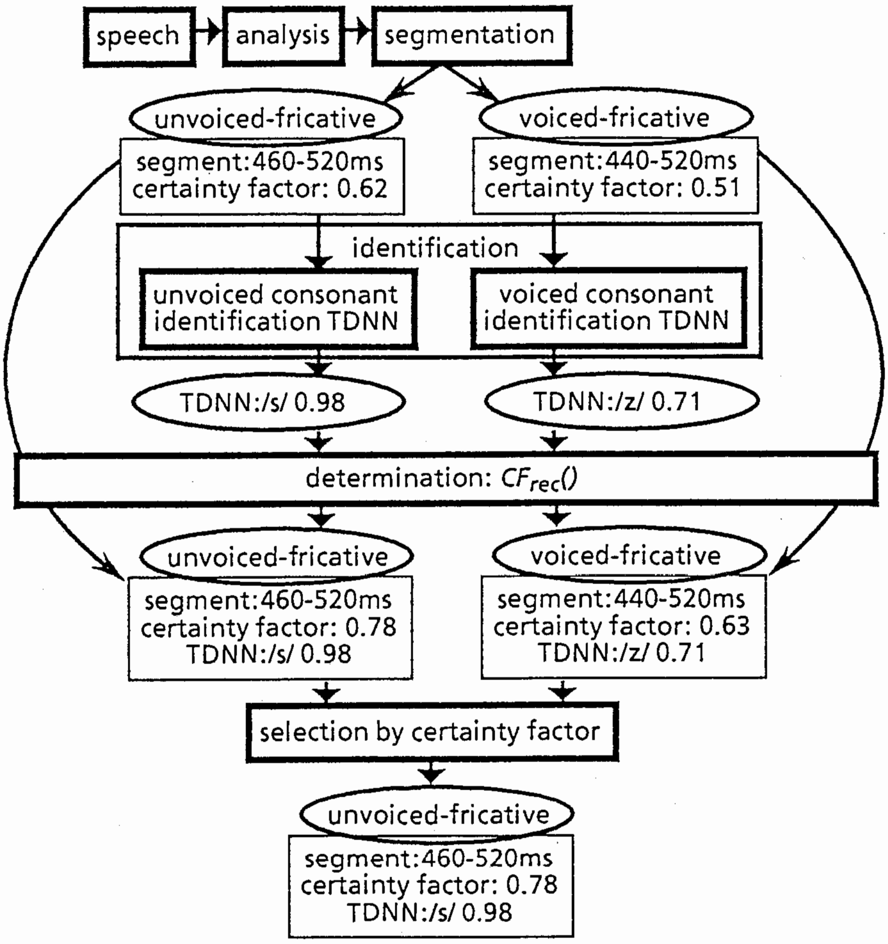


Figure 5-1d: Close Combination of Knowledge and Selective TDNNs

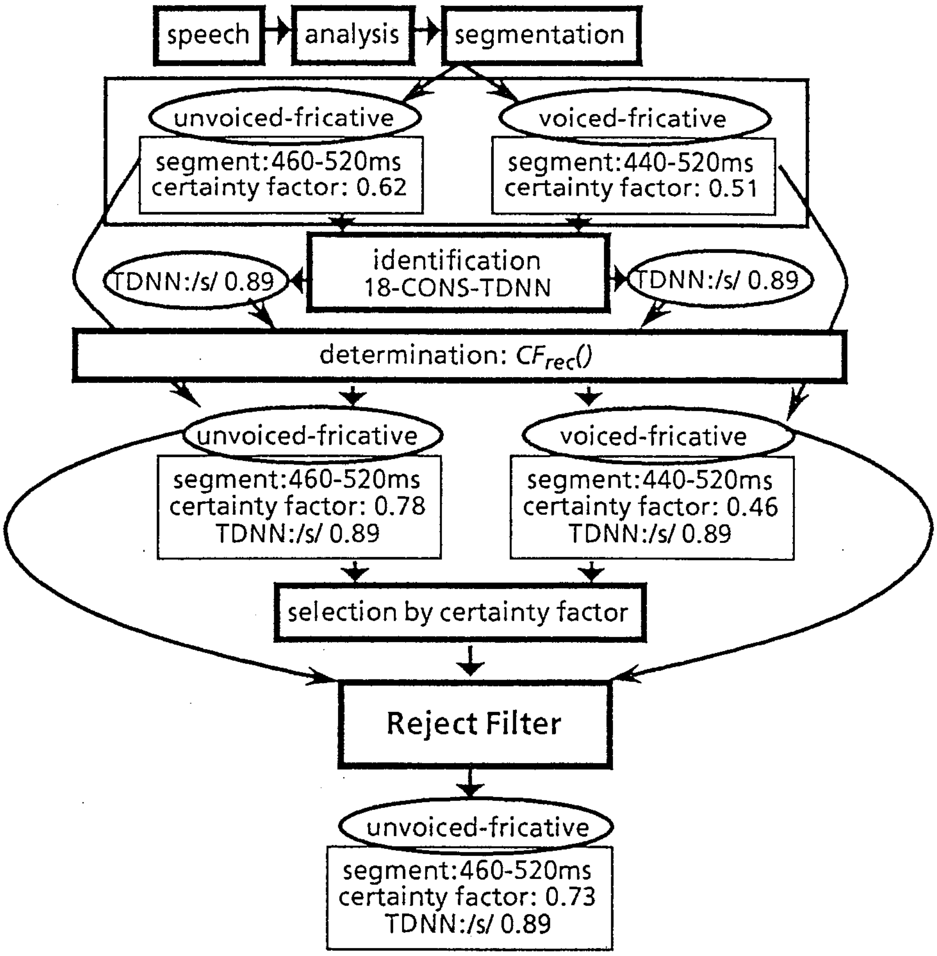


Figure 5-1e: Integration of a Reject Filter

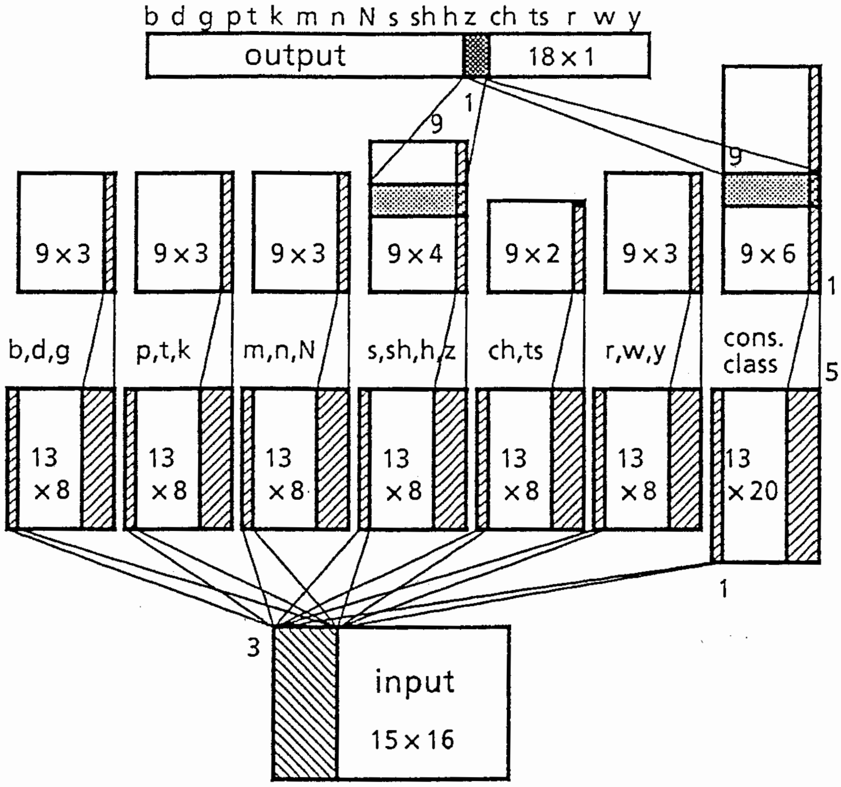


Figure 5-2a: 18-Consonant Identification TDNN

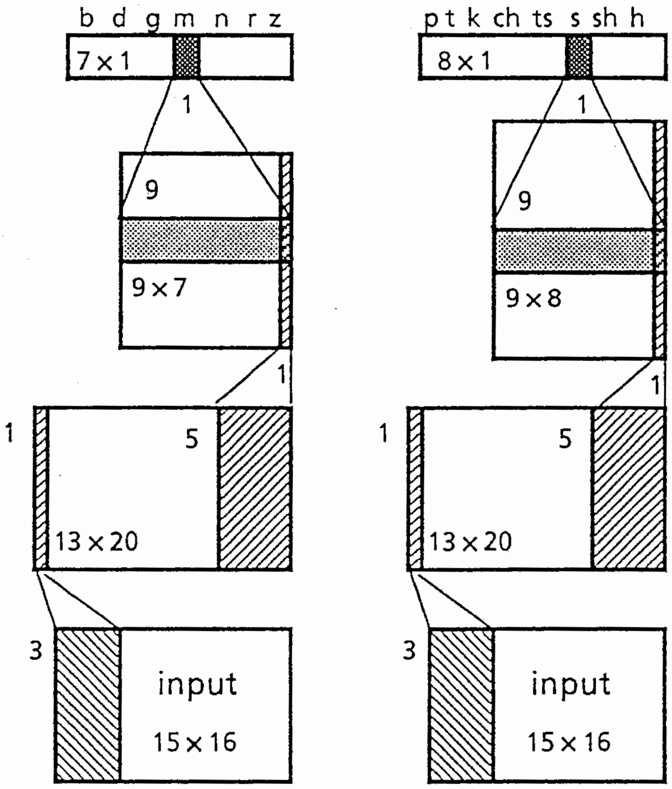


Figure 5-2b: Voiced & Unvoiced Consonant Identification TDNNs

Table 5-1: Integration Mechanism Comparison of Human Knowledge and Neural Networks

Combination Method	Condition			Result				
	Num.	TDNN	TDNN Reliability	Recog.	Ins.	Seg.	Boundary Ave. Error	Ident.
Simple	①	18-cons	-	86.8	27.8	93.3	5.75	93.0
Simple with Selective TDNNs	②	V/UV	-	87.7	27.8	93.3	5.75	94.0
Close	③	18-cons	k = 0.4	88.8	22.0	94.6	5.43	93.9
	④	18-cons	k = 0.8	88.8	18.6	94.5	5.38	93.9
Close with Selective TDNNs	⑤	V/UV	k = 0.4	88.8	22.6	93.7	5.43	94.8
	⑥	V/UV	k = 0.8	88.4	22.6	93.1	5.40	94.9
With a Reject Filter	⑦	18-cons	k = 0.8	89.4	12.4	94.1	5.42	95.0

Table 5-2: Phoneme Recognition, Segmentation and Identification Results

Phoneme		Segmentation Result		Insertion Error	Indentification Result		TDNN Ability
Category	Number	Correct [%]	Average Boundary Error [ms]		Correct [%]	In Correct Segment	
p	28	92.9	4.6	11.9	89.3	96.2	100.0
t	461	98.3	4.3		95.4	97.1	94.5
k	1300	96.6	5.7		91.0	94.2	93.5
ch	141	91.5	5.8		81.6	89.1	87.4
ts	220	93.6	5.0		87.6	93.6	93.5
s	572	92.8	3.3	3.4	89.9	96.8	93.5
sh	387	94.3	5.4		93.8	99.5	97.5
h	313	91.7	8.9	0.6	86.6	94.4	94.0
z	315	87.0	9.4	1.0	86.7	99.6	97.5
b	230	96.5	4.7	12.8	95.2	98.6	93.5
d	177	98.3	3.5		90.4	92.0	92.2
g	263	79.8	8.9		73.0	91.4	90.5
m	485	94.0	5.7	33.8	86.8	92.3	93.5
n	273	95.2	5.2		86.4	90.8	89.0
r	760	95.4	4.0	13.0	91.2	95.6	97.5
total	5925	94.1	5.4	12.4	89.4	95.0	93.3

Chapter 6

VOWEL RECOGNITION

6.1 Introduction

The vowel recognition part in this expert system utilizes a neural network for detecting vowel candidates. The spectrogram reading knowledge is utilized for verifying the vowel categories and for detecting boundaries. In this part, five vowels /a,i,u,e,o/, one syllabic nasal /N/ and two semivowels /y,w/ are recognized. The neural network for vowel recognition is also a TDNN which is used as a phoneme-spotting method. The time-shift tolerance capability of the TDNN is expected to be a good phoneme-spotting method to detect vowels, syllabic nasal or semivowels whose spectral features are stable or change smoothly.

6.2 Vowel Recognition

Vowel recognition is performed in the following steps:

Vowel Region Detection

Possible region for vowels is determined using the power of low frequency range (0-1,500Hz and 500-1,000Hz).

Vowel Region Division

The vowel region is divided at the point of a large spectral change peak in the low and middle frequency range (0-3,000Hz) by assuming that a vowel change exists at that point.

TDNN Vowel Spotting

Vowels and their rough locations are detected using TDNN for phoneme-spotting in the divided vowel regions. TDNN, which detects vowels and semivowels, is shifted over the input speech frame-by-frame, as shown in Figure 6-1, which spots five vowels /a,i,u,e,o/, one syllabic nasal /N/ and two semivowels.

TDNN is trained by adjusting the center of the vowel and semivowel samples to the center frame of the input layer. The frame-by-frame vowel outputs which have values over a certain threshold are blocked together, and the block is hypothesized as a vowel candidate. The certainty factor of the vowel candidate depends on the block duration and the sum of the activating values within the block. If this certainty factor is not large enough, the hypothesis will be rejected.

Boundary Detection and Category Evaluation

The boundaries of the vowels /a,i,u,e,o/ and /N/ are detected by searching for points where the spectral difference in the low and middle frequency range (0-3,000Hz) rises over a fixed threshold. The search is conducted from the middle of the hypothesized vowel candidate toward its left and right sides. The existence of the semivowels /y/ and /w/ is evaluated by phoneme context. In Japanese, /y/ and /w/ can appear only in a very limited phoneme context. Phoneme /y/ appears at the utterance initial position or between the phonemes /p,b,k,g,z,m,n,r,h/, the five vowels /a,i,u,e,o/, and /a,u,o/. Phoneme /w/ only appears at the utterance initial position or between the five vowels and /a/.

Additionally, the certainty factor for the candidate is recalculated by using duration information. When a conflicting region exists among these vowel candidates, each certainty factor is recalculated using the lowest frequency spectral peak in the conflicting region, by assuming the lowest frequency spectral peak as a first formant. The vowel given the largest certainty factor is determined to be the vowel recognition result.

Without the integration of this knowledge, i.e. if the system directly uses the results of the vowel-spotting TDNN, a large number of insertion errors may occur.

6.3 Vowel Recognition Example

Figure 6-2 shows a vowel recognition example and Figure 6-3 shows the spectrogram for this utterance. The utterance is a Japanese word /omowazu/. The horizontal axis indicates the time scale and the vertical axis indicates the frequency scale. The phonemes and acoustic events of hand-label lie at the top of this figure. The segments under the hand-labels are the recognition results obtained by the expert system. The dotted segments just below the recognition result with vowel characters are the vowel candidates produced by TDNN vowel-spotting. Between 600ms to 700ms, two candidates of vowel /a/ and /o/ are obtained by the TDNN-spotting. Spectrum peaks are calculated between the conflicting region. In the figure, formant peaks 1,218.8Hz and 2,625.0Hz were obtained. Using these formant peaks, the certainty factors of the two vowel candidates are recalculated. The formant of vowel /o/ is generally lower. Thus, the certainty of the vowel /o/ candidates is decreased and finally the vowel /a/ is obtained as a vowel result for this region.

6.4 Vowel Detection Experiment

Here, the effectiveness is shown through a vowel detection experiment.

6.4.1 Experimental Condition

The training data for the vowel-spotting TDNN are selected from the odd numbered words in the ATR 5,240 isolated word database, up to 500 phonemes for each category. The acoustic analysis for the input speech conditions are exactly the same as those used in the consonant recognition experiments in chapter 4. The neural networks were trained using the fast back-propagation training method "Dynet" [Haffner 89]. The task given to the expert system in this vowel detection experiment was to determine what kind of, and how many, vowels will be detected in the vowel regions of the input speech. In this task, mis-detection in the consonant regions of the input speech is of no consequence.

Here, the blocked vowel candidates were used to evaluate vowel detection performance. The blocked vowel candidate with the largest certainty factor is selected from among the candidates which overlap each other in the middle of their block candidates.

6.4.2 Result

Table 6-1 shows the result of the vowel detection experiment. The column **Num.** is the number of vowels for testing. The column **Rate** is the ratio of correct detection to the number of testing vowels. The bold number on the diagonal of this matrix is the number of vowels which are correctly detected. And the number in the parentheses () are the insertion errors. The total detection score was comparatively good, 96.1% [7,726/8,043]. The top five causes for detection error were: 1) vowel /a/ mis-activated in /y/ following /a/, 2) /i/ in /y/, 3) /u/ in /N/, 4) /o/ in /u/, 5) /u/ in /y/ following /u/.

6.5 Conclusion

The vowel recognition part in this expert system is described. This part utilizes the TDNN as a vowel-spotting method for vowel candidate detection. The spectrogram reading knowledge is used for category verification and for boundary detection. The effectiveness of the proposed method is shown by the vowel and semivowel detection experiment.

6.6 Figures & Tables

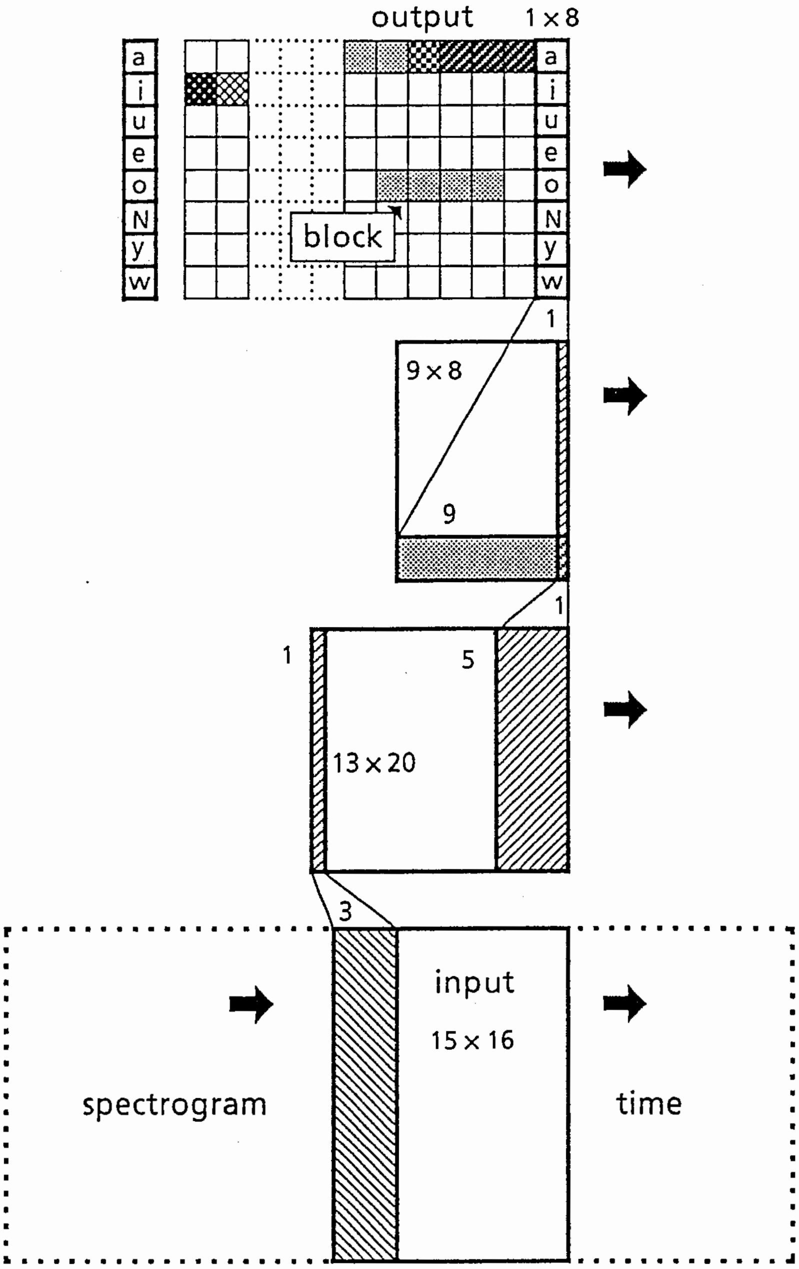
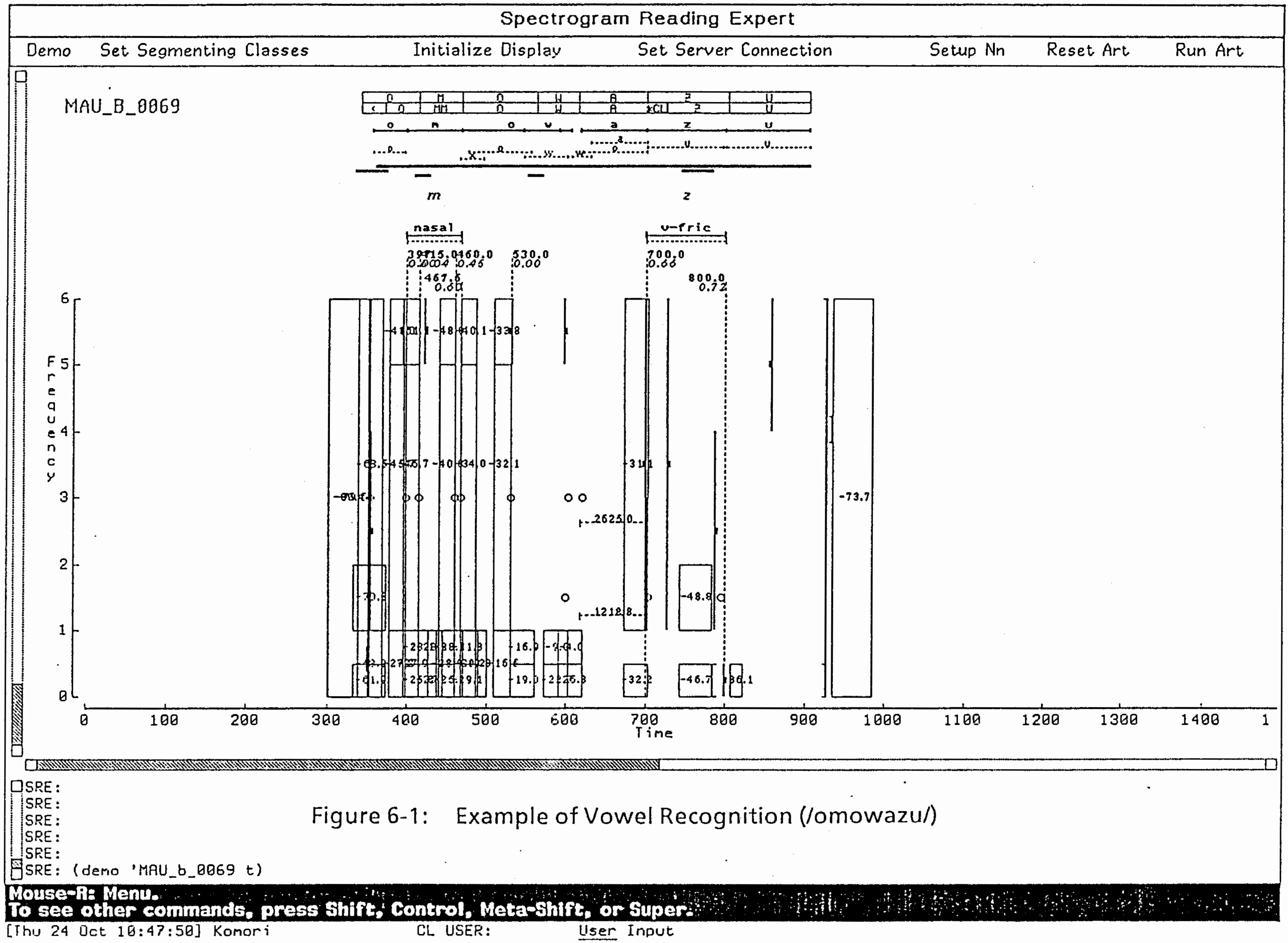


Figure 6-1: Vowel-spotting TDNN



```
SRE:
SRE:
SRE:
SRE:
SRE:
SRE: (demo 'MAU_b_0069 t)
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Thu 24 Oct 10:47:50] Komori          CL USER:          User Input
```

Figure 6-1: Example of Vowel Recognition (/omowazu/)

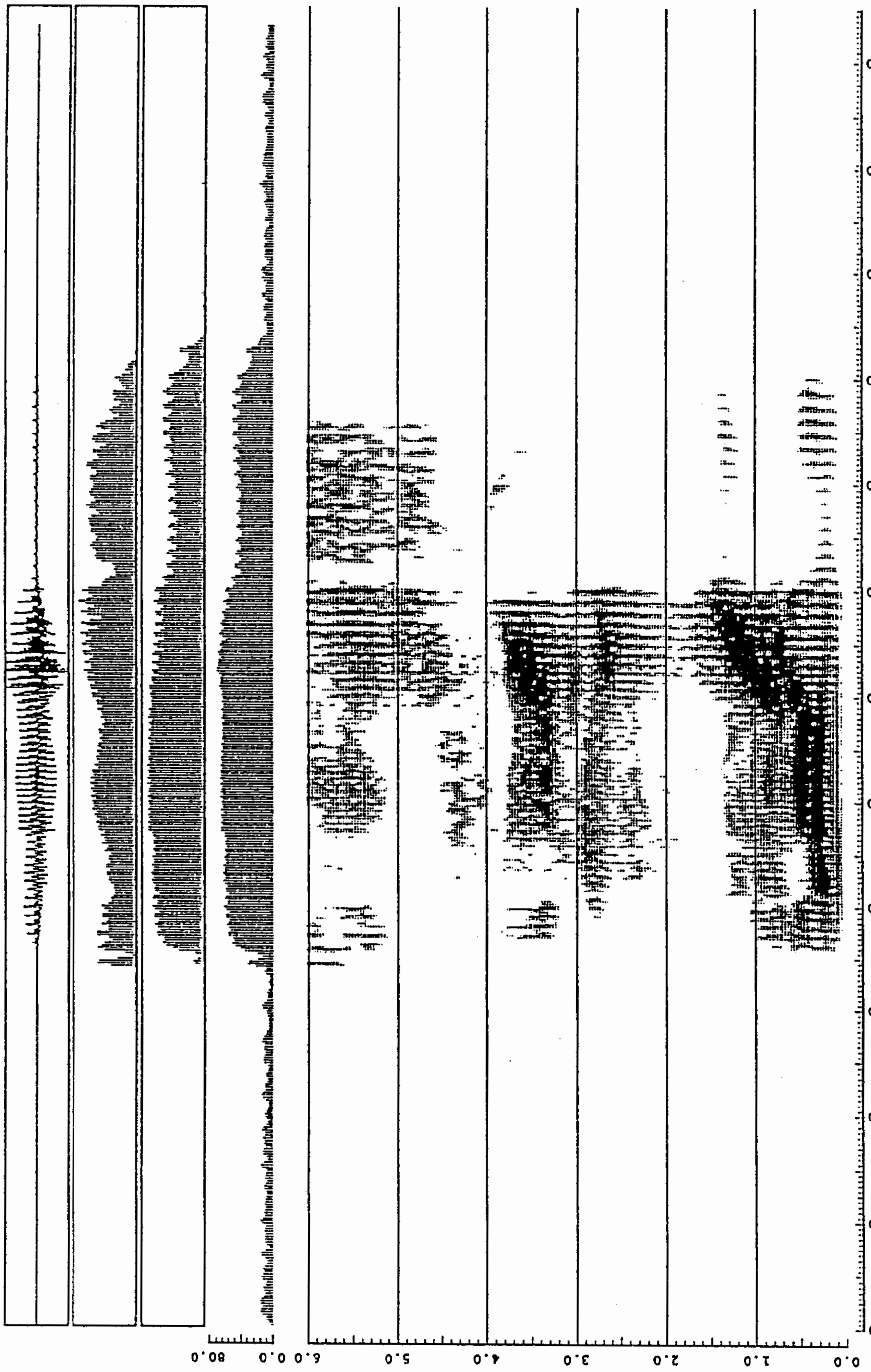


Figure 6-2: Example of Speech Spectrogram (/omowazu/)

Table 6-1: Vowel Detection

	Num.	a	i	u	e	o	N	y	w	Rate %
a	1772	**1771** (9)	-	(4)	-	(8)	(1)	(1)	-	99.9
i	1333	-	**1186** (51)	(10)	(4)	-	(2)	(23)	(1)	89.0
u	1615	(9)	(20)	**1521** (70)	(20)	(54)	(11)	(6)	-	94.2
e	829	(5)	(5)	(6)	**820** (18)	-	(1)	(3)	-	98.9
o	1352	(28)	-	(26)	-	**1339** (34)	(4)	-	-	99.0
N	488	(17)	(17)	(78)	(4)	(3)	**464** (15)	-	-	95.1
y	573	(88)	(85)	(44)	(21)	(16)	-	**554** (12)	(1)	96.7
w	81	(33)	(3)	(2)	-	(17)	(1)	(1)	**71** (0)	87.7

Average Detection Rate : 96.1% [7726 / 8043]
Number in () Indicates the Mis-spotting Vowel

Chapter 7

FULL SYSTEM EVALUATION

7.1 Introduction

This chapter presents a phoneme recognition example using the proposed expert system and discusses all phoneme recognition experiments without using any language model.

7.2 Recognition Example

Figure 7-1 shows an example recognized using the current expert system. The input speech is /subete/, whose spectrogram is shown in Figure 7-2. In the figure, ① shows the spectrogram plane where the horizontal axis indicates the time axis(ms) and the vertical axis indicates the frequency axis (kHz). The blocks ② with the alphabetic labels at the top of the figure are the hand-labels. The upper ones are the phoneme labels and the lower ones are the event labels. Immediately below are the final recognition results ③of the system for this input. The dotted-line segments ④ with vowel labels are the vowel-spotting results from the vowel identification TDNN. The next bars ⑤ are the global acoustic features for searching for the rough location of phonemes. The characters ⑥ under the bars are the consonant candidates from the 18-consonant identification TDNN. The segments ⑦ with the phoneme classes are the segmentation results. The dotted-line segments ⑧ are the phoneme segment candidates. The vertical dotted-lines ⑨ are the candidates for the phoneme boundaries and the number indicates their position in the time scale. The a) rectangles, b) bold vertical lines, c) small circles, d) bold rectangles and e) horizontal lines on the spectrogram plane ① are the acoustic features ⑩ used in the current system:

a) spectral power in certain frequency ranges.

b) time when the spectral power increases or decreases across thresholds determined according to the phoneme context.
c) time and magnitude of spectral power change peaks in certain frequency ranges.
d) frequency and magnitude of spectrum peaks.
e) cutoff frequency of fricative power.

In this example, the phonemes are correctly recognized both in segmentation and identification.

7.3 All Phoneme Recognition Experiment

All phoneme recognition experiments were performed for total system evaluation purposes.

7.3.1 Experimental Condition

The experimental conditions are exactly in the same conditions previously described. The acoustic analysis for the input speech conditions are the same. The knowledge for the rules is created using an ATR database of 216 phonetically balanced words uttered by a single male speaker (MAU). Both TDNNs for the 18-consonant identification and for vowel-spotting are trained on half (the even numbered words) of the ATR 5,240 isolated word database, uttered by the same speaker. The neural networks were trained using the fast back-propagation training method "Dynet" [Haffner 89].

All phoneme recognition experiments were performed using the other half of the 5,240 isolated word database (the odd numbered words). The task given to the system was to find phonemes in the words and to recognize their categories.

In the consonant recognition part, the proposed close integration of knowledge and TDNN with a reject filter, which showed the best performance in the consonant recognition experiment, is adopted. The consonant recognition part and the vowel recognition part are combined in a very simple fashion. In the current system, the regions which have consonant segments are assumed to be correct and the results obtained by the consonant recognition part is determined as consonant regions. The other remaining regions are assumed to be vowel, syllabic nasal or semivowel segments.

7.3.2 Evaluation Criteria

The criteria of the evaluation is as follows:

correct recognition: The correct phoneme is found inside the region of that phoneme in the input speech.

substitution error: The correct phoneme is not found inside the region of the input speech phoneme, instead, another incorrect phoneme is found.

deletion error: No phoneme is found inside the region of the input speech phoneme.

insertion error: The phoneme produced by the system is neither a correct recognition nor a substitution error.

7.3.3 Result

Table 7-1 shows the confusion matrix for all phonemes appearing in the testing data. There are 23 Japanese phonemes. The column **Num** shows the total numbers of each phoneme, **Del.** the number of deletion errors and **Rate** the percentage of phonemes correctly recognized by the system. The row Ins. shows the number of insertion errors. The number of phonemes correctly recognized lies on the diagonal in this confusion matrix. The numbers which lie off the diagonal are the substitution errors. Thus, the sum of "correct recognition", "substitution error" and "deletion error" is equal to the number of the input phonemes. All the results obtained by the system have certainty factors (CF) over a certain threshold ($CF \geq 0.2$).

Overall, the phoneme recognition experiment produced a 91.4% [11,612/12,710] recognition rate for all Japanese phonemes. The deletion error rate was 3.6%, the substitution error rate 5.0% and the insertion error rate 20.7%.

The phoneme recognition results for /i/,/y/,/w/,/ch/ and /g/ were not sufficient. The main reason is that the vowel detection by TDNN-spotting was not sufficient for the phonemes /i/,/y/,/w/. Moreover, when the phoneme /i/ is uttered between an unvoiced phoneme or the phoneme /z/, the duration becomes very short. Thus, the phoneme /i/ has many deletion errors. This is also true in the case of the phoneme /u/. The reason for errors in recognizing the phoneme /ch/ is substitution errors with the phoneme /sh/. The acoustic features of the phoneme /ch/ and /sh/ are very similar, particularly in utterance initial positions. For the phoneme /g/, the reason for the errors was that the segmentation knowledge was not well-tuned. Thus, the current system cannot determine the phoneme boundary of the phoneme /g/ with sufficient accuracy.

Most of the insertion errors were vowels, semivowels, unvoiced-stops and the phonemes /g/ and /r/. The main cause of the insertion errors were the vowels. Many short vowels, but whose duration exceeded 30ms, appeared at the transitional part of the voiced regions. Insertion errors caused by the unvoiced-stops /p/,/t/,/k/, mainly appeared in the utterance initial vowel position. These appeared because the acoustic features at the burst point of the initial vowel in the utterance is very similar to the features of the unvoiced-stop, and of the buzz-bar-less voiced-stop. Most of the insertion errors caused by the phonemes /g/ and /r/ occurred in the last vowel in the utterance, where the acoustic features, such as spectrum and power, are not stable. These insertion errors appeared because the rules to detect these boundaries use very precise spectral features and power changes.

Some of these errors, especially those which occurred due to the insufficient rules, for instance phoneme /g/ deletions, insertions and deletions of short vowels, can be improved by adding more knowledge. Although errors occurring by TDNN mis-identification are fatal now, TDNN, itself, has to be improved.

7.4 Conclusion

The proposed phoneme recognition expert system, which is realized by a close combination mechanism of human knowledge and neural networks, is evaluated using the ATR isolated word database. The experimental result showed that high phoneme recognition performance can be achieved without any language model, using this approach.

A phoneme recognition experiment showed a 91.4% recognition rate for all Japanese phonemes. The deletion error rate was 3.6%, the substitution error rate 5.0% and the insertion error rate 20.7%. This phoneme recognition performance was realized without using any language model.

7.5 Figures & Tables

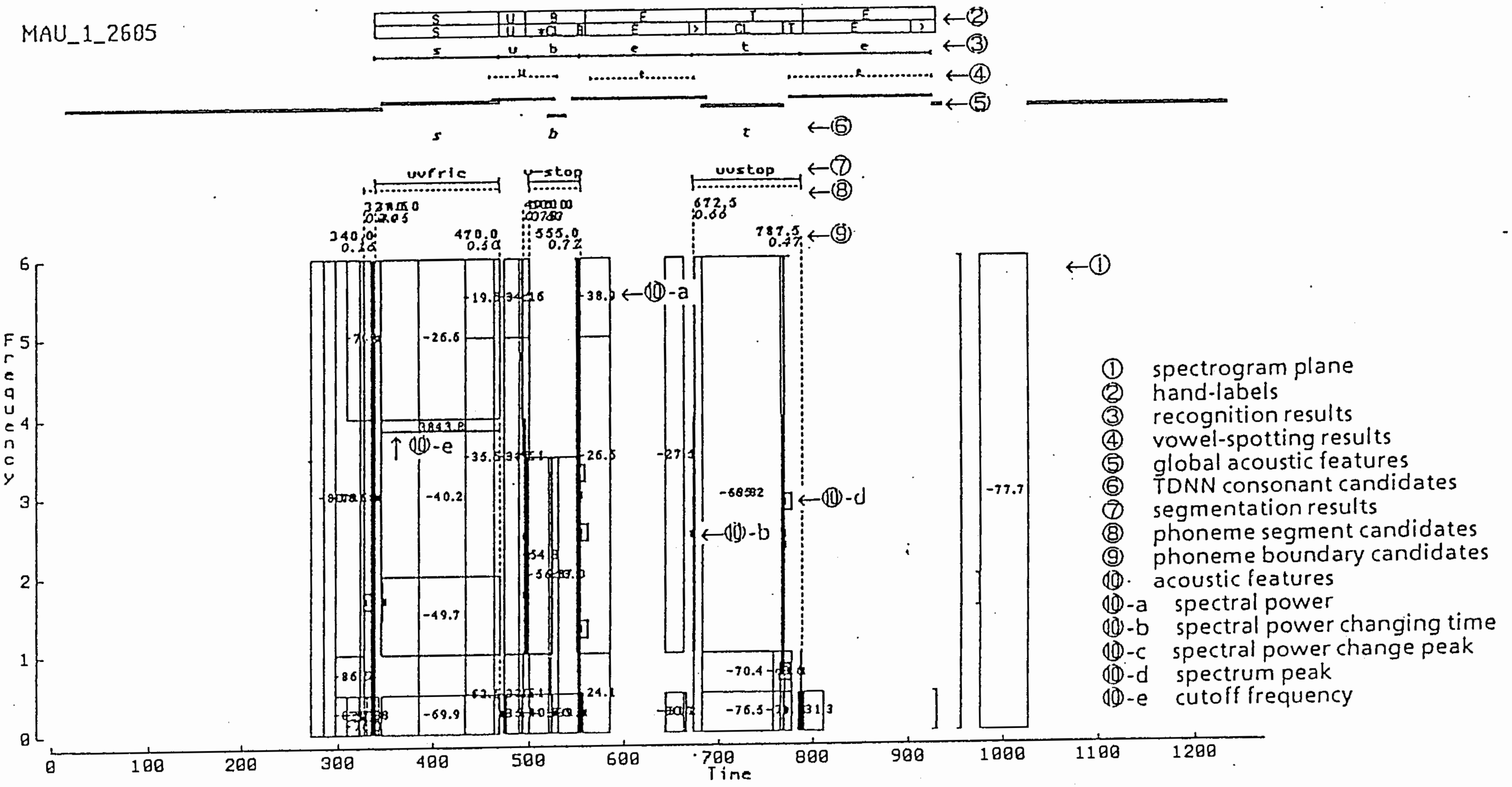


Figure 7-1: Example of Phoneme Recognition (/subete/)

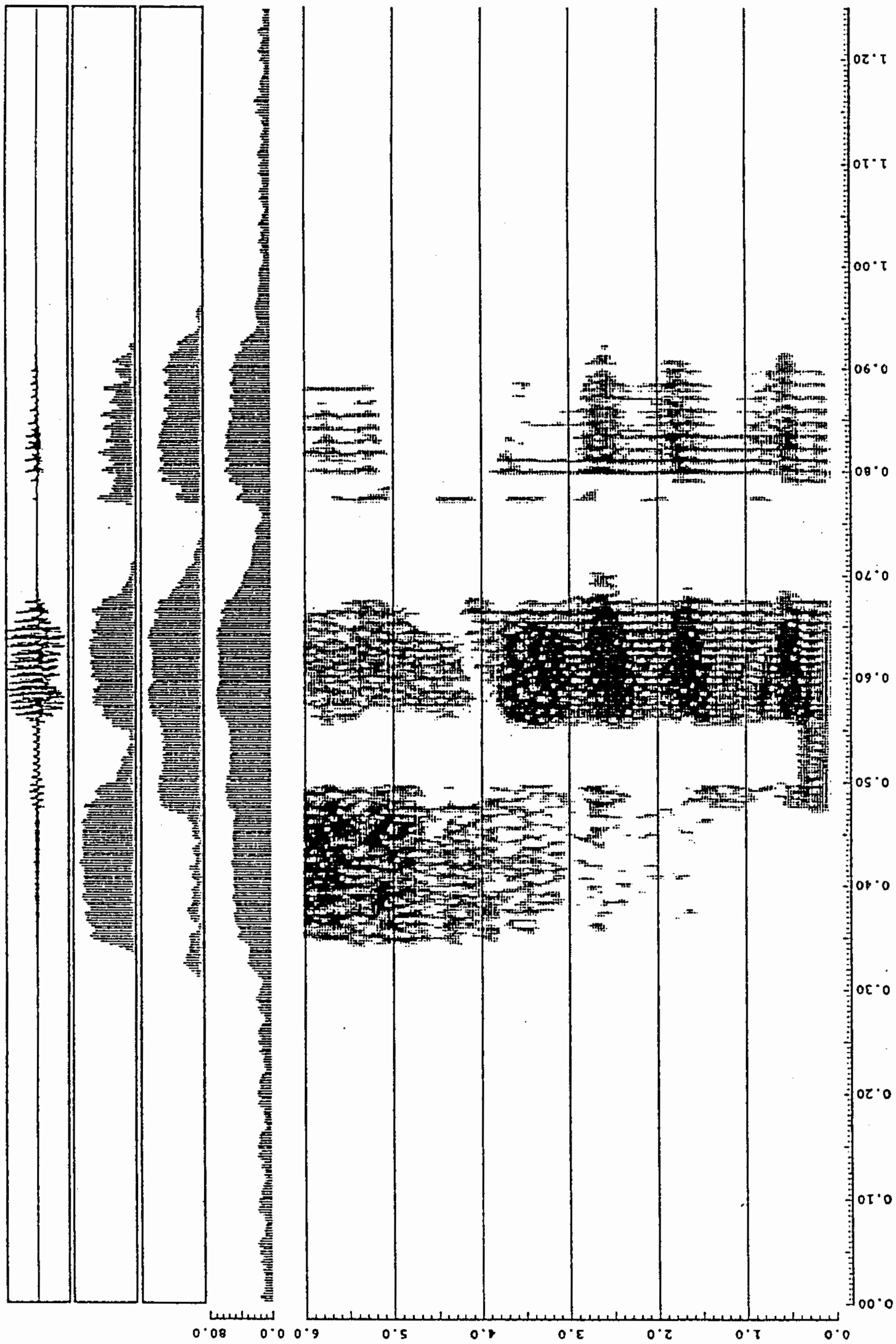


Figure 7-2: Example of Speech Spectrogram (/subete/)

Table 7-1: Confusion Matrix

I/O	a	i	u	e	o	N	y	w	p	t	k	ch	ts	s	sh	h	z	b	d	g	m	n	r	Del.	Rate	Num
a	1747		1		2											1								8	99.3	1759
i	1	1053	9	2		18	22		4	1	3	1		2			11			5		1	3	150	81.9	1286
u		6	1330	1	10	16	11		1	6	6	1	2	1		1	1			6		1		62	91.0	1462
e			2	659	1	1																		14	97.3	677
o	1		5		819	2					1		1	1				1	1			1		22	95.8	855
N		2	11			454			1											1			2	17	93.0	488
y	3	1		1		1	138	1								1				6		6	1	15	79.3	174
w	7				1			57												4	1			8	73.1	78
p									24	1														2	88.9	27
t					2				6	440	7					1								4	95.7	460
k			1		1	1			2	18	1188	4	5	4	9	35								32	91.4	1300
ch							1					117			17		1							5	83.0	141
ts			1							1	2	2	196	9										6	89.9	218
s			3			1	13	2			1		6	522	4	2	1							17	91.3	572
sh		1					6					1	2		373									5	96.1	388
h	5		2				1	5			11				4	274	1			1				12	86.7	316
z		3	8								2	1					287		1		1			12	91.1	315
b			1			1												220			1		2	6	95.2	231
d										2								2	160	2		1	9	4	88.9	180
g		1	8	3	1	1	6	2	1		1							2	1	197	13	1		27	74.3	265
m		1	5		3	1												1		17	425	15	1	16	87.6	485
n		3	2			2													1	7	14	237	1	6	86.8	273
r	1	5	13			1												11	5	10	1	5	695	13	91.4	760
Ins.	21	145	353	166	574	90	633	70	65	50	52	3	7	13	8	49	2	29	3	135	30	30	106	----	20.7	2634

Total: 91.4% [11612/12710]

Chapter 8

ROBUSTNESS OF FEATURE BASED SEGMENTATION

8.1 Introduction

A phoneme recognition expert system by integrating spectrogram reading knowledge and neural networks has been described. In the previous chapters, the effectiveness of a phoneme recognition expert system integrating spectrogram reading knowledge and phoneme identification based on neural networks is shown.

The spectrogram reading knowledge is mainly used for segmentation because phoneme segmentation is not as difficult as identification using a feature based expert system. The neural networks are mainly used for phoneme identification after the segmentation because of the high identification performance on pre-segmented phonemes. Through phoneme recognition experiments, it is shown that the integrated system is one of the most promising ways to recognize continuous speech. However, all these experiments were performed under the condition of speaker dependent isolated word speech. This expert system should be expanded to a speaker independent continuous speech recognition system. As the first step in this expansion, the robustness of this segmentation module to speaker independent speech and continuous speech is tested.

This chapter presents the performance of a feature based phoneme segmentation expert system, tested on speaker independent and continuous speech. The experiments were performed both on isolated word speech uttered by six male speakers and on speaker dependent continuous speech. The additional and modified knowledge for this expansion is also reported by the difference of the rules and fuzzy membership functions.

8.2 Rule Expansion

Details of the feature based consonant segmentation part have already been described in Chapter 4. The system consists of about 250 rules in the current expert speech recognizer and produces the left and right phoneme boundaries and its phoneme classes.

The rule creation and brushing up have been performed in a style shown in Figure 8-1, using an ATR database of 216 phonetically balanced words uttered by one male speaker (MAU). The basic rule creation is performed in the following steps:

(1) Pick up several data having typical spectrogram pattern for creating segmentation rules.

(2) Describe the rule by carefully observing the spectrogram, then testing and modifying iteration is performed until the described rule is correctly segmented.

(3) Test the rule using the hole database, and pick up the unsuccessful data, go to (2).

(4) Modify the rule until it segments the error data sufficiently. If the acoustic features or acoustic environments strongly differ, a new rule must be described, goto (1).

To expand the expert system from speaker dependent to speaker independent and/or from isolated word to continuous speech, the same iteration of rule creation will be applied, as shown in Figure 8-2. The speaker dependent rules are utilized for the initial rules for speaker independent rule training. The process of rule creation and knowledge expansion is exactly the same except the data to be trained.

8.3 Segmentation Experiment

The evaluation of phoneme segmentation is performed using Japanese consonants in the ATR database. The robustness of this segmentation module of the expert system is tested on the ATR 216 phonetically balanced word speech database uttered by six male speakers. The robustness to continuous speech is tested on the ATR short and long Japanese phrase continuous speech database uttered by one male speaker (MAU).

8.3.1 Experimental Condition

The task given to the system is to find consonants in the utterances and to determine their phoneme boundaries, both start and end points. The knowledge for consonant segmentation (about 250 rules), used in the current system, is basically created from the ATR 216 phonetically balanced isolated word speech database uttered by one male speaker (MAU). The phoneme segmentation rules have been enhanced so that the success rate is over 96% using the same MAU training data. When the testing is performed on six male speakers, some knowledge is added and/or modified using another male speaker (MNM). Details of acoustic analysis and evaluation criteria are exactly the same which are described in section 4.4.2 and 4.4.3, respectively.

8.3.2 Result

Table 8-1 shows the consonant segmentation experiment results on a) 2,620 isolated words uttered by one male speaker, b) Japanese short and long phrases uttered by one male speaker and c) 216 isolated words uttered by six male speakers. In the case of the 2,620 isolated words and phrase utterances, knowledge was trained by one male speaker (MAU) on 216 isolated word utterances. In the case of six male speaker utterances, two results are shown:

(1) knowledge trained by one male speaker (MAU).
(2) knowledge trained by two male speakers (MAU and MNM).

The result for 2,620 isolated words is already reported in Chapter 4. In the table, the column **Task** indicates the utterance style. **Data** consists of **Speaker** for speaker information, **Number** for number of data. **Result** consists of **Segmentation**, **Boundary Error** and **Insertion Error**. **Segmentation** indicates the percentage of phonemes whose start and end boundaries were detected within 50ms of the hand-labeled boundaries. **Boundary Error** is the average of the boundary alignment errors compared with hand-labeled boundaries. **Insertion Error** is the ratio of extra segments to the number of phonemes.

Figure 8-3 shows the distribution of the boundary alignment error compared with the hand-labeled boundaries in the database for short and long phrases uttered by one male speaker and 216 isolated words uttered by six male speakers. From these distributions, it can be seen that, in any case, most of the errors lie between −15ms and +15ms.

The overall experimental results are as follows. The average result, which is correctly segmented by the system on the six male speakers, is 91.1% [2,938/3,226] with an average boundary alignment error of 6.2ms. The result on the short phrase utterance was 89.2% [2,374/2,661] and 5.6ms. The result on the long phrase utterance is 87.6% [2,336/2,667] and 5.5ms. These results are as good as, or slightly worse than, the previous experiment result on the speaker dependent 2,620 isolated word speech, which is 93.3% [5,530/5,925] and 5.7ms. These results, especially the boundary alignment errors, are as good as those achieved by human labeling.

8.3.3 Discussion

This section discusses the difference in acoustic features between utterances or between speakers through knowledge which was added and/or modified in this experiment.

Table 8-2 shows the average segmentation results for each phoneme tested on 1) 216 isolated words uttered by six male speakers, 2) phrases (short and long) uttered by one male speaker and 3) on 2,620 isolated words by one male speaker. **Num.** indicates the number of phonemes. **Rate** indicates the ratio of correct segments to the number of phonemes. **Bndry Error** indicates the average of the boundary alignment errors compared with hand-labeled boundaries. **Ins. Error Rate** indicates the rate of extra segments to the number of phonemes.

The tendency of the performance for each phoneme is very similar whether tested on 2,620 isolated words, on phrase utterances or on six male speaker 216 words. Results which were not so good on 2,620 isolated words such as /g/, /z/, /h/ and /s/ worsened, especially in the case of phrase utterances. For the phoneme /g/, the knowledge itself is not enough even for the isolated word utterances in the training data. For the other phonemes /z/, /h/ and /s/, the influence of the fricative increases the high frequency power of the next vowel, which mismatches the inbuilt phoneme contextual knowledge trained by the isolated words. This mismatch reduces the certainty factors of the correct segmentation hypothesis.

From the result of six male speakers trained by one male, the results of three speakers were over 90% and others were about 80% (see Table 8-1). This indicates that three speakers have some different acoustic phonetic features from the training speaker. Moreover, it is very interesting that the performance of the speakers (MHT, MSH, MTK) who were not trained also improves when additional knowledge for another speaker (MNM) was added and/or modified.

Only some knowledge is modified or added to the multiple speaker expansion. Only three kinds of knowledge described as rules out of about 250 are directly added or modified. Moreover, only seven kinds of knowledge described as fuzzy membership functions, out of about 120, are modified. They are shown in Table 8-3. Two examples of the modified fuzzy membership functions are shown in Figures 8-4 and 8-5.

From the number of rules and fuzzy membership functions for expansion, it can be said that most of the additional speaker MNM knowledge is the fuzzy membership functions. And the modification of these fuzzy membership functions were very small as shown in Figures 8-4 and 8-5. These fuzzy membership functions map the acoustic measurements to the certainty factors, which represent the suitability of the measurements in their phonetic contexts, e.g., the power level function for the unvoiced-stop closure, the power level for voiced-stop buzz-bar and the power level for burst, etc. Also, a few rules were added, e.g., the rule of searching for the rough location of the nasal. In this case, the balance of low and high frequency power was different between speaker MAU and MNM. Also, in the case of the unvoiced-stop in speaker MNM, the formant of the previous vowel with the largest power comes into the closure part. This kind of acoustic feature was rarely observed in speaker MAU utterances.

Finally, from these results, it can be said that most of the rough contextual knowledge of phonemes for segmentation can be obtained from one speaker. However, more precise adaptation or modification of knowledge should be done for each speaker to achieve good performance.

8.4 Conclusion

The expansion of a feature based phoneme segmentation module of the expert system toward speaker independent continuous speech were presented. This system utilizes spectrogram reading knowledge and the strategy used by a human expert when reading spectrograms, and determines the phoneme boundary along with the phoneme class. The experiments were performed both on isolated word speech uttered by six speakers and on speaker dependent continuous speech. The results were as good as, or slightly worse than the result tested on speaker dependent isolated word speech.

8.5 Figures & Tables

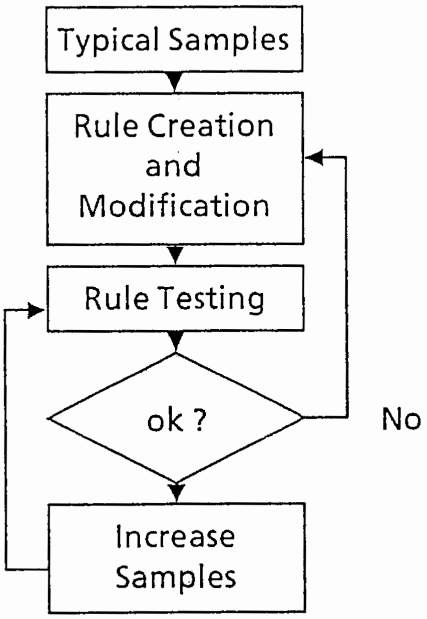


Figure 8-1: Rule Creation and Modification

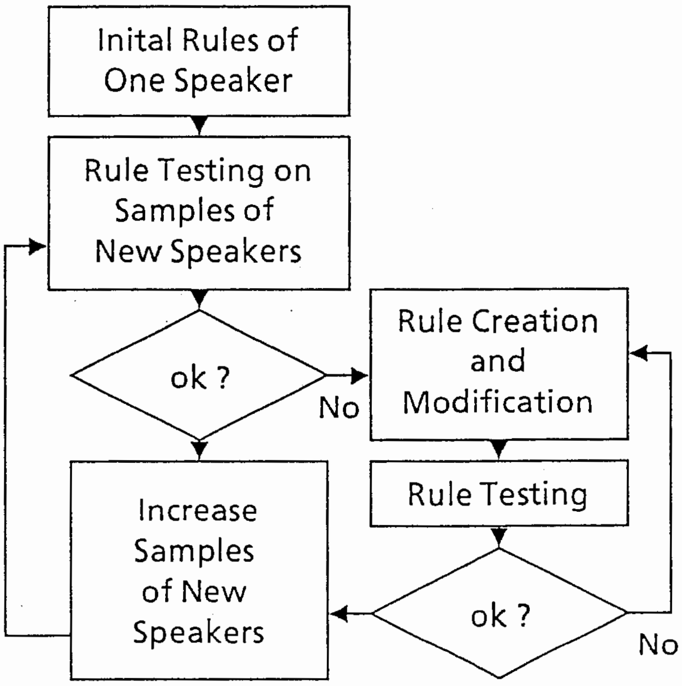

Inital Rules of One Speaker
Rule Testing on Samples of New Speakers
ok ?
No
Rule Creation and Modification
Rule Testing
ok ?
No
Increase Samples of New Speakers

Figure 8-2: Rule Expansion for Multiple Speakers

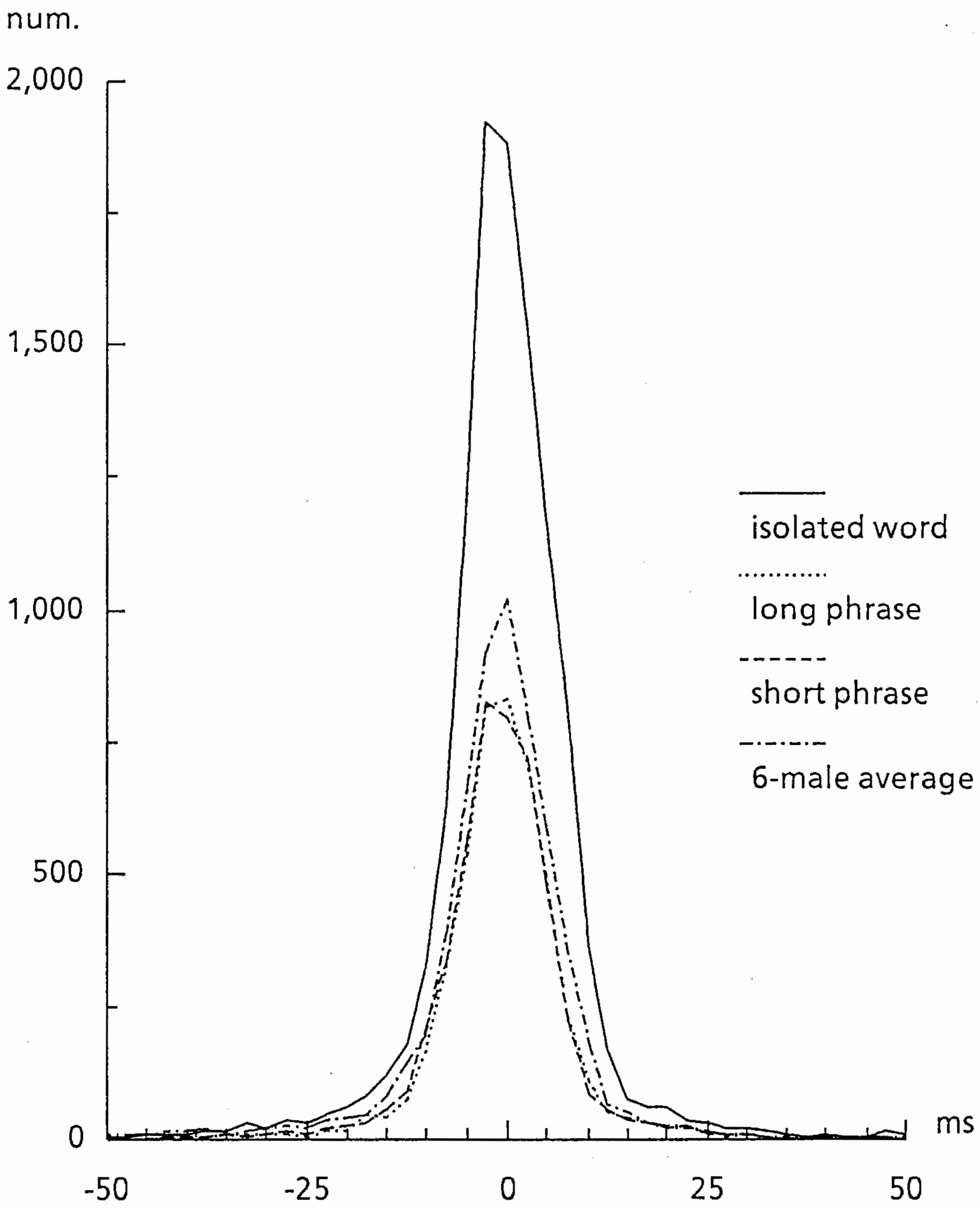
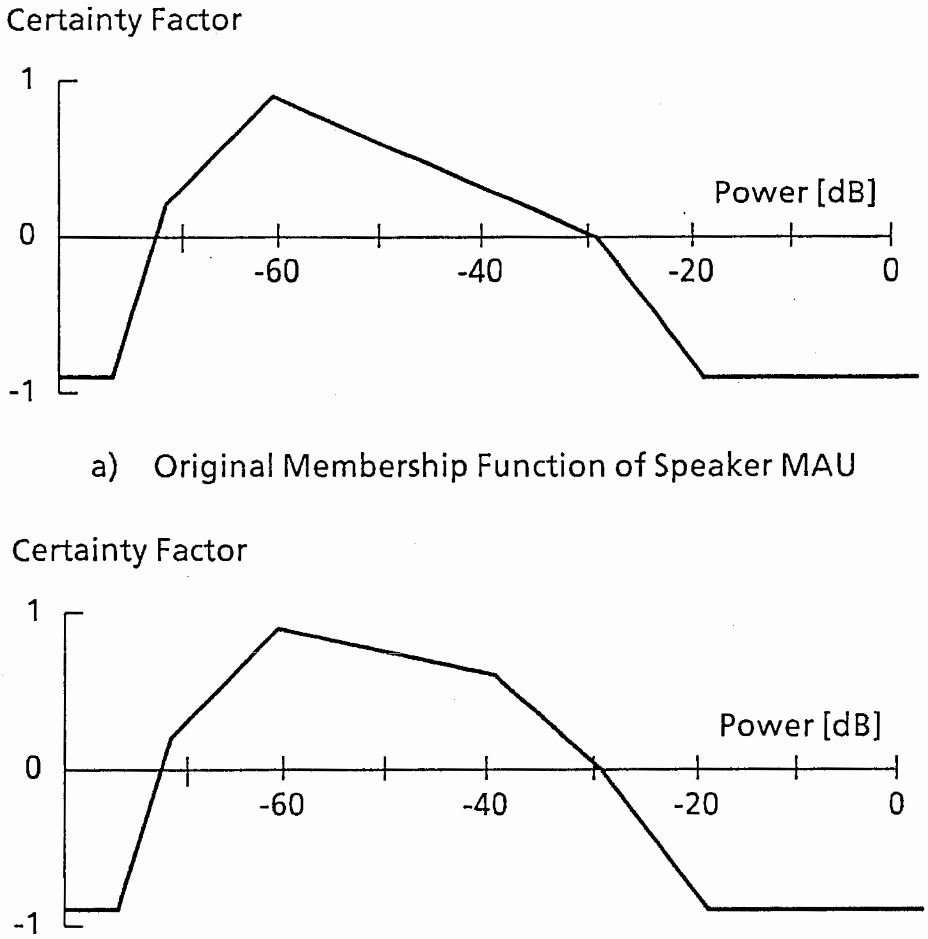


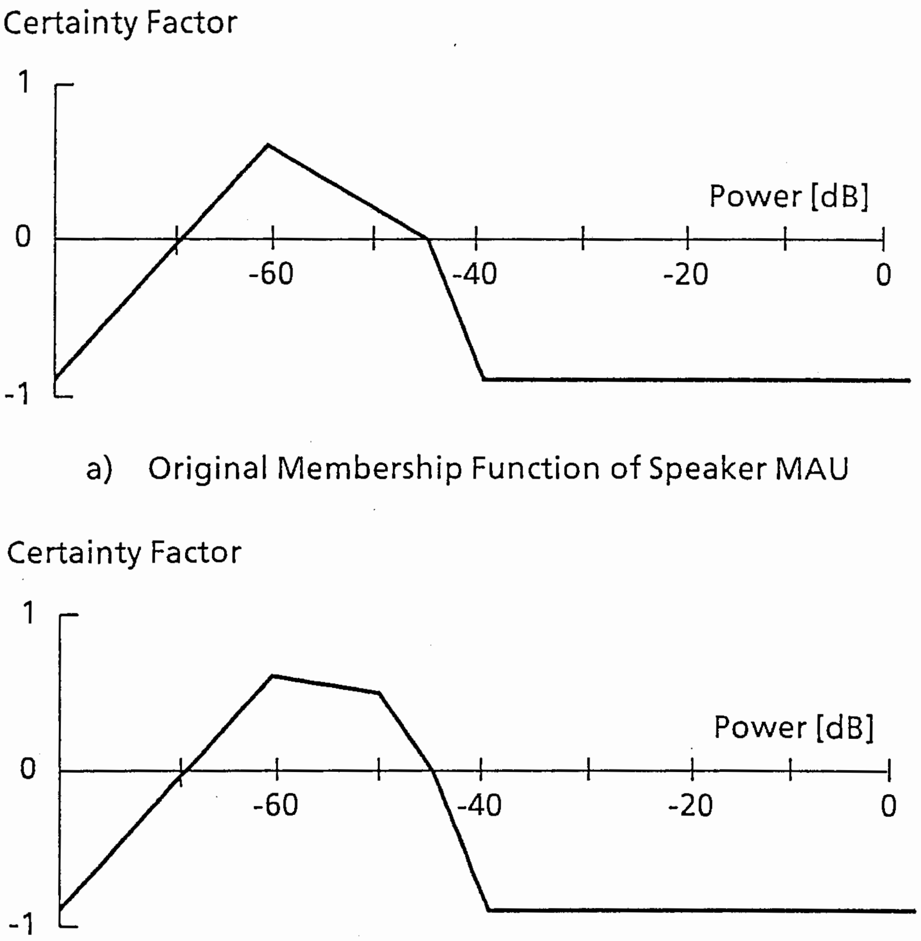
Figure 8-3: Distribution of Boundary Alignment Errors



a) Original Membership Function of Speaker MAU

b) Modified Membership Function of Additional Speakers

Figure 8-4: Membership Function of Voiced-fricative 1000-2000Hz Power



a) Original Membership Function of Speaker MAU

b) Modified Membership Function of Additional Speakers

Figure 8-5: Membership Function of Unvoiced-fricative 0-500Hz Power

Table 8-1: Performance of Phoneme Segmentation

Task	Data		Result		
	Speaker	Number	Segmentation	Boundary Error	Insertion Error
2,620 words	MAU	5925	93.3	5.75	27.8
short phrase	MAU	2661	89.2	5.58	25.7
long phrase	MAU	2667	87.6	5.47	24.5
Isolated 216 Word Using Rules Trained by ***MAU***	***MAU***	546	96.3	5.08	33.3
	MHT	539	90.0	5.88	33.2
	MMY	535	91.8	5.49	31.8
	MNM	542	80.4	7.25	63.3
	MSH	538	80.9	7.10	45.9
	MTK	546	83.2	6.78	42.1
	all	3226	87.1	6.26	41.9
Isolated 216 Word Using Rules Trained by ***MAU*** & ***MNM***	***MAU***	546	94.9	5.03	32.8
	MHT	53.9	92.9	6.00	33.2
	MMY	535	91.6	5.54	30.7
	MNM	542	93.9	7.07	57.9
	MSH	538	82.0	7.16	38.1
	MTK	546	89.6	6.62	41.6
	all	3226	91.1	6.24	39.3

Table 8-2: Phoneme Segmentation on Phrase and Multiple Speaker Speech

	Segmentation on Phrase			Ins. Error Rate		Segmentation on Six Speakers			Ins. Error Rate		Segmentation on 2,620 Words			Ins. Error Rate
	Num.	Rate [%]	Bndry Error [ms]	[%]		Num.	Rate [%]	Bndry Error [ms]	[%]		Num.	Rate [%]	Bndry Error [ms]	[%]
p	53	100.	4.8	12.4	p	132	93.2	5.7	14.2	p	28	96.4	4.2	17.4
t	540	93.9	4.2		t	186	94.1	4.7		t	461	98.0	4.3	
k	1121	91.7	4.7		k	525	93.9	5.7		k	1300	97.8	5.8	
ch	88	81.8	4.4		ch	131	93.1	5.7		ch	141	91.5	5.8	
ts	117	89.7	4.4		ts	48	89.6	7.4		ts	220	93.2	5.6	
s	510	77.3	4.9	1.1	s	189	93.1	4.1	3.9	s	572	88.3	3.5	3.5
sh	343	92.4	4.7		sh	145	93.8	3.9		sh	387	92.0	4.5	
h	169	75.1	8.5	13.6	h	190	83.7	9.6	24.7	h	313	88.8	8.3	7.7
z	149	83.2	7.6	24.2	z	233	79.0	7.9	7.7	z	315	85.4	9.6	11.1
b	95	98.9	3.9	14.0	b	206	96.6	6.5	35.1	b	230	98.3	4.7	15.7
d	472	91.5	4.8		d	124	96.8	4.5		d	177	98.3	3.4	
g	242	69.0	9.5		g	208	82.7	8.7		g	263	83.7	8.9	
m	380	90.8	7.0	72.9	m	263	86.7	7.1	125.3	m	485	95.3	6.0	94.2
n	554	93.3	6.4		n	216	94.0	6.2		n	273	97.8	5.7	
r	497	85.7	6.5	47.9	r	430	93.7	6.3	59.5	r	760	90.7	6.2	47.4
Total	5328	88.4	5.5	25.1	Total	3226	91.1	6.2	39.3	Total	5925	93.3	5.8	27.8

Table 8-3: Expansion Knowledge for Multiple Speaker Speech.

Sort of Knowledge	Knowledge and Expansion	
	No.	Rule and Function Name
Rules (about 250)	1)	Unvoiced-fricative after vowel (2) → Threshold
	2)	Unvoiced-stop → Searching point of burst
	3)	Nasal-peak candidates → Add 0-500/3000-4000Hz
Membership Functions (about 120)	1)	Voiced-stop burst-1000-6000Hz start-change
	2)	Voiced-frictive 1000-2000Hz power
	3)	Unvoiced-stop closure-0-6000Hz power
	4)	Unvoiced-stop 0-500Hz power before-vowel
	5)	Unvoiced-stop 0-500/500-1000Hz power-ratio before vowel
	6)	Unvoiced-fricative 0-500Hz power
	7)	Unvoiced-fricative 0-200Hz power at word initial

Chapter 9

TIME-STATE NEURAL NETWORKS

9.1 Introduction

In order to expand the proposed system to continuous speech recognition, it is necessary to improve phoneme identification performance of neural networks for continuous speech. There are two points for improving the neural network performance: 1) neural network structure, 2) neural network training. This chapter focuses on the structure of phoneme classification-type neural networks to improve the phoneme identification performance against continuous speech.

Phonemes in Japanese have certain rough temporal structures of phonemic features. With phoneme /b/ in the medial part of the utterance, for example, first a transition from the previous vowel is observed, next a buzz-bar, then a /b/ burst, and finally transition to the next vowel. Each of these features contains information which contributes to identifying the phoneme. Moreover, this kind of rough temporal manner does not greatly change even if the utterance is an isolated word or continuous speech. Thus, if the neural network is to treat this kind of temporal manner, it would be very helpful in order to identify phonemes, whatever the utterance style.

Since the back-propagation algorithm was developed, many neural network applications to speech recognition have been proposed. However, there are few neural networks whose structure considered the temporal structure of phonemic features. Some neural network approaches, such as the Neural Prediction Models (NPM) [Iso 90], Dynamic Neural Networks (DNN) [Sakoe 89] and Time-Delay Neural Networks (TDNN) [Waibel 89], attempt to deal with this problem. NMP is able to deal with the time warping of speech features even though it is classified as a prediction-type neural network. DNN and TDNN are classified as classification-type neural networks. Although DNN considers the temporal structure, it is proposed for word recognition. Moreover, the time-shift tolerance capability is unknown. This time-shift tolerance capability is very significant when combined with the segmentation part in the proposed expert system. TDNN has a time-shift tolerance capability, but on the other hand, its structure forces it to suppress the temporal structure of the phonemic feature.

In this chapter, several new structures for phoneme identification neural networks, Time-State Neural Networks (TSNN) which are able to deal with the temporal structure of phonemic features, are proposed. Phoneme identification performance of the proposed TSNN on Japanese phonemes /b,d,g,m,n,N/ compared with that of a conventional TDNN is also described.

9.2 Time-State Neural Networks

In this section, the structures of classification-type neural networks, TDNN, and several types of TSNNs, are described.

9.2.1 Time-Delay Neural Networks

Time-Delay Neural Networks (TDNN) can easily be trained using the back-propagation training algorithm. Moreover, it is shown to be a very high performance phoneme classifier. The main advantage of TDNN is the time-shift tolerance capability derived from its time-shifted and tied-connected weight architecture. This is an important property in combination with phoneme segmentation, because slight errors always occur in phoneme segmentation. It is also very important when the TDNN is used for phoneme-spotting in speech.

Figure 9-1 shows the TDNN architecture for 6-phoneme /b,d,g,m,n,N/ identification. This TDNN is made up of four layers. The lowest layer corresponds to spectral input values, the two next layers are hidden layers and the topmost layer, which is the output layer, corresponds to each phoneme output. The input layer has 15 frames (150ms) × 16 spectral coefficient units. The window structure of the connections between the layers is time-shifted and tied-connected. The connection in the time-shifted window from input layer to hidden layer 1 is 3 frames to 1 frame, and from hidden layer 1 to hidden layer 2 is 5 frames to 1 frame. The tied-connection to the output layer has the same weight for each unit. All weights are adjusted using the back-propagation training procedure. The phoneme corresponding to the highest activated output unit is defined as the classification result.

9.2.2 Simple Time-State Neural Networks

Figure 9-2 shows the simplest TSNN, which is a 6-phoneme classifier. Each output directly concerns each phoneme. This four layer neural network has four states in the time sequence and is able to capture the phonemic features in each state. The first state is considered to capture the transition feature from vowel into consonant, the second to capture the buzz-bar or stable part of the nasal, the third to capture the burst or the nasal transition to the next vowel and the last to capture the next vowel. The connections from the input layer to hidden layer 1 are time-shifted windows (3 frames to 1 frame) and are tied-connected in the manner of the TDNN connection. The time-shifted tied-connected windows are also shifted over the input speech but differ from the TDNN in the point of its shifting range. In this TSNN, each window is shifted between the indicated size as shown by ⟷ in the diagrams, (6 frames). This structure allows the TSNN to capture the phonemic features at any point within the windows and considers the temporal structure of the phoneme features.

9.2.3 All Tied-Connection TSNN

The simple TSNN described above, has a time-shifted and tied-connected window only between the input layer and the hidden layer 1. Thus, the capability of the time-shift tolerance will not be achieved by this neural network architecture sufficiently. Here, another TSNN, which has time-shifted and tied-connected windows between every layer, is proposed. This TSNN, shown in Figure 9-3, has three states in the time sequence and is also able to capture the phoneme features in each state. The first state is considered to capture the transition feature from vowel to consonant, the second to capture the buzz-bar or stable part of the nasal, the third to capture the burst with next vowel or the nasal transition with the next vowel. This TSNN has time-shifted windows from the input layer to each state of hidden layer 1 and from each hidden layer 1 to each hidden layer 2. Moreover, the connections are tied-connected between every layer. The windows are only shifted over each layer between the indicated window size as shown by ⟷ in the diagrams, (7 frames in the input layer, 3 frames in the hidden layer 1). The connection to the output units is separated into three weights. This architecture may improve the time-shift tolerance capability over than that of the simple TSNN.

9.2.4 Compressed TSNN

In Figure 9-4 a compressed-type TSNN is shown. Basically, the structure of this compressed TSNN is exactly the same as that of the conventional TDNN shown in Figure 9-1, except for the weights connected to the output layer. In the conventional TDNN, all 9 weights are tied-connected, which means that all connections have the same weights. In this TSNN, the connections are also tied-connected, but they are separated into three weights: the front three, the middle three and the back three.

This weight separation may separate the hidden layers into three states and may capture the phonemic features in each state. The first three weights are considered to capture the transition feature from vowel to consonant, the second three to capture the buzz-bar or stable part of the nasal, the third three to capture the burst and next vowel or the nasal transition to the next vowel. Thus, this TSNN can be observed as the compressed-type TSNN shown in Figure 9-3. The time states in the hidden layers, which are considered to capture the temporal phonemic features, are compressed into one hidden layer. However, the weight separation in the output layer may represent the time states for the temporal phonemic features in the units of the hidden layers through the back-propagation training algorithm.

9.3 Experiment Using TDNN and TSNNs

Japanese phoneme /b,d,g,m,n,N/ identification experiments are performed using TDNN and several types of TSNNs which are proposed.

9.3.1 Experimental Condition

The neural networks were trained on half (even numbered words) of the ATR 5,240 isolated word database, recorded by one male speaker. For testing, various styles of utterance are used such as the other half (odd numbered words) of the 5,240 isolated words, short and long phrase utterances and continuous utterances in the ATR speech database.

Two types of data taken from the ATR database are used:

(a) 150ms fixed samples.
(b) Samples linearly normalized by each phoneme duration.

This linearly normalized data, which aligns its phoneme temporal structure, is used to show the effectiveness of considering the temporal structure of phonemic features. Also, in the isolated word utterance, data shifted from −20ms to +20ms in 10ms steps, are used to evaluate the time-shift tolerance capability of each neural network.

For input to the neural networks, the speech was sampled at 12kHz and analyzed by FFT using a 21.3ms Hamming window every 5ms. 16 mel-scaled coefficients were computed and merged for a 10ms frame rate, and normalized to fall between −1.0 and +1.0 with the average at 0.0.

The neural networks were trained using the fast back-propagation training method "Dynet" [Haffner 89]. In this process, every end point of the hand-segmented datum is aligned at the center frame of the input layer. Similarly, in the process of classification, the end point of the datum is also adjusted at the center frame of the input layer.

9.3.2 Result

Table 9-1 shows the experimental results using TDNN, simple TSNN, all tied-connection TSNN and compressed TSNN. The number in the table indicates the percentage of phonemes correctly identified in each experimental condition.

(a) From the result, the conventional **TDNN** is capable of a time-shift tolerance of about 30ms, or a little more. However, the recognition rate is drastically reduced when the utterance changes from isolated word speech to phrase and sentence speech. This is because the TDNN does not have enough flexibility to capture the temporal structure of the acoustic feature. This can also be confirmed from the results on the linearly normalized data of the phoneme duration.

(b) The **Simple TSNN** recognition rate improved drastically compared with that of the TDNN, especially as regards phrase and sentence utterances. This improvement is more evident for linearly normalized data. However, no time-shift tolerance capability is obtained in this simple TSNN.

(c) The result of **Shift Training TSNN**, which is the simple TSNN trained using shift data, shows that the time-shift tolerance capability can be obtained by training using the shifted data. However, in this case, three times more training data is necessary to train the neural networks, which also means that the training cost is three times more than that of a conventional TDNN.

(d) The phoneme identification performance of the **TSNN All-Tied**, which indicates the results of TSNN with tied-connection in the hidden layer, and **Comp. TSNN**, which indicates the results of compressed TSNN, lies between that of the **Shift Training TSNN** and **Simple TSNN**. This result shows that the time-shifted and tied connection is not as good as that of the TDNN. However, the capability was slightly improved over that of the **Simple TSNN**.

Additionally, the recognition rate was better than that of the conventional TDNN for various utterances. This result indicates that the time-shifted and tied-connected weights in the conventional TDNN is so strong that it suppresses the ability to capture the temporal structure of the acoustic phonemic features.

Finally, from these results, it can be said that to incorporate some kind of temporal structure into neural networks is necessary for improved identification performance. Moreover, an additional experiment shows that the TSNN can obtain time-shift tolerance capability by making time-shifted and tied-connected weights in the hidden layers and/or by using shifted data for training.

9.4 Conclusion

This chapter proposed a new structure of phoneme identification neural networks, Time-State Neural Networks (TSNN). TSNNs are able to deal with the temporal structure of phonemic features, which does not greatly change according to utterances such as isolated word or continuous speech. Thus, TSNN is well able to identify phonemes, whatever the utterance style. Some types of TSNNs are tested on Japanese phonemes /b,d,g,m,n,N/. Their phoneme identification performance was much better than that of the conventional TDNN, especially on continuous speech.

9.5 Figures & Tables

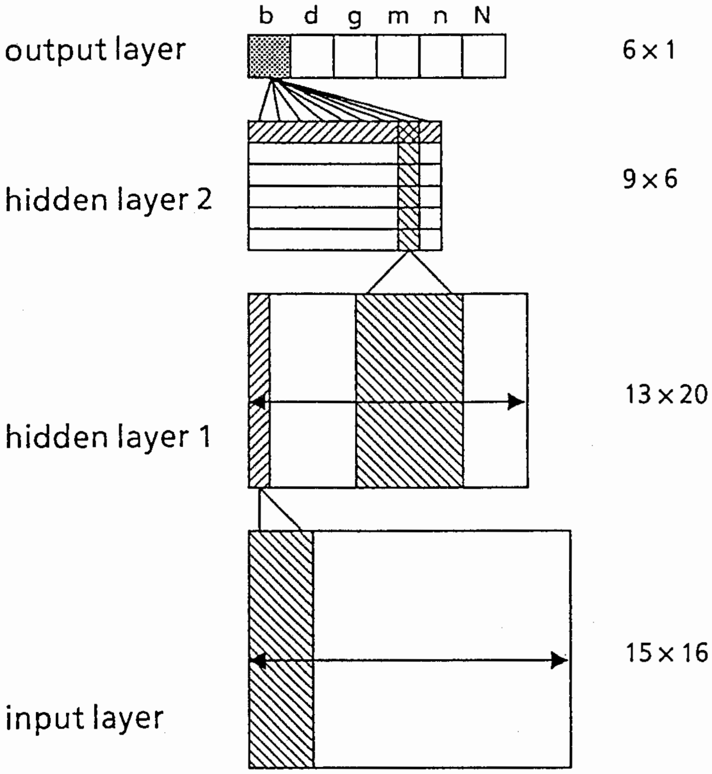


Figure 9-1: Time-Delay Neural Network

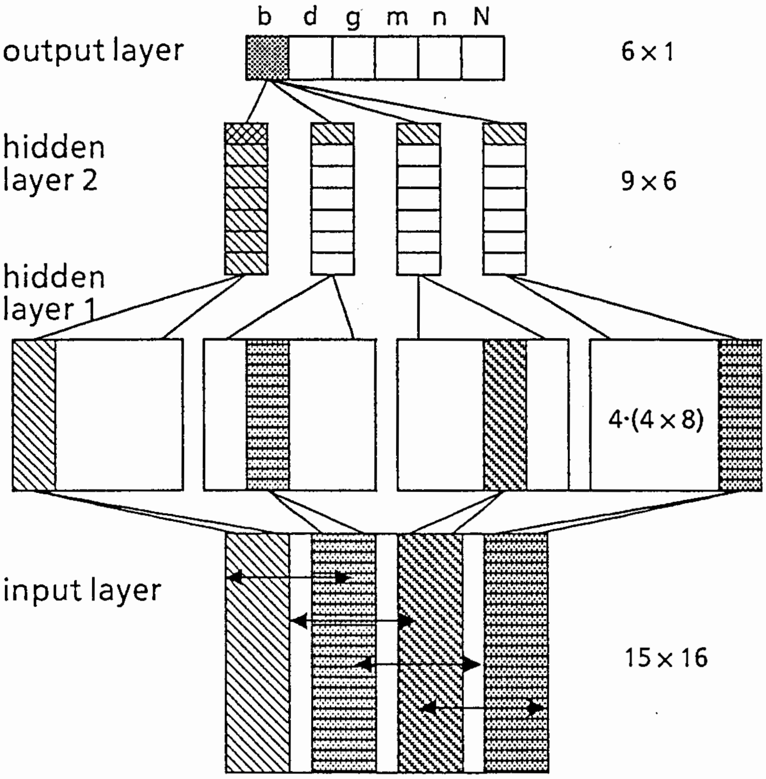


Figure 9-2: Simple Time-State Neural Network

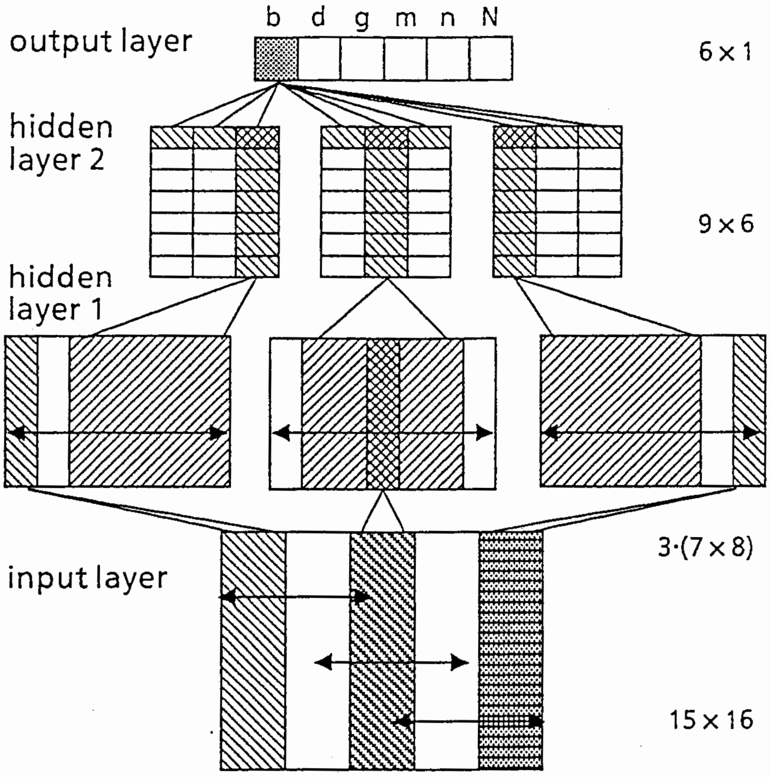


Figure 9-3: All Tied-Connection TSNN

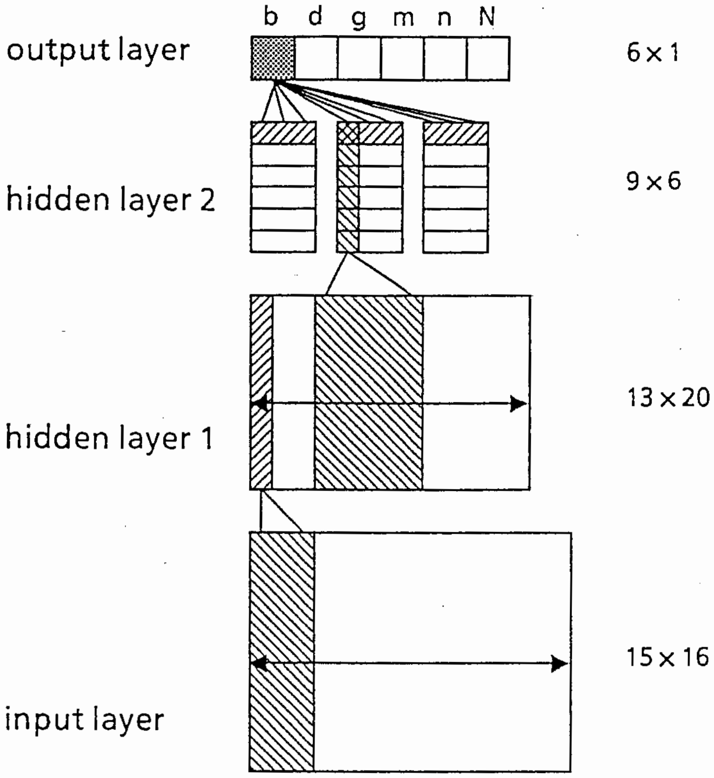


Figure 9-4: Compressed TSNN

Table 9-1: Performace of Time-State Neural Networks.

Utterance Style	Testing	150ms Fixed Samples					Samples Normalized by Duration	
	Neural Networks	TDNN	Simple TSNN	Shift Training TSNN	All-Tied TSNN	Comp. TSNN	TDNN	Simple TSNN
Isolated Word	-20ms	91.2	52.7	81.6	58.7	54.4	-	-
	-10ms	94.7	86.7	**95.7**	92.6	93.0	-	-
	0ms	**95.7**	**97.6**	**97.0**	**97.0**	**97.6**	**95.6**	**98.0**
	+10ms	94.1	84.4	**95.0**	89.9	90.3	-	-
	+20ms	85.9	56.5	77.5	65.8	58.6	-	-
Continuous	Short Phrase	76.6	82.7	79.9	79.0	80.7	74.6	83.8
	Long Phrase	75.9	77.6	77.6	77.7	77.7	74.8	81.6
	Sentence	61.8	70.7	71.1	69.0	71.7	58.5	72.0

Chapter 10

NEURAL FUZZY TRAINING

10.1 Introduction

In order to expand the proposed expert system to continuous speech recognition, it is necessary to improve phoneme identification performance of neural networks for continuous speech. There are two points for improving the neural network performance: 1) neural network structure, 2) neural network training. This chapter focuses on the training method of phoneme classification-type neural networks to improve the phoneme identification performance against continuous speech.

Recently, in the research field of speech recognition, it has become possible to deal with a large amount of data because of the incredible improvement of the computer. Moreover, methods which use a lot of data such as statistical models like HMM and neural networks, become one of the main resources in studying speech recognition.

Among these methods, since the back-propagation algorithm, a powerful neural network training algorithm [Rumelhart 86] [Lippmann 87], was developed, many applications for speech recognition have been proposed using feed-forward identification-type neural networks. Time-Delay Neural Networks (TDNN) [Waibel 89] showed good phoneme identification performance. The TDNN is presented as a good speech recognition neural model not only for its performance but also for its time-shift tolerance capability.

Through the continuing study of neural speech recognizers, a generalization problem has arisen, especially in the phoneme identification-type feed-forward neural networks trained with the conventional back-propagation algorithm such as TDNN. In other words, the robustness of the neural networks trained by the conventional back-propagation algorithm are not as adequate as expected. The generalization problem is essentially an over-learning of the training data which causes a drastic performance reduction when a slight difference arises in the testing data, (e.g. speaking rate differences). This problem arose because the conventional training method creates very sharp boundaries between classes in the neural networks.

Another problem also arises when combining the phoneme identification-type neural networks with a language model, in which the top-N candidate performance is not required. This problem derives from simply giving the phoneme class information of the training sample, 1 to the phoneme class which the sample belongs and 0s to the other phoneme classes, to the target values of the neural network in the conventional method. Thus, the neural network is trained only to produce the top phoneme candidate but not the top-N candidates. In other words, the neural networks are not trained to produce the likelihood for each phoneme class. As a result, the output values of the neural networks for the 2nd, 3rd, and top-N candidates are suppressed at almost zero, which reduces the top-N recognition performance. However, this top-N phoneme candidate information is very important when combined with a language model for continuous speech recognition. Once the lack of the phoneme candidate information occurs, it may lead to a fatal error in continuous speech recognition.

There are several approaches to overcoming these problems for phoneme identification-type neural networks. The most famous is to avoid over-learning the training data by stopping the training iteration using an additional cross validation data set. There is another method for creating robust neural networks by adding some noise to the training data. Minami proposed a method to improve the top-N candidates by smoothing the values of the output or the hidden layer units in the neural network [Minami 90]. Kawabata proposed the "KNIT" training method which avoids over-learning of the training data by imposing the constraints between the input data and target values using a K-nearest neighbor interpolation training [Kawabata 90]. Also, Takami has proposed a pairwise discriminant approach to improve the robustness by using multiple neural networks [Takami 90].

In this paper, a new fuzzy training method for phoneme identification neural networks, quite different from the aforementioned approaches, called "Neural Fuzzy Training", is proposed. The difference between the proposed and the conventional method is that the target values of the training datum are given as fuzzy phoneme class information instead of discrete phoneme class information. By giving the fuzzy phoneme class information instead of the discrete phoneme class information, it is expected that the top-N candidate performance of phoneme identification neural networks will improve and more robust neural networks will be created by overcoming the over-learning problem.

The basic idea of the proposed Neural Fuzzy Training method is described in the next section. Then, the phoneme identification experiments using /b,d,g,m,n,N/ identification task, 18-consonant identification task is shown using the ATR isolated word database, phrase database and sentence database. Finally, continuous speech recognition experiments by means of TDNN-LR speech recognizer using the ATR isolated word database and phrase database are presented. From the experimental result, the effectiveness of the proposed Neural Fuzzy Training method on training speed, on generalization and on untrained speakers are discussed.

10.2 Neural Fuzzy Training

Neural Fuzzy Training is realized using the back-propagation algorithm, but differs how the target values are given to the neural network. In the conventional method, target values are given as discrete phoneme class information. In the proposed method, the target values are given as fuzzy phoneme class information between 0 and 1, which inform the phoneme class likelihood of the input sample to the neural network.

The conventional method is realized by the use of the back-propagation algorithm, whose target values are given as discrete phoneme class information, i.e. 1 to the phoneme class which the sample belongs and 0s to the other phoneme classes.

On other other hand, the proposed Neural Fuzzy Training method is also realized by the use of the back-propagation algorithm, either. However, the target values are given as fuzzy phoneme class information whose values are given as between 0 and 1. The fuzzy class information informs the neural network likelihood of the training sample to each phoneme class, in other words the possibility of belonging to each phoneme class. The reliability of belonging to the phoneme classes can be considered using the idea of distance between training samples, for instance Euclidean distance measure. Here, there is an assumption that "when the distance of two samples is small, these two samples are considered to be similar." This leads to each sample having the possibility of belonging to the class of the other sample. On the contrary, "when the distance of two samples is large, these two samples are considered to be very different." This leads to each sample having less (or no) possibility of belonging to the class of the other sample. To model this likelihood using the distance d, a likelihood transformation function $f(d)$ is adopted. By the use of monotonous decreasing function such as $f(d) = exp(-\alpha \cdot d^2)$ where $\alpha \geq 0$, as shown in Figure 10-1, it can easily model the idea that "the larger the distance is the lower the possibility is and the smaller the distance is the larger the possibility is." Thus, fuzzy phoneme class information can be computed according to the distance between the input sample and the nearest sample of each phoneme class in the training data set.

Figure 10-2 gives a brief idea of the conventional training method (CT) and the proposed Neural Fuzzy Training method (NFT). The target values of the conventional method are given as discrete phoneme class information, i.e. the target values of sample B (●) is given as $\{0, 1, 0\}$. The target values of the Neural Fuzzy Training method are given as fuzzy phoneme class information, i.e. the target values of sample B (●) is given as $\{f(d_{AB}), f(d_{BB}), f(d_{CB})\}$ where $f(d)$ is a likelihood transformation function of a distance d.

Considering the computational cost for creating the target values for every training sample, if this method is adopted in a straightforward manner, distance calculation $\{N \cdot (N-1)\}/2$ where N is the number of training samples, is required because the nearest samples belonging to each class of the training sample have to be selected. This is very expensive if the training set is very large.

To avoid this problem, pre-selection of training samples for likelihood calculation is possible. The computational cost reduces from $N \cdot (N-1)/2$ to $C \cdot M \cdot N$ when $C \cdot M$ is much smaller than $N/2$, where N is the number of training samples, C is the phoneme class number and M is the pre-selected sample number.

In the following section, experimental results are described showing the effectiveness of the proposed Neural Fuzzy Training method compared with the conventional training method.

10.3 Phoneme Identification Experiment

In this section, /b,d,g,m,n,N/ and 18-consonant identification experiments using the ATR database [Takeda 88] are discussed to show the effectiveness of the proposed Neural Fuzzy Training method for phoneme identification. The Japanese 18 consonants are /b/,/d/,/g/, /p/,/t/,/k/, /ch/,/ts/, /s/,/sh/,/h/,/z/, /m/,/n/,/N/, /r/,/w/ and /y/.

10.3.1 Experimental Condition

Phoneme samples for neural network training are culled using the hand-labels from half of the ATR isolated word database (even numbered words; 5.7 mora/s). In the /b,d,g,m,n,N/ identification task, 1,857 training samples are selected up to 500 samples for each phoneme class. In the 18 consonant identification task, 3,638 training samples are selected up to 250 samples for each phoneme class.

For input to the neural networks, the speech was sampled at 12kHz and analyzed by FFT using a 21.3ms Hamming window every 5ms. 16 mel-scaled coefficients were computed and merged for a 10ms frame rate, and normalized to fall between -1.0 and $+1.0$ with the average at 0.0.

Phoneme samples for neural network testing are also culled using the hand-labels from the other half of the ATR isolated word database (odd numbered words; 5.7 mora/s). Additionally, to evaluate the robustness to the speaking rate, testing samples are also culled from the phrase database (7.1 mora/s) and from the sentence database (9.6 mora/s).

TDNNs, as shown in Figures 10-3 and 10-4, are adopted for the /b,d,g,m,n,N/ identification experiment and for the 18-consonant identification experiment, respectively. The structures of TDNNs are feed-forward neural networks of four layers. The input for the TDNNs is 16 mel-scaled spectral power of 7 frames (70ms). All the end points of the phoneme labels for training and testing samples are adjusted so as to be at the center of the input layer. The neural networks are trained using the fast back-propagation training method "Dynet" [Haffner 89].

Two conventional training methods and the proposed Neural Fuzzy Training method are compared. The two conventional methods differ in the point of the error function of the back-propagation algorithm. They are 1) mean square error (M.S.E.) function and 2) McClelland error function $ln(1 - \varepsilon^2)$. M.S.E. function is adopted for the Neural Fuzzy Training method. The McClelland error function, which back-propagates emphasized errors, is well-known as a very fast training method when the number of classes to be identified is very large.

The Euclidean distance measure d of the 7-frame input samples is adopted. To model the likelihood using the distance d, $f(d) = exp(-\alpha \cdot d^2)$ where $\alpha = 0.005$, is adopted as a likelihood transformation function. The value $\alpha = 0.005$ is chosen by experience in order that the target values of the 2nd and the 3rd candidates may have certain values.

The weight which has the highest performance on the testing data culled from the isolated word database is chosen for experiments from 100 training iterations.

10.3.2 Result

The phoneme identification results of /b,d,g,m,n,N/ identification and those of the 18-consonant identification are shown in Figures 10-5 and 10-6, respectively. They show the identification performance of the first candidate and the top-N candidates on the training data and the testing data (phonemes culled from isolated word, phrase, sentence database. The vertical axis indicates the identification rate (%) and the horizontal axis the top-N candidates.

Comparing the two conventional and the Neural Fuzzy Training methods, there were no big differences in identification performance on phonemes cut out from the isolated word database in both the training and testing tasks. However, the identification performance trained by the proposed Neural Fuzzy Training method improved on the phoneme culled phrase (7.1 mora/s) and sentence database (9.6 mora/s) in which the speaking rate differs from the training data (5.7 mora/s). The figures indicate that not only the first candidate result but also the top-N results improved. Especially on the sentence data, the top-N results improved drastically.

For continuous speech recognition, the top-N phoneme candidate information is very important when combined with a language model. Thus, from the improvement shown in these figures, it can be expected that the proposed Neural Fuzzy Training method will improve the overall recognition performance on phrase and/or sentence speech in combination with a language model.

Comparing the two conventional methods of the top-5 candidate performance on 18-consonant identification task, the result trained by the McClelland error function was worse than that trained by the M.S.E. function. This performance reduction derives from the characteristics of the McClelland error function. The output values are forced to be almost 0 or 1 for the strongly emphasized error back-propagation training. As a result, the output values of the neural networks for the top-N candidates are strongly suppressed at almost zero which reduces the top-N identification performance.

From this point of view, the conventional training method with McClelland error function will not perform sufficiently on continuous speech recognition which performs in combination with a language model, even if the training method is well-known as a fast training method.

These experiments indicate the effectiveness of the proposed Neural Fuzzy Training method compared with conventional methods. However, there is a problem in the proposed Neural Fuzzy Training method. A very high computational cost for creating the target values is required if the training set is very large.

10.4 Continuous Speech Recognition Experiment

In this section, isolated word recognition and phrase recognition experiments using TDNN-LR continuous speech recognizer [Sawai 91] were performed using the same ATR database applying 25-phoneme identification TDNN.

10.4.1 TDNN-LR Speech Recognizer

TDNN-LR speech recognizer consists of two main techniques:

(1) Generalized LR-parser.
(2) TDNN+DTW phoneme verifier.

A brief idea of each technique is introduced in this section.

Generalized LR-Parser

The LR-parser [Aho 86] is originally developed for programming languages, and is known as an effective parser for a large class of context-free grammar.

The grammar rules described in a context-free grammar style are automatically pre-compiled into an LR-table (**action table** and **goto table**) by the LR-table generator. The LR-parser is deterministically guided by the LR-table with the two subtables (**action table** and **goto table**) and is processed left-to-right without back-tracking.

The action table determines the next parser action **ACTION**[s,a] from the state s currently on top of the stack and the current input symbol a. There are four kinds of actions: **shift**, **reduce**, **accept** and **error**. The action **shift** means one word from input buffer onto the stack. The action **reduce** means constituents on the stack using the grammar rule. The action **accept** means input is accepted by the grammar. And the action **error** means input is not accepted by the grammar. The goto table determines the next parser state **GOTO**[s,A] from the state s and the grammar table symbol **A**.

The standard LR-parser cannot handle ambiguous grammars. In order to cope with natural language processing, which includes speech processing, this ambiguous grammars have to be handled. This ambiguous grammars is able to handled by incorporating a multiple entries (conflicts). And as a general method, stack-splitting mechanism can be used to cope with multiple entries. Whenever a multiple entry is encountered, the stack is divided into two stacks, and each stack is processed in parallel. The Generalized LR-parser is proposed [Tomita 86] in order to handle this ambiguous grammar for natural language processing by incorporating a multiple entries (conflicts) into the LR-table. Thus, this mechanism makes it possible to use LR-parser to handle an ambiguous grammar which is very effective to handle natural language processing.

TDNN-LR Procedure

TDNN-LR speech recognizer [Sawai 91] is realized as an integrated system of the Generalized LR-parser [Tomita 86] and the TDNN phoneme identifier [Waibel 89]. The system architecture of the TDNN-LR is very effective and it is a sophisticated speech recognizer which can deal simultaneously with phoneme verification using the linguistic information constraints and language analysis using the grammar. The block diagram of the TDNN-LR speech recognizer is shown in Figure 10-7.

The process of TDNN-LR speech recognizer performs as follows:

(1) Acoustic analysis is performed for the input speech.

(2) Phoneme identification is performed frame-by-frame by shifting TDNN phoneme identifier over the analyzed input speech.

(3) Phoneme verification, symbol reduction to symbol or acceptance is requested by the LR-parser according the LR-table. As for the phoneme verification request, the phonemes which might come after the current state under the linguistic constraint should be verified. As for symbol reduction, the symbol

will be reduced to another symbol using the grammar table. As for the acceptance, the input is accepted under the linguistic constraint and later end the process.

(4) Verification of the aforementioned phonemes is performed using the DTW (Dynamic Time Warping) algorithm. The score for the DTW is computed as the log value of the frame-by-frame identified phoneme values of the requested phoneme. Each phoneme reference has a frame length of an average duration estimated using the training samples of the isolated word database. The window for DTW calculation is in between $1/2 \cdot t$ and $2 \cdot t$ where t is number of frames in time.

The DTW is realized in the following equation:

$$g(i,t) = max \left\{ \begin{array}{l} g(i-1,t-1) \\ \quad +log(TDNN(t,p)), \\ g(i-2,t-1) \\ \quad +log(TDNN(t,p)) + log(TDNN(t-1,p)), \\ g(i-1,t-2) \\ \quad +0.5log(TDNN(t,p)) + 0.5log(TDNN(t-1,p)) \end{array} \right\}$$

where

p is the requested phoneme.

i is the position in the reference phoneme sequence.

t is the frame number.

$TDNN(t,p)$ is the TDNN activating value of p at t.

The duration control performs after the phoneme scoring by DTW. The duration control is realized by multiplying a penalty into the DTW score in the form of Gaussian distribution using the difference between the average duration μ of the phoneme and the estimated duration d of phoneme obtained by the DTW. The penalty $P(d)$ for the phoneme duration control is given in the following equation:

$$P(d) = exp(-\frac{(d-\mu)^2}{2\pi\sigma^2})$$

In the case of phrase recognition, the average duration μ and the deviation σ for each phoneme are re-estimated from the isolated word phoneme duration by the phrase duration transformation function [Hanazawa 90].

(5) Scores at the current state are sorted to realize beam search. The top-N candidates are saved and the others are pruned. Then shift is performed to go to the next state in the LR-table, goto 3).

The whole process of this TDNN-LR speech recognizer is similar to the level-building DTW speech recognition system with a context-free grammar [Myers 81] with its end point free, which builds up subsequent phonemes.

10.4.2 Experimental Condition

To recognize isolated words and phrases in Japanese using the TDNN-LR speech recognizer, 25-phoneme identification TDNN is adopted. The Japanese 25 phonemes are /b/,/d/,/g/, /p/,/t/,/k/, /ch/,/ts/, /s/,/sh/,/h/,/z/,/zh/, /m/,/n/,/N/, /r/,/w/,/y/, /a/,/i/,/u/,/e/,/o/ and silence /Q/. Input to the neural networks is analyzed under the same conditions used in the phoneme identification experiment.

A 25-phoneme identification TDNN is shown in Figure 10-8. The structure of the TDNNs is a feed-forward neural network of four layers. The input for the TDNNs is 16 mel-scaled spectral power of 7 frames (70ms). The neural networks are trained using the fast back-propagation training method "Dynet" [Haffner 89].

In this experiment, phoneme samples for the TDNN training are culled from half of the ATR isolated word database (even numbered words; 5.7 mora/s) using the hand-labels. However, the condition of selecting training samples differs from the previous experiments. Samples are culled from a phoneme not at the end of the hand-label but from several points in the phoneme as shown in Figure 10-9. One sample is selected from the center of the phoneme, two from the edge of the phoneme whose center of the sample is located 15ms inside the phoneme boundaries, and others by shifting 15ms toward the boundaries inside the two edge samples. Up to 2,000 training samples for each phoneme class are selected.

For the isolated word recognition experiment, the other half of the training data in the ATR isolated word database (odd numbered words; 5.7 mora/s), and for the phrase recognition experiment, the 278 ATR phrase database (7.1 mora/s), are used.

Two word dictionaries and two grammars are used to evaluate the isolated word recognition and the phrase recognition performance, respectively.

For isolated word recognition, two word dictionaries are used:

(1) Small vocabulary task using a 500 word dictionary.
(2) Large vocabulary task using a 2,620 word dictionary.

For phrase recognition, two context-free grammars are used:

(1) Small task using a task specific grammar.
(2) Large task using a general grammar.

The complexity of the phrase grammar is shown in Table 10-1.

As previously mentioned, the computational cost is very high for creating the target values for every training sample in the Neural Fuzzy Training method, if this method is adopted in a straightforward manner. Here, about 50,000 samples have to be trained. The distance calculation is about $(50,000 \cdot 50,000)/2$ in this large training set. To avoid this problem, pre-selection of training samples, 200 random samples per phoneme class, for likelihood calculation is performed. The computational cost is about 1/5 that without pre-selection.

Here, two conventional training methods and the proposed Neural Fuzzy Training method are compared. The two conventional methods are the back propagation algorithm with 1) mean square error (M.S.E.) function and 2) McClelland error function $ln(1 - \varepsilon^2)$. M.S.E function is adopted in the Neural Fuzzy Training method.

Other experimental conditions are almost the same as in the previous phoneme identification experiments, such as 7-frame Euclidean distance measure d between samples, likelihood transformation function $f(d) = exp(-\alpha \cdot d^2)$ where $\alpha = 0.005$, and so on. The only difference is the weight selection. The weight of 100 training iterations is chosen. At 100 training iterations, the TDNN training almost converged.

10.4.3 Result

Table 10-2 and Table 10-3 show the recognition results of all tasks (500 and 2,620 isolated word recognition and 278 phrase recognition using task specific grammar and general grammar) using the TDNN-LR speech recognizer for speaker MAU and MHT, respectively.

The result for the conventional method is obtained by using the weight trained by the McClelland error function, because the weight trained by M.S.E. function was not sufficiently estimated within 100 iterations. Discussions of the training speed will appear in the next section.

Figure 10-10 shows the output values obtained by each TDNN for input speech /to:jitsuno/. At the top of the figure, the input spectrogram is shown. The second shows the output values obtained by the TDNN trained using a conventional training method with McClelland error, and the bottom shows the output values obtained by the TDNN trained using the Neural Fuzzy Training method.

In the conventional result, several deletion errors, such as /ts/ and /n/, can be observed, which increase the fatal error possibility. On the other hand, few deletion errors can be observed in the Neural Fuzzy trained result, but many insertion errors can be observed. However, in the regions of these insertion errors, the correct phoneme result can also be observed. Thus, it will not lead a fatal error.

In practice, the recognition result for this input /to:jitsuno/ appears in the second candidate in the Neural Fuzzy Training case. Though in the conventional training case it appears in the fifth candidate. The top result is mistaken as /to:jitsuo/ in the Neural Fuzzy Training case, because the duration control is not well-tuned in the current system.

10.4.4 Effectiveness of Neural Fuzzy Training

In this section, the effectiveness of the Neural Fuzzy Training method 1) of training speed, 2) against over-training and 3) against untrained speakers, is discussed.

Effectiveness of Training Speed

Figures 10-11a, 10-11b, 10-11c show the training speed of each method for the 6-phoneme, 18-phoneme and 25-phoneme identification TDNN of the speaker MAU data. 6-phoneme are the /b,d,g,m,n,N/, 18-phoneme are for the 18 consonants and 25-phoneme are of all phonemes. The training methods are: 1) conventional training with M.S.E., 2) conventional training with McClelland error and 3) the proposed Neural Fuzzy Training method. The McClelland error function is well-known as a very fast training error function in the back-propagation algorithm when the number of the identification classes is large.

In the 6-phoneme training, there are no speed differences between each training method. In the 18-consonant training and in the 25-phoneme training, the training speed of the conventional training method with M.S.E error is somewhat slower than the others.

The effect of the proposed method and the McClelland error is evident, especially in Figure 10-11c compared with that of the M.S.E. The training speed of the proposed method is almost the same as that of the McClelland error. Thus, the Neural Fuzzy Training method proved to be a very fast training method.

Effectiveness against Over-Training

Figure 10-12 shows the training speed for the 25-phoneme identification TDNN of the speaker MHT. In the conventional method with the McClelland error function, the training converged around 96%, however in the Neural Fuzzy Training method, it converged around 92%.

From these results, the neural fuzzy training method does not seem to be a good training algorithm for neural networks. However, the result shown in Table 10-3 is good compared with the conventional trained results. Thus, from this point of view, the result indicates that the Neural Fuzzy Training method can avoid over-training the training data.

Effectiveness against Untrained Speakers

Tables 10-4, 10-5 and Tables 10-6, 10-7 show the recognition results for untrained speakers. Tables 10-4, 10-5 show the results using the TDNN trained by speaker MAU and Tables 10-6, 10-7 trained by speaker MHT. Tables 10-4, 10-6 show the results on a 2,620 word recognition task and Tables 10-5, 10-7 on phrase recognition task using a general grammar. There are some slight performance reductions, albeit very small, however, most results have improved. The improvement can be considered evidence of a good generalization of the neural network using the proposed Neural Fuzzy Training method. This also indicates that the Neural Fuzzy Training will be more effective when it is applied to a speaker independent approach.

10.5 Conclusion

A new fuzzy training method for neural network classifiers, called "Neural Fuzzy Training", has been described. The effectiveness of the proposed method compared with the conventional method is shown using both phoneme identification experiments and continuous speech recognition experiments. Furthermore, the proposed "Neural Fuzzy Training" method is also shown to be a very fast training algorithm.

10.6 Figures & Tables

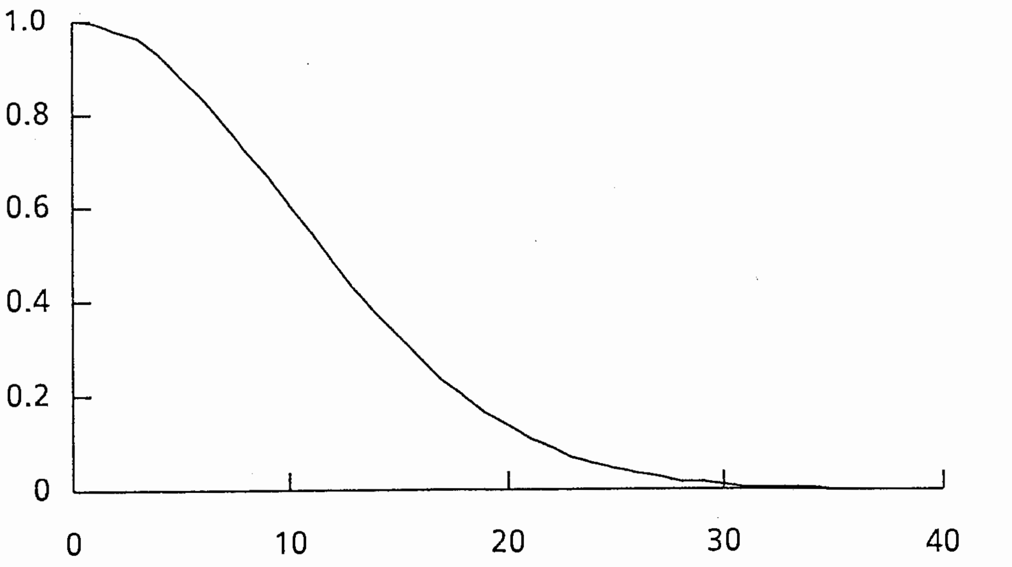
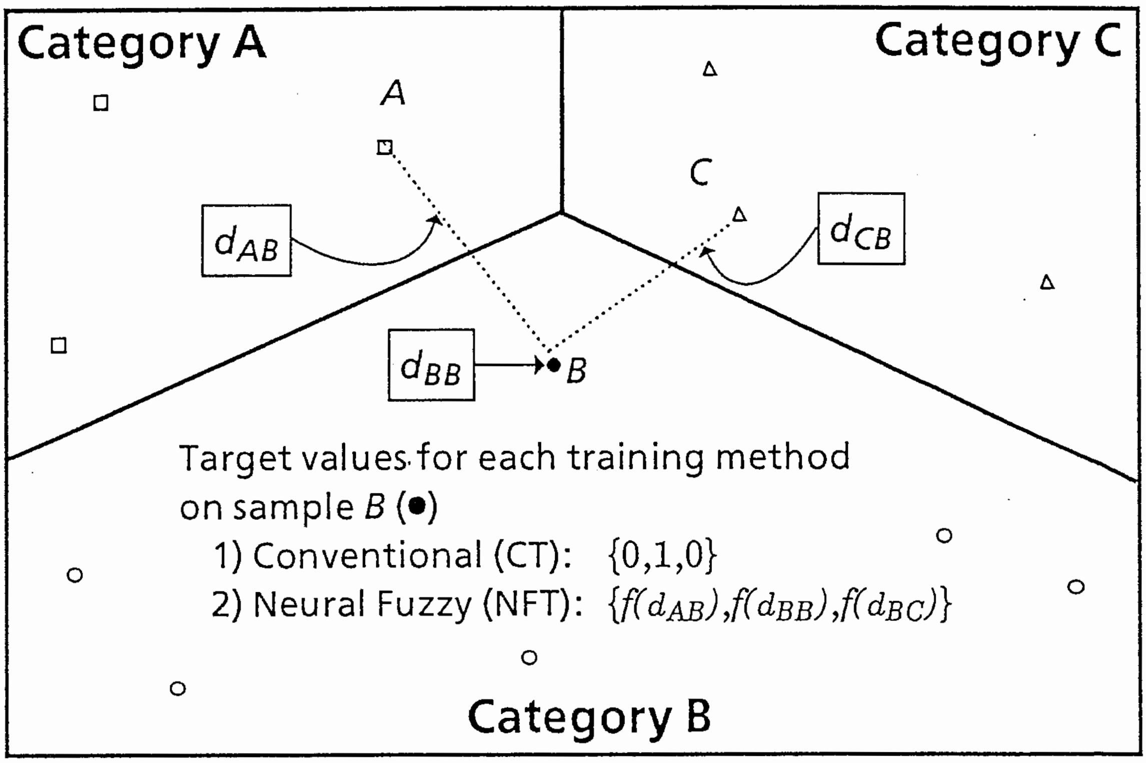


Figure 10-1: Likelihood Transformation Function $f(d) = exp(-a{\cdot}d2)$



Target values for each training method on sample B (●)

1) Conventional (CT): {0,1,0}

2) Neural Fuzzy (NFT): $\{f(d_{AB}),f(d_{BB}),f(d_{BC})\}$

Figure 10-2: General Idea of Neural Fuzzy Training

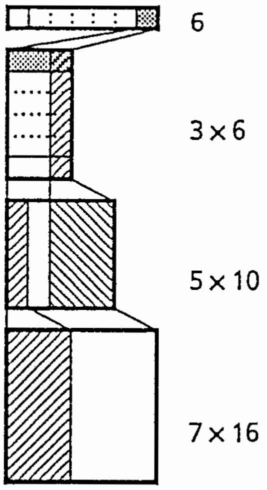


Figure 10-3: /b,d,g,m,n,N/ Identification TDNN

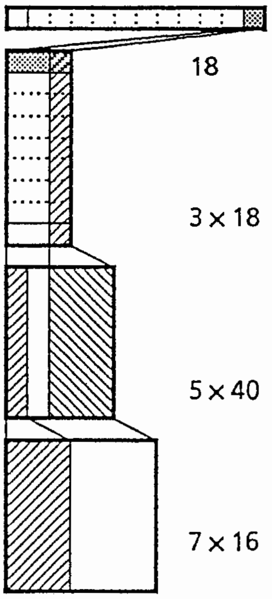


Figure10-4: 18-Consonant Identification TDNN

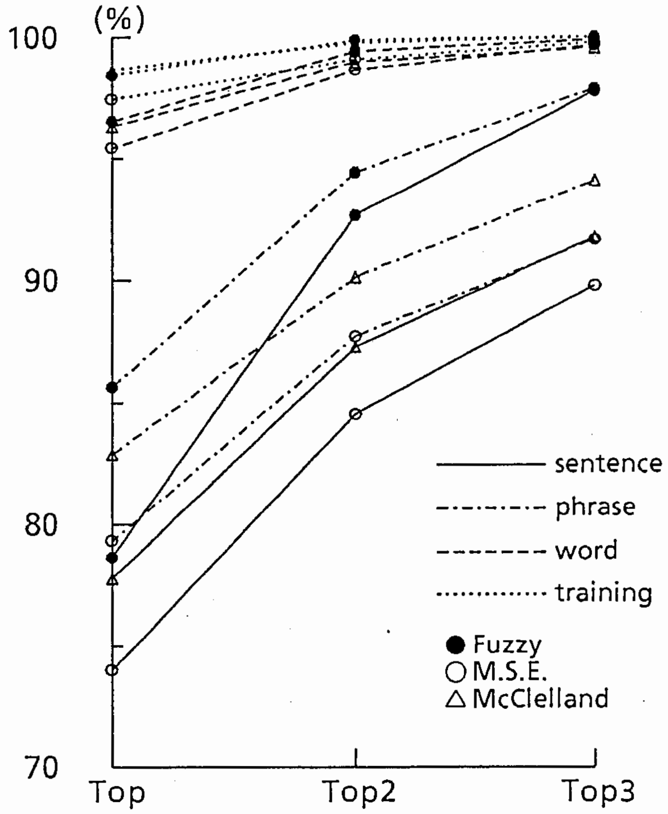


Figure 10-5: /b,d,g,m,n,N/ Identification Result

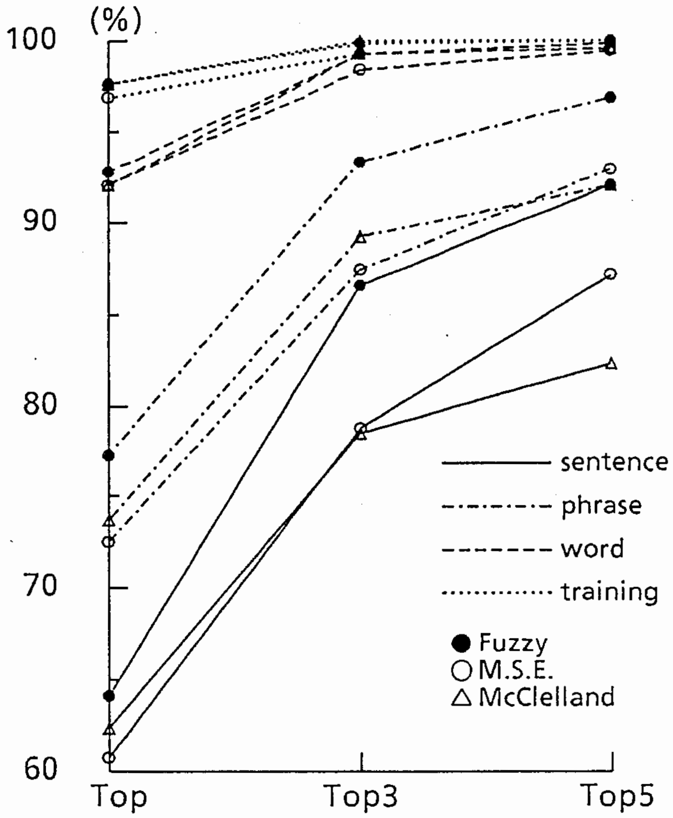


Figure 10-6: 18-Consonant Identification Result

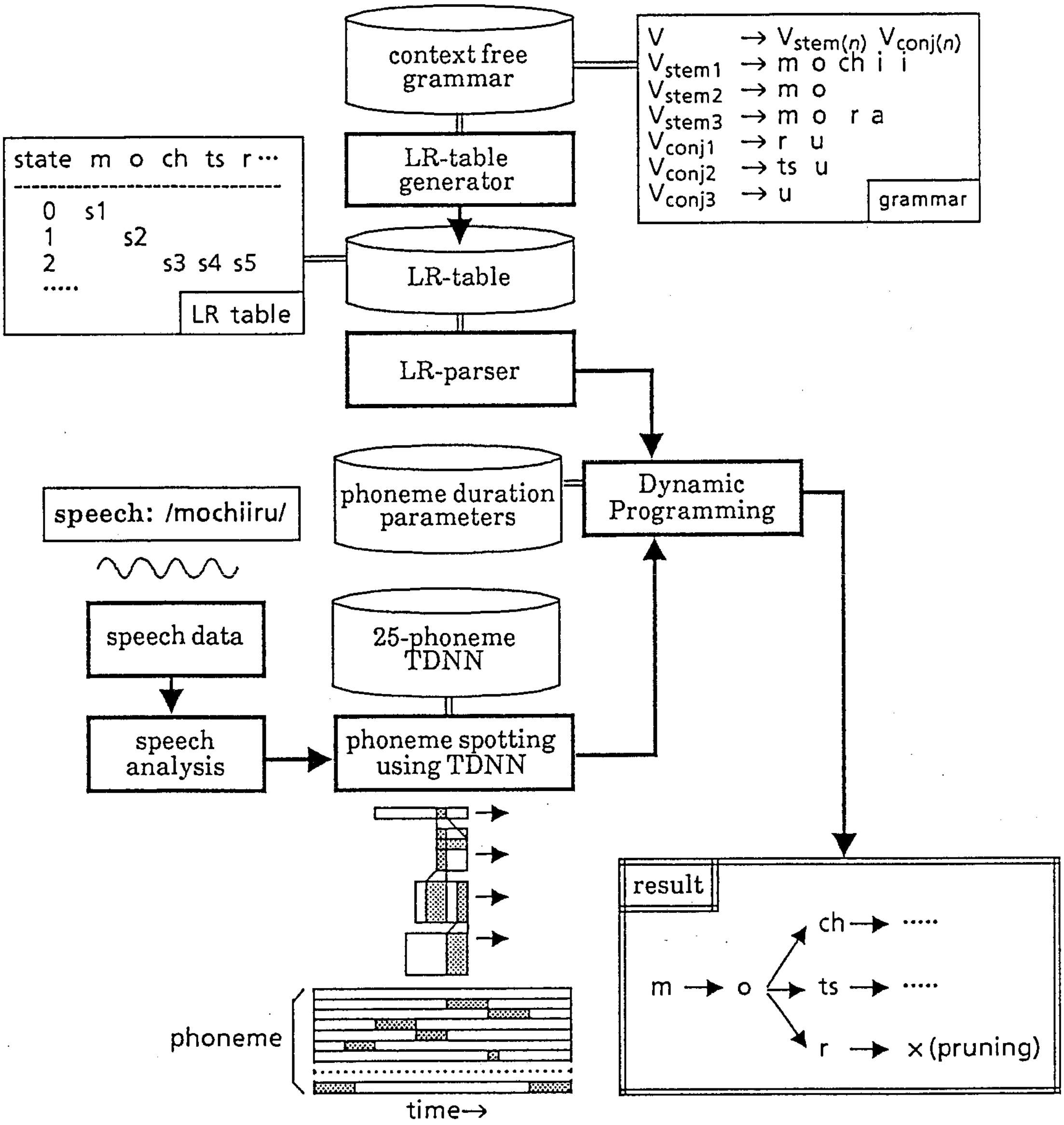


Figure 10-7: TDNN-LR Continuous Speech Recognizer

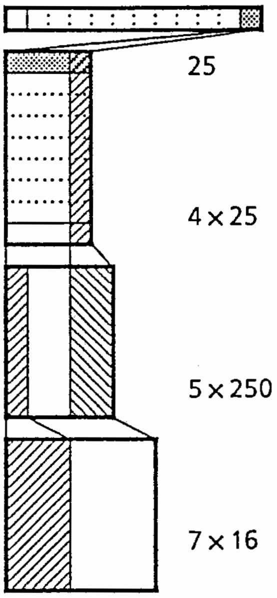


Figure 10-8: 25-Phoneme Identification TDNN

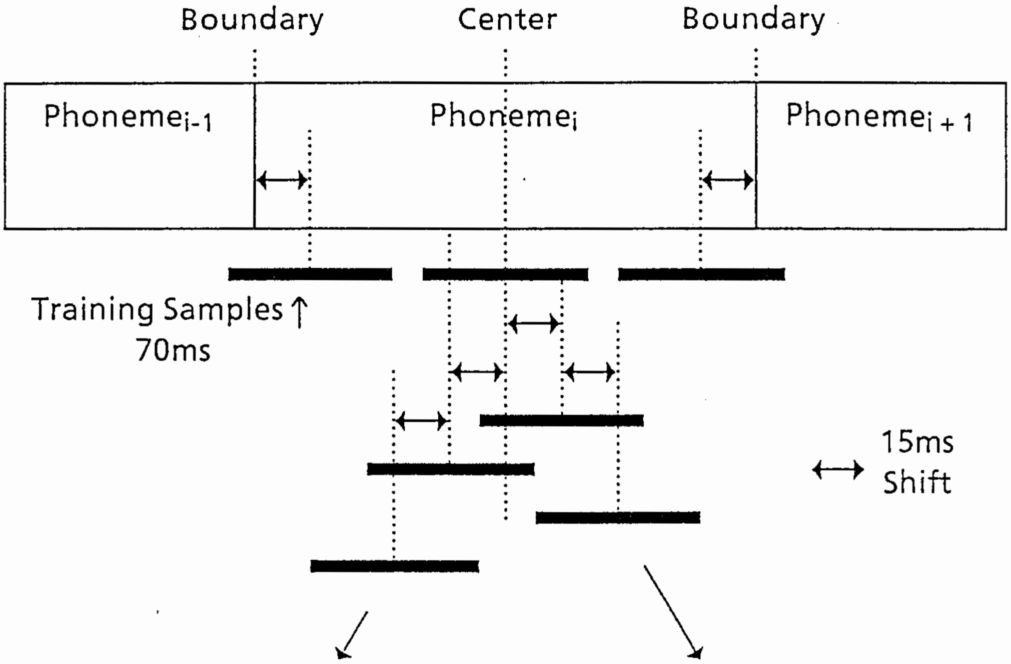


Figure 10-9: Training Samples for TDNN

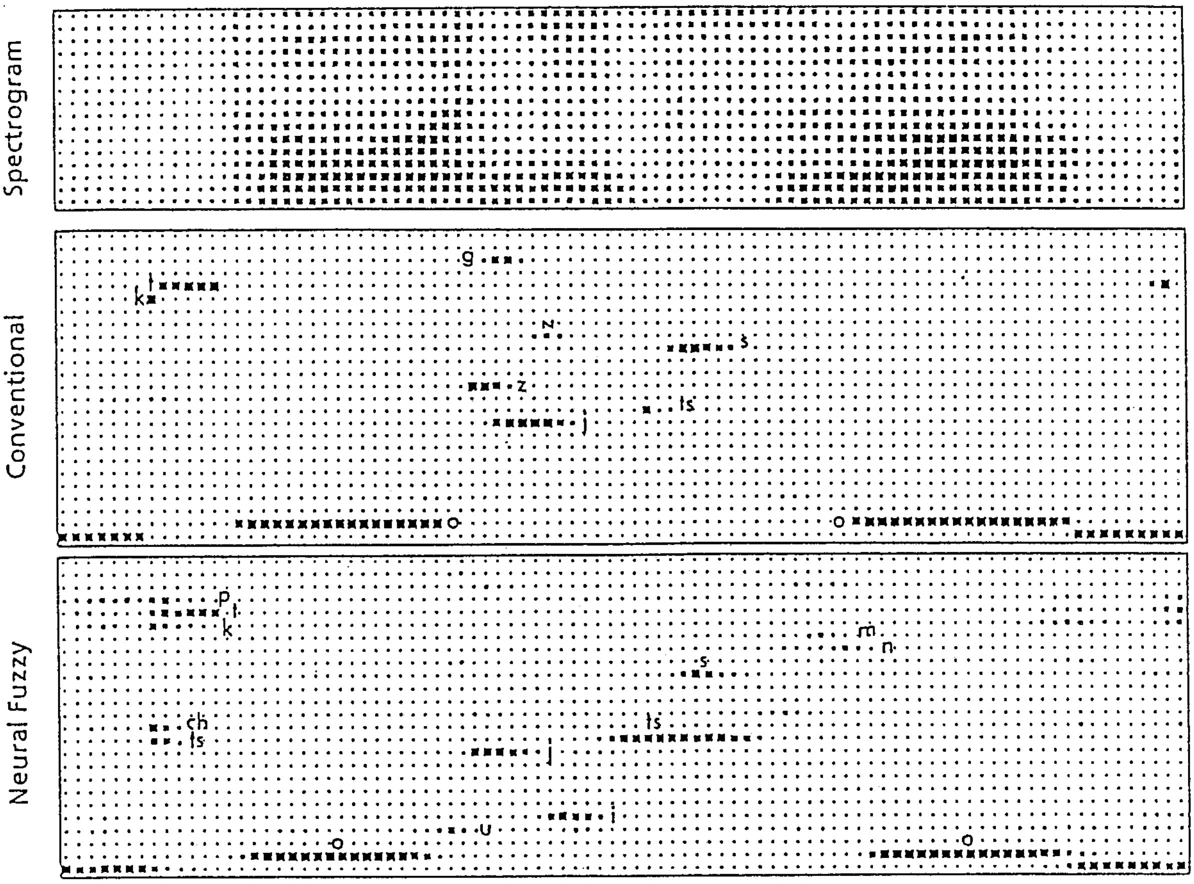


Figure 10-10: TDNN-Scanning Activation Values (/to:jitsuno/)

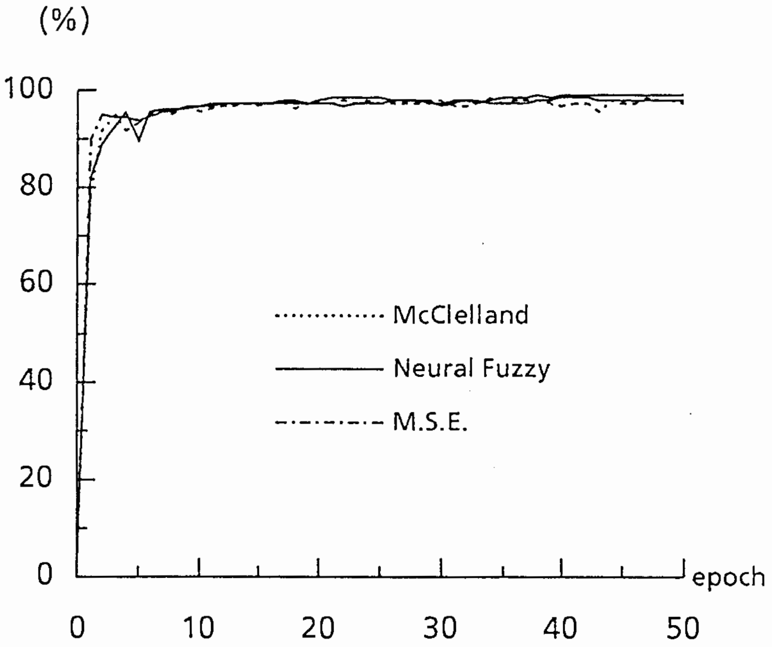


Figure 10-11a: Training Speed of /b,d,g,m,n,N/ Identification TDNN for Speaker MAU

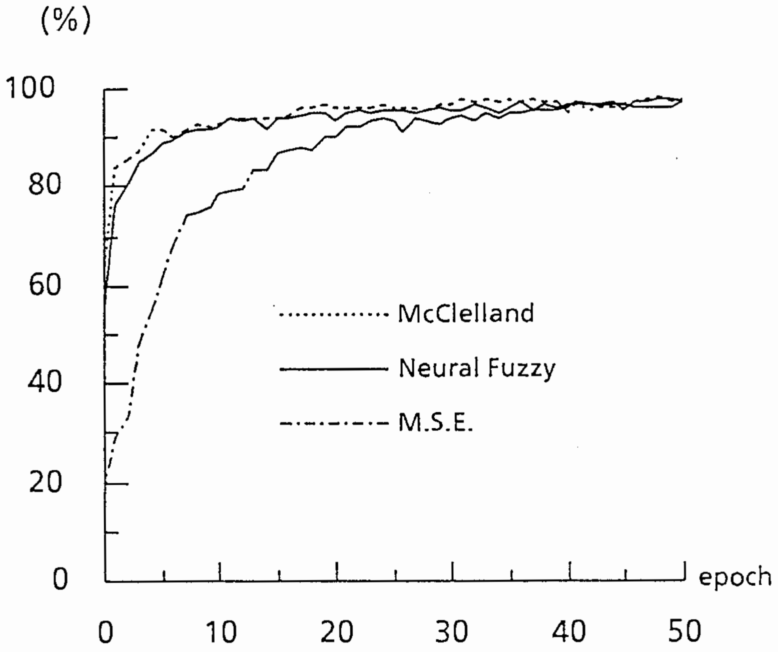


Figure 10-11b: Training Speed of 18-Consonant Identification TDNN for Speaker MAU

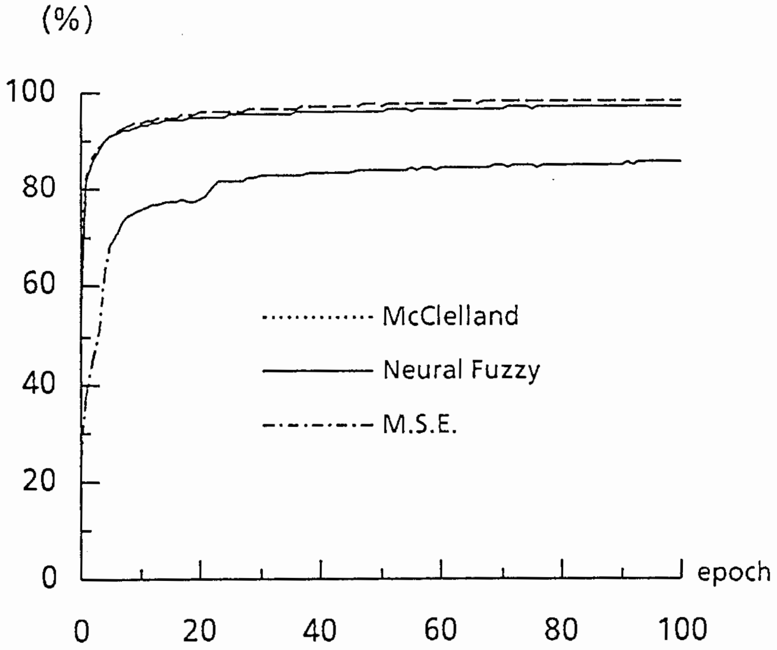


Figure 10-11c: Training Speed of 25-Phoneme Identification TDNN for Speaker MAU

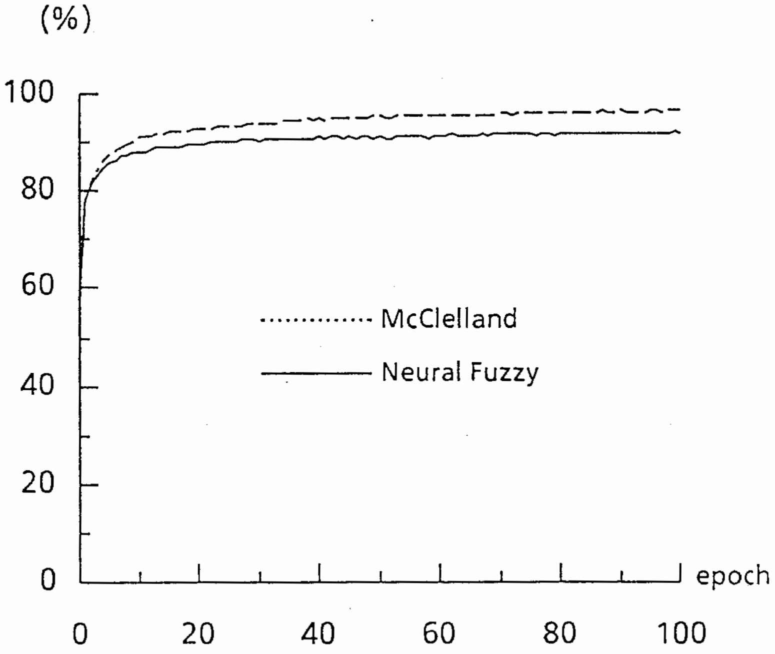


Figure 10-12: Training Speed of 25-Phoneme Identification TDNN for Speaker MHT

Table 10-1: Size of Grammar for Phrase Recognition Using TDNN-LR

Task	Grammar	Number of Rules	Size of Vocabulary	Number of States in LR-table
Small	Task Specific	607	275	1341
Large	General	1672	1035	4866

Table 10-2: Speaker Dependent Speech Recognition MAU (%)

Task & Method	500 words		2,620 words		*small		*large	
	CT	NFT	CT	NFT	CT	NFT	CT	NFT
Top-1	97.8	98.0	96.1	_94.7*_	71.2	80.9	64.3	71.2
Top-3	99.2	99.6	99.4	_99.0*_	86.0	93.5	81.7	88.8
Top-5	99.2	99.6	99.5	_99.4*_	92.8	96.0	87.1	92.1

CT: _Conventional Training (McClelland),_ NFT: _Neural Fuzzy Training (M.S.E.)_
*small: _Phrase using Small Grammar,_ *large: _Phrase using Large Grammar_

Table 10-3: Speaker Dependent Speech Recognition MHT (%)

Task & Method	500 words		2,620 words		*small		*large	
	CT	NFT	CT	NFT	CT	NFT	CT	NFT
Top-1	98.0	_97.0*_	96.6	_94.2*_	71.2	76.6	66.9	71.2
Top-3	98.8	99.6	99.3	_98.2*_	91.0	_89.2*_	85.3	88.8
Top-5	98.8	99.6	99.5	_98.9*_	93.5	93.9	89.2	92.8

CT: _Conventional Training (McClelland),_ NFT: _Neural Fuzzy Training (M.S.E.)_
*small: _Phrase using Small Grammar,_ *large: _Phrase using Large Grammar_

Table10-4: Recognition on Untrained Speakers (%) (*Trained by* MAU)

Speaker & Method	278 **Phrase** Recognition Using **Large** Grammar							
	MAU		MHT		MNM		FSU	
	CT	NFT	CT	NFT	CT	NFT	CT	NFT
Top-1	64.3	71.2	32.0	40.6	32.0	37.4	2.9	7.2
Top-3	81.7	88.8	51.1	55.4	54.4	59.4	8.2	9.4
Top-5	87.1	92.1	59.0	59.7	61.5	67.6	10.1	10.8

CT: *Conventional Training (McClelland),* NFT: *Neural Fuzzy Training (M.S.E.)*

Table 10-5: Recognition on Untrained Speakers (%) (*Trained by* MTH)

Speaker & Method	278 **Phrase** Recognition Using **Large** Grammar							
	MAU		MHT		MNM		FSU	
	CT	NFT	CT	NFT	CT	NFT	CT	NFT
Top-1	36.7	47.5	66.9	71.2	37.4	46.8	2.9	23.4
Top-3	55.4	64.4	85.3	88.8	54.0	66.9	5.0	33.8
Top-5	59.4	72.7	89.2	92.8	60.4	71.9	6.8	36.7

CT: *Conventional Training (McClelland),* NFT: *Neural Fuzzy Training (M.S.E.)*

Table 10-6: Recognition on Untrained Speakers (%) (*Trained by* MAU)

Speaker & Method	2,620 isolated word Recognition							
	MAU		MHT		MNM		FSU	
	CT	NFT	CT	NFT	CT	NFT	CT	NFT
Top-1	96.1	94.7*	61.6	65.4	52.4	57.7	12.5	10.6*
Top-3	99.4	99.0*	78.2	83.2	72.8	76.4	22.0	20.0*
Top-5	99.5	99.4*	83.9	87.9	79.1	81.3	27.3	25.8*

CT: *Conventional Training (McClelland),* NFT: *Neural Fuzzy Training (M.S.E.)*

Table 10-7: Recognition on Untrained Speakers (%) (*Trained by* MTH)

Speaker & Method	2,620 isolated word Recognition							
	MAU		**MHT**		MNM		FSU	
	CT	NFT	CT	NFT	CT	NFT	CT	NFT
Top-1	78.5	78.0*	96.6	94.2*	63.1	76.3	14.5	35.9
Top-3	91.4	91.0*	99.3	98.2*	80.5	89.9	24.6	52.4
Top-5	94.3	94.0*	99.5	98.9*	85.7	92.7	29.9	60.0

CT: *Conventional Training (McClelland),* NFT: *Neural Fuzzy Training (M.S.E.)*

Chapter 11

CONCLUSIONS

11.1 Summary

This report proposed a phoneme recognition expert system aiming the two purposes:

(1) Simulation of spectrogram reading behavior of a human expert using an expert system.
(2) Development of a speech recognizer by integrating human knowledge and neural networks.

In general, conventional expert systems for phoneme recognition are realized by a separate structure of a) acoustic feature extraction and b) phoneme verification, aiming at constructing a full rule-based system. Although, most of these systems have the following problems:

- Only the static human knowledge is utilized.
- Dynamic human knowledge (i.e. human strategy) is not utilized.
- Impossible to pre-process all acoustic feature extraction.
- Impossible to extract precise features according to phoneme context.
- Impossible to describe all knowledge in explicit rules.
- Difficult to manage context dependent acoustic features.
- Unextractable precise features exist on a spectrogram.

In order to overcome these problems and to realized the two purposes described above, the following techniques are incorporated in the proposed expert system.

(1) Spectrogram Reading Process Simulation

– Human strategy is adopted as dynamic knowledge.

- Hypothesis and evaluation as human behavior.
- Representation of explicit knowledge.
- Non-deterministic strategy (ATMS).
- Representation of uncertainty.
- Representation of fuzziness.
- On-demand contextual top-down acoustic feature extraction.

(2) Time-Delay Neural Networks

- Representation of implicit knowledge.
- Extraction of unextractable precise features.
- High performance phoneme identifier.
- Tolerance capability for slight segmentation errors.
- Good vowel detector as a phoneme-spotting method.

(3) Total System

- Integration of human knowledge and neural networks by considering suitability.
- Full bottom-up style speech recognizer without a language model.

With these techniques, the proposed expert system achieved:

(1) Accurate feature based phoneme segmentation.
(2) Robust feature based phoneme segmentation for continuous speech and multiple speaker utterances.
(3) Powerful neural network based phoneme identification.
(4) Good phoneme recognition performance by a close integration of knowledge and neural networks.

Moreover, a) Time-State Neural Networks (TSNN) by considering the temporal structure of a phonemic feature, and b) Neural Fuzzy Training method for a robust neural network creation, are proposed in order to expand the expert system toward continuous speech recognition.

In Chapter 1, the purpose of this study was described along with the background of recent studies on speech recognition.

In Chapter 2, the framework of the expert system, in order to simulate human expert behavior naturally and easily: a) spectrogram reading knowledge for explicit knowledge, b) non-deterministic strategy, c) representation of uncertainty and fuzziness, d) on-demand top-down control feature extraction under phoneme context constraints, e) Time-Delay Neural Networks representing implicit knowledge were described.

In Chapter 3, the hardware configuration and the architecture of the proposed speech recognition expert system, which was realized as an integration of human knowledge and neural networks, was described.

In Chapter 4, details of consonant recognition part was described which consisted of two main parts, 1) feature based phoneme segmentation, 2) neural network based phoneme identification. The experimental result tested on speaker dependent 15-consonant task using an ATR database was also reported. The expert system correctly recognized 86.8% of the total number of phonemes, both in phoneme segmentation and phoneme identification.

In Chapter 5, five mechanisms of integrating knowledge and neural networks were proposed. Consonant recognition experiments were carried out and the proposed mechanisms were compared. The experiment showed that the close integration of knowledge and TDNN by considering their suitability with a reject filter improved the overall system performance. Not only the identification performance but also segmentation accuracy, and the reject filter showed an effective reduction in insertion errors. A phoneme recognition experiment showed an 89.4% recognition rate for 15 consonants using the best integration mechanism.

In Chapter 6, vowel and semivowel recognition utilizing a phoneme-spotting TDNN for vowel detection was described. Human knowledge was mainly utilized for verifying the vowel category and boundaries. The effectiveness was shown through a vowel detection experiment, whose detection rate of 96.1% was comparatively good.

In Chapter 7, the overall phoneme recognition expert system was evaluated using the best integration of knowledge and neural networks without using any language model. An experiment showed a 91.4% recognition rate for all Japanese 23 phonemes. The deletion error rate was 3.6%, the substitution error rate 5.0% and the insertion error rate 20.7%.

In Chapter 8, the performance of a feature based phoneme segmentation of the proposed expert system, tested on speaker independent and continuous speech, were presented. The experiments were performed both on isolated word speech uttered by six speakers and on speaker dependent continuous speech. The results were as good as the result tested on speaker dependent isolated word speech. In order to achieve this performance by the expert system, only slight modification of knowledge need to be done, which indicates that the feature based approach is a robust method of for phoneme segmentation.

In Chapter 9, a new structure for phoneme identification neural networks which took account of temporal structures of phonemic features, Time-State Neural Networks (TSNN) was proposed. Several types of TSNNs were described along with their phoneme identification experimental results on Japanese phonemes /b,d,g,m,n,N/ culled from isolated word, phrase and sentence utterances. The performance of the proposed TSNNs proved better that that of the conventional TDNN.

In Chapter 10, a Neural Fuzzy Training method for phoneme identification neural networks was proposed whose general idea was to give a fuzzy phoneme class information to target values. The experiments of phoneme identification and of continuous speech recognition using the TDNN-LR speech recognizer showed dramatic improvement especially on continuous speech data compared with the conventional training method. The improvement of the Neural Fuzzy Training method was not only on identification or recognition performance but also on the training speed.

11.2 Further Research

The proposed system showed a good performance for phoneme recognition under the condition of speaker independent isolated word utterance.

However, for speaker independent and continuous speech, the performance of the proposed system has not yet been evaluated. The main reason is that the performance of the neural network phoneme identifier is not particularly significant for an untrained speaker, and also the computational cost for speaker independent neural network training is very high. Thus, the speaker adaptation method for neural network [Nakamura 90] [Iso 89] [Fukuzawa 91] or how to train a speaker independent neural network [Sawai 91] must be studied. After investigation of these neural networks, the system may show a good performance for speaker independent continuous speech, because its phoneme segmentation part is robust.

However, the current system expansion, in other words the rules of knowledge creation and modification, have to be performed by hand, which is a very big problem for the future expansion. Humans are not able to look at a large amount of data for rule creation and modification. To overcome this problem, an automatic rule creation mechanism and a system adaptation mechanism for new data must be developed. Moreover, the increase in the number of rules will lead to another difficult problem of complex rule management. Therefore, a more sophisticated rule management system must be developed.

Note that, in recent speech processing research, the speech database has played an important role. In particular, phoneme labeled speech databases have contributed to the improvement of speech recognition systems, and the larger the better. Thus, an accurate automatic labeling system is required. By utilizing the advantage of the feature based phoneme segmentation system in combination with the phonetic transcription of the utterance, an accurate automatic labeling system [Fujiwara 91] for creating a database can be realized. This can be another contribution of this report in the study of speech recognition.

Bibliography

[ART 87]
"ART Reference Manual"
Inference Corp., 1987.

[Aho 86] Aho, A. V., Sethi, R., Ullman, J. D.
"Compilers, Principles, Techniques, and Tools"
Addison-Wesley, Massachusetts, 1986.

[Asidi 90] Asidi, A., Schwartz, R., Makhoul, J.
"Automatic Detection of New Words in a Large Vocabulary Continuous Speech Recognition System"
IEEE ICASSP90, pp. 125-128, 1990-04.

[Baker 75] Baker, J. K.
"The DRAGON system - An Overveiw"
IEEE Trans. ASSP, Vol. ASSP-23, 1975-02.

[Baum 70] Baum, L. E., Petrie, T., Soules, G., Weiss, N.
"A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains"
Ann. Math. Stat., vol. 41, pp. 164-171, 1970.

[Buchanan 85] Buchanan, B. G., Shortliffe, E. H.
"Rule-Based Expert Systems"
Addison-Wesley Publishing Company, California, 1985.

[Carbonell 86] Carbonell, N., Damestory, J. P., Forth, D., Haton, J. P., Lonchamp, F.
"APHODEX, Design and Implimentation of an Acoustic-Phonetic Decoding Expert System"
IEEE ICASSP86, pp. 1201-1204, 1986.

[Chow 87] Chow, Y. L., Dunham, M. O., Kimball, O. A., Krasner, M. A., Kubala, G. F., Makhoul, J., Price, P. J., Roucos, S., Schwarts, R. M.
"BYBLOS: The BBN Continuous Speech Recognition System"
IEEE ICASSP87, pp89-92, 1987-04.

[Connolly 86] Connolly, J. H., Edomonds, E. A., Guzy, J. J., Johnson, S. R., Woodcock, A.
"Automatic Speech Recognition Based on Spectrogram Reading"
Int. J. Man-Machine Studies, 24, pp. 661-621, 1986.

[Davis 52] Davis, K. H., Biddulph, R., Balashek, S.
"Automatic Recognition of Spoken Digits"
Journal of the Acoustic Society of America, 24, 1952-11.

[de Kleer 86] de Kleer, J.,
"An Assumption-Based TMS"
Artificial Intelligence, 28, pp. 127-162, 1986.

[Dempster 77] Dempster, A. P., Laird, N. M., Rubin, D. B.
"Maximum-Likelihood from Incomplete Data via the EM Algorithm"
J.Royal Statist. Soc. Ser. B (methodological), vol. 39, pp. 1-38, 1977.

[Dudley 40] Dudley, H.
"The Carrier Natural of Speech"
Bell Sys. Tech. Journal, 19, 1940.

[Fant 60] Fant, G.
"Acoutic Theory of Speech Production"
Monton & Co's - Gravenhage, 1960.

[Franagan 55] Flanagan, J. L.
"A Difference Limen for Vowel Formant Frequency"
Journal of the Acoustic Society of America, 27, 1955.

[Fukuzawa 91] 福沢圭二, 小森康弘, 沢井秀文, 杉山雅英
"セグメント話者適応ニューラル・ネットワークを用いた文節音声認識"
音学講論, 秋季研究発表会, 2-5-11, 1991-10.

[Fujiwara 91] 藤原紳吾, 岩橋直人, 小森康弘, 杉山雅英
"HMM とスペクトログラム・リーディング知識に基づく
ハイブリッド音素セグメンテーションシステム"
音学講論, 秋季研究発表会, 2-5-20, 1991-10.

[Furui 85] 古井
"ディジタル音声処理"
ディジタルテクノロジーシリーズ 6, 東海大学出版会, 1985.

[Haffner 89] Haffner, P.
"Fast Back-propagation Learning Methods for Large Phonemic Neural Networks"
Neurospeech89, 1989-05.

[Hanazawa 90] Hanazawa, T., Kita, K., Nakamura, S., Kawabata, T., Shikano, K.

"ATR HMM-LR Continuous Speech Recognition System"
IEEE ICASSP90, pp. 53-56, 1990-04.

[Hatazaki 87] 畑崎香一郎, 田村震一, 川端豪, 鹿野清宏
"連続音声中の音韻認識エキスパートシステムの検討"
情報処理学会 35 回全国大会 講論集, pp. 1443-1444, 1989-09.

[Hatazaki 88] Hatazaki, K., Tamura, S., Kawabata, T., Shikano, K.
"Phoneme Segmentation by an Expert System based on Spectrogram Reading Knowledge"
Proc. of Speech'88, 7th FASE Symposium, pp. 927-934 1988.

[Hatazaki 90] 畑崎香一郎, 小森康弘, 川端豪, 鹿野清宏
"スペクトログラム・リーディング知識を用いた 音韻セグメンテーション・エキスパートシステム"
信学会論文誌, D-II, Vol. J73-D-II, No.1, pp. 1-9, 1990-01.

[Ishizuka 85] 石塚満
"曖昧な知識の表現と利用"
情報処理, 26, 12, pp. 1481-1486, 1985-12.

[Iso 89] 磯健一, 麻生川稔, 吉田和永, 渡辺隆夫
"ニューラルネットワークによる話者適応"
音学講論, 1-6-16, 1989-03.

[Iso 90] Iso, K., Watanabe, T.
"Speaker-Independent Word Recognition Using a Neural Prodiction Model"
IEEE ICASSP90, pp. 441-444, 1990-04.

[Itakura 69] 板倉, 斉藤
"偏自己相関系数による音声分析合成系"
音学講論, 2-2-6, 1969-10.

[Itakura 71] 板倉, 斉藤
"統計的手法による音声スペクトル蜜度とホルマント周波数の推定"
信学会論文誌, 53A, 1971-01.

[Jelinek 85] Jelinek, F.
"The Development of an Experimental Discrete Dictation Recognizer"
IEEE, Proc. of IEEE 73(11) pp1616-1624, 1985-11.

[Kawabata 90] 川端豪
"k- 近傍内挿学習による音韻認識"
音学講論, 2-P-21, pp. 161-162, 1990-03.

[Kawabata 91] Kawabata, T., Hanazawa, T., Ito, A., Shikano, K.
"Japanese Phonetic Typewriter Using HMM Phone Recognition and Stochastic Phone-Sequence Modeling"

IEICE Trans. Vol. E74, No. 7, pp.1783-1787, 1991-07.

[Klatt 77] Klatt, D. H.
"Review of the ARPA Speech Understanding Project"
Journal of Acoustic Society of America, Vol. 62, No. 6, 1977-12.

[Kobayashi 85] 小林哲則
"音素識別部をもった連続音声認識に関する研究"
早稲田大学博士論文, 1985-03.

[Kohonene 88] Kohonen, T.
"The Neural Phonetic Typewriter"
IEEE Computer pp. 11-22, 1988-03.

[Lee 89] Lee, K., F., Hon, H. W., Raddy, R.
"An Overview of the SPHINX Speech Recognition System"
IEEE Trans. ASSP, 1990-01.

[Lesser 75] Lesser, V. R., Fennel, R. D., Erman, D. R., Reddy, D. R.
"Organization of the Hearsay II Speech Understanding System"
IEEE Trans. ASSP, Vol. ASSP-23, 1975-02.

[Lippmann 87] Lippmann, R. P.
"An Introduction to Computing with Neural Nets"
IEEE ASSP Mag., pp. 4-22, 1987-4.

[Lowerre 76] Lowerre, B. T.
"The HARPY Speech Recognition System"
Ph.D. thesis, Carnegie-Mellon Univ., 1976.

[Mercier 77] Mercier, G., Quinton, P., Vives, R.
"Man-Machine Dialogue with KEAL"
Centre National D'etudes des Telecommunications, Vol. 4, 1977.

[Minami 90] 南泰弘, 田村震一, 沢井秀文, 鹿野清宏
"入力層, 中間層におけるベクトルの近傍情報を利用した
TDNN 出力の平滑化"
音学講論, 1-3-18, pp.3 35-36, 1990-03.

[Minsky 69] Minsky, M. Papert, S.
"Parceptrons"
Cambridge, MA., MIT Press. 1969.

[Miyatake 90] Miyatake, M., Sawai, H., Minami, Y., Shikano, K.
"Integrated Training for Spotting Japanese Phoneme Using Large
Phonemic Time-Delay Neural Networks"
IEEE ICASSP90, S8.10, pp. 449-452, 1990-05.

[Mizoguchi 87] 溝口理一郎, 田中康宣, 福田尚行, 辻野克彦, 角所収
”連続音声認識エキスパートシステム - SPREX -”
信学論, J70-D, 6, pp. 1189-1198, 1987-06.

[Nakagawa 76] Nakagawa, S.
”A Machine Understanding System for Spoken Japanese Sentence”
Dr. Thesis, Kyoto Univ., 1976-10.

[Nakagawa 88] 中川聖一
”確率モデルによる音声認識”
電子情報通信学会, 1988.

[Nakamura 90] Nakamura, S., Shikano, K.
”A Comparative Study of Speactral Mapping for Speaker Adaptation”
IEEE ICASSP90, pp. 157-160, 1990-05.

[Nakata 77] 中田和男
”音声”
日本音響学会編, 音響工学講座 7, コロナ社, 1977.

[Nakata 78] 中田和男
”パターン認識とその応用”
現代自動制御双書 11, コロナ社, 1978.

[Niimi 77] 新美, 小林, 浅見, 三木
”「SPOKEN BASIC」の認識システム”
情報処理, 5, 1977-5.

[Niimi 79] 新美康永
”音声認識”
情報科学講座 E 19-3, 共立出版, 1979.

[Olson 56] Olson, H. F., Belar, H.
”Phonetic Typewriter”
Journal of Acoustic Society of America, 28, 1956-11.

[Potter 47] Potter, R. K., Kopp, G. A., Green, H. C.
”Visible Speech”
D. Van Nostrand Co. New York, 1947.

[Myers 81] Myers, C. S., Rabiner, L. R.
”A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition”
IEEE Trans. ASSP, Vol. ASSP-29, pp. 284-297, 1981-04.

[Rabiner 86] Rabiner. L. R, Juang, B. H.
”An Introduction to Hidden Markov Models”
IEEE ASSP Mag., pp.416, 1986-1.

[Rumelhart 86] Rumelhart, D. E., McClelland, J. L.
"Parallel Distributed Processing:
Explorations in the Micro Structure of Cognition"
MIT Press, 1986.

[Sakai 63] 坂井, 堂下
"会話音声識別装置"
信学誌, 46, 1963-11.

[Sakoe 71] 迫江, 千葉
"動的計画法を利用した音声の時間軸正規化に基づく連続単語認識"
音響学会誌, 27, 1971-09.

[Sakoe 89] Sakoe, H. Isotani, R. Yashida, K., Iso, K.
"Speaker-Independent Word Recognition Using Dynamic Programming Neural Networks"
IEEE ICASSP89, pp. 29-32, 1989-05.

[Sawai 91] Sawai, H., Minami, Y., Miyatake, M., Waibel. A. H., Shikano, K.
"Connectionist Approaches to Large Vocabulary Continuous Speech Recognition"
IEICE Trans. Vol. E74, No. 7, pp.1834-1843, 1991-07.

[Sawai 88] 沢井秀文, Waibel, A. H., 宮武正典, 鹿野清宏
"モジュール構成ニューラルネットワークのスケールアップによる音声認識"
通信学会, SP88-105, 1988-12.

[Sawai 91] Sawai, H., Nakamura, S.
"Time-Delay Neural Network Architectures for High-Performance Speaker-Independent Recognition"
ESCA Eurospeech'91, pp. 1011-1014, 1991-09.

[Shikano 81] 鹿野清宏
"会話音声自動認識システムに関する研究"
名古屋大学博士論文, 1981-02.

[Stern 86] Stern, P. E., Eskenazi, M.,Memmi, D.
"An Expert System for Speech Spectrogram Reading"
IEEE ICASSP86, pp. 1193-1196, 1986.

[Takami 90] Takami, J., Sagayama, S.
"A Pairwise Discriminat Approach to Robust Phoneme Recognition by Time-Delay Neural Networks"
IEEE ICASSP90, S2.13, pp. 89-92, 1990-05.

[Takeda 88] 武田一哉, 匂坂芳典, 片桐滋, 桑原尚夫
"音韻ラベルを持つ日本語音声データベースの構築"

音響学会誌, 44, 10, 1988.

[Tomita 86] Tomita, M.
"Efficient Parsing for Natural Language:
A Fast Algorithm for Practical Systems"
Kluwer Academic Publishers, Boston, 1986.

[Waibel 89] Waibel, A. H., Hanazawa, T., Hinton, G., Shikano, K., Lang, K. J.
"Phoneme Recognition Using Time-Delay Neural Netorks"
IEEE ICASSP89, 1989-05.

[Waibel 90] Waibel, A., Lee, K. F.
"Readings in Speech Recognition"
Morgan Kaufmann Publishers Inc., California, 1990.

[Winograd 72] Winograd, T.
"Understanding Natural Language"
Academic Press, 1972.

[Woods 70] Woods, W. A.
"Transition Network Grammars for Natural Language Analysis"
Commun. ACM, 13, 1970.

[Zadeh 65] Zadeh, L. A.
"Fuzzy Set"
Inform. Contro, 8, pp 338-353, 1965.

[Zue 79] Zue, V. W., Cole, R.A.
"Experiments on Spectrogram Reading"
IEEE ICASSP79, pp.116-119, 1979.

[Zue 86] Zue, V. W., Lamel, F.
"An Expert Spectrogram Reader:
A Knowledge-Based Approach to Speech Recognition"
IEEE ICASSP86, pp. 1187-1200, 1986.

[Zue 90] Zue, V, Glass, J., Phillips, M., Seneff, S.
"The SUMMIT Speech Recognition System:
- Phonological Modelling and Lexical Access"
IEEE ICASSP90, pp.49-52, 1990-04.

Appendix A

Typical Segmentation Knowledge

This appendix shows some consonant segmentation knowledge in the proposed expert system. A typical consonant spectrogram and its segmentation knowledge is shown using consonants appearing in a vowel-consonant-vowel phoneme context. In practice, more precise and various kinds of knowledge, considering several phoneme contexts, are incorporated to realize the total system,

Unvoiced-stop

Figure A-1 shows a typical spectrogram with its automatic segmentation result of unvoiced-stop /k/ at utterance initial and /ch/ between vowels. The utterance is /kachi/.

- find utterance initial or closure where 0-6000Hz power < silence threshold.
- find burst at the utterance initial or at left of closure.
- find the power increasing point of 0-500Hz power toward next vowel as the end boundary.
- find the power increasing point of 0-500Hz power toward previous vowel as the start boundary.
- evaluate the vowel possibility using 0-500Hz power of both side of the boundary

Unvoiced-fricative

Figure A-2 shows the typical spectrogram of unvoiced-stop /s/ which appears between vowels with its automatic segmentation result. The utterance is /asa/.

- find region where 4000-6000Hz power > fricative threshold.
- find power decreasing point of 4000-6000Hz power toward next vowel as the end boundary.
- find power decreasing point of 4000-6000Hz power toward previous vowel as the start boundary.

- evaluate the vowel possibility using 0-500Hz power of both side of the boundary.

Voiced-stop

Figure A-3 shows the typical spectrogram of unvoiced-stop /b/ which appears between vowels with its automatic segmentation result. The utterance is /oba/.

- find closure where 1000-6000Hz power < voiced-closure threshold.
- find burst to left of closure.
- find the power increasing point of 0-500Hz power toward the next vowel as the end boundary.
- find the power increasing point of 0-500Hz power toward the previous vowel as the start boundary.
- evaluate the vowel possibility using 0-500Hz power of both sides of the boundary.

Voiced-fricative

Figure A-4 shows the typical spectrogram of unvoiced-stop /z/ which appears between vowels with its automatic segmentation result. The utterance is /kaze/.

- find the region where both 4000-6000Hz power > fricative threshold and 0-500Hz power > voicing threshold.
- find the power increasing point of 0-500Hz power toward next vowel.
- find power decreasing point of 4000-6000Hz power toward the next vowel.
- select the earlier time as the start boundary.
- find the power increasing point of 0-500Hz power toward the previous vowel.
- find the power decreasing point of 4000-6000Hz power toward the previous vowel.
- select the later time as the end boundary.
- evaluate the vowel possibility using 0-500Hz power of both sides of the boundary.

Nasal

Figure A-5 shows a typical spectrogram of unvoiced-stop /n/ which appears between vowels with its automatic segmentation result. The utterance is /ana/.

- find the power dip using the 4000-6000Hz / 0-500Hz power ratio < nasal dip threshold.
- find the spectral change point of 0-6000Hz toward the next vowel as the end boundary.
- find the spectral change point of 0-6000Hz toward the previous vowel as the start boundary.

- evaluate nasal possibility of this region using the 0-500Hz power and 500-1000Hz power around the boundary.
- evaluate the vowel possibility using 0-500Hz power of both sides of the boundary.

Liquid

Figure A-6 shows a typical spectrogram of liquid /r/ which appears between vowels with its automatic segmentation result. The utterance is /kara/.

- find the region where the 2000-4000Hz power decreases and after the 2000-4000Hz power suddenly increases within 30ms.
- evaluate the liquid possibility using the sum of decreasing and increasing values.
- the decreasing and the increasing points are detected as the boundaries.
- evaluate the vowel possibility using 0-500Hz power of both sides of the boundary.

Glottal

Figure A-7 shows a typical spectrogram of glottal /h/ which appears at utterance initial and between vowels with its automatic segmentation result. The utterance is /haha/.

- find the power region where the 0-1000Hz power < glottal threshold.
- evaluate the glottal possibility around this region using the 1000-5000Hz power > glottal threshold.
- find the power increasing point of 0-500Hz power toward the next vowel as the end boundary.
- find the utterance initial or find the power increasing point of 0-500Hz power toward the previous vowel as the start boundary.
- evaluate the vowel possibility using 0-500Hz power of both sides of the boundary.

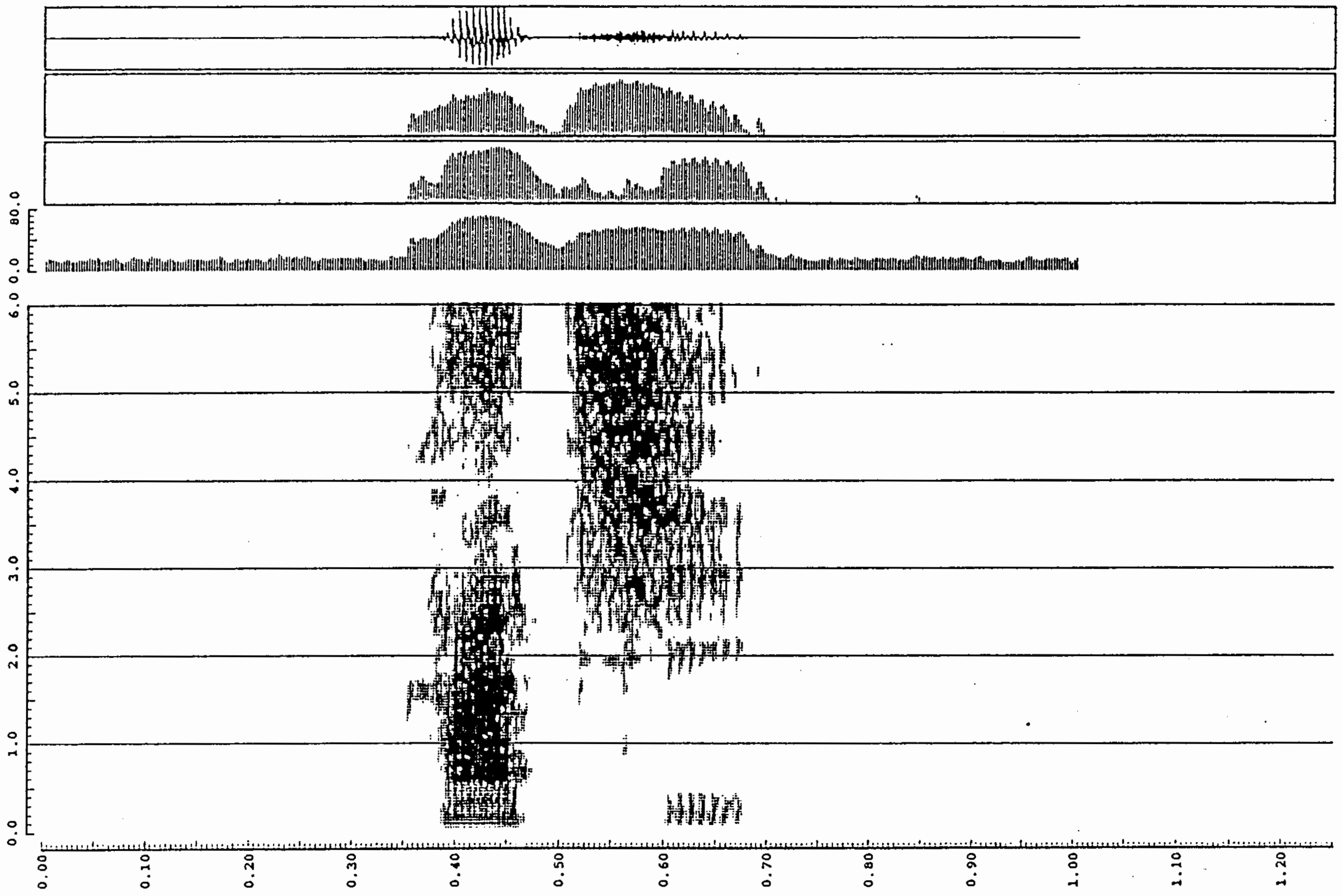
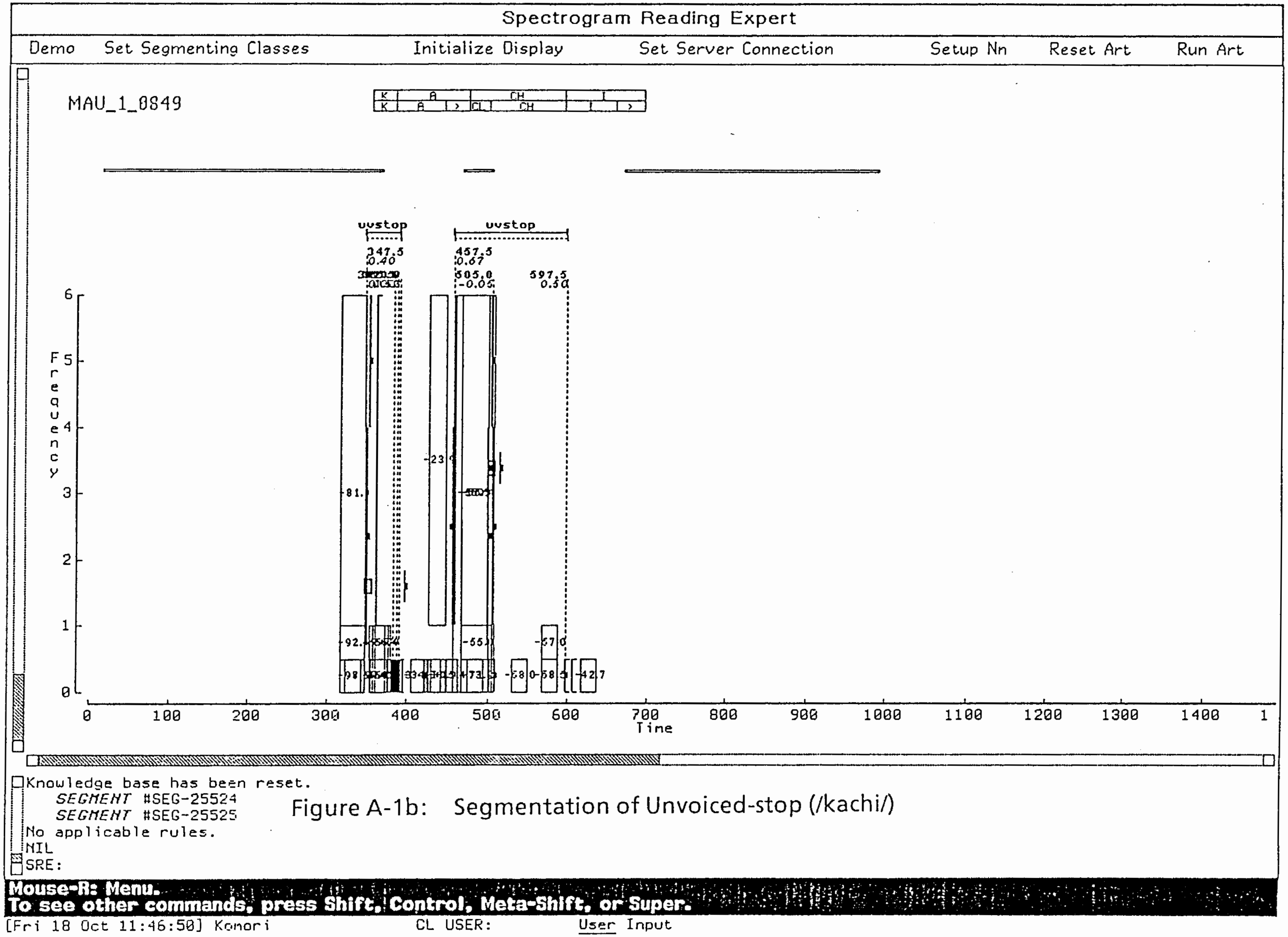


Figure A-1a: Typical Spectrogram of Unvoiced-stop (/kachi/)



```
Knowledge base has been reset.
    SEGMENT #SEG-25524
    SEGMENT #SEG-25525
No applicable rules.
NIL
SRE:
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 18 Oct 11:46:50] Komori          CL USER:          User Input
```

Figure A-1b: Segmentation of Unvoiced-stop (/kachi/)

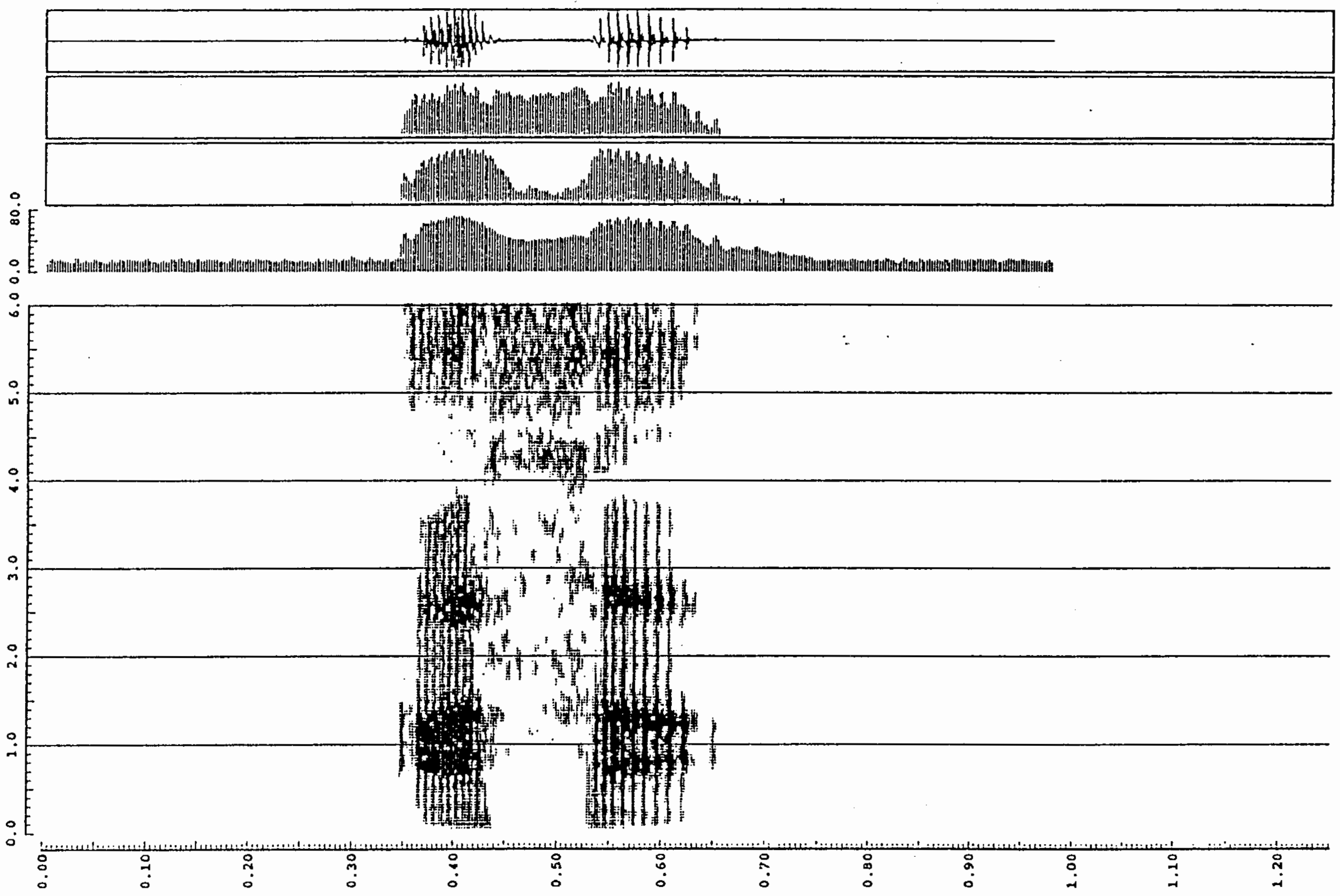
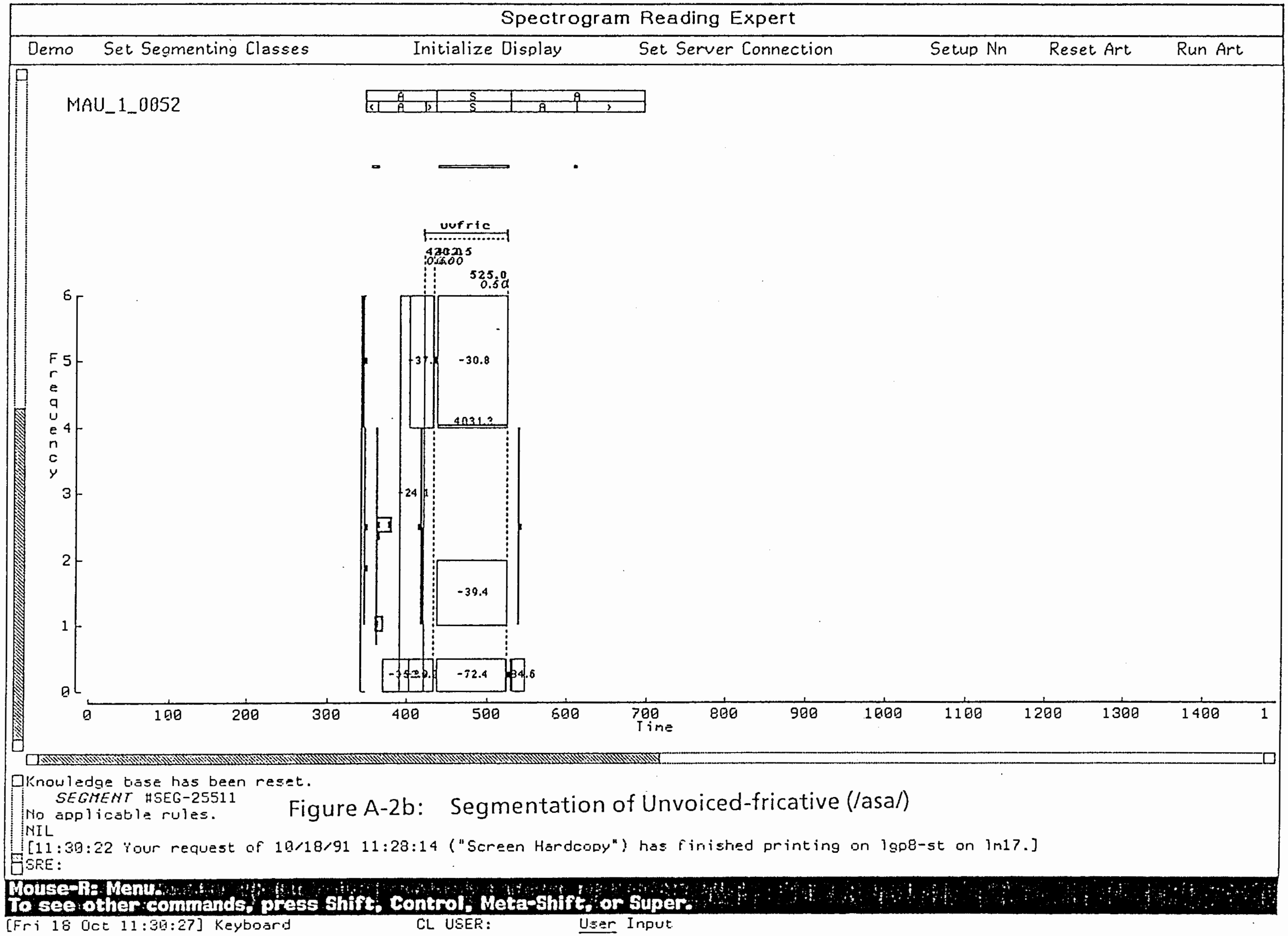


Figure A-2a: Typical Spectrogram of Unvoiced-fricative (/asa/)



```
Knowledge base has been reset.
   SEGMENT #SEG-25511
No applicable rules.
NIL
[11:30:22 Your request of 10/18/91 11:28:14 ("Screen Hardcopy") has finished printing on lgp8-st on ln17.]
SRE:
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 18 Oct 11:30:27] Keyboard          CL USER:          User Input
```

Figure A-2b: Segmentation of Unvoiced-fricative (/asa/)

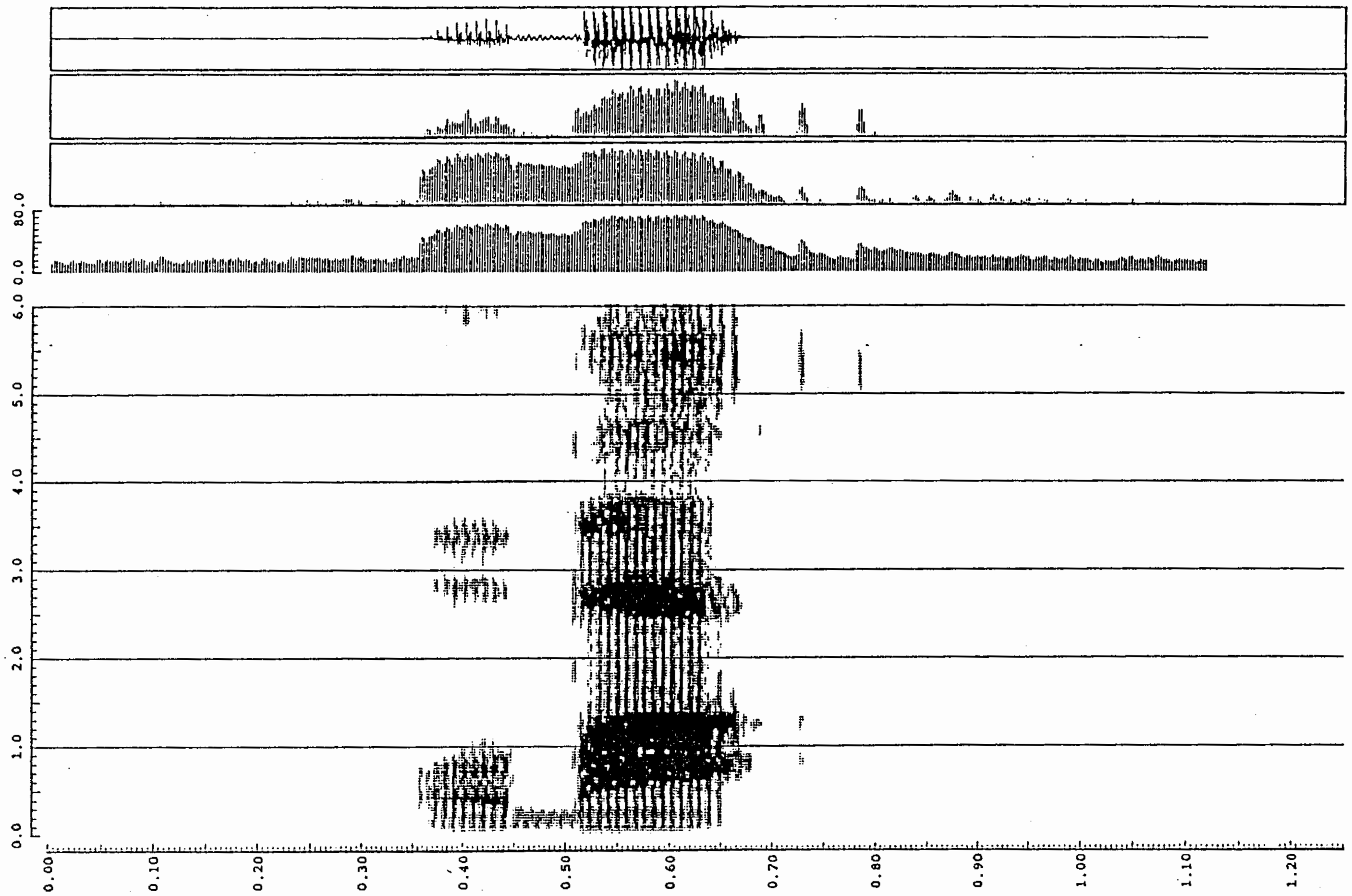
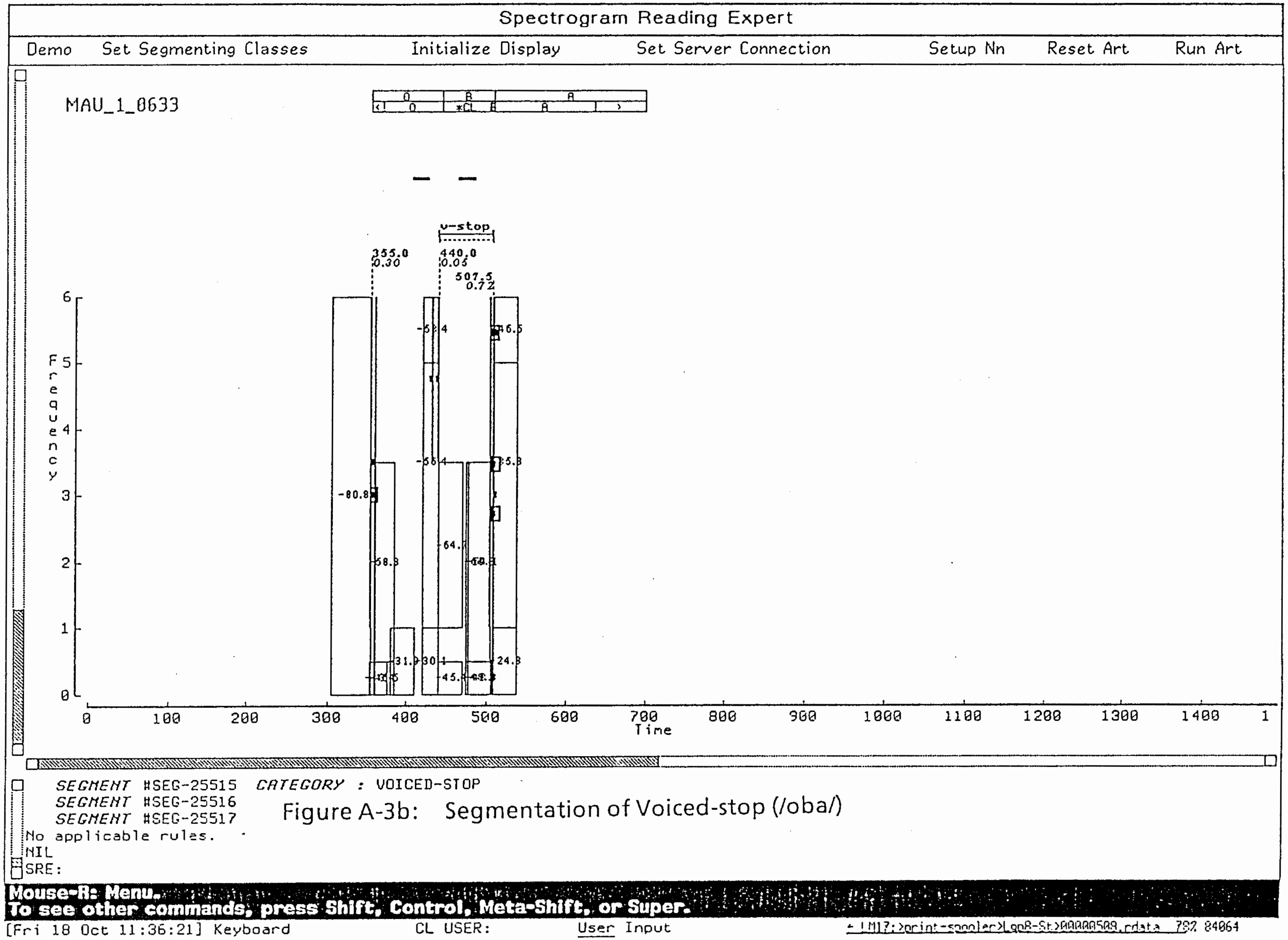


Figure A-3a: Typical Spectrogram of Voiced-stop (/oba/)



```
SEGMENT #SEG-25515  CATEGORY : VOICED-STOP
SEGMENT #SEG-25516
SEGMENT #SEG-25517
No applicable rules.
NIL
SRE:
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 18 Oct 11:36:21] Keyboard          CL USER:        User Input          ← LMI7:>print-spooler>Lgp8-St>00000509.rdata  792 84064
```

Figure A-3b: Segmentation of Voiced-stop (/oba/)

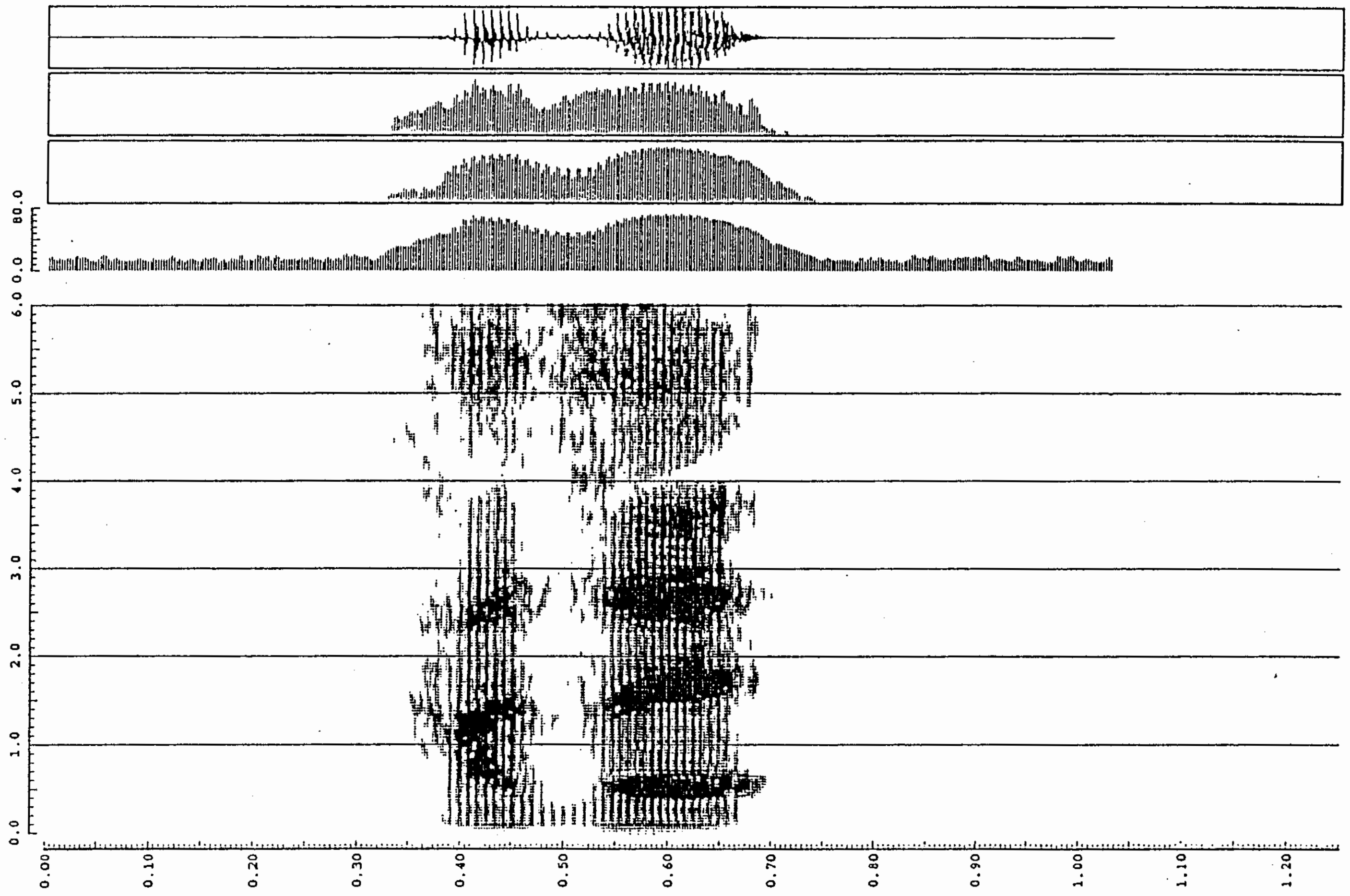
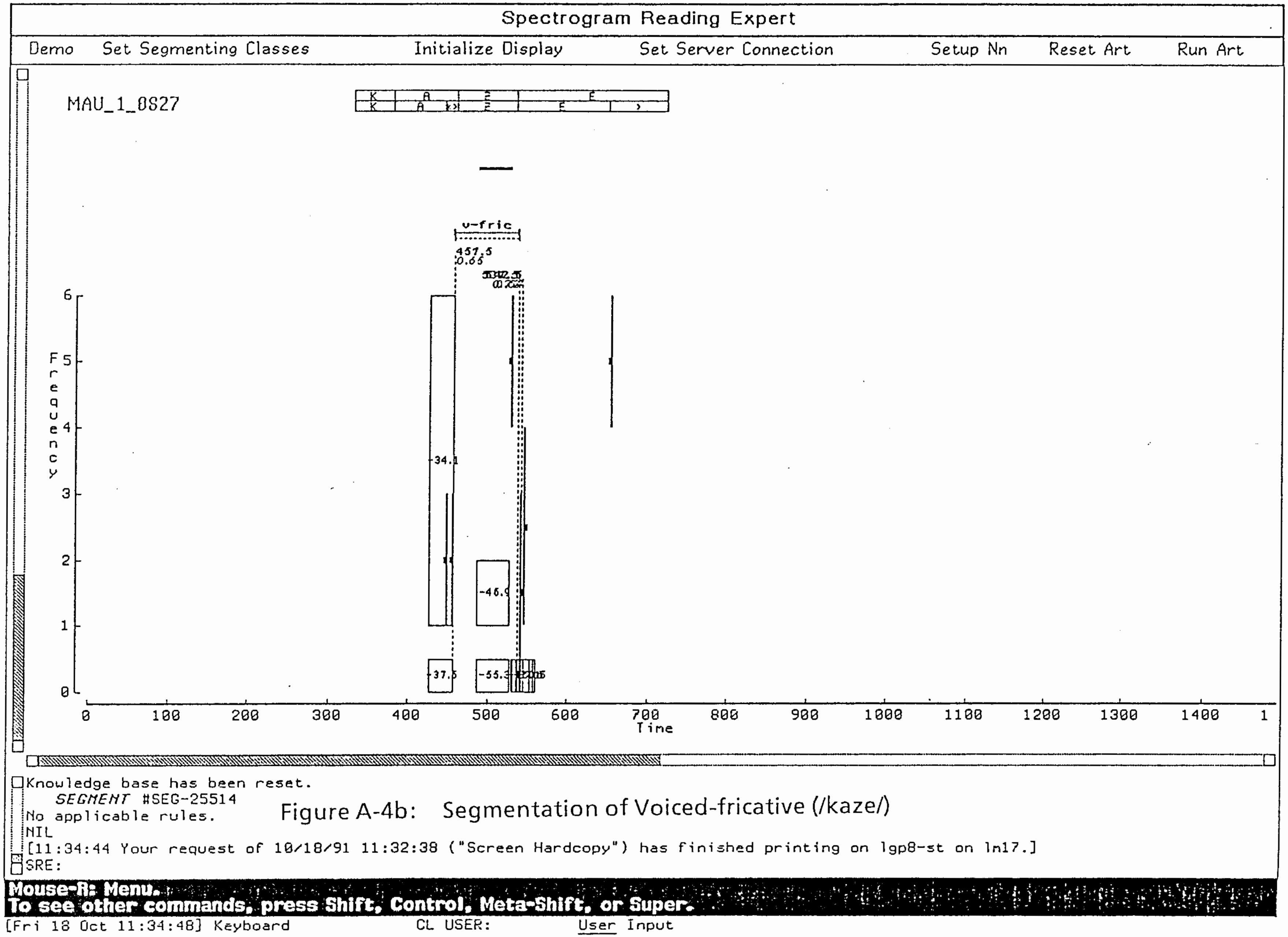


Figure A-4a: Typical Spectrogram of Voiced-fricative (/kaze/)



```
Knowledge base has been reset.
    SEGMENT #SEG-25514
No applicable rules.
NIL
[11:34:44 Your request of 10/18/91 11:32:38 ("Screen Hardcopy") has finished printing on lgp8-st on ln17.]
SRE:
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 18 Oct 11:34:48] Keyboard          CL USER:          User Input
```

Figure A-4b: Segmentation of Voiced-fricative (/kaze/)

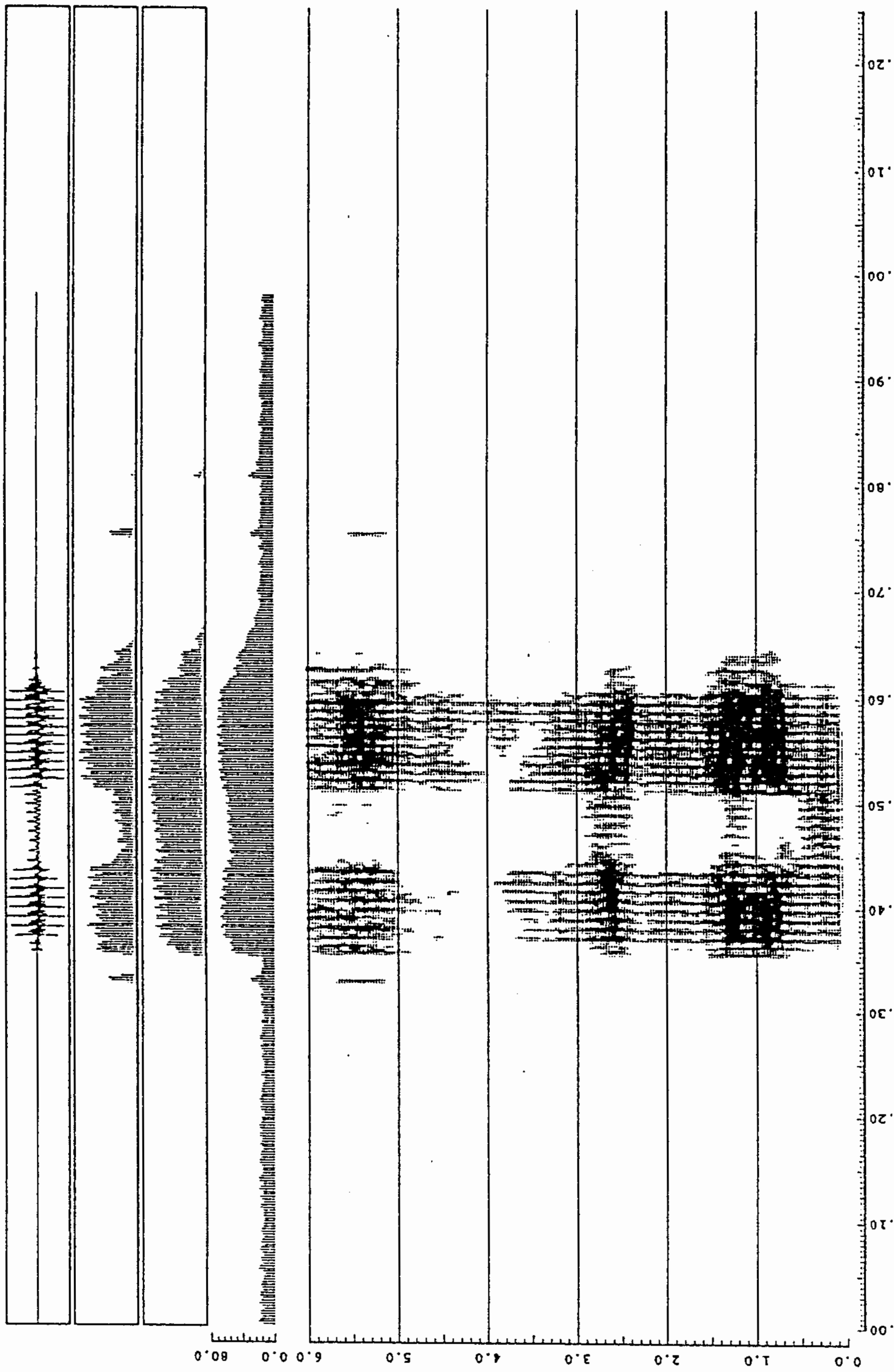
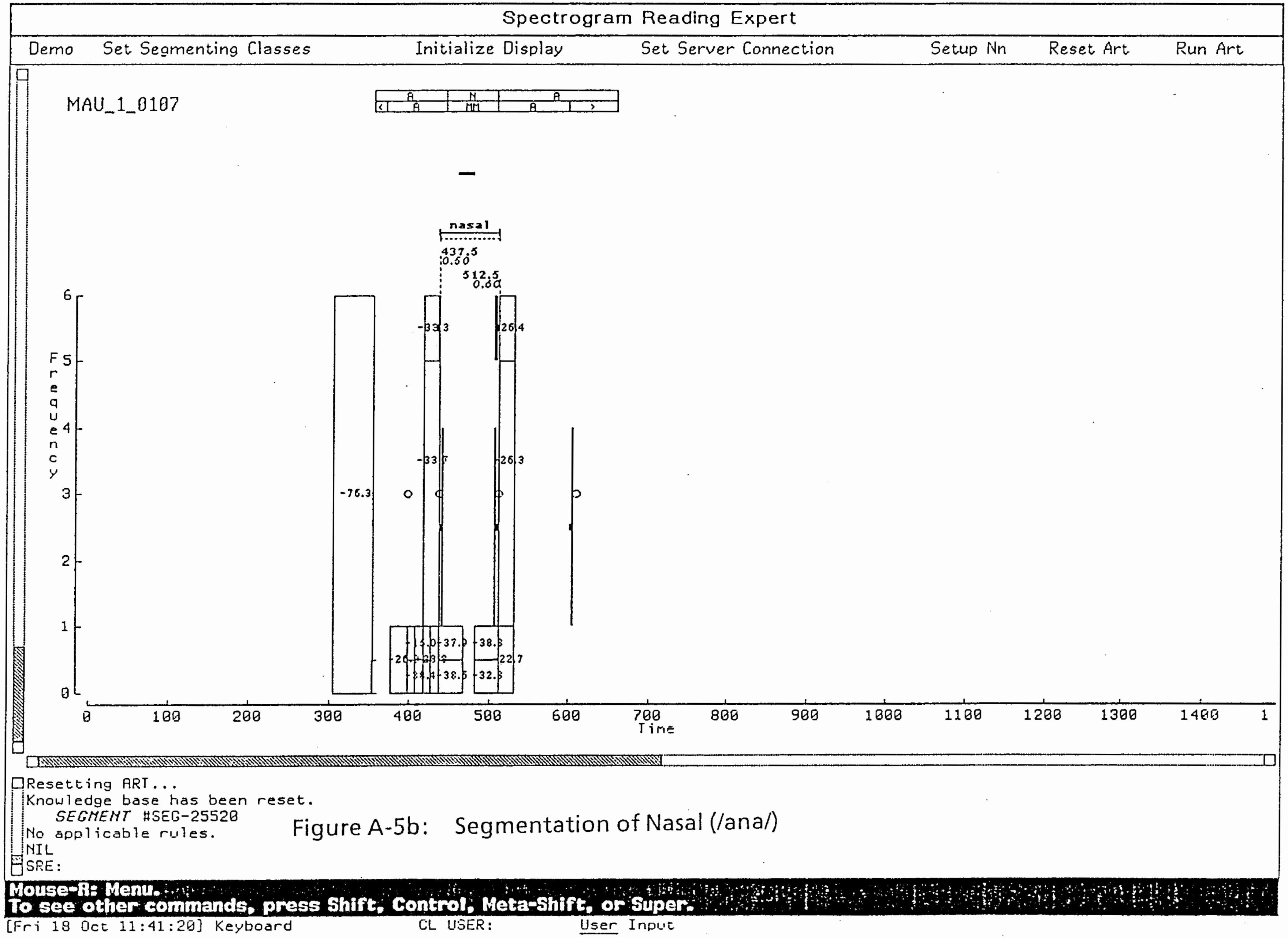


Figure A-5a: Typical Spectrogram of Nasal (/ana/)



```
Resetting ART...
Knowledge base has been reset.
   SEGMENT #SEG-25520
No applicable rules.
NIL
SRE:
```

```
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 18 Oct 11:41:20] Keyboard          CL USER:          User Input
```

Figure A-5b: Segmentation of Nasal (/ana/)

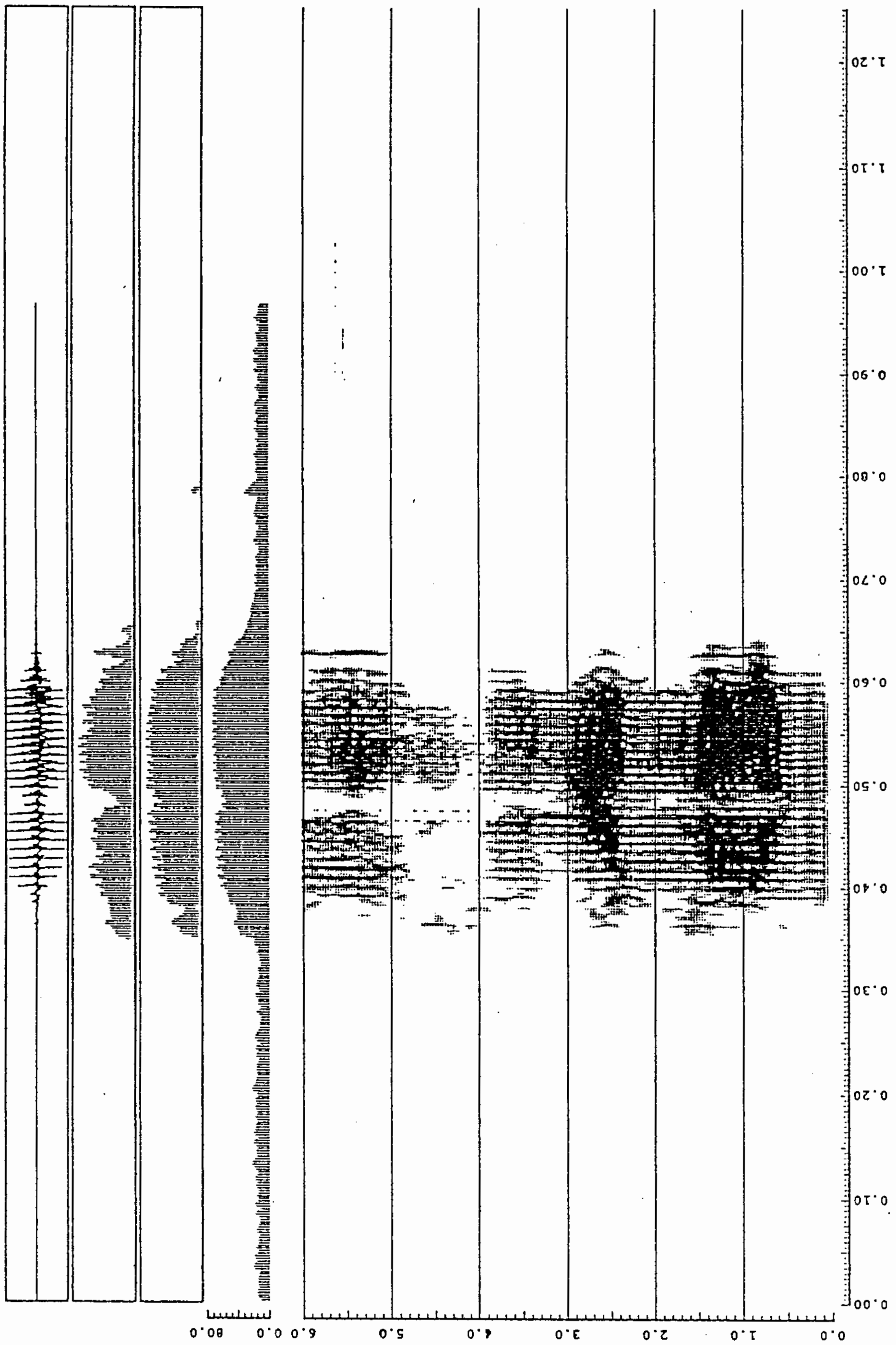
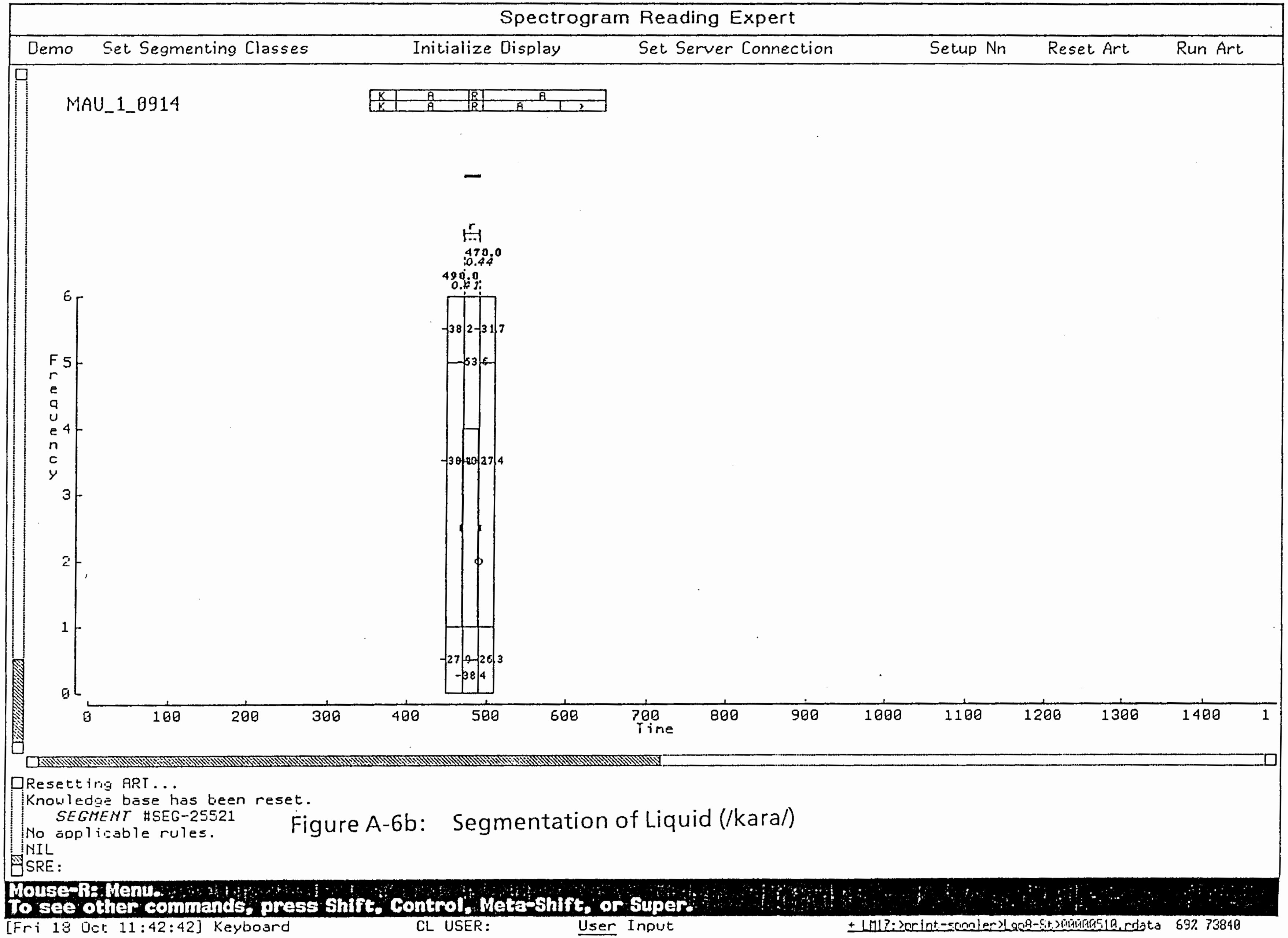


Figure A-6a: Typical Spectrogram of Liquid (/kara/)



```
Resetting ART...
Knowledge base has been reset.
   SEGMENT #SEG-25521
No applicable rules.
NIL
SRE:
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 18 Oct 11:42:42] Keyboard              CL USER:        User Input              + LM17:>print-spooler>Lgp8-St>00000510.rdata  69% 73840
```

Figure A-6b: Segmentation of Liquid (/kara/)

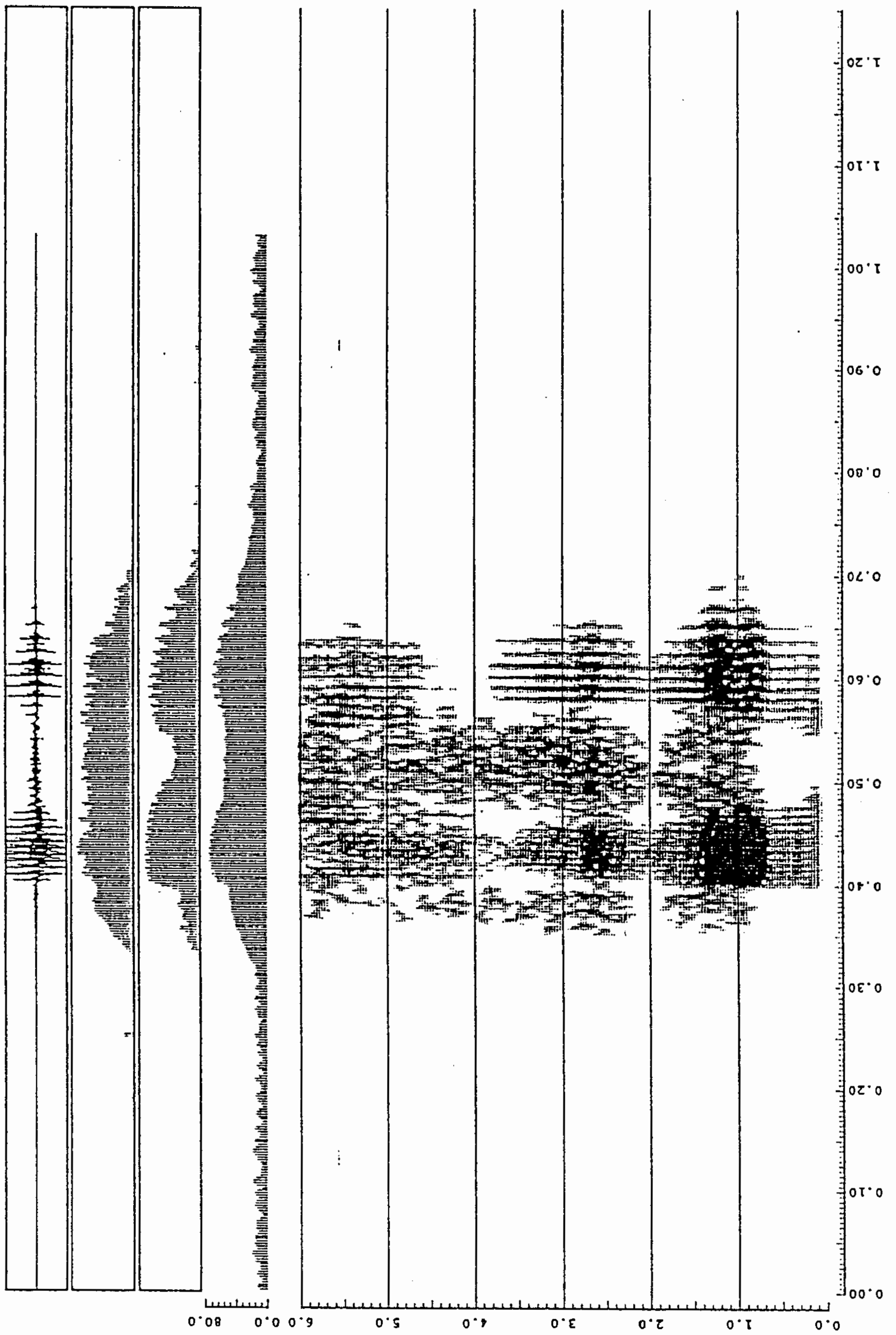
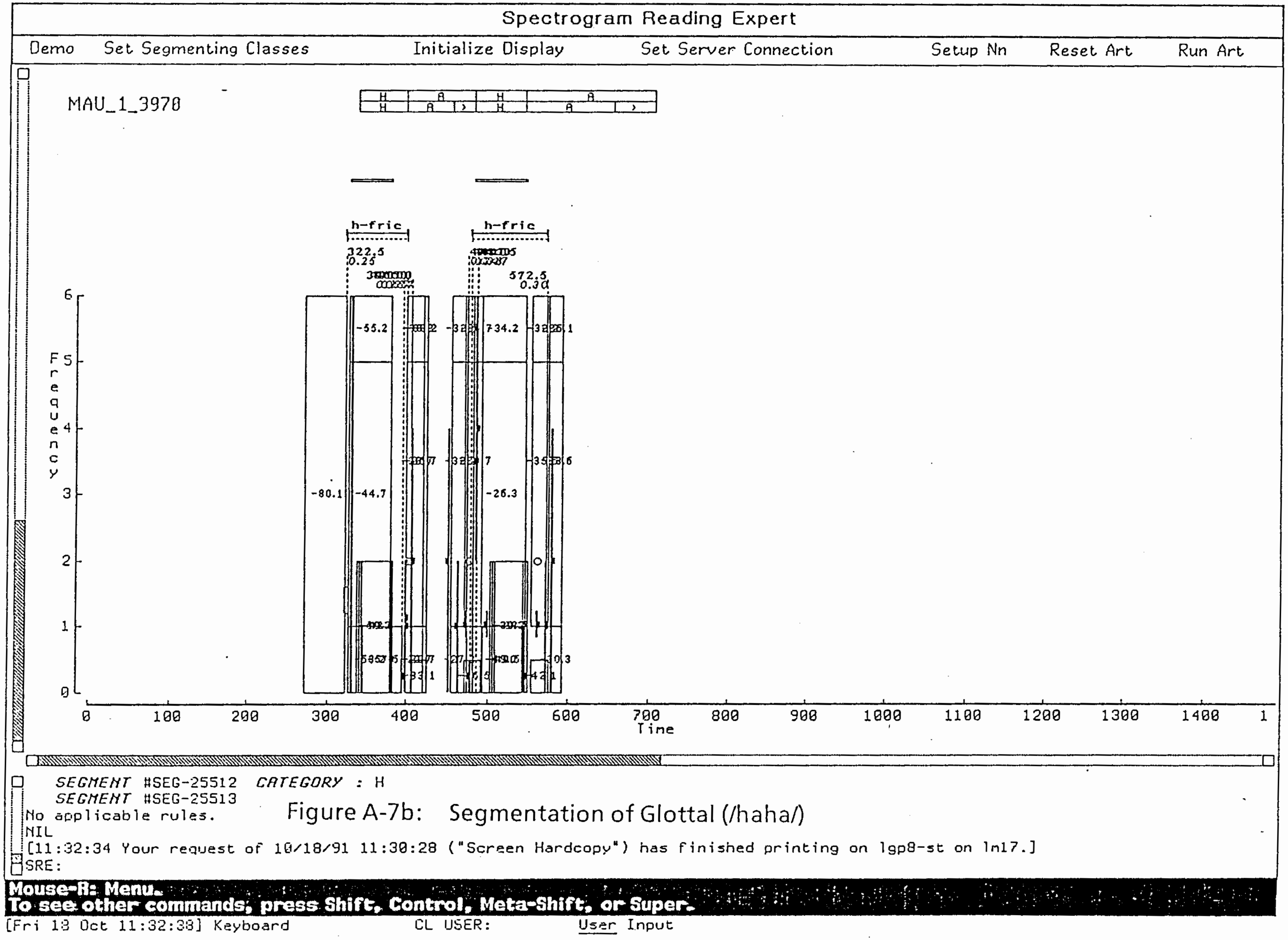


Figure A-7a: Typical Spectrogram of Glottal (/haha/)



SEGMENT #SEG-25512 CATEGORY : H
SEGMENT #SEG-25513
No applicable rules.
NIL
[11:32:34 Your request of 10/18/91 11:30:28 ("Screen Hardcopy") has finished printing on lgp8-st on ln17.]
SRE:

Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.

[Fri 18 Oct 11:32:38] Keyboard CL USER: User Input

Figure A-7b: Segmentation of Glottal (/haha/)

Appendix B

Multiple Speaker Speech Segmentation

Figure B-1(a-f) shows the segmentation results of utterance /subete/ for multiple speakers (MAU, MHT, MNM, MTK, MMY, MXM).

Figure B-1a: /subete/ of speaker MAU
(spectrogram & recognition).

Figure B-1b: /subete/ of speaker MHT (segmentation).

Figure B-1c: /subete/ of speaker MNM (segmentation).

Figure B-1d: /subete/ of speaker MTK (segmentation).

Figure B-1e: /subete/ of speaker MMY (segmentation).

Figure B-1f: /subete/ of speaker MXM (segmentation).

Figure B-2(a-f) shows the segmentation results of utterance /misebirakasu/ for multiple speakers (MAU, MHT, MNM, MTK, MMY, MXM).

Figure B-2a: /misebirakasu/ of speaker MAU
(spectrogram & recognition).

Figure B-2b: /misebirakasu/ of speaker MHT (segmentation).

Figure B-2c: /misebirakasu/ of speaker MNM (segmentation).

Figure B-2d: /misebirakasu/ of speaker MTK (segmentation).

Figure B-2e: /misebirakasu/ of speaker MMY (segmentation).

Figure B-2f: /misebirakasu/ of speaker MXM (segmentation).

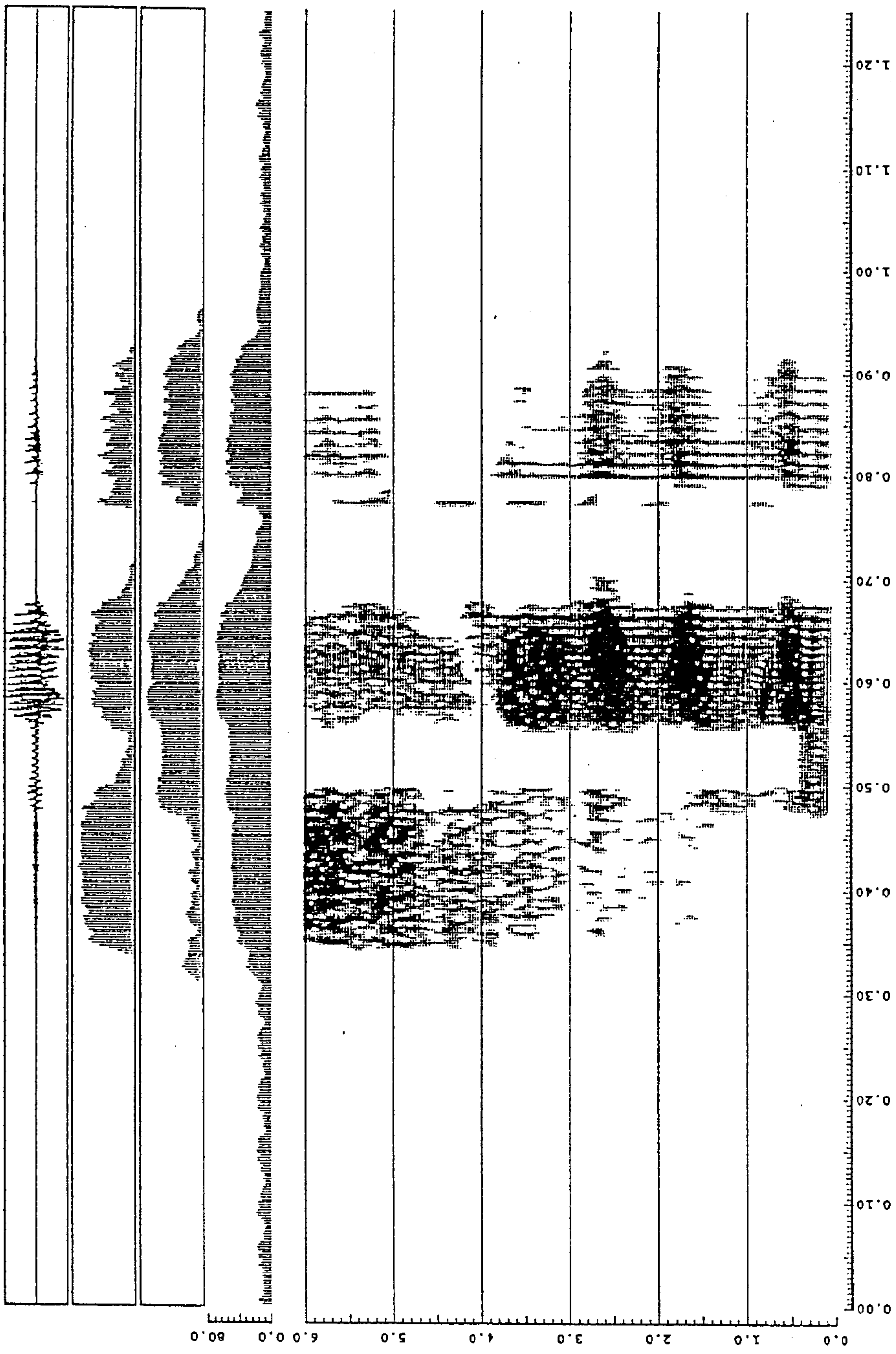
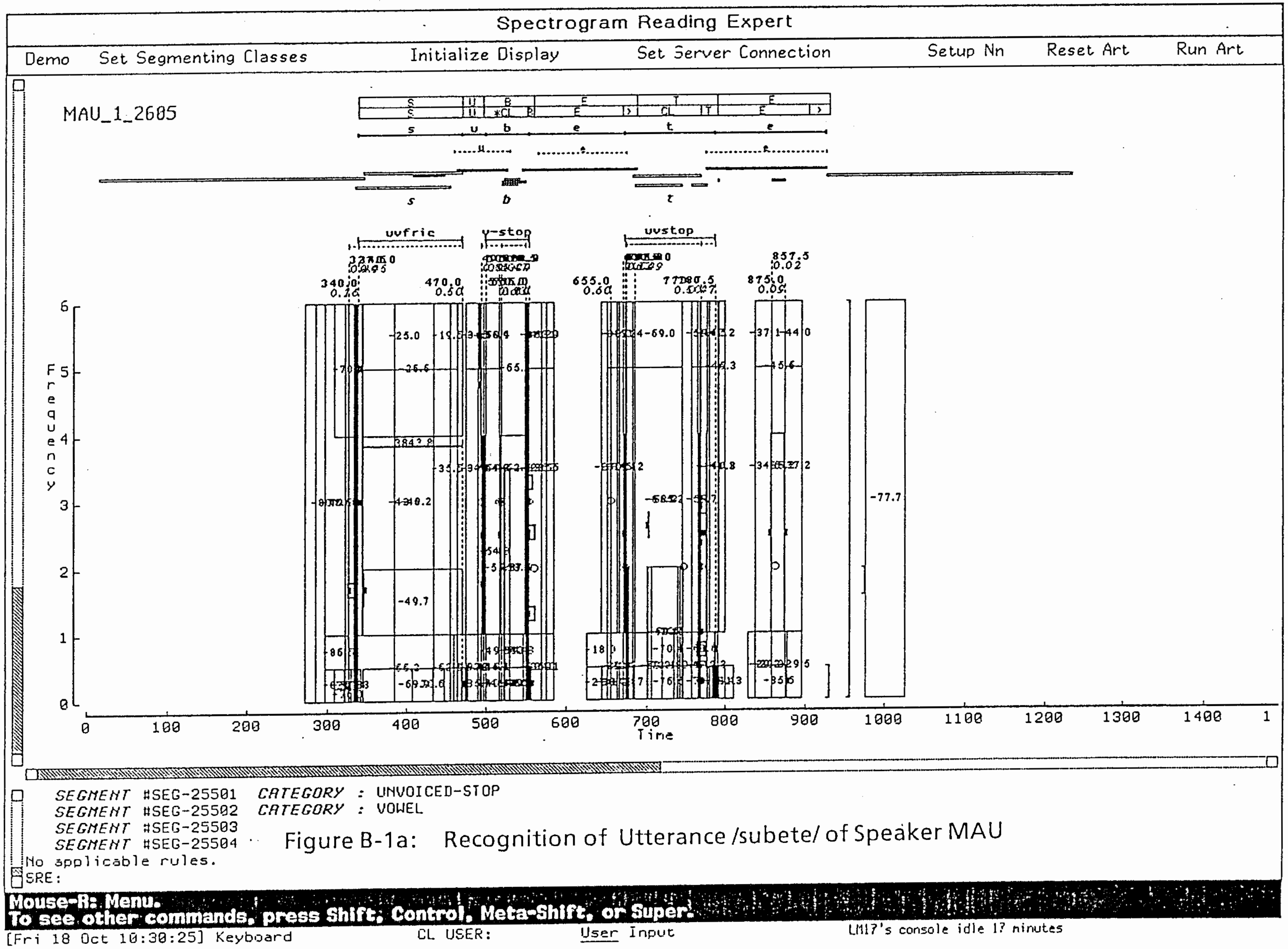
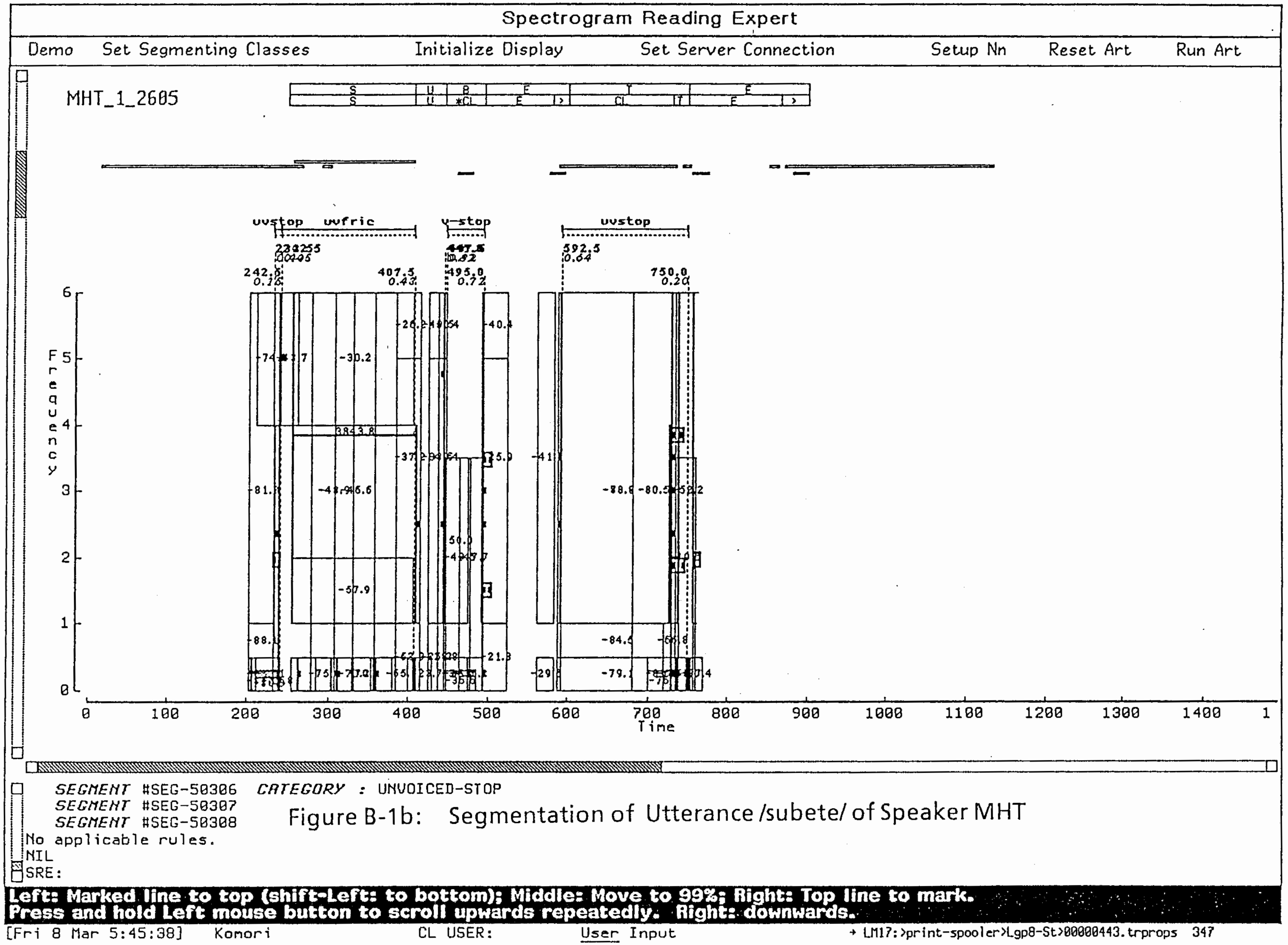


Figure B-1: Spectrogram of Utterance /subete/ of Speaker MAU



```
SEGMENT #SEG-25501  CATEGORY : UNVOICED-STOP
SEGMENT #SEG-25502  CATEGORY : VOWEL
SEGMENT #SEG-25503
SEGMENT #SEG-25504
No applicable rules.
SRE:
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 18 Oct 10:30:25] Keyboard          CL USER:     User Input              LM17's console idle 17 minutes
```

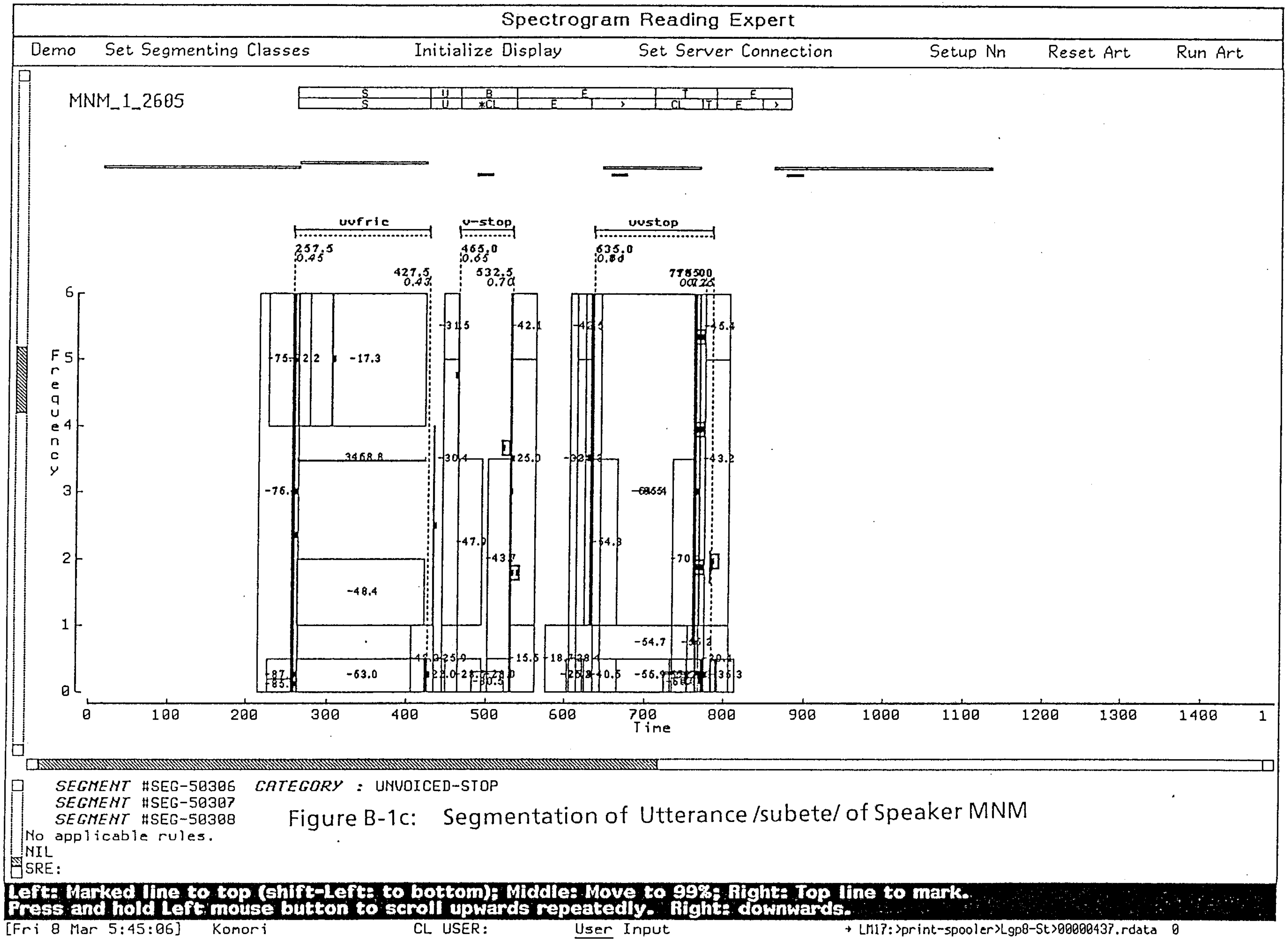
Figure B-1a: Recognition of Utterance /subete/ of Speaker MAU



```
SEGMENT #SEG-50306  CATEGORY : UNVOICED-STOP
SEGMENT #SEG-50307
SEGMENT #SEG-50308
No applicable rules.
NIL
SRE:
```

Figure B-1b: Segmentation of Utterance /subete/ of Speaker MHT

```
Left: Marked line to top (shift-Left: to bottom); Middle: Move to 99%; Right: Top line to mark.
Press and hold Left mouse button to scroll upwards repeatedly.  Right: downwards.
[Fri 8 Mar 5:45:38]    Komori          CL USER:         User Input              + LM17:>print-spooler>Lgp8-St>00000443.trprops  347
```

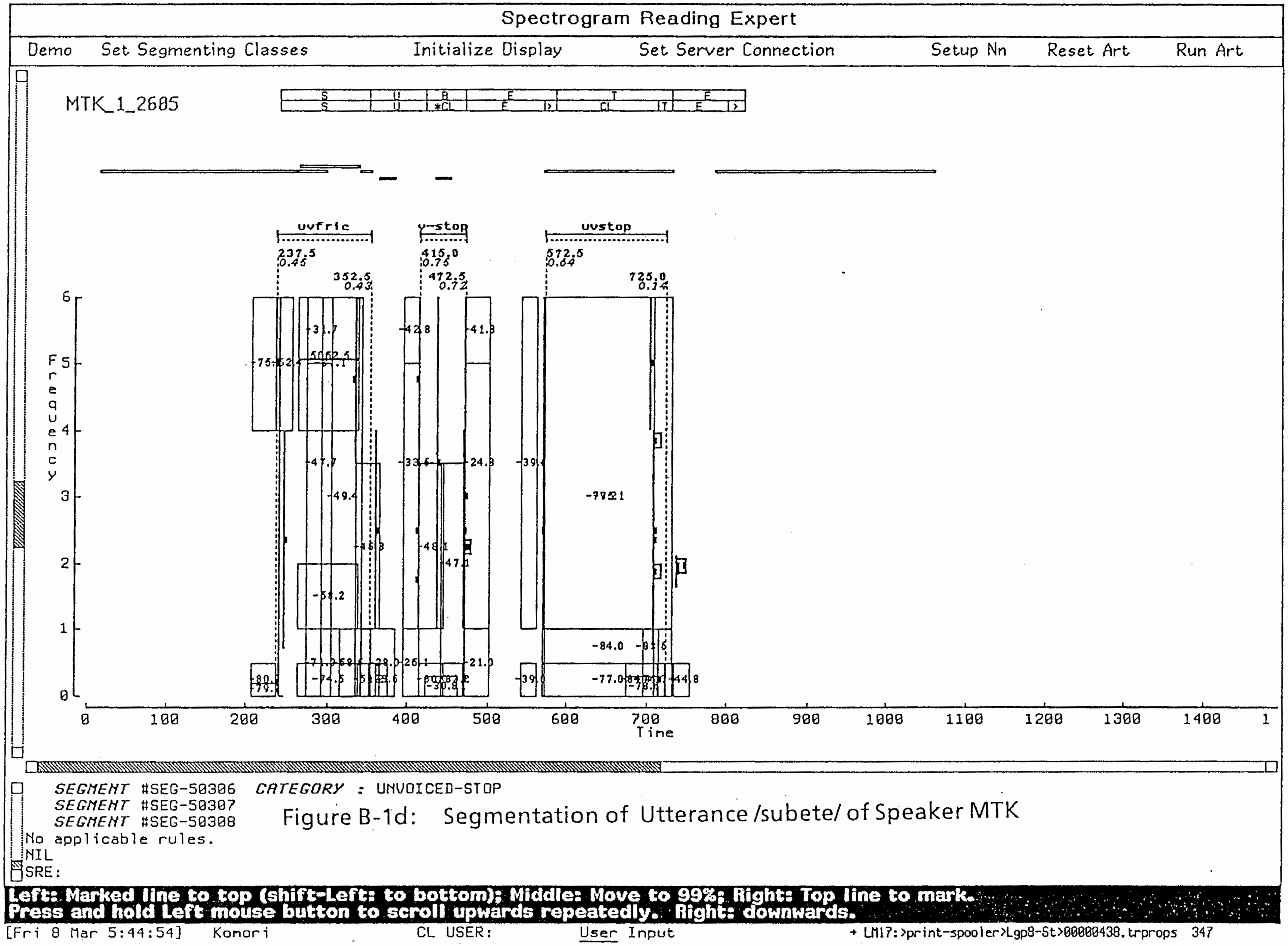


```
SEGMENT #SEG-50306  CATEGORY : UNVOICED-STOP
SEGMENT #SEG-50307
SEGMENT #SEG-50308
No applicable rules.
NIL
SRE:
```

Left: Marked line to top (shift-Left: to bottom); Middle: Move to 99%; Right: Top line to mark.
Press and hold Left mouse button to scroll upwards repeatedly. Right: downwards.

```
[Fri 8 Mar 5:45:06]    Konori          CL USER:       User Input              → LM17:>print-spooler>Lgp8-St>00000437.rdata  0
```

Figure B-1c: Segmentation of Utterance /subete/ of Speaker MNM



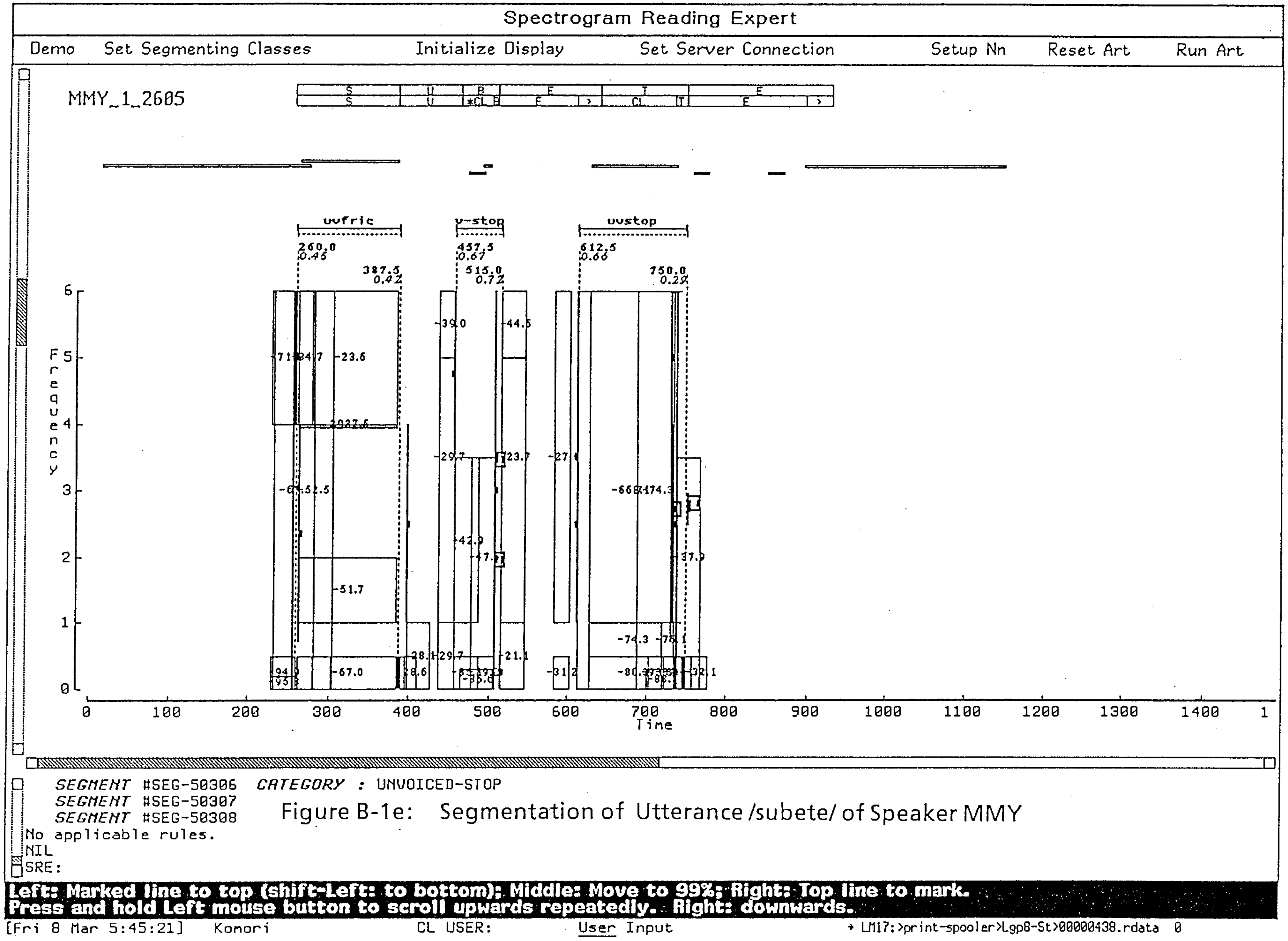
SEGMENT #SEG-50306 CATEGORY : UNVOICED-STOP
SEGMENT #SEG-50307
SEGMENT #SEG-50308

No applicable rules.
NIL
SRE:

Left: Marked line to top (shift-Left: to bottom); Middle: Move to 99%; Right: Top line to mark.
Press and hold Left mouse button to scroll upwards repeatedly. Right: downwards.

[Fri 8 Mar 5:44:54] Konori CL USER: User Input → LM17:>print-spooler>Lgp8-St>00000438.trprops 347

Figure B-1d: Segmentation of Utterance /subete/ of Speaker MTK

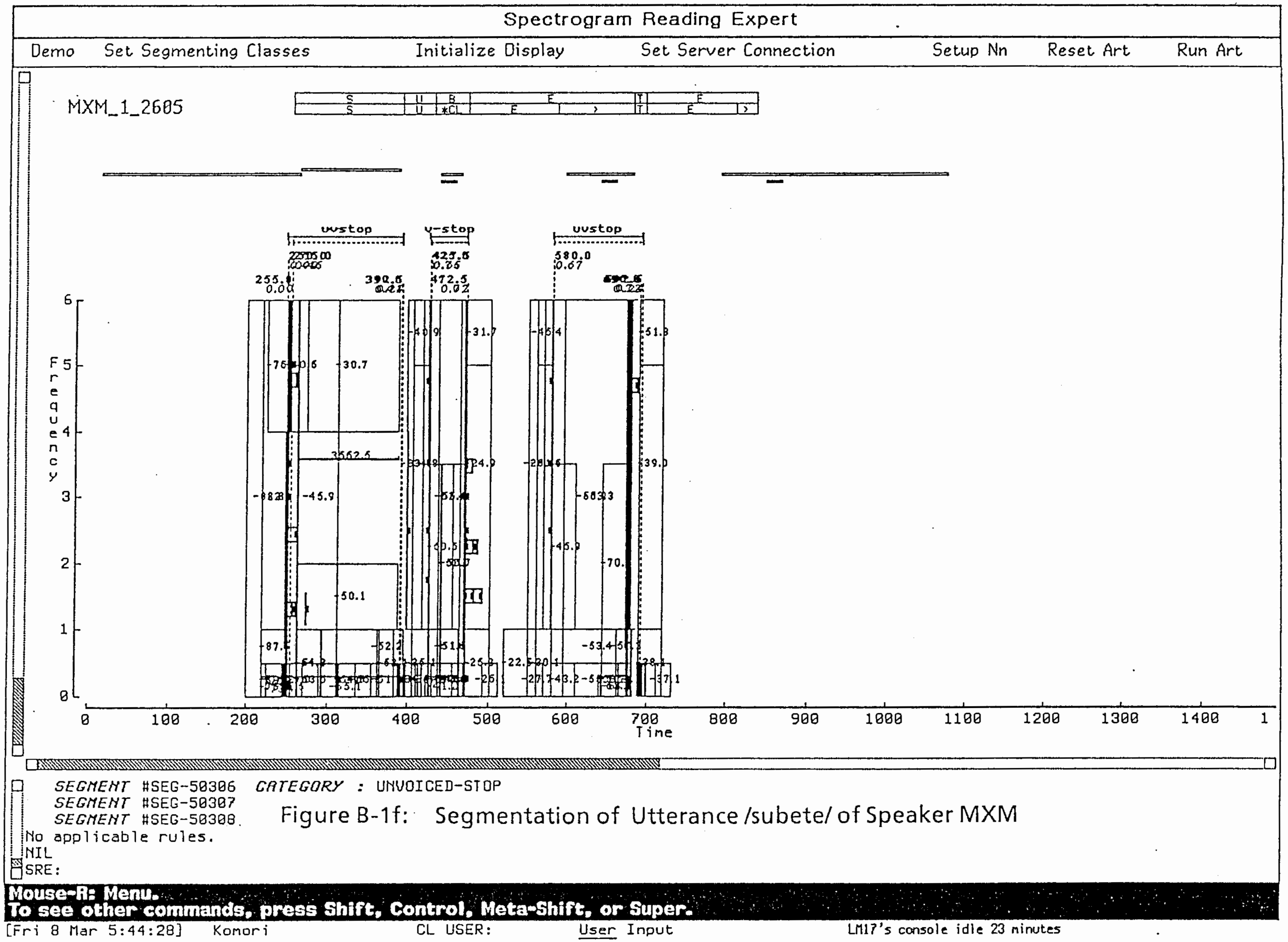


```
SEGMENT #SEG-50306  CATEGORY : UNVOICED-STOP
SEGMENT #SEG-50307
SEGMENT #SEG-50308
No applicable rules.
NIL
SRE:
```

Figure B-1e: Segmentation of Utterance /subete/ of Speaker MMY

Left: Marked line to top (shift-Left: to bottom); Middle: Move to 99%; Right: Top line to mark.
Press and hold Left mouse button to scroll upwards repeatedly. Right: downwards.

```
[Fri 8 Mar 5:45:21]    Komori              CL USER:          User Input                  → LM17:>print-spooler>Lgp8-St>00000438.rdata  0
```



```
SEGMENT #SEG-50306  CATEGORY : UNVOICED-STOP
SEGMENT #SEG-50307
SEGMENT #SEG-50308
No applicable rules.
NIL
SRE:
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 8 Mar 5:44:28]    Komori          CL USER:        User Input            LM17's console idle 23 minutes
```

Figure B-1f: Segmentation of Utterance /subete/ of Speaker MXM

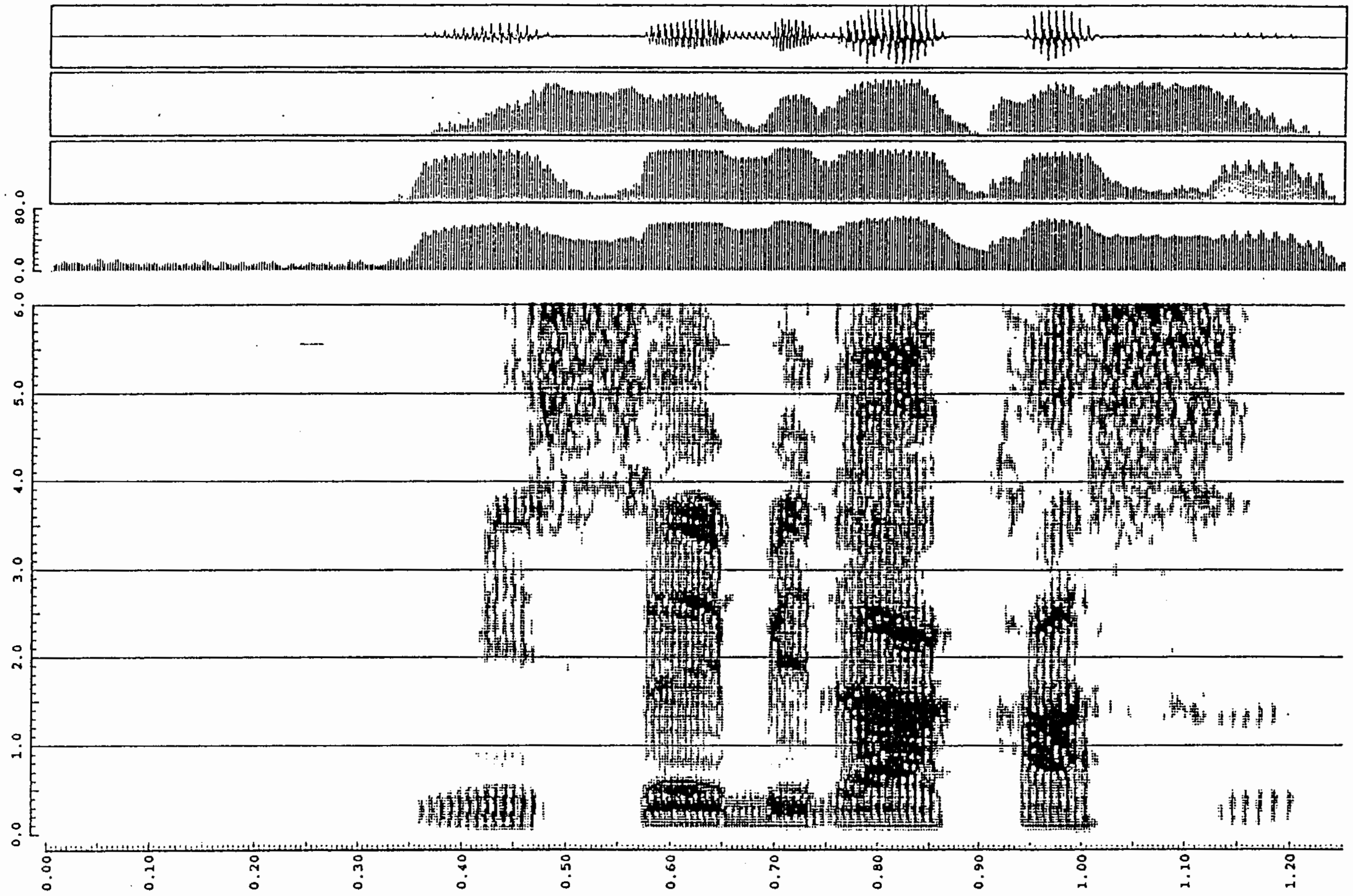
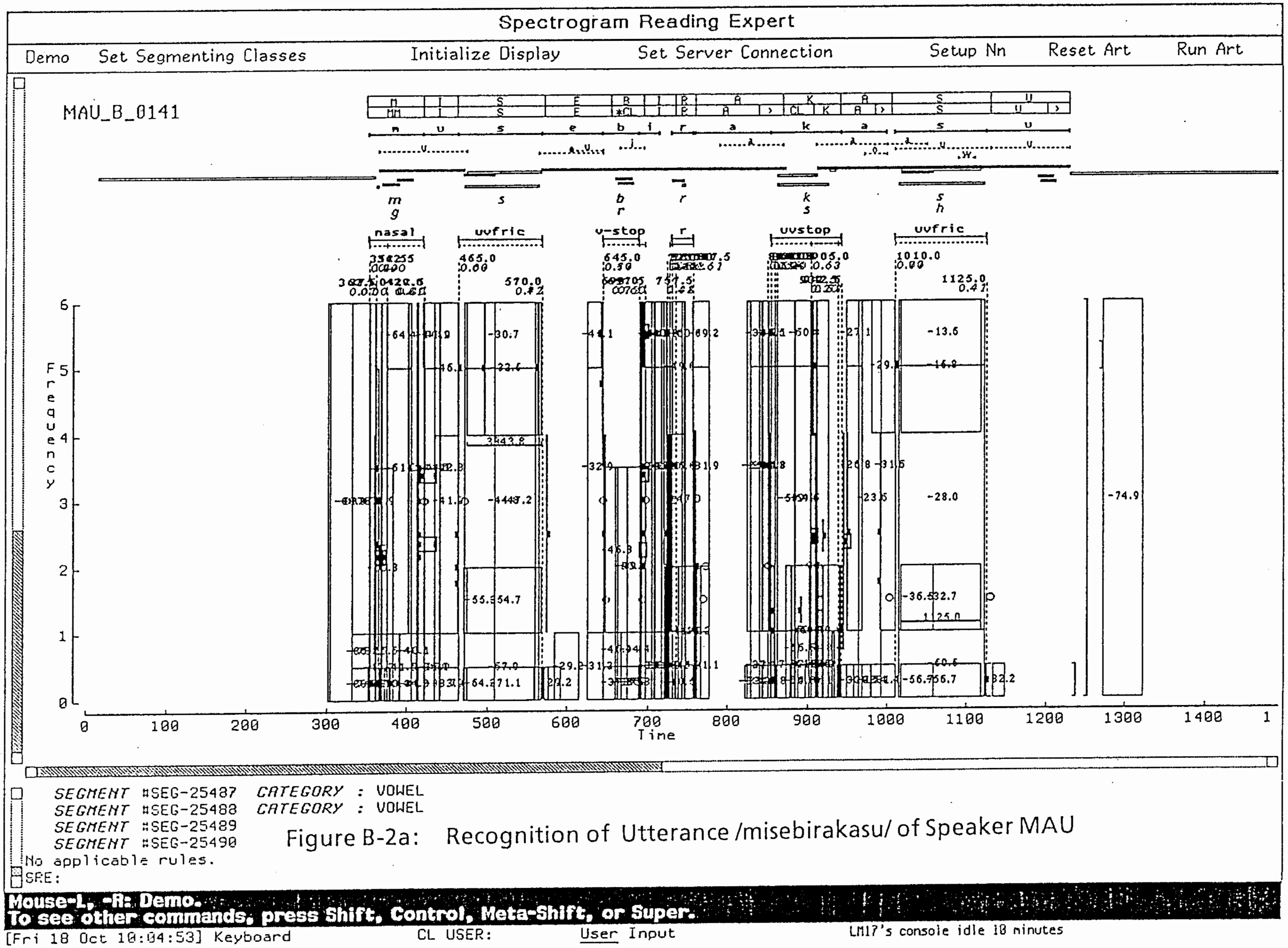


Figure B-2: Spectrogram of Utterance /misebirakasu/ of Speaker MAU

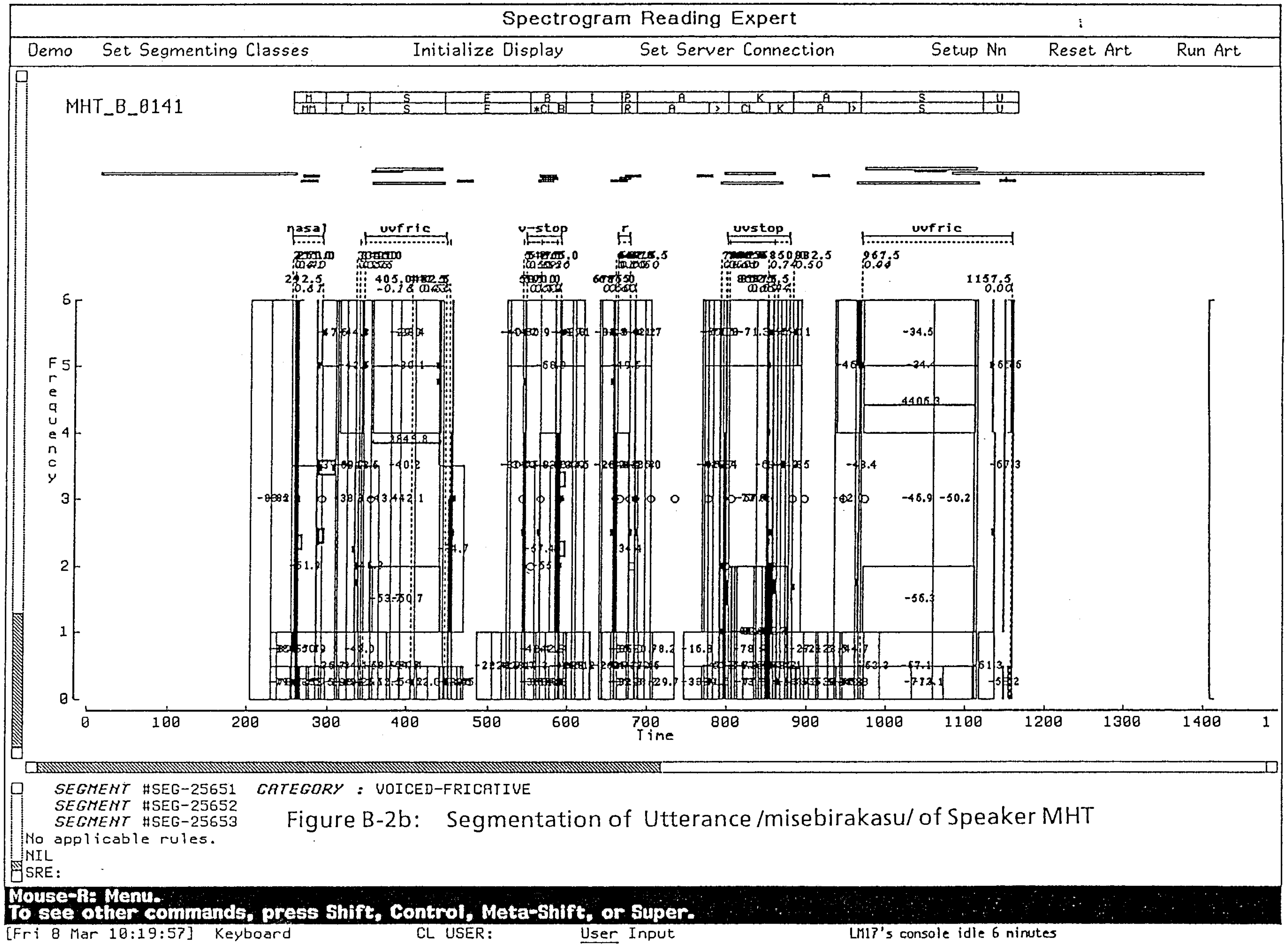


```
SEGMENT #SEG-25487  CATEGORY : VOWEL
SEGMENT #SEG-25488  CATEGORY : VOWEL
SEGMENT #SEG-25489
SEGMENT #SEG-25490
No applicable rules.
SRE:
```

Mouse-L, -R: Demo.
To see other commands, press Shift, Control, Meta-Shift, or Super.

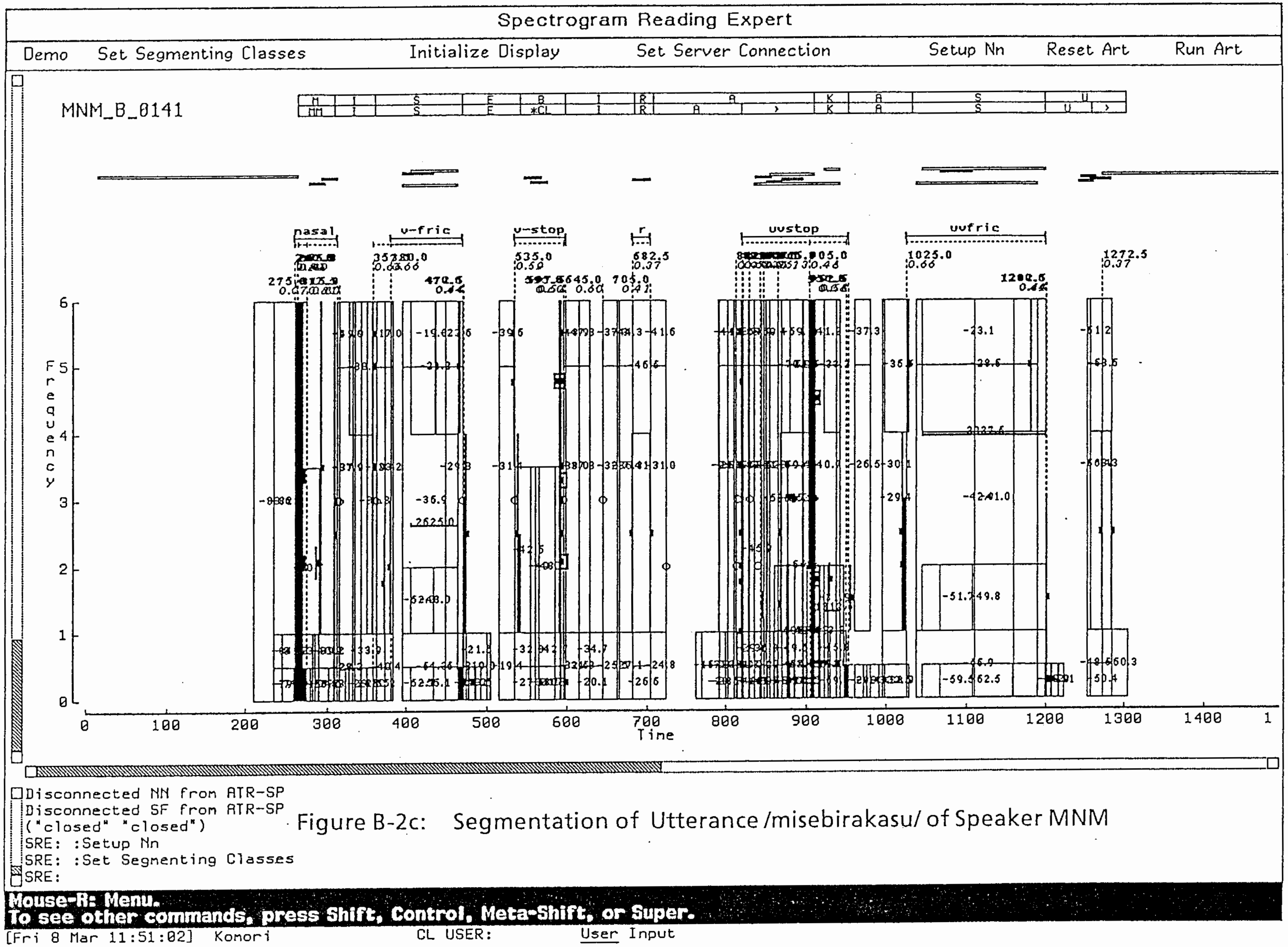
```
[Fri 18 Oct 10:04:53] Keyboard          CL USER:     User Input          LM17's console idle 10 minutes
```

Figure B-2a: Recognition of Utterance /misebirakasu/ of Speaker MAU



```
SEGMENT #SEG-25651 CATEGORY : VOICED-FRICATIVE
SEGMENT #SEG-25652
SEGMENT #SEG-25653
No applicable rules.
NIL
SRE:
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 8 Mar 10:19:57]  Keyboard          CL USER:          User Input                    LM17's console idle 6 minutes
```

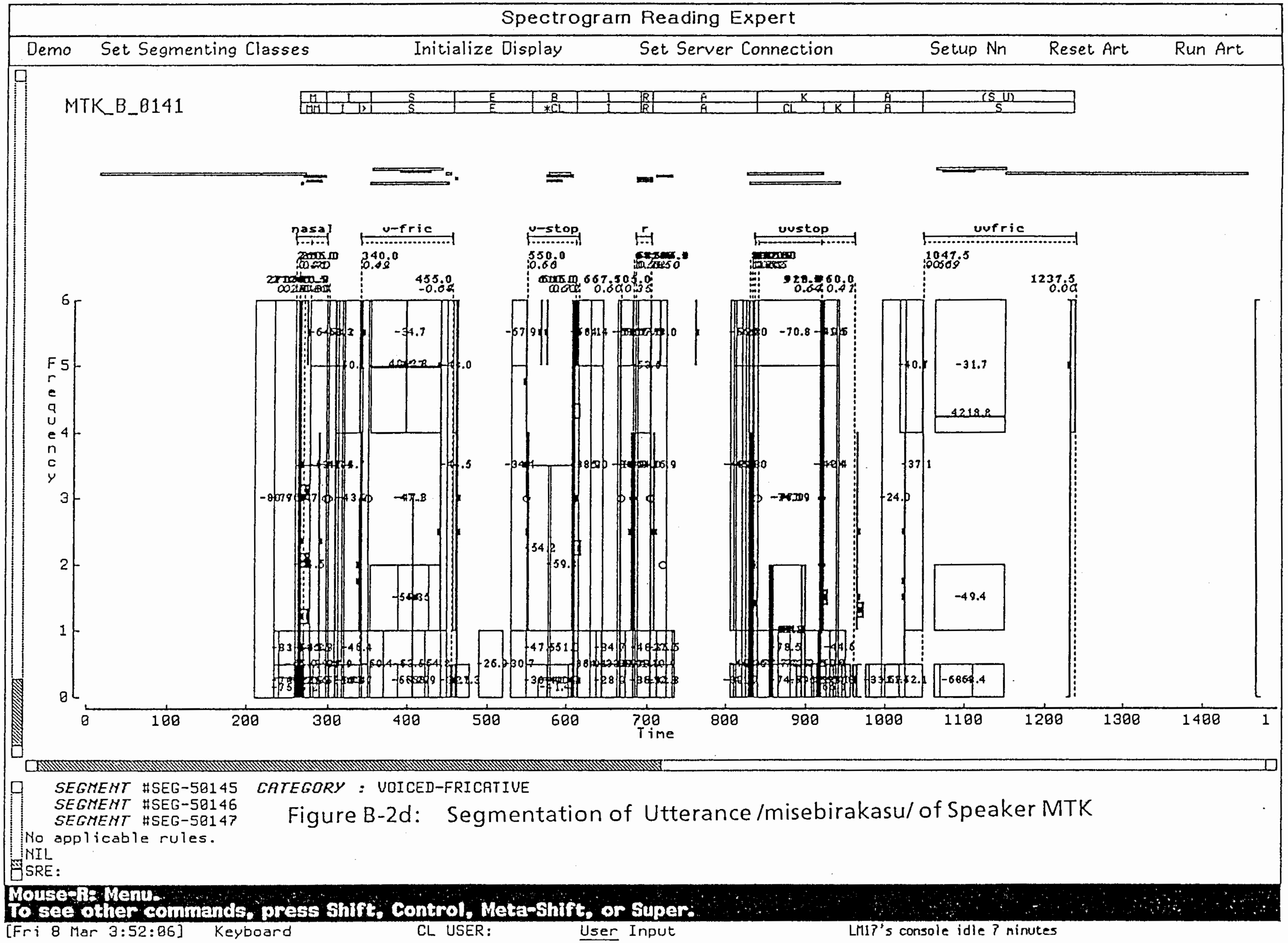
Figure B-2b: Segmentation of Utterance /misebirakasu/ of Speaker MHT



```
Disconnected NN from ATR-SP
Disconnected SF from ATR-SP
("closed" "closed")
SRE: :Setup Nn
SRE: :Set Segmenting Classes
SRE:
```

Figure B-2c: Segmentation of Utterance /misebirakasu/ of Speaker MNM

```
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 8 Mar 11:51:02]  Komori          CL USER:          User Input
```



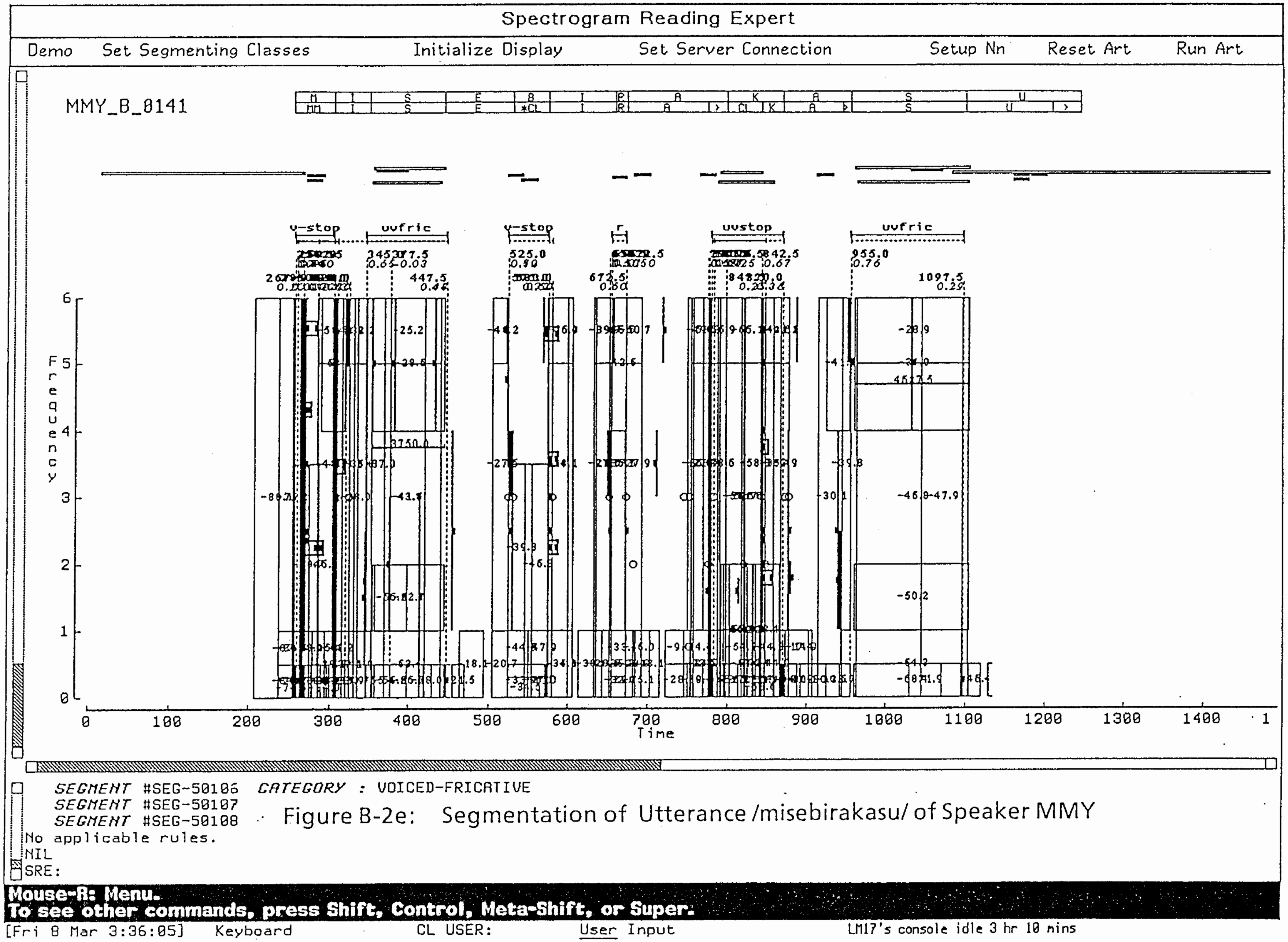
SEGMENT #SEG-50145 CATEGORY : VOICED-FRICATIVE
SEGMENT #SEG-50146
SEGMENT #SEG-50147

Figure B-2d: Segmentation of Utterance /misebirakasu/ of Speaker MTK

No applicable rules.
NIL
SRE:

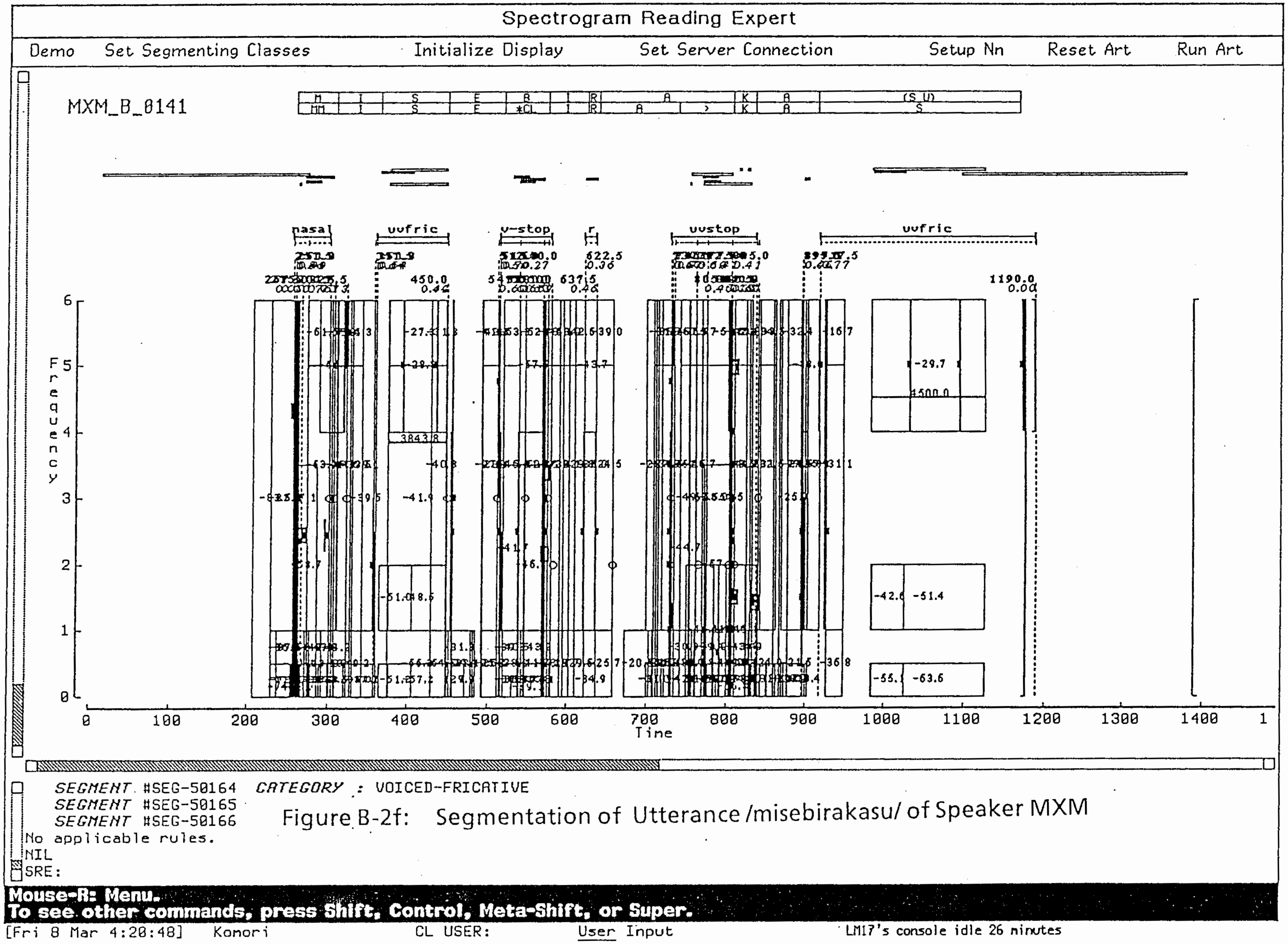
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.

[Fri 8 Mar 3:52:06] Keyboard CL USER: User Input LM17's console idle 7 minutes



```
SEGMENT #SEG-50106  CATEGORY : VOICED-FRICATIVE
SEGMENT #SEG-50107
SEGMENT #SEG-50108
No applicable rules.
NIL
SRE:
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 8 Mar 3:36:05]    Keyboard          CL USER:        User Input              LM17's console idle 3 hr 10 mins
```

Figure B-2e: Segmentation of Utterance /misebirakasu/ of Speaker MMY



```
SEGMENT #SEG-50164  CATEGORY : VOICED-FRICATIVE
SEGMENT #SEG-50165
SEGMENT #SEG-50166
No applicable rules.
NIL
SRE:
```

Figure B-2f: Segmentation of Utterance /misebirakasu/ of Speaker MXM

```
Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Fri 8 Mar 4:20:48]   Konori            CL USER:        User Input              LM17's console idle 26 minutes
```

Index