

論文 / 著書情報  
Article / Book Information

|                   |                                                                                                                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 題目(和文)            | コーパスベーステキスト音声合成における最適化の研究                                                                                                                                                                 |
| Title(English)    | A study on optimization in Corpus-based text-to-speech synthesis                                                                                                                          |
| 著者(和文)            | 岩橋直人                                                                                                                                                                                      |
| Author(English)   |                                                                                                                                                                                           |
| 出典(和文)            | 学位:博士(工学),<br>学位授与機関:東京工業大学,<br>報告番号:乙第3551号,<br>授与年月日:2001年7月31日,<br>学位の種別:論文博士,<br>審査員:                                                                                                 |
| Citation(English) | Degree:Doctor (Engineering),<br>Conferring organization: Tokyo Institute of Technology,<br>Report number:乙第3551号,<br>Conferred date:2001/7/31,<br>Degree Type:Thesis doctor,<br>Examiner: |
| 学位種別(和文)          | 博士論文                                                                                                                                                                                      |
| Type(English)     | Doctoral Thesis                                                                                                                                                                           |

# A Study on Optimization in Corpus-based Text-to-speech Synthesis

Naoto Iwahashi

April 2001

## Acknowledgements

First and foremost, I would like to express my sincere appreciation to Dr. Yoshinori Sagisaka for his encourage, guidance, discussion and help. His wide knowledge and keen insight provided me with substantial support.

I would like to express my sincere gratitude to Prof. Takao Kobayashi, Tokyo Institute of Technology, my thesis advisor, for his kind encouragement, guidance, suggestions, advice and arrangements concerning the thesis submission. Also, I would like to express my gratitude to Prof. Sadaoki Furui, Prof. Makoto Satoh, Prof. Hiroshi Nagahashi, and Prof. Toshiya Itoh, Tokyo Institute of Technology, for their kind suggestions.

I wish to express my sincere appreciation to Mr. Kenzo Akagiri, Sony Corp., for giving me the opportunity for working at ATR.

I wish to express my sincere thank to Prof. Akira Kurematsu, the President of ATR Interpreting Telephony Research Labs. (currently with University of Electro-Communications), for giving me a chance to achieve this study.

I thank to Dr. Toshitada Doi, Dr. Mario Tokoro and Dr. Hitoshi Iida, Sony Computer Science Labs for their continuous support.

I also owe a great deal to the members of the ATR speech processing department. I would like to thank Dr. Nobuyoshi Kaiki, Mr. Katsuhiko Mimura and Mr. Toshio Hirai for their kind cooperation. I wish to express my sincere appreciation to Prof. Nick Campbell for his encouragement, help and suggestion. I would like to thank Prof. Shigeki Sagayama (currently with Tokyo University), Prof. Masahide Sugiyama (currently with Aizu University) and Dr. Norio Higuchi for their kind encouragement and discussion.

I thank to Dr. Mary Ostendorf, Boston University, for her kind suggestion and help.

Finally, I am thankful to all the people whom I have met in my life, in particular, to my family. To my deceased father, Shunji, and my mother, Nobuko, for raising me to be as I am, to my sister, Ayako, for supporting me, to my sons, Takuya and Daisuke, for their smile, which encourage me every day, and to my wife, Akiko, for her smile, patience and encouragement.

# Abstract

Corpus-based text-to-speech (TTS) synthesis converts input text into speech taking advantage of a large speech corpus. In order to synthesize the natural-sounding and intelligent speech in this method, necessary information should be appropriately selected, extracted, or modeled in the corpus. On this problem, the approach based on diagnostic manual tuning by human experts, which has been used in the development of rule-based TTS synthesis systems, does not lead to enough optimality.

This thesis addresses the algorithmic optimization approach based on objective criteria, which excludes the empirical process. The optimization methods are presented for each of the five main factors influencing speech-quality and applicability of the corpus-based TTS synthesis: unit selection, unit set construction, prosody control, the adaptation of speaker characteristics, and classifier training.

The novel method proposed for unit selection minimizes acoustic distortions between the selected segment and the desired target spectrum without using heuristics. The acoustic quantities formulated and used as measures for minimization are four types of distortion, (1) the spectral prototypicality of a unit, (2) the spectral difference between the source and target contexts, (3) the degradation resulting from concatenation of phonemes, and (4) the acoustic discontinuity between the concatenated units. An efficient search method for selecting units from a large speech database is also described, and a perceptual test shown that it produces synthesized speech of high quality.

The novel method proposed for construction of the unit set uses a speech segment network so that high-quality speech can be synthesized but its size is small enough to be of practical use. This method selects a synthesis unit set in which segmental or intersegmental distortion, or both, are minimized by using combinatorial optimization methods. Experimental results using diphone units have shown that this method can be used to construct suitable diphone unit sets whose total and maximum intersegmental distortion are respectively reduced. A listening test also confirmed the effectiveness of this method.

The novel method presented for statistical modeling in prosody control is referred to as Constrained Tree Regression (CTR) and requires only a moderate amount of learning data for making suitable representation of the complex effects of control factors. It is based on recursive splits of predictor variable spaces and partial imposition of constraints of linear independence among predictor variables. A hierarchical error function is also presented so that hierarchical structure can be taken into ac-

count in prosody control. Experimental results shown that the models of speech segment duration obtained with this new method were better than those obtained with previous methods.

The novel method proposed for controlling speaker characteristics is a speech spectrum transformation method that uses Radial Basis Function networks for multi-functional representation and interpolates the spectral patterns of multiple speakers. The interpolation between the spectral parameters of the stored utterance data of multiple speakers generates new spectrum patterns, and only a small amount of training data is needed for generating speech adapted to a target speaker. Experiments using the data stored for four speakers confirmed that a speech spectrum close to that of the target speaker could be generated by adaptation.

The novel method proposed for training classifiers makes it possible to obtain the classifiers that will perform well after they are adapted to changes in speaker and other input conditions. Such classifiers can be used to improve the performance of speech recognition as well as speech alignment in both the adaptation of speaker characteristics and the automatic phoneme segmentation for creating speech corpus. The validity of the method was confirmed by experiments in speech recognition with speaker adaptation.

The proposed algorithmic optimization approach to TTS synthesis was shown to be suitable for generating speech that sounds natural, has been widely used, and has resulted in a level of speech quality appropriate for many useful applications.

# Contents

|          |                                                                       |           |
|----------|-----------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Overview</b>                                                       | <b>1</b>  |
| 1.1      | Introduction . . . . .                                                | 1         |
| 1.2      | Outline . . . . .                                                     | 2         |
| <b>2</b> | <b>Unit Selection</b>                                                 | <b>5</b>  |
| 2.1      | Introduction . . . . .                                                | 5         |
| 2.2      | Distortion Measure . . . . .                                          | 7         |
| 2.3      | Estimation of Contextual Spectral Difference (CSD) . . . . .          | 10        |
| 2.4      | Segment Selection Algorithm . . . . .                                 | 11        |
| 2.5      | Evaluation on Spectral Distortion Minimization . . . . .              | 13        |
| 2.5.1    | Validity of CSD Estimation . . . . .                                  | 13        |
| 2.5.2    | Perceptual Evaluation on Acoustic Measures . . . . .                  | 14        |
| 2.6      | Conclusion . . . . .                                                  | 16        |
| <b>3</b> | <b>Unit Set Construction</b>                                          | <b>19</b> |
| 3.1      | Introduction . . . . .                                                | 19        |
| 3.2      | Unit Set Construction using Speech Segment Network Approach . . . . . | 22        |
| 3.2.1    | Speech Segment Network . . . . .                                      | 22        |
| 3.2.2    | Optimization methods . . . . .                                        | 23        |
| 3.3      | Experiments for Constructing a Diphone Unit Set . . . . .             | 26        |
| 3.3.1    | Diphone Segment Network . . . . .                                     | 27        |
| 3.3.2    | Optimization . . . . .                                                | 27        |
| 3.3.3    | Results . . . . .                                                     | 28        |
| 3.4      | Perceptual Evaluation . . . . .                                       | 28        |
| 3.5      | Conclusion . . . . .                                                  | 34        |
| <b>4</b> | <b>Duration Control</b>                                               | <b>37</b> |
| 4.1      | Introduction . . . . .                                                | 37        |
| 4.2      | Preliminary . . . . .                                                 | 38        |

|          |                                                                                            |           |
|----------|--------------------------------------------------------------------------------------------|-----------|
| 4.2.1    | Linear regression . . . . .                                                                | 39        |
| 4.2.2    | Tree regression . . . . .                                                                  | 39        |
| 4.3      | A New Regression Method . . . . .                                                          | 40        |
| 4.3.1    | Linear Regression as a Tree Model . . . . .                                                | 40        |
| 4.3.2    | Constrained Tree Regression: CTR . . . . .                                                 | 41        |
| 4.3.3    | Algorithm for Generating a CTR Tree . . . . .                                              | 43        |
| 4.4      | Hierarchical Representation in Regression . . . . .                                        | 46        |
| 4.5      | Experiments on Speech Segmental Duration Modelling . . . . .                               | 48        |
| 4.5.1    | Speech Corpus and Control Factors . . . . .                                                | 48        |
| 4.5.2    | Obtained Statistical Models . . . . .                                                      | 48        |
| 4.5.3    | Discussion . . . . .                                                                       | 50        |
| 4.6      | Conclusion . . . . .                                                                       | 54        |
| <b>5</b> | <b>Adaptation of Speaker Characteristics</b>                                               | <b>57</b> |
| 5.1      | Introduction . . . . .                                                                     | 57        |
| 5.2      | Spectrum Transformation by Speaker Interpolation . . . . .                                 | 59        |
| 5.3      | Speaker Adaptation by Speaker Interpolation . . . . .                                      | 60        |
| 5.4      | Multi-functional Representation with Weighting by Radial Basis Function Networks . . . . . | 62        |
| 5.4.1    | Advantages of using multiple linear functions . . . . .                                    | 62        |
| 5.4.2    | Weighting by Radial Basis Function networks . . . . .                                      | 63        |
| 5.4.3    | Optimization procedure . . . . .                                                           | 64        |
| 5.5      | Evaluation and Discussion . . . . .                                                        | 66        |
| 5.5.1    | Characteristics of speaker interpolation . . . . .                                         | 66        |
| 5.5.2    | Experiments of adaptation by a single interpolating function . . . . .                     | 68        |
| 5.5.3    | Experiments of adaptation by multiple linear functions . . . . .                           | 72        |
| 5.6      | Conclusion . . . . .                                                                       | 73        |
| <b>6</b> | <b>Classifier Training</b>                                                                 | <b>77</b> |
| 6.1      | Introduction . . . . .                                                                     | 77        |
| 6.2      | Formulation for Minimum Error Classification . . . . .                                     | 78        |
| 6.3      | Training Objective . . . . .                                                               | 80        |
| 6.3.1    | Adaptation Criterion . . . . .                                                             | 80        |
| 6.3.2    | Empirical Average Loss for Classifier Training . . . . .                                   | 80        |
| 6.4      | Optimization Method . . . . .                                                              | 82        |
| 6.5      | Experiments . . . . .                                                                      | 83        |
| 6.5.1    | Arrangements . . . . .                                                                     | 83        |
| 6.5.2    | Results . . . . .                                                                          | 85        |

|          |                                                                                      |           |
|----------|--------------------------------------------------------------------------------------|-----------|
| 6.6      | Discussion . . . . .                                                                 | 88        |
| 6.7      | Conclusion . . . . .                                                                 | 89        |
| <b>7</b> | <b>Conclusion</b>                                                                    | <b>91</b> |
| <b>A</b> | <b>Computational Complexity in Speech Segment Network Optimiza-<br/>tion Problem</b> | <b>95</b> |
| <b>B</b> | <b>Efficient Search in Growing a CTR Tree</b>                                        | <b>99</b> |

# Chapter 1

## Overview

### 1.1 Introduction

Speech is the primary modality in human communication, and man-machine interfaces that are natural and easy to use must be speech-based ones. Interactive systems using speech might consist of several modules processing spoken language, such as modules for automatic speech recognition, for spoken language understanding, for dialog management, and for text-to-speech (TTS) synthesis. TTS technology converts machine-readable text into speech and can convey information to users in a natural way. While the short-term goal of those working on TTS synthesis is simply to produce natural-sounding speech that satisfies the requirements of a specific application, their ultimate goal is to produce speech that cannot be distinguished from human speech.

As a result of intensive research that has been carried out since the 1960s, TTS systems with many applications have been developed. We can, for example, already listen to a machine that reads e-mail and news while we drive our cars. And over a telephone line we can obtain simple services from automatic dialog systems producing intelligible synthetic speech. The speech quality obtained in TTS synthesis has had a great influence on the effectiveness of the man-machine interface in such systems and even though the level of quality available today is still not high enough for complicated and sophisticated tasks, many speech-based interface systems have been put to practical use by combining TTS synthesis with speech recognition technology and natural language technology.

The research on TTS synthesis has been primarily devoted to describing human speech production mechanisms from the viewpoint of cognitive research, and TTS systems have usually been implemented as expert systems in which knowledge is described as declarative rules. In the course of developing such traditional

rule-based TTS systems, their rules and control parameters are manually tuned in a time-consuming cyclic procedure including perceptual evaluation and empirical modification of the rules and control parameters. Because this tuning process depends on the subjective judgment of experts, it is almost impossible to analyze the problems quantitatively and seek better solutions systematically.

In contrast to this rule-based approach to TTS synthesis, the corpus-based approach widely used currently is an algorithmic optimization process that is based on objective measures and that takes advantage of a large speech corpus. The corpus-based TTS synthesis made it possible to investigate the methods of TTS synthesis systematically using algorithmically well-defined procedures with objective measures and eventually resulted in the production of natural-sounding synthetic speech that could be put to use in many applications.

This thesis addresses the algorithmic optimization methods for corpus-based TTS synthesis. The proposed methods triggered the further intensive researches of corpus-based TTS synthesis and have contributed to the current success of TTS synthesis.

## 1.2 Outline

The five main topics in this thesis are a unit-selection method based on spectral distortion minimization, the unit set construction based on segment network optimization, the statistical modeling for prosody control, the adaptation of speaker characteristics, and the training of classifiers. There is a chapter for each of these topics, and each contains its own introduction and review of related work because the backgrounds of the topics are different. Anyone with sufficient knowledge of TTS synthesis should be able to work their way through this thesis, while this may require considerable effort for a novice. Anyone who wants to know the basic technologies of TTS synthesis should be kindly recommended to refer to [Dutoit 96].

Chapter 2 describes a new unit-selection scheme for unit-concatenative speech synthesis. In this scheme a unit sequence for concatenation is selected, without the use of heuristics, by minimizing acoustic distortions between the selected units and the desired spectrum for the target. The acoustic quantities formulated and used as measures for minimization are four types of distortion: (1) the spectral prototypicality of a unit, (2) the spectral difference between the source and target contexts, (3) the degradation resulting from concatenation of phonemes, and (4) the acoustic discontinuity between the concatenated units. Also described here is a search method for selecting units from a large speech corpus. In this method the four

types of distortion are minimized by using a three-step optimization. This chapter is based on the work reported in [Iwahashi et al. 92, Iwahashi et al. 93], which have been cited in many papers, such as [Hunt and Black 96, Huang et al. 96]. The proposed optimization scheme with objective measures has been studied intensively and used in many current TTS systems, such as AT&T NEXTGEN synthesis system [Syrdal et al. 00].

Chapter 3 describes a speech segment network approach for the construction of a synthesis unit set that is suitable for the synthesis of high-quality speech yet small enough to be of practical use. This approach selects a synthesis unit set in which segmental or intersegmental distortion, or both, are minimized by using combinatorial optimization methods such as iterative improvement and simulated annealing. This chapter is based on the work reported in [Iwahashi and Sagisaka 95b].

Chapter 4 describes a new method for the statistical modeling of prosody control in speech synthesis. Referred to as Constrained Tree Regression (CTR), this method requires only a moderate amount of learning data for making a suitable representation of the complex effects of factors controlling prosody. It splits predictive variable spaces recursively while imposing linear independence in the multiple split spaces. It includes both linear and tree regressions with categorical predictor variables, which have been conventionally used for prosody control, and extends them to more general models. This chapter also presents a hierarchical error function so that hierarchical structure can be taken into account in prosody control. These new methods have been applied to the modeling of speech segmental duration. This chapter is based on the work reported in [Iwahashi and Sagisaka 00].

Chapter 5 describes a new method for the adaptation of speaker characteristics that uses Radial Basis Function networks and interpolates the spectral patterns of multiple speakers. The interpolation among the stored speech spectral patterns of multiple speakers, which may be synthesized by a TTS system using each speaker's speech corpus, is carried out to generate new spectrum patterns. Adaptation to a target speaker can be performed to generate a new speech spectrum sequence close to that of the target speaker using only a small amount of training data. This chapter is based on the work reported in [Iwahashi and Sagisaka 95a].

Chapter 6 describes a novel training method of classifiers, which can improve the performance of speech recognition as well as speech alignment. The precise alignment of target speaker's speech with synthesized speech is desired in the method for adapting speaker characteristics. The alignment of speech with its phoneme transcription is also desired in the automatic creation of speech corpus [Iwahashi et al. 91, Makashay et al. 00]. In the proposed method, a classifier is trained while adapta-

tion is being carried out so that the trained classifier will perform well after it is adapted to changes in speaker and other input conditions. This chapter is based on the work reported in [Iwahashi 00].

Chapter 7 presents closing discussions and discussions of possible future work.

# Chapter 2

## Unit Selection

### 2.1 Introduction

In concatenative speech synthesis it is not always easy to select a suitable segment sequence for a given context. So far, this problem has been studied from various points of view, such as size of synthesis unit, type of segment concatenation and measures of segment sequence optimality.

As for synthesis units, various types of unit, such as CV (Consonant-Vowel), diphone, and triphone have been proposed. The CV type unit is simple and efficient in the case of Japanese because of the small number needed for Japanese syllables, but is not satisfactory for high quality of synthesized speech [Abe and Imai 79, Tohkura and Sagisaka 80]. Diphone units can effectively represent the spectral dynamics, and produce comparatively good quality speech [Moulines and Charpentier 90].

The combination of CVC and diphones units have also been proposed [Sato 84]. However, to represent the allophonic variation for better quality synthesis, the number of units tends to increase for different phoneme environments. It is difficult therefore to prepare all the necessary combinations as a unit set. There is a trade-off between the number of prepared combinations and the quality of the synthesized speech. Non-uniform unit selection [Sagisaka 88] is very flexible, and can represent spectral patterns specific to the various contexts if given enough source data.

As for segment concatenation, it has been reported that degradation from concatenation is smaller in consonants than in vowels [Sato 84], and consideration of spectral smoothness at the concatenation points improves naturalness and intelligibility [Abe et al. 89, Nomura et al. 90]. In [Abe et al. 89], spectral smoothness is improved by automatic boundary adjustment. In [Nomura et al. 90], the segment sequence for a target text is selected by minimizing the sum of all the distortions at fixed concatenation points. However, these concatenation methods

were not yet incorporated systematically in non-uniform unit selection procedures [Takeda et al. 92].

Moreover, as for the measurement of a segment sequence's optimality, phonological knowledge and other heuristic methods have been used [Hirokawa and Hakoda 90, Takeda et al. 92]. These measures improved the speech quality to a certain extent. However, it is time-consuming to develop segment selection procedures for such a synthesis scheme, because to improve speech quality iterative adjustment and listening has to be done. These measures represent only qualitative characteristics in each case, therefore it is difficult to optimize their parameters. More general measurement for a segment sequence's optimality is needed.

As mentioned above, in previous studies attention to both suitable spectral patterns for target context and smooth concatenation of segments has improved speech quality to a certain extent. Whereas different measures have been used for these factors, it would be preferable if they could be evaluated consistently in the same domain. It is best to utilize acoustic measurement for allophonic variation and segment concatenation in the segment selection procedure consistently. In the case that segment sequence is selected for a target text from a large amount of speech data, at least two different types of spectral distortion have to be reduced:

- a) **spectral pattern difference:** The difference between the desired spectral patterns of the target and those of the candidate segments selected from a database for concatenation.
- b) **segment discontinuity:** The spectral discontinuity across the boundaries of the segments that have been selected for concatenation.

It is important to select a synthesis segment sequence which minimizes these acoustic distortions for optimal synthesized speech. From the view point of these acoustic properties, non-uniform unit (NUU) concatenation [Sagisaka 88] [Takeda et al. 92] takes into account both the above distortions, but spectral pattern difference is typically defined using only heuristic measures on the phoneme string, and segment continuity is not considered in the selection procedure. Automatic boundary adjustment [Abe et al. 89] according to phoneme environment is carried out only after the segments have been selected. The method of selecting waveform phoneme segments [Hirokawa and Hakoda 90] takes both distortions into account, but the spectral pattern difference between segments from different contexts is considered only by checking the phonemic contextual coincidence. The context-oriented clustering method (COC) [Nakajima and Hamada 88] deals with spectral

pattern differences acoustically, using contextual averaging, but does not always realize smooth concatenation at the segment boundaries.

To improve speech quality, spectral pattern difference and segment discontinuity should be minimized. To sum up, two further problems have to be solved for minimization of spectral distortions.

The first problem concerns estimation of the contextual spectral difference (CSD) when the target spectral pattern is unknown. If all ideal target spectral patterns were available in the database, there wouldn't be any problem, but in practice it is almost impossible to prepare all spectral patterns for all possible contexts. In the COC method, this problem is avoided by simply averaging the spectral patterns of similar contexts. In the NUU method, categorical measures have been employed instead of acoustic measures.

The second problem is the search problem in a large database for suitable speech segment sequences. It is difficult to know how best to search for an appropriate segment sequence which minimizes spectral distortion. It is impractical to consider the acoustic distortions of all possible unit combinations. Thus, an efficient search method is necessary.

Spectral distortion has to be measured quantitatively. However, it is difficult to define spectral distortions in a single measure, because a non-linear function is required from the view point of perceptual characteristics. Therefore, we propose four types of acoustic measure for different portions of the synthesized speech to predict degradation of synthetic speech. These will be described in the next section.

In this chapter, the CSD estimation method is shown to overcome the first problem noted above. A segment selection algorithm which minimizes spectral distortions with non-uniform units is shown to solve the second problem. Finally, perceptual results for the proposed scheme are shown.

## 2.2 Distortion Measure

The following four distortion measures quantify the spectral distortions described in Sect. 2.1. There are two measures for spectral pattern difference and two for segment discontinuity. Fig. 2.1 shows the points at which these measures would be applied for the three segment target word /tsuibamu/.

In the definitions of the four distortions, cepstrum parameters are used because speech segments are represented by cepstral parameter sequences in our system. An improved cepstrum method[Imai and Abe 79] is used to extract the envelope of the speech spectrum. Manually labelled phonemic information is used to segment the

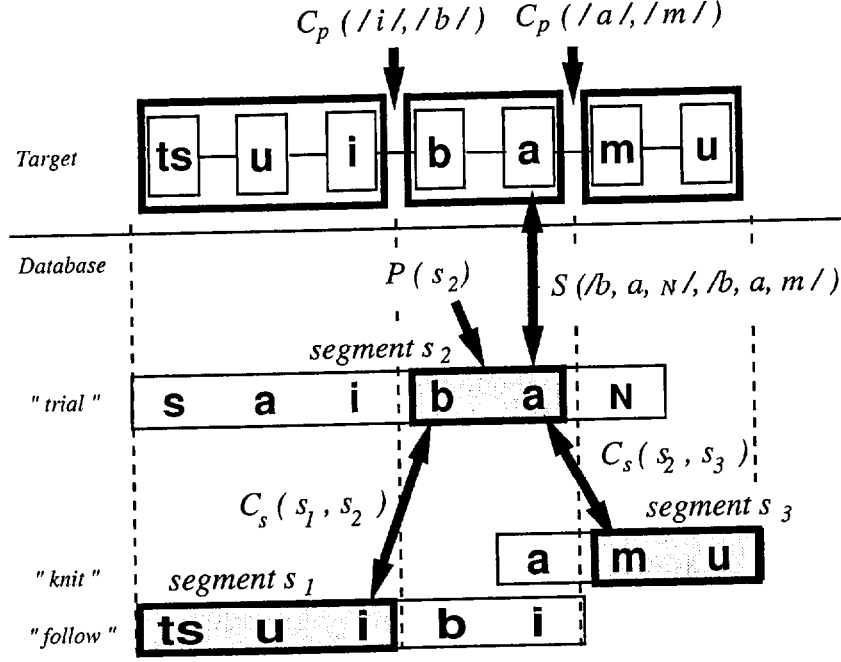


Figure 2.1: Four acoustic measures to evaluate spectral distortion

speech data.

### CSD of Source Context to Target Context

It is usually impossible to prepare all possible contexts in a database of synthesis units, and it is not necessarily optimal simply to use the same context as the target because of possible interference from other factors. Therefore segments in different contexts from the target one sometimes need to be used. In this case, the CSD  $S(e_s, e_t)$  between the source context  $e_s$  and target context  $e_t$  is important. Here, the source context means the context of the segment in the database. This is calculated as the Euclidean distance between the centroids of the sets of time-normalized cepstral parameters of the segments in each of two contexts. Only the immediately preceding and following phonemes are currently considered as context. Evaluation is carried out only on the vowel portions of the units as in the following equation:

$$S(e_s, e_t) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m (c_{s_{i,j}} - c_{t_{i,j}})^2 \quad (2.1)$$

where  $n$  and  $m$  represent the frame number in time-normalized vowel portions in the units and the order of cepstral parameters respectively.  $c_{s_{i,j}}$  and  $c_{t_{i,j}}$  represent the  $j$ th cepstral parameter of the  $i$ th frame of the time-normalized centroids of the

source context in the database and the target context. In the proposed acoustic measures, time-normalization into mean frame length of middle phoneme in each context is carried out linearly on the spectrum sequence of each segment because the selected segment is linearly expanded or contracted to hold suitable phoneme duration.

If there are no contexts the same as the target contexts in the database, then the segments in the target contexts cannot be used. In this case, CSD can not be calculated either, and the CSD estimation described in the next section has to be carried out instead.

### Prototypicality of segment

Speech signals show considerable variation, even in the same context, but irregular or uncharacteristic speech samples should not be used as synthesis segments. It is thus important to measure the typicality  $P(s)$  of a segment among segments from the same context. This prototypicality is calculated as the Euclidean distance between time-normalized cepstral parameters of segment  $s$  and the centroid of the set of segments from the context in which segment  $s$  is taken. Evaluation is carried out only on the segments of vowel portions.

### Disconcatability of phonemes

This is a measure of how much concatenation of two phonemes will degrade speech quality. It has been reported [Sato 84, Abe et al. 89] that degradation is less at the point where the speech spectrum changes rapidly than at the point where spectrum is stable. For an approximation of such degradation, the value is calculated by subtracting the mean of the spectral change at the boundaries of phonemes  $/p_p/$  and  $/p_f/$  in a database from the maximum among these mean values.

### Discontinuity between segments

In order to assess the spectral dynamics, discontinuity  $C_s(s_p, s_f)$  between the preceding segment  $s_p$  and the following segment  $s_f$  is calculated as the spectral distance between  $s_p$  and  $s_f$  around the concatenation point, in the following way:

$$C_s(s_p, s_f) = \sum_{i=-w_c/2}^{w_c/2} h_c(i) \sum_{j=1}^m (c_{p_{i,j}} - c_{f_{i,j}})^2 \quad (2.2)$$

where  $c_{p_{i,j}}$  and  $c_{f_{i,j}}$  represent the  $j$ th cepstral parameter of the  $i$ th frame from the concatenation frame of the preceding segment  $s_p$  and that from the following segment  $s_f$  respectively. A triangular window of length  $w_c$  is used as  $h_c(i)$  to consider the spectral dynamics.

## 2.3 Estimation of Contextual Spectral Difference (CSD)

If a segment in the same context as the target does not exist in the database, then CSD  $[S(e_s, e_t)]$  cannot be calculated as described in the previous section. CSD estimation method has been developed to overcome this problem.

Context  $e$  is expressed as  $/p_i, p_j, p_k/$  which represents the spectral pattern of phoneme  $/p_j/$  when preceded by phoneme  $/p_i/$  and followed by phoneme  $/p_k/$ . When target context  $/p_1, p_2, p_4/$  does not exist in the speech database, CSD between the source context  $/p_1, p_2, p_3/$  and the target context  $/p_1, p_2, p_4/$  can be estimated by the following method. In this example only the following phonemes are different.

1. Find a  $/p_x/$  which satisfies the following equation with the proviso that  $/p_x, p_2, p_4/$  exists in the database. (\* indicates the portion of the context which is not considered.)

For all possible  $p_i$  in the database:

$$\begin{aligned} S(/p_x, p_2, */ , /p_1, p_2, */ ) \\ = \min_{/p_i/} S(/p_i, p_2, */ , /p_1, p_2, */ ) \end{aligned}$$

2. Substitute  $/p_x, p_2, p_4/$  for missing  $/p_1, p_2, p_4/$ .

$$\begin{aligned} S(/p_1, p_2, p_3/ , /p_1, p_2, p_4/ ) \\ \approx S(/p_1, p_2, p_3/ , /p_x, p_2, p_4/ ) \end{aligned}$$

This method replaces an unknown context by an appropriate known context in Eq. (2.1). Then calculation for CSD is carried out between the source and substituted contexts as before. In the above case, only the effect of the following phoneme  $/p_3/$  and  $/p_4/$  need be considered. The following context can be tested if the preceding phoneme is substituted by another similar phoneme. As a first step, a replacement phoneme is chosen so that its effect on the middle phoneme is similar to that of the preceding phoneme  $/p_1/$  in the target. As a second step, CSD between the source and the substituted target context is calculated, using Eq. (2.1), to produce the estimated CSD between the source and the target contexts. The same procedure is applied in the reverse direction swapping preceding and following for

the case where the preceding phonemes are different.

## 2.4 Segment Selection Algorithm

A segment sequence is selected for a target from a database of about five thousands words [Takeda et al. 87] so that spectral distortions are minimized between the desired target sequence and the one chosen for synthesis. The segment selection scheme, using non-uniform units, allows use of any size phoneme sequence as a synthesis unit. Thus, an efficient search method for suitable segment sequences is needed because the number of possible combination is very large.

On the other hand, to calculate the spectral distortion of the generated speech from the view point of perceptual characteristics, a non-linear function with the above four measures seems to be required, but it is difficult to determine this non-linear function. Since it is rare to find a segment sequence which minimizes all four measures in a database, a search algorithm is also needed which reduces all measures simultaneously. A good solution is facilitated in the segment search stage.

The proposed search method reduces the four types of distortion gradually and efficiently by a three-step procedure and so avoids exhaustive search. Moreover, by empirically assigning priority to each distortion measure, this method eliminates the need for a non-linear function.

As a first step, the search space is reduced by evaluating prototypicality  $P(s)$  to reject all unusual or irregular speech segments of only vowel portions. This is carried out off-line in advance.

As a second step, the weighted sum of CSD  $S(e_s, e_t)$  and disconcatability of phonemes  $C_p(/p_p/, /p_f/)$  at potential unit concatenation locations is minimized over a sentence to locate the optimal concatenation points for the target text, and thereby divide the whole target text into smaller fragments. Possible candidate segments for each fragment of the optimal sequence are then selected. The number of possible candidate segments is thus reduced by these pre-processes. A small weighting factor  $\alpha$  is introduced as a tie-breaker so that values of disconcatability of phonemes would produce an optimal unit sequence even when there is not enough difference using only CSD values over a sentence. The value of  $\alpha$  is determined heuristically. The cost function  $F_2(T_p, U)$  for this step is calculated as the weighted sum of phoneme disconcatability and contextual spectral difference over sentence, where  $T_p$  and  $U$  denote the phoneme sequence of target sentence and the sequence of its divided fragments respectively.  $F_2(T_p, U)$  is minimized in terms of  $U$  by dynamic programming as follows:

1. Set initial value,

$$f_2(0) = 0$$

2. Repeat the following procedure for  $i = 1, 2, \dots, N_p$

$$f_2(i) = \min_{0 \leq j < i} \{f_2(i-1-j) + d(i-j, i)\}$$

$$\begin{aligned} d(i-j, i) &= \min_{seg(i-j, i, p_p, p_f)} \{S(/p_p, p_{i-j}, p_{i-j+1}/, /p_{i-j-1}, p_{i-j}, p_{i-j+1}/) \\ &\quad + S(/p_{i-1}, p_i, p_f/, /p_{i-1}, p_i, p_{i+1}/)\} \\ &\quad + \alpha \cdot C_p(/p_{i-j-1}/, /p_{i-j}/) \end{aligned}$$

on condition that  $seg(i-j, i, p_p, p_f)$  exists in a database.

3. Get the minimum value of  $F_2$  by

$$\min_U F_2(/p_1, p_2, \dots, p_{N_p}/, U) = f_2(N_p).$$

where  $N_p$  and  $p_i$  represent the number of phonemes and the  $i$ th phoneme in the target text respectively.  $seg(i, j, p_p, p_f)$  represents the segment in the phoneme sequence  $/p_i, p_{i+1}, \dots, p_j/$  extracted from the context preceded by  $/p_p/$  and followed by  $/p_f/$ . In above procedure,  $S(e_s, e_t)$  is set to zero in the case that middle phoneme is not vowel. Concatenation points in the target speech are selected so that the minimum cost is achieved, and possible candidate segments for each fragment in the selected fragment sequence are obtained at once.

As a third step, an optimal segment sequence which minimizes the discontinuity of segments  $C_s(s_p, s_f)$  over the sentence is selected from the reduced candidate set. Automatic boundary adjustment[Abe and Sato 92] according to the phoneme environment is carried out to evaluate discontinuity at each potential segment boundary. By evaluating  $S(e_s, e_t)$  and  $C_p(/p_p/, /p_f/)$  in the second step, discontinuities between segments are reduced considerably.

The cost function  $F_3(S_s)$  for this step is calculated as the sum of segment discontinuity at all concatenation points, and minimized in terms of segment sequence  $S_s$  by dynamic programming as follows:

1. Set initial value,

$$f_3(0, 0) = 0.$$

2. Repeat the following procedure  
for  $i = 1, 2, \dots, N_s$

$$f_3(i, j) = \min_k \{f_3(i-1, k) + C_s(s_{i-1,k}, s_{i,j})\}$$

3. Get the minimum value of  $F_3$  by following:

$$\min_{S_s} F_3(S_s) = \min_j f_3(N_s, j).$$

where  $N_s$  represents the number of text fragments determined for the target text.  $s_{i,j}$  represents a  $j$ th segment candidate for  $i$ th fragment. An optimal sequence of segments is chosen from the segments which minimize the cost.

## 2.5 Evaluation on Spectral Distortion Minimization

To evaluate the proposed method of minimizing spectral distortions, both the segment selection algorithm for minimizing spectral distortions and the spectral distortion minimization for improving speech quality perceptually have to be checked. First, CSD estimation is evaluated, and second, the effects of minimizing CSD and spectral discontinuity between segments were evaluated as explained below.

### 2.5.1 Validity of CSD Estimation

To confirm the validity of the CSD estimation method, an evaluation experiment was conducted using ATR isolated word speech database [Takeda et al. 87] in which there are 1,516 vowel contexts. The average CSD between each sample and the sample selected by minimizing CSD by estimation **after removing the target sample from the database** is calculated, and compared with the following two values :

**Mean-CSD** : the average of the mean CSD between each sample in the database and all others (This represents a value assuming the substituted context had been selected randomly), and

**Min-CSD** : the average of the minimum CSD between each sample and every other

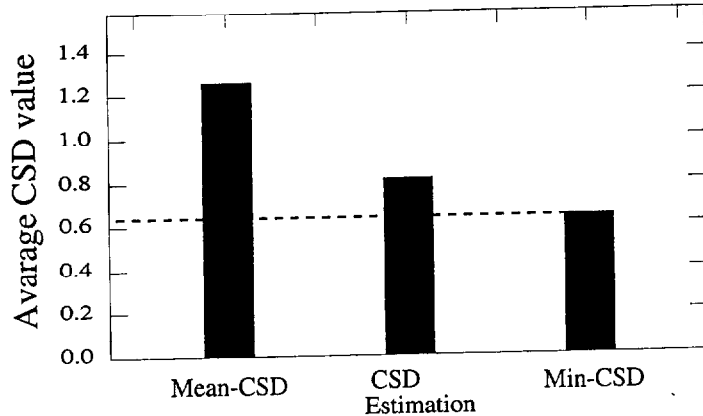


Figure 2.2: Evaluation of CSD Estimation

in the database. (This is an ideal value that we could expect if the target were in the database).

To produce these CSD values, pairs of phoneme environments in which only following phonemes differ are used. That is, the CSD between  $/p_i, v_j, p_k/$  and  $/p_i, v_j, p_x/$  is calculated, where,  $p_i$ ,  $p_k$ , and  $p_x$  stand for any phonemes, and  $v_j$  stands for any vowel, subject to the condition that both  $/p_i, v_j, p_k/$  and  $/p_i, v_j, p_x/$  exist in the database.

As shown in Fig. 2.2, the average CSD by minimum CSD estimation is much closer to the ideal value **Min-CSD**. It was much lower than the value of **Mean-CSD**. This result suggests that CSD estimation reduces the CSD value 70% more than in the case of randomly selecting a substituted context, and supports the usefulness of CSD estimation for calculation of  $S(e_s, e_t)$ .

### 2.5.2 Perceptual Evaluation on Acoustic Measures

Two comparative tests were carried out in order to confirm the validity of the acoustic factors of the proposed selection algorithm.

[Utterances] 30 phrases(utterances) extracted from 12 sentences.

[Subjects] 7 males.

[Procedure] Subjects were asked to make a subjective preference choice between each pair of tokens in response to the question : "Which has the better quality?".

Synthetic speech was generated with sampling frequency of 12kHz. 30th order cepstrum parameters were used for each 5 ms interval. A Log Magnitude Approximation filter[Imai 80] was used for waveform synthesis. Pulse and noise excitation

were used. Prosodic parameters such as pitch, duration and power were controlled using statistical methods[Kaiki et al. 91, Mimura et al.91, Sagisaka and Kaiki 92].

### Test for CSD

Test A is for confirming the validity of the CSD evaluation. Four speech samples with different CSD values were used for a comparative test as follows: ‘Following’ contexts were selected according to different CSD values in the second step of the segment selection method.

**Sample A1:** Optimal contexts are selected. (proposed method)

**Sample A2:** Optimal contexts are selected after removing the ‘following’ target context from the database.

**Sample A3:** Contexts which have medium CSD as possible candidates are selected.

**Sample A4:** Contexts which have maximum CSD are selected.

**A2** assumes that the following context for the target doesn’t exist in the database, and this is used particularly for the perceptual evaluation of CSD estimation.

The relation between average CSD values and preference scores from test A is shown in Fig. 3.3. It confirmed that the use of segments which have small CSD values improves output speech quality. This confirms the validity of CSD minimization. Because **A2** received higher preference scores than **A3** and **A4**, the perceptual validity of CSD estimation was confirmed. This shows that the above segment selection procedures can select suitable candidates by CSD estimation even if the required target context doesn’t exist in the database.

### Test for Segment Discontinuity

Test B checked the validity of evaluating the discontinuity between segments. Four samples were used. In Table. 3.1, the column **Segment Selection** indicates whether DP is carried out to maximize discontinuity as a worst case, or to minimize it. **Boundary Adjustment** indicates whether optimal adjustment was carried out in the DP and concatenation processes automatically, or not at all.

The results of test B (shown in Fig. 2.4) show that as discontinuity values get smaller, quality is higher. It suggests that minimizing discontinuity in both segment selection and boundary adjustment achieves highest quality. Reduction of discontinuity by segment selection was better than one by boundary adjustment.

In Fig. 2.4, we can see that slopes of lines from B1 to B2 and from B3 to B4 are flatter compared with the overall slope, which indicates that segment selection elicits more positive judgements if reduction rate is the same.

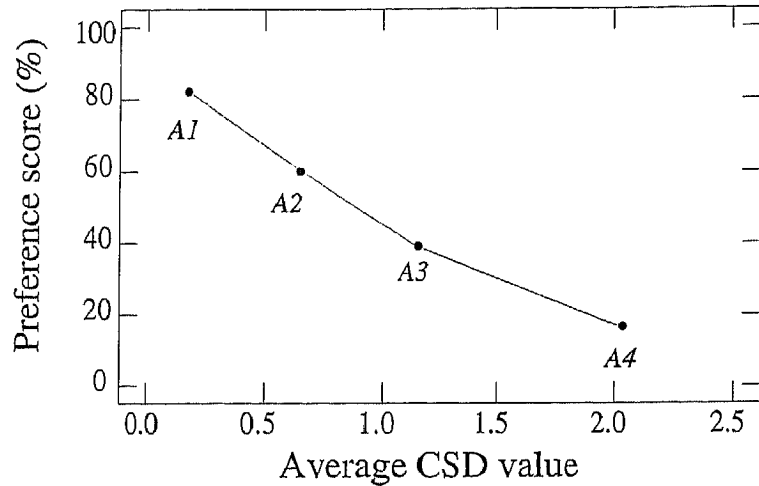


Figure 2.3: Relation between preference score and average CSD value

Table 2.1: Speech Samples of test B

|            | Segment Selection | Boundary Adjustment |
|------------|-------------------|---------------------|
| Sample B1† | min               | yes                 |
| Sample B2  | min               | no                  |
| Sample B3  | max               | yes                 |
| Sample B4  | max               | no                  |

†proposed method

## 2.6 Conclusion

A new scheme for the selection of segments for concatenative synthesis with non-uniform units was proposed. It employs minimum distortion criteria to select an acoustically optimal segment sequence for a target utterance from a large segment database. It copes with two problems, acoustical evaluation of synthesized speech degradation, and search for suitable segment sequences in a large database. It therefore effectively combines the three problems described in the introduction i.e., size of synthesis unit, type of segment concatenation and measures of segment sequence optimality.

To estimate quality degradation from several spectral aspects of the synthesized speech, four acoustical measurements were defined on spectral distortions between

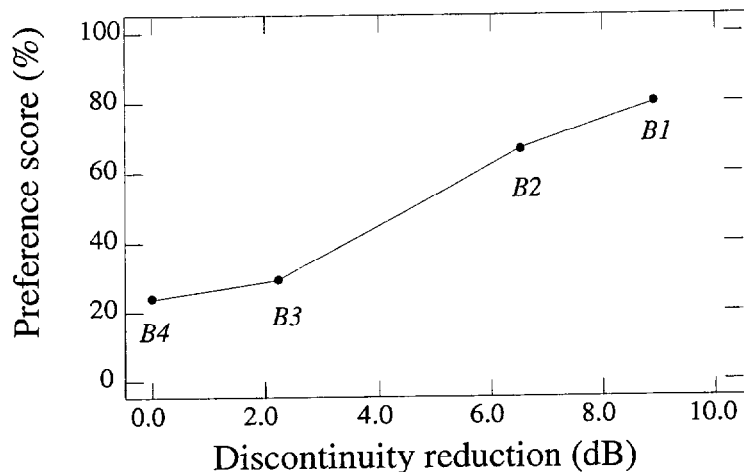


Figure 2.4: Relation between preference score and Discontinuity reduction

a target utterance and the candidate segments for synthesis. CSD estimation was proposed for the case in which the required target context doesn't exist in the database. Validity of this CSD estimation method was shown by the "leave-one-out" test.

An efficient search method which minimizes acoustical cost functions was described. Three step optimization was employed to avoid the necessity of a perceptual non-linear function on the four acoustical measures. Dynamic programming was used for efficient search in a large database.

A perceptual test confirmed that minimization of both CSD and discontinuity values improves speech synthesis quality. It was shown by perceptual test that CSD estimation for segment selection can produce suitable candidates even if the required target context doesn't exist in the database. For continuity, the use of both the search for an optimal segment sequence and boundary adjustment between each segment is effective. Segment prototypicality was not always effective because the database which was used consists of very carefully spoken words. However, this measure might become valid if the size of the database has to be reduced. As for phoneme concatenability, at the second step of segment selection a small value of  $\alpha$  improved speech quality. However, an approximation of this defined measure seems rough: further perceptual investigation might be needed.

Although more precise control might be effective, speech synthesized by the proposed scheme with non-uniform units was of considerably high quality. This distortion minimization scheme can be applied to any type of concatenative synthesis system.

# Chapter 3

## Unit Set Construction

### 3.1 Introduction

TTS speech synthesis achieved by concatenating small segments of analyzed natural speech has been studied to attain high-quality synthetic speech. In segment concatenative scheme, segments for concatenation are stored in a synthesis system as synthesis units, such as diphone, demi-syllable and polyphone. From set of stored units, unit sequence is selected for arbitrary target utterance, then speech signal is generated controlling prosody, which contains pitch, phoneme duration and segmental power. While to get high quality of synthetic speech, all processes in synthesis, which contain synthesis unit set construction, unit sequence selection, prosody control and waveform synthesis, should be suitably implemented, this chapter will be focused on construction of set of synthesis units. It is important to prepare suitable units to ensure the quality of the resultant synthesized speech. Preferably, units should both represent allophonic characteristics properly and be concatenated naturally.

In the search for better speech quality, many kinds of synthesis units have been proposed [Sagisaka and Sato 86, Klatt 87]. CV:Consonant-Vowel syllable [Abe and Imai 79] has been commonly used in Japanese speech synthesis, because a small number of combinations of CV (about 150 types) can cover all possible Japanese utterances. In order to appropriately achieve the natural spectral dynamics of detailed coarticulation, longer units such as VCV, CVC, demi-syllable, triphone and polyphone are becoming popular as synthesis units [Sagisaka et al. 86, Ichikawa et al. 87, Salza et al. 87, Sagisaka 88]. On the other hand, in representing allophonic variation around concatenation portions, the number of units shows a tendency to increase for different phoneme contexts [Sagisaka 88, Olive 90]. Additionally, on perceptually reducing degradation at concatenation, concatenation at consonant portions proved

to be effective [Sato 84]. To satisfy these demands, a larger number of units is necessary. However, it is difficult to prepare all necessary phoneme contexts as a unit set.

Recently, appropriate use of a large set of units has proved to yield high quality speech to a certain extent, if a suitable procedure for selecting unit sequences for target utterances is employed [Takeda et al. 92, Hirokawa and Hakoda 90, Iwahashi et al. 92]. In selecting unit sequences, several criteria were used to measure the suitability of unit sequences for input phoneme sequence, in terms of unit length, phoneme context around units, attribute of phoneme at concatenation point, pitch in segment, and so on. Based on such criteria, suitable segment sequences for target utterances could be selected by a dynamic programming technique [Iwahashi et al. 92].

Even so, the unit set stored in a synthesis system should be constructed adequately so as to include suitable units phonemically and acoustically. No matter what kind of units are used, a well-designed unit set is of great importance for synthesized speech quality. On the other hand, the size of this set is expected to be small in a practical synthesis system. The size requirement of the unit set is due to two factors: computational cost for searching the best unit sequence from the unit set and storage cost for the unit set. A unit set that satisfies both speech quality and size requirements is preferable.

Hitherto, when constructing such a unit set, the unit set has been built by iterative replacement and listening (with heuristics) using a large source database. This operation is time-consuming, and provides no guarantee that a better unit set can be obtained, because it is impractical to listen to all unit combinations and select a better unit set. Constructing a unit set can be taken as a combinatorial optimization process: the best unit subset is chosen from the large source unit set. It is difficult to find the best solution to this combinatorial problem using heuristic means, partly because the number of combinations is vast when the source unit set is large, and partly because suitability of a unit set is not measured objectively. An automatic procedure is needed for the selection of an optimal synthesis unit set.

To automatically construct a suitable unit set for high quality synthetic speech, optimality of the unit set has to be evaluated objectively by some means. Similar objective evaluation has been studied for the selection of the best unit sequence for target utterance. The segmental optimality of unit sequences for arbitrary input phoneme sequences can be considered by breaking it into two parts: suitability of spectral characteristics in each unit and suitability of concatenation between adjacent units. Improving smoothness of concatenation leads to higher quality synthe-

sized speech, as well as use of units each of which represent appropriate spectral characteristics for its context [Nomura et al. 90, Hirokawa and Hakoda 90, Takeda et al. 92]. However, these suitabilities are measured separately, and an inclusive measurement has not yet been found. We have reported that at the stage of unit sequence selection, reducing the following types of distortion, which correspond to the above-mentioned suitabilities, is effective in obtaining high quality speech [Iwahashi et al. 92], although these two distortions were not integrated.

- a) **Intra-segmental distortion:** It is preferable if each unit suitably represents acoustical characteristics on contextual variations. This distortion is applied to measurement of typicality of segments to corresponding contextual effects of allophonic characteristics. It is calculated by the difference between the spectral pattern of a segment and the typical pattern for the target context.
- b) **Inter-segmental distortion:** It is preferable that adjacent units be connected acoustically in a smooth manner. This distortion is applied to measurement of smooth concatenation between segments. It is calculated by the difference of the spectral patterns of adjacent units at the concatenation point.

Correlation between these objective criteria and subjective scores on the quality of synthesized speech has been already shown [Iwahashi et al. 92]. For more precise measurements, prosodic effects on segmental spectra, such as effects of duration and pitch, can be considered in intra-segmental distortion as well as the effects of the adjacent phoneme context. However, modifying pitch and duration in segment by adequate method causes a quite smaller spectral distortion than the difference of the adjacent phoneme context does. Besides, discontinuity of the pitch contours of the two adjacent units has to be considered strictly in the inter-segmental distortion, if segments are concatenated without modifying pitch. While in a synthesis scheme in which pitch is modified by adequate rules, such a consideration is not necessary.

Because the construction of a synthesis unit set can be taken as a process for pre-selection of units for all possible target utterances, the above-mentioned distortions as objective criteria can be used for evaluating the optimality of a synthesis unit set as well as the optimality of a sequence of units for each target utterance. As a method for constructing a synthesis unit set by objective criteria, the contextual-oriented clustering method [Nakajima and Hamada 88], which reduces intra-segmental distortion, has been proposed. However, a method which reduces inter-segmental distortion has not yet been proposed. A method which reduces not only intra-segmental but also inter-segmental distortion is necessary to ensure both the high quality. Reducing intra-segmental distortion can be taken as basic problem of clustering and calculating centroids in each cluster; reducing inter-segmental

distortion is a different type of problem. As a method for coping with this problem using a graph scheme, Speech Segment Network (SSN) approach is described in the following section.

## 3.2 Unit Set Construction using Speech Segment Network Approach

### 3.2.1 Speech Segment Network

In order to construct a synthesis unit set, speech segments are selected from a source database which consists of a large amount of speech segments, where each speech segment is extracted from natural speech according to manually labelled phonetic boundaries. The segments are represented by analyzed parameters, such as cepstrum parameters and pitch synchronous waveform. Optimality of a unit set is evaluated by calculating the sum of intra-segmental distortions of all segments in the unit set and the sum of inter-segmental distortions between possible concatenated segments. The intra-segmental distortion of each segment is measured as the mean of distances from the controversial segment to the all other segments which could be used in the same context at a synthesis stage. Inter-segmental distortion is measured by the spectral difference at a concatenated point between units.

To minimize these distortions in a unit set as cost, we considered the selection of a synthesis unit set as a combinatorial optimization problem. The problem is defined as follows:

*Find a unit set which consists of segments selected one by one from each subset of segments in a source database so as to minimize the cost for the unit set,*

where subsets of segments in a source database are given beforehand, and consist of segments which are used for representing same unit inventories, such that all unit inventories corresponding to the subsets can cover all possible phoneme sequences in speech. The cost value is calculated by using intra-segmental and/or inter-segmental distortions. Not only minimizing the sum of these distortions, but also suppressing their maximum is of importance when considering perceptual characteristics. It is preferable that a cost function be able to cope with this.

To consider intra-segmental and inter-segmental distortions in a unit set geometrically, Speech Segment Network (SSN) is defined as a network of segments with values representing the degree of distortion that would occur both in each segment and between segments, if concatenated in a synthesis process. Let set  $S$  and its element  $s_i$  be a source database and the  $i$ th segment in it, respectively, where a

collection  $A$  of disjoint subsets of the set  $S$  is given. Each subset in  $A$  is distinguished by unit inventories represented by segments in them. The SSN is defined as a directed graph  $G(V, E)$ ,  $E = E_1 \cup E_2$ . Vertices  $v_i \in V$ , and edges  $e_{i,j} = (v_i, v_j)$ , ( here, if  $i = j$ , then  $e_{i,j} \in E_1$ , else  $e_{i,j} \in E_2$  ) are defined as follows :

$v_i$  : The  $i$ th vertex that is identical to element  $s_i \in S$ .

$e_{i,i} \in E_1$  : Self-loop that has values corresponding to intra-segmental distortions  $d_{i,i}$  of  $i$ th segment.

$e_{i,j} \in E_2$  : Edge that has values  $d_{i,j}$  of inter-segmental distortion by concatenation of the  $j$ th segment following the  $i$ th segment, which could be connected in a synthesis process.

Each vertex in the graph  $G$  is put into a subset of  $V$  that is identical to a subset in  $A$ . Moreover, a directed subgraph  $C(V', E') \subset G$  is defined as a network which consists of vertices selected one by one from each subset of  $V$ , and edges  $(v_i, v_j)$  such that  $v_i, v_j \in V'$  and  $(v_i, v_j) \in E$ .

A cost function for the directed subgraph  $C(V', E')$  is defined as

$$f(C) = \left\{ \sum_{\substack{i,j \\ e_{i,j} \in E'}} d_{i,j}^\eta \right\}^{1/\eta}.$$

If  $\eta = 1$ ,  $f(\cdot)$  is the sum of distortions, and if  $\eta = \infty$ ,  $f(\cdot)$  is the maximum of distortions. The desired synthesis unit set which constructs  $C^*$  is obtained by searching  $C^*$  for the following:

$$f(C^*) = \min_C f(C).$$

This unit set has small intra-segmental distortion and/or inter-segmental distortion. A major advantage of the SSN approach is that it can reduce inter-segmental distortions in the unit set, as well as intra-segmental distortions. An optimal speech unit set can be obtained if the selected segments set minimizes cost. For instance, an outline SSN for Japanese diphones consisting only of vowels is shown in Fig. 3.1 and Fig. 3.2.

### 3.2.2 Optimization methods

To get the best solution for the defined optimization problem in the previous subsection, enormous computational time is generally required. The computational time

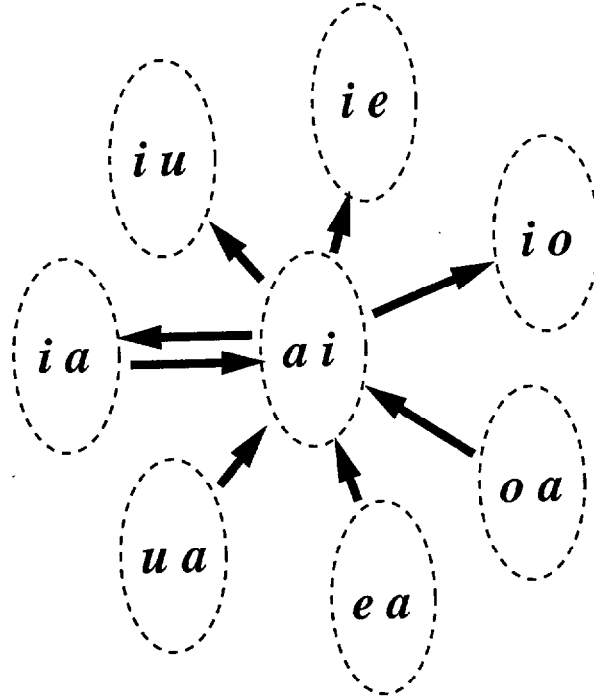


Figure 3.1: Outline of the relationship between cluster / ai / and others

for searching  $C^*$  in the SSN approach increases rapidly as the number of subsets and the number of segments in each subset increase. We consider this computational complexity from the view point of NP-completeness (i.e. completeness in nondeterministic polynomial time) in the Appendix A.

The kinds of unit inventories attributed to subsets in a source database depend on the kinds of segments to be used as synthesis units, such as CV, diphone, triphone and polyphone. For instance, in the case of CV units, the number of CV subsets is about one hundred and fifty for Japanese. Generally, more units need to be used for adequate speech quality. Thus, a larger number of subsets are used in the SSN approach. Even if a CV unit set is used, it would take an enormous amount of time to solve the problem by checking all combinations. As an efficient solution, there are useful techniques, such as the branch-and-bound method, simulated annealing and iterative improvement, which could all be used to solve this problem sub-optimally. We used an iterative improvement method and simulated annealing.

In iterative improvement, each selected combination is slightly changed at each step so that the cost value decreases in an iterative process. This method decreases the cost value deterministically and yields a minimal value within a local search. Simulated annealing [Kirkpatrick et al. 83, Laarhoven and Aarts 87] decreases the cost probabilistically and is able to achieve a global minimum. This technique

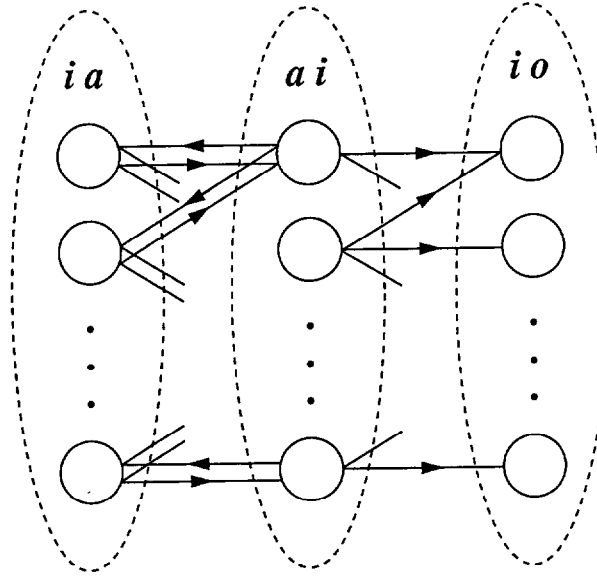


Figure 3.2: Outline of links between vertices for segments

is based on simulation of the annealing of solids. As temperature  $T$  decreases, the Boltzmann distribution concentrates on the states with lowest energy. The simulated annealing can be implemented as the following algorithm :

- Get an initial network  $C$ .
- Get initial temperature  $T_0$ .
- While not yet “frozen”, do the following:
  - Repeat  $L$  times the following:
    - Pick a random cluster  $V_x$
    - Select  $C_y$  which includes vertex  $v_y$  ( $\in V_x$ ) with probability

$$\frac{e^{-f(C_y)/T}}{\sum_{v_i \in V_x} e^{-f(C_i)/T}}$$

- Set  $C = C_y$ .
  - Update temperature  $T$ .
- Return  $C$ .

Table 3.1: Analysis conditions

|                    |                            |
|--------------------|----------------------------|
| Sampling frequency | 12 kHz                     |
| Analysis period    | 5 ms                       |
| Window length      | 21.3 ms ( Hamming window ) |
| Cepstrum           | 30th order                 |

### 3.3 Experiments for Constructing a Diphone Unit Set

Experiments were performed to evaluate whether a unit set could be selected by the SSN approach so as to reduce distortions within it. In the experiments, diphone unit sets for Japanese, consisting of two hundred and sixty-nine units, were selected from a source database which included diphone segments. Each segment consists of a time sequence of low-order cepstrum parameters analyzed from digitized natural speech, which represent the spectral envelope of speech. Analysis conditions for speech segments are shown in Table 3.1. Cepstrum parameters were obtained by the improved cepstrum method [Imai and Abe 79]. Segments were selected one by one from subsets for each diphone inventory in the synthesis unit set.

In the following experiments, reduction of only inter-segmental distortion by the SSN approach was evaluated, partly because the main advantage of the SSN approach is the reduction of inter-segmental distortions, and partly because the weighting between inter-segmental distortion and intra-segmental distortion should be considered from the view point of the perceptual characteristics; and we would first like to separate this problem from the SSN optimization algorithm. This restriction does not cause any loss of generality in the SSN optimization algorithm.

Inter-segmental distortion  $d_{ij}$ , by concatenation of segment  $s_j$  following segment  $s_i$ , is calculated as follows:

$$d_{ij} = \sum_{k=1}^K (c_{i,k}^{(e)} - c_{j,k}^{(s)})^2, \quad (3.1)$$

where  $c_{i,k}^{(e)}$  and  $c_{j,k}^{(s)}$  represent the  $k$ th cepstral parameter of the end frame of the preceding segment  $s_i$  and the start frame of the following segment  $s_j$ , respectively;  $K$  is the order of cepstrum.

### 3.3.1 Diphone Segment Network

The number  $M_s$  of segments in each subset was an experimental variable; 4, 6, 8 and 10 were tried in order to evaluate the effects of various source database sizes. For a source of diphone units, an isolated-word database [Takeda et al. 87] consisting of 5,456 words was used. Because the number of segments for some diphone inventories in this database was less than  $M_s$ , the total number of segments were 990, 1441, 1879 and 2310, respectively. To create the SSN, inter-segmental distortion values between segment  $s_i$  and segment  $s_j$ , which could be connected in a synthesis process, were given as the value  $d_{i,j}$  to edges  $e_{i,j}$  between the vertices  $v_i$  and  $v_j$ . Inter-segmental distortions were calculated at the center portions of vowels, semi-vowels and nasals, and zero values were given at the others, because it is effective for improving speech quality to reduce inter-segmental distortions at these portions [Takeda et al. 92]. To check simply whether the SSN approach could reduce inter-segmental distortions in the synthesis unit set, zero values were given to edges  $e_{ii}$ .

### 3.3.2 Optimization

For optimization procedures in SSN, both iterative improvement and simulated annealing were tested. Unit sets were selected by the following methods:

[Min\_Sum\_II] By iterative improvement; networks were selected to minimize the sum of inter-segmental distortion.

[Min\_Sum\_SA] By simulated annealing; networks were selected to minimize the sum.

[Min\_Max\_II] By iterative improvement; networks were selected to minimize the maximal inter-segmental distortions using a maximum norm. For simplicity of calculation, the minimization was carried out using the approximated value  $\sum_{i,j} d_{i,j}^4$ . Network  $G'$  in which the values of edges were  $d_{i,j}^4$  was used instead of  $G$ .

[Min\_Max\_SA] By simulated annealing; networks were selected to minimize the maximum.

In addition, a unit set which minimizes distortion within subsets was selected for comparison with the unit sets derived by the SSN approach. This unit set can be obtained by choosing the nearest segment to a centroid in each subset (quoted as the [Centroid] method in follows). This synthesis unit set minimizes the sum of intra-segmental distortion, but does not deal with inter-segmental distortion.

### 3.3.3 Results

First, the distribution of mean inter-segmental distortion of unit sets selected randomly 100,000 times in the case of  $M_s = 10$  is depicted in Fig. 3. In this figure, the mean segmental distortion of the unit set selected by SSN optimization (**Min\_Sum\_SA**,  $M_s = 10$ ) is also shown. This figure shows that the unit set selected by SSN optimization has half as much mean inter-segmental distortion as those selected randomly.

The mean, maximum values and variances of inter-segmental distortions in the selected unit sets are shown in Figs. 3.4, 3.5 and 3.6. **Min\_Sum\_II** and **Min\_Sum\_SA** methods reduced the mean of inter-segmental distortions by about 20 ~ 35% of that achieved by the **Centroid** method (Fig. 3.4). **Min\_Max\_II** and **Min\_Max\_SA** methods reduced the maximum value of inter-segmental distortions by about 30% of that achieved by the **Centroid** method (Fig. 3.5). Inter-segmental distortions decreased as  $M_s$  increased. Variance by minimizing the maximum was smaller than by minimizing the sum (Fig. 3.6). There is not a large difference between iterative improvement and simulated annealing methods for reducing inter-segmental distortions. Distributions of inter-segmental distortion in unit sets are shown in Figs. 3.7, 3.8 and 3.9 for **Centroid**, **Min\_Sum\_SA** and **Min\_Max\_SA** respectively, for  $M_s = 10$ . It was proved that the distribution moved into a lower area in inter-segmental distortion by using **Min\_Sum\_SA** and **Min\_Max\_SA**. In addition, the change of inter-segmental distortions can be seen clearly in Fig. 3.10, which shows the relationship between inter-segmental distortions in unit sets selected by **Centroid** and **Min\_Sum\_SA** methods for  $M_s = 10$ . Finally, the convergence process of the cost value by simulated annealing in **Min\_Sum\_SA** ( $M_s = 10$ ) is shown in Fig. 3.11 in contrast with values of temperature  $T$ . In this case, parameters in simulated annealing were set as  $L = 6750$ ,  $T_0 = 6000$ , and  $T$  was updated by  $T_{new} = 0.99 \cdot T_{old}$ .

## 3.4 Perceptual Evaluation

Reduction of inter-segmental distortions by the SSN approach was verified in the experimental results in the previous section. It has been already reported [Iwahashi et al. 92] that reducing the value of this objective measure can improve the quality of synthetic speech as mentioned in Sect. 3.1. In speech synthesis, however, any difference of possibly related factors can affect the quality of synthetic speech.

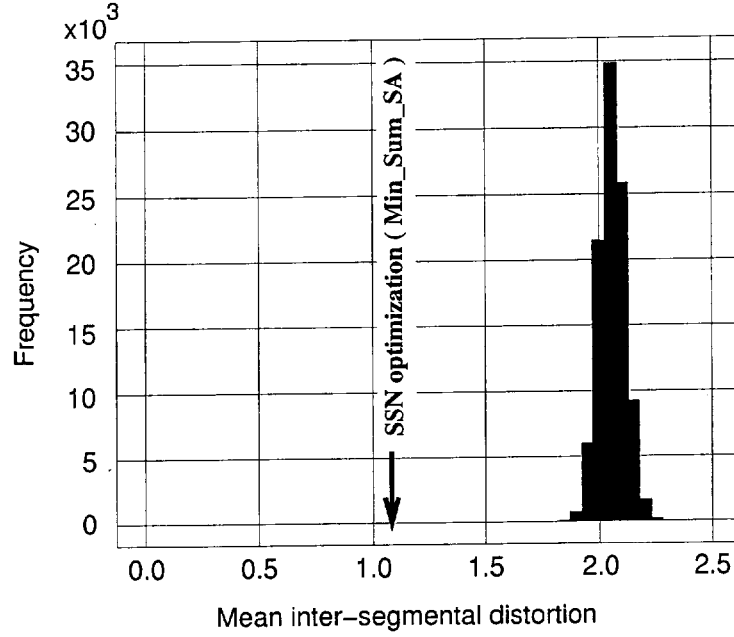


Figure 3.3: Distribution of mean inter-segmental distortions at random trials ( 100,000 times )

So, we conducted a listening test to ensure that the constructed unit set could produce synthetic speech of perceptually better quality. The listening test was carried out by comparing synthesized speech with the unit set selected using the SSN approach, and one using the **Centroid** method.

Speech samples were synthesized by LMA filter [Imai 80], with thirty-order cepstrums. The pitch pattern, phoneme duration and power component were kept the same as in natural utterances. In the process of connecting segments corresponding to phonemes except for vowels and semi-vowels, automatic boundary adjustment [Takeda et al. 92] was carried out in order to minimize cepstral discontinuity.

[Subjects] 7 females

[Procedure] Preference judgment of randomly presented pairs of speech samples

[Speech samples] Forty-eight words synthesized with unit sets selected by a) **Centroid** method, and b) **Min\_Sum\_SA** method for  $M_s = 10$ .

The relationship (Fig. 3.4) between inter-segmental distortions in synthetic words for the test with **Centroid** and **Min\_Sum\_SA** is similar to that of the whole inter-segmental distortions in the unit sets shown in Fig. 6. This shows that speech

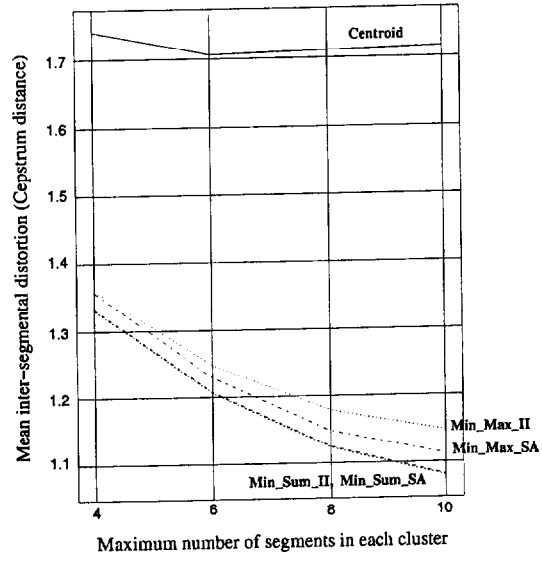


Figure 3.4: Mean of inter-segmental distortions in unit sets selected by different methods

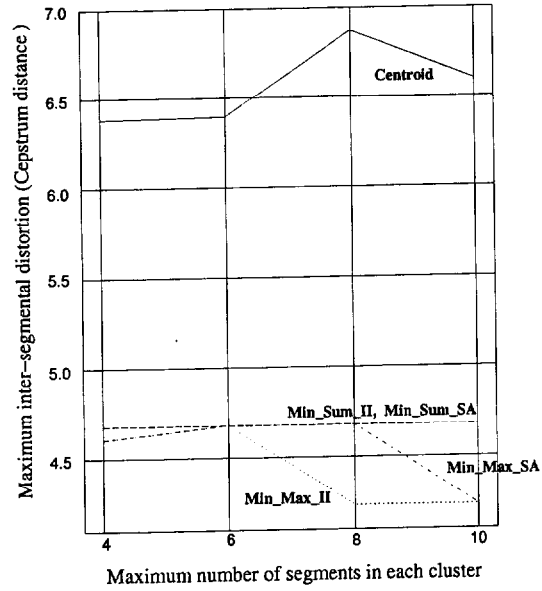


Figure 3.5: Maximum of inter-segmental distortions in unit sets selected by different methods

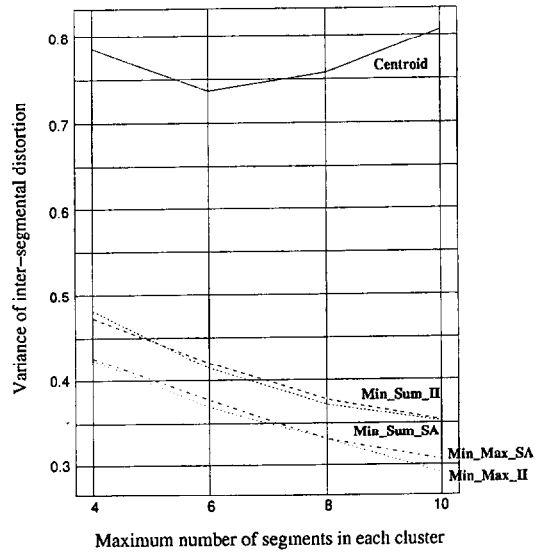


Figure 3.6: Variance of inter-segmental distortions in unit sets selected by different methods

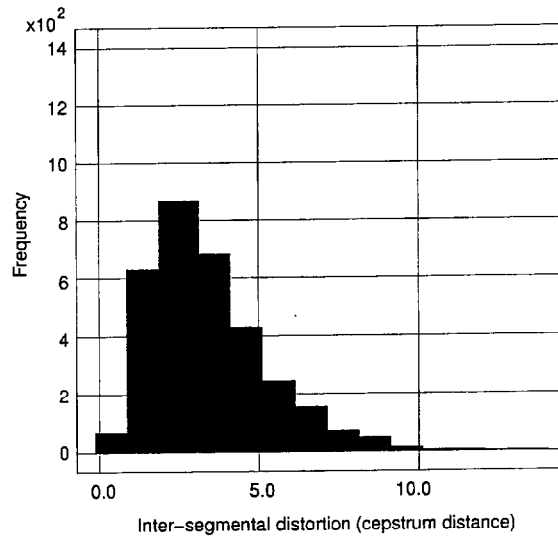


Figure 3.7: Distributions of inter-segmental distortions in unit sets selected by Centroid method (  $M_s = 10$  )

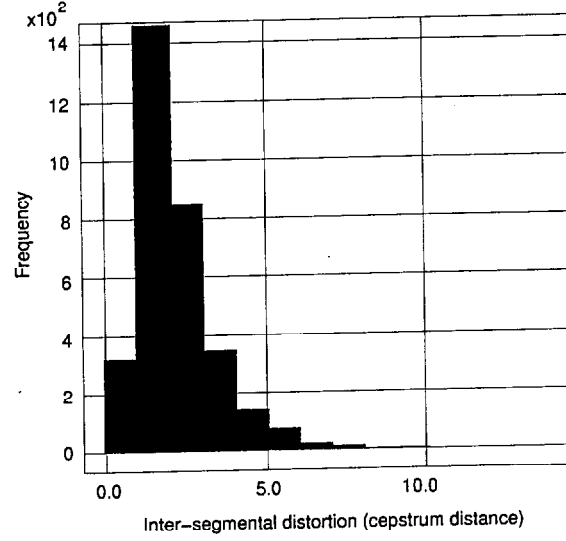


Figure 3.8: Distributions of inter-segmental distortions in unit sets selected by **Min\_Sum\_SA** method (  $M_s = 10$  )

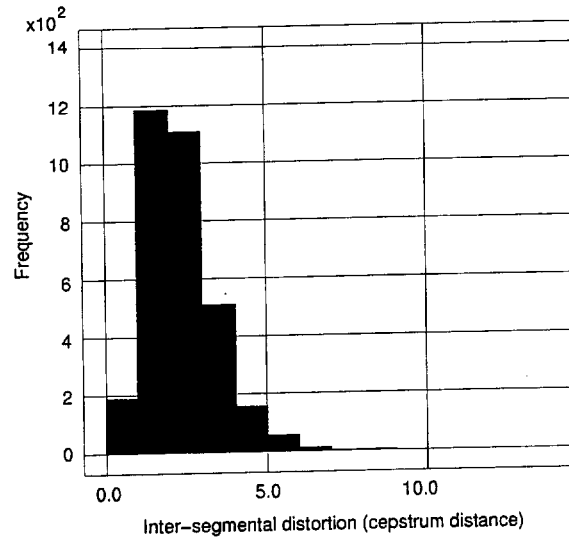


Figure 3.9: Distributions of inter-segmental distortions in unit sets selected by **Min\_Max\_SA** method(  $M_s = 10$  )

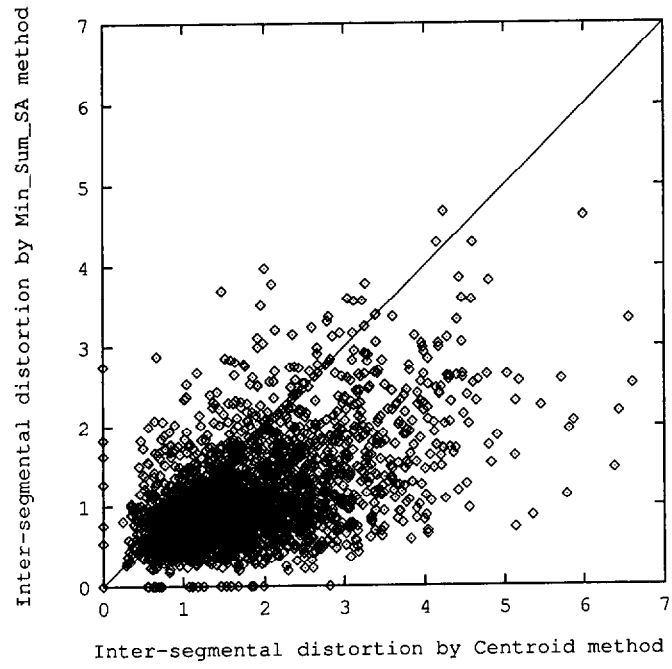


Figure 3.10: Relationship between inter-segmental distortions ( Cepstrum distances ) in units sets selected by **Centroid** and **Min\_Sum\_SA** methods ( $M_s = 10$ )

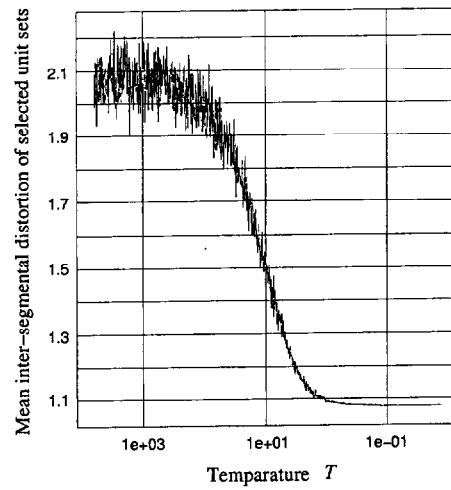


Figure 3.11: Decrease of mean inter-segmental distortion at simulated annealing in **Min\_Sum\_SA** ( $M_s = 10$ )

samples used as stimuli can represent two unit sets without bias in terms of inter-segmental distortion distribution.

In the results, a preference score of 64% was obtained for **b**). This proves that the SSN approach **Min\_Sum\_SA** produces a better quality output than the **Centroid** method. That is, reducing inter-segmental distortions in a unit set improves the quality of synthetic speech perceptually, compared with the usual method.

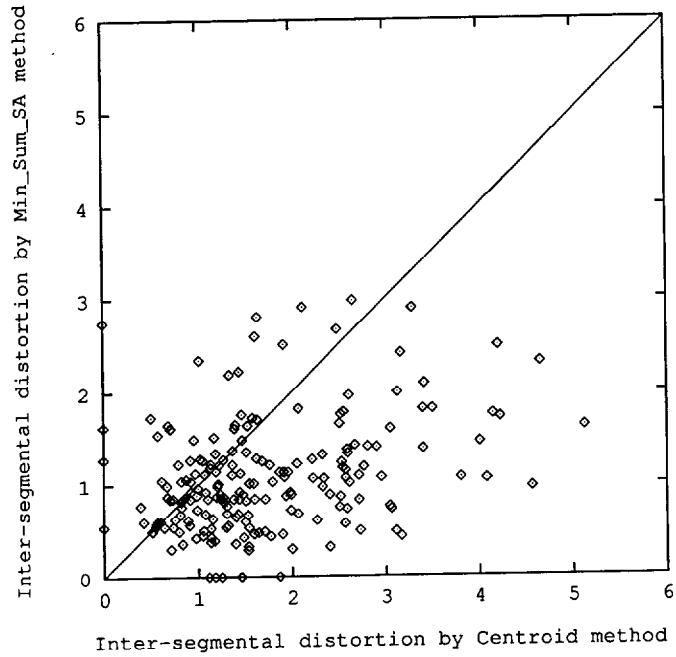


Figure 3.12: Relationship between inter-segmental distortions ( Cepstrum distances ) in synthetic words for perceptual test by using unit sets selected by **Centroid** and **Min\_Sum\_SA** methods ( $M_s = 10$ )

### 3.5 Conclusion

The Speech Segment Network ( SSN ) approach has been proposed for obtaining a synthesis unit set from which high-quality speech can be synthesized, and yet which is small enough to be practical. Such a unit set is selected from a large source database by minimizing segmental and/or inter-segmental distortions. The SSN approach selects an optimal subset from the whole SSN made from a large source

database. Use of simulated annealing or iterative improvement methods overcomes the combinatorial difficulty.

Experimental results for the selection of a diphone unit set showed that the SSN approach efficiently selects a unit set in which the mean and/or maximum of inter-segmental distortion is small. Effectiveness for reducing inter-segmental distortion was enhanced as the segment candidate population increased, while it is not clear whether inter-segmental distortions converge into zero or not.

In the experiments, there is not a large difference between the simulated annealing and iterative improvement methods. However, in finer unit set building with context-dependent units and other perceptually relevant measures, the simulated annealing method has a potential to achieve better quality.

Unit inventory type, which is attributed to the subsets in a source database, is of importance for the quality of synthetic speech. Although diphone units were used in the experiments to check basic performance of the SSN approach, we think that the SSN approach is applicable to longer units, such as varying size units. Besides, in the SSN approach described in this chapter, subsets in a source database, *i.e.* unit type, are given in advance. Preferably, the organization of unit inventories should be optimized as well as the segments in each inventory. Contextual-oriented clustering [Nakajima et al. 88] is a such method, but only intra-segmental distortion can be considered in it. The SSN approach might be able to be extended to the simultaneous optimization of both subsets and segments; both organization of unit inventories and segment in each inventory are optimally chosen. This is one of the subjects of future research.

Furthermore, the procedure of selecting the suitable unit sequence from many possible unit sequences in a large set of units, has been studied to get high quality speech [Takeda et al. 92, Hirokawa and Haokoda 90, Iwahashi et al. 92]. Preferably, unit set should be optimized for the synthesis system with the unit sequence selection scheme, so that suitable unit sequence can be selected from it for arbitrary input utterance. However, the described SSN optimization algorithm assumes that there is only one candidate of unit sequence for each input phoneme sequence, that is, one unit sequence corresponds to each input phoneme sequence without any choice. The extension of the SSN approach to the optimal construction considering the unit sequence selection procedure, is also the subject of future research.

In speech synthesis, automatic optimization methods like the proposed SSN approach depend on the objective measures of the quality of synthetic speech, such as inter-segmental distortion, intra-segmental distortion, etc. However, perfect objective measure which represents the quality of synthetic speech directly has been not

achieved. When considering automatic optimization methods, insufficient objective measure is always a problem. This makes the research on optimization of speech synthesis system difficult. It is necessary to pursue a perceptually more relevant measure. The SSN approach provides the algorithmic scheme for the optimization of a unit set. We think that a better objective measure could enhance the effectiveness of the SSN approach.

# Chapter 4

## Duration Control

### 4.1 Introduction

In text-to-speech synthesis, fine control of prosody, such as segmental duration, fundamental frequency and segmental power, is required for naturalness and intelligibility of synthetic speech. It is well known that the prosody is affected by several inter-dependent factors such as phonemic context, syntactic structure, semantic and pragmatic knowledge, etc. [Klatt 87]. In corpus based synthesis, prosody control modelling has been studied to minimize the difference between observed and predicted values.

Since the number of possible combinations of influential factors for prosody is vast, it is difficult to give quantities for the effects of these factors by manual tuning. Recently, as large databases became more common, statistical approaches have been used for modelling the prosody control to overcome this difficulty. From the view point of the statistical approach, the modelling of the prosody control can be regarded as a problem of regression. In previously proposed methods, simple statistical quantities were used [Bartkova and Sorin 87] to begin with. Further, traditional regression methods such as linear regression and tree regression with categorical predictor variables [Hayashi 50, Breiman et al. 84] have been used [Sakayori 87, Mimura et al. 91, Kaiki et al. 92, Riley 92, Sagisaka and Kaiki 92, Yamashita et al. 93]. However, these regression methods are not necessarily satisfactory as pointed out by [van Santen and Olive 90]. Because control factors often interact with each other in affecting the prosody (e.g. [Port 81]), the regression method should represent such dependence among effects of factors. In addition, it is necessary that a suitable statistical model of prosody control can be obtained with a moderate number of learning samples. Thus, desired is a regression method which can represent properties of the prosody control with as small a number of free parameters as possible.

Linear regression is based on an assumption of linear independence among effects of control factors. This assumption brings reliable inference of the regression model with a small amount of learning samples, but does not model the dependence among the effects. On the other hand, tree regression is based on clustering by a binary recursive split of the factor space, which can represent the dependence among the effects, but can not put constraints of linear independence for reliable inference. These two regression methods have complementary merits and demerits, and merits of both are necessary in a regression for prosody control. To cope with this problem, we propose a new regression method: Constrained Tree Regression (CTR). This regression includes both linear and tree regressions as special cases, and extends them in a model that can represent efficiently both dependence and independence of factors using a small number of free parameters. It uses the constraints of linear independence and a clustering technique, simultaneously. CTR will be described in Sect. 4.3.

In addition, for more adequate control of prosody, hierarchical characteristics should also be considered. It has been pointed out that duration is controlled under constraints among phonologically different hierarchies such as phoneme and syllable [Campbell 91, Campbell 93]. Syllable timing has an important role perceptually as well as phoneme duration [Huggins 68]. Moreover, the hierarchical control model of fundamental frequency has also been proposed [Abe and Sato 92]. In these hierarchical models, however, statistical optimizations have been performed for each level separately. The proposed method can provide overall optimization in multiple hierarchies with an arbitrary error function, and will be described in Sect.4.4. Finally, in Sect. 4.5 we will present results of experiments of modelling segmental duration in English, using both CTR and the hierarchical error function.

## 4.2 Preliminary

Suppose that observed data as learning samples consisting of  $n$  cases  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$  is used to infer a regression model as predictor  $d(\mathbf{x})$ , where  $\mathbf{x}_i : [x_{i1}, \dots, x_{iM}]$  is a vector of values of predictor variables, and  $y_i$  is a real-valued number, the value of the  $i$ th observed data sample. Predictor variable space  $\mathbf{X}$  consists of all possible predictor variable vectors  $\mathbf{x}$  in  $M$ -dimensional space. A predictor is a function  $d(\mathbf{x})$  defined on  $\mathbf{X}$  taking real-value as a response. Assume each predictor variable is categorical, that is:

$$x_{im} \in \{c(m, 1), \dots, c(m, K_m)\}, \quad (4.1)$$

where  $c(m, j)$  is the  $j$ th category for the  $m$ th predictor variable;  $K_m$  is the number of categories for the  $m$ th predictor variable. Predictor  $d(\mathbf{x})$  is constructed so as to minimize the average squared error

$$\frac{1}{n} \sum_{i=1}^n (y_i - d(\mathbf{x}_i))^2. \quad (4.2)$$

### 4.2.1 Linear regression

Linear regression method predicts the value of observed data for predictor variables  $\mathbf{x}$  as a response under the assumption of linear independence among the effects of the predictor variables. Using categorical predictor variables, linear regression predicts value  $y$  of observed data as  $\hat{y}$  according to:

$$\hat{y} = \sum_{j=1}^M \sum_{k=1}^{K_j} \alpha(c(j, k)) \delta_j(c(j, k), \mathbf{x}), \quad (4.3)$$

where  $\alpha(c(j, k))$  is value of the model parameter which is given to the category  $c(j, k)$ .  $\delta_j(c, \mathbf{x})$  is binary function:

$$\delta_j(c, \mathbf{x}) = \begin{cases} 1, & \text{if } x_j = c \\ 0, & \text{otherwise.} \end{cases} \quad (4.4)$$

Because of the assumption of independence, the number of free parameters is represented by  $\sum_{j=1}^M (K_j - 1) + 1$ . The least squared error estimate  $\hat{\alpha}(c(j, k))$  for  $\alpha(c(j, k))$  can be obtained by the solution of the following normal equations:

$$\sum_{j=1}^M \sum_{k=1}^{K_j} f(uv, jk) \hat{\alpha}(c(j, k)) = \sum_{i=1}^n y_i \delta_u(c(u, v), \mathbf{x}_i), \quad (4.5)$$

$$u = 1, 2, \dots, M, \quad v = 1, 2, \dots, K_u.$$

Here, coefficients  $f(uv, jk)$  of the left side of the equation represent the number of samples which are in both  $v$ th category for the  $u$ th predictor variable and the  $k$ th category for the  $j$ th predictor variable simultaneously:

$$f(uv, jk) = \sum_{i=1}^n \delta_u(c(u, v), \mathbf{x}_i) \delta_j(c(j, k), \mathbf{x}_i). \quad (4.6)$$

### 4.2.2 Tree regression

A regression tree is generated by successive splitting of the measurement space  $\mathbf{X}$ . Each subspace  $\mathcal{X}$  obtained by a split is represented as a node  $t$ . The tree regression

method predicts a value of the observed data for predictor variables  $\mathbf{x}$  as values given to terminal nodes as follows:

$$\hat{y} = \sum_{j=1}^N \beta(t_j) \delta(T(\mathbf{x}), t_j). \quad (4.7)$$

Here,  $N$  is the number of terminal nodes in the tree.  $\beta(t_1), \dots, \beta(t_N)$  is the set of model parameters which are given to the terminal nodes  $t_1, \dots, t_N$ , respectively.  $T(\mathbf{x})$  is the terminal node corresponding to predictor variable  $\mathbf{x}$ .  $\delta(T(\mathbf{x}), t)$  is binary function:

$$\delta(T(\mathbf{x}), t) = \begin{cases} 1, & \text{if } T(\mathbf{x}) = t, \\ 0, & \text{otherwise.} \end{cases} \quad (4.8)$$

The number of free parameters is same as the number of terminal nodes,  $N$ . The least squared error estimate  $\hat{\beta}(t_1), \dots, \hat{\beta}(t_N)$  for model parameter  $\beta(t_1), \dots, \beta(t_N)$  can be obtained separately as the mean value of the observed data whose predictor variable vector responds to each terminal node  $t_1, \dots, t_N$ :

$$\begin{aligned} \hat{\beta}(t_j) &= \frac{1}{n_j} \sum_{i=1}^n y_i \delta(T(\mathbf{x}_i), t_j), \\ j &= 1, 2, \dots, N, \end{aligned} \quad (4.9)$$

where,

$$n_j = \sum_{i=1}^n \delta(T(\mathbf{x}_i), t_j).$$

Here,  $n_j$  represents the number of samples which respond to terminal node  $t_j$ .

## 4.3 A New Regression Method

### 4.3.1 Linear Regression as a Tree Model

To facilitate the development of a more desirable regression model, tree structured linear regression is considered here. Although linear and tree regression methods are superficially different, it is possible to make a tree representation equivalent to a linear regression by modification in the formula of tree regression.

For example, in Fig. 4.1, a tree representation equivalent to a linear regression with two predictor variables which include four and two categories respectively is shown. To obtain this representation, model parameters given only at terminal nodes in a conventional tree are first transformed to parameters at all nodes. After this reallocation of parameters, the response of the tree is obtained as a sum of

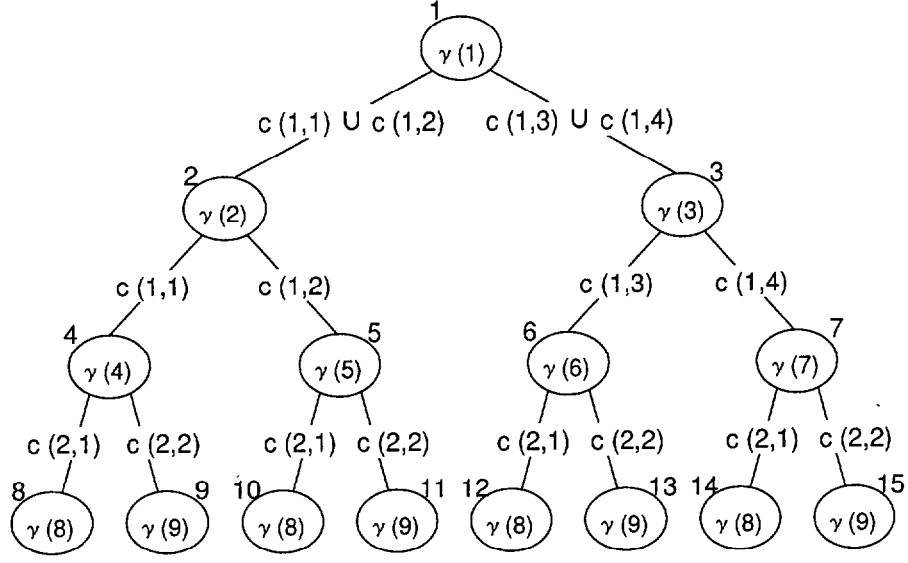


Figure 4.1: *Modified tree representation equivalent to a linear model with categorical predictor variables*

values of parameters given to all nodes through a path from root to corresponding terminal. In addition, particular reallocated parameters should be shared in the tree. For example, parameter  $\gamma(8)$  is shared among nodes 8, 10, 12, 14, and parameter  $\gamma(9)$  is shared among nodes 9, 11, 13, 15. Finally, values of parameters  $\alpha(c(i, j))$  for categories  $c(i, j)$  in the linear regression model can be directly calculated from values of parameter  $\gamma(k)$ ,  $k = 1, \dots, 9$ , in the parameter-reallocated tree using the following formula:

$$\begin{aligned}
 \alpha(c(1, 1)) &= \gamma(1) + \gamma(2) + \gamma(4), \\
 \alpha(c(1, 2)) &= \gamma(1) + \gamma(2) + \gamma(5), \\
 \alpha(c(1, 3)) &= \gamma(1) + \gamma(3) + \gamma(6), \\
 \alpha(c(1, 4)) &= \gamma(1) + \gamma(3) + \gamma(7), \\
 \alpha(c(2, 1)) &= \gamma(8), \\
 \alpha(c(2, 2)) &= \gamma(9).
 \end{aligned}$$

Inversely, we note that a tree regression model can be taken for a non-linear regression with considering sums and products of categories in the predictor variables.

### 4.3.2 Constrained Tree Regression: CTR

The above interpretation between linear and tree regressions is an important step for a desirable regression method. Although the modified tree shown in Fig. 4.1 repre-

sents linear independent effects of two predictor variables for responses by parameter sharing, it is able to represent the dependence in the subspaces, which consist of sums and products of categories, by appropriate changing the combination of the shared parameters. For example, in Fig. 4.2, parameters  $\gamma(10)$  and  $\gamma(11)$  are shared in nodes 10,12 and 11,13, respectively; parameters  $\gamma(10)$  and  $\gamma(11)$  are given to the subspaces  $\{c(1,3) \cup c(1,4)\} \cap c(2,1)$  and  $\{c(1,3) \cup c(1,4)\} \cap c(2,2)$ , respectively, where category  $c(i,j)$  refers to the subspace in which the  $i$ th predictor variable in  $\mathbf{x}$  is  $c(i,j)$ .

Based on this property of the modified tree, we propose a new regression method, Constrained Tree Regression (CTR), which splits predictor variable space successively in a binary manner where allocated parameters can be shared. By parameter sharing, both linear independence and dependence among effects of predictor variables in subspaces can be represented, thereby reducing the number of free parameters needed to represent phenomena affected by inter-dependent factors. This regression method includes linear and tree regressions as special cases and extends them.

The structure of a tree for CTR is similar to a traditional regression tree. The difference is that parameters are given not only to terminal nodes, but also to internal nodes in CTR. It predicts value  $y$  of observed data as the sum of values given to all nodes through a path from root to corresponding terminal node as follows:

$$\hat{y} = \sum_{j=1}^V \gamma(v_j) \delta(\mathcal{X}(v_j), \mathbf{x}). \quad (4.10)$$

Here,  $V$  is the number of nodes in a tree for CTR, and  $\gamma(v_j)$  denotes the value given to node  $v_j$ . Node  $v_j$  is associated to subspace  $\mathcal{X}(v_j)$  in whole predictor variable space  $\mathbf{X}$ .

$$\delta(\mathcal{X}, \mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \mathcal{X} \\ 0, & \text{otherwise.} \end{cases} \quad (4.11)$$

Subspace  $\mathcal{X}$  is represented by the intersection of subspaces specified by each predictor variable, and defined as follows:

$$\mathcal{X} = \bigcap_{m=1}^M C_m(\mathcal{X}), \quad (4.12)$$

where

$$C_m(\mathcal{X}) = \bigcup_{i \in B_m(\mathcal{X})} c(m, i), \quad (4.13)$$

$$B_m(\mathcal{X}) \subset \{1, 2, \dots, K_m\}. \quad (4.14)$$

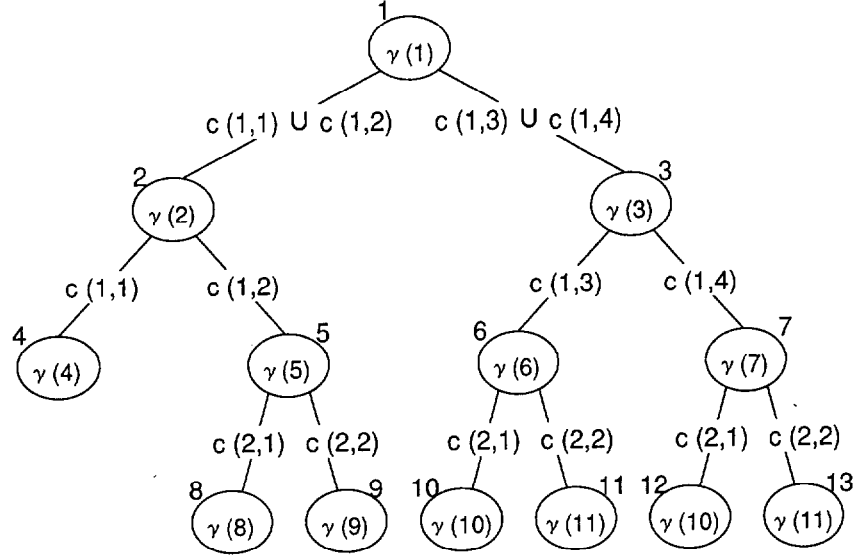


Figure 4.2: *Representation of dependence among effects of categories*

A structure of CTR tree, subspaces for nodes and their corresponding values are decided by learning with data samples. The learning methods will be described in the following section.

### 4.3.3 Algorithm for Generating a CTR Tree

#### Growth of a CTR Tree

For generating a CTR tree, first, the tree grows from a root node which represents whole predictor variable space  $\mathbf{X}$  as in conventional tree regression. In growth of a CTR tree, splits are considered not only at terminal nodes, but also at internal nodes. Split of a node refers to a split of predictor variable space as in conventional tree regression. Split at a terminal node, which is the same process as for tree regression substantially, captures dependence among factors corresponding to the terminal node. Split at an internal node, which is a new process, captures linear independence among factors in the subspace that corresponds to the internal node. Split at an internal node invokes splits of the terminal nodes which are descendants of the split node, generating new terminal nodes. Split of subspace  $\mathcal{X}$  is carried out by splitting the category set,  $\lambda = \{c(m, i) | i \in B_m(\mathcal{X})\}$ , in one of predictor variables, which represents one of subspaces  $C_m(\mathcal{X})$ ,  $m = 1, \dots, M$ .

Let  $T^{(k)}$  denote a tree for CTR after  $k$  successive splits. Split of category set  $\lambda_i$  into category sets  $\lambda_j$  and  $\lambda_k$  is denoted by  $s_c = (\lambda_i, \lambda_j, \lambda_k)$ .  $\lambda_j$  and  $\lambda_k$  are selected

to be disjoint with  $\lambda_j \cup \lambda_k = \lambda_i$ . Split of subspace  $\mathcal{X}_i$  into two subspaces  $\mathcal{X}_j$  and  $\mathcal{X}_k$  is denoted by  $s_s = (\mathcal{X}_i, \mathcal{X}_j, \mathcal{X}_k)$ . Split of node  $v_i$ , which generates two nodes  $v_j$  and  $v_k$ , is denoted by  $s_n = (v_i, v_j, v_k)$ .

Consider the split at node  $v$  in CTR tree  $T^{(k)}$ . The category split  $s_c = (\lambda, \lambda_l, \lambda_r)$ , invokes split  $s_s$  of subspace  $\mathcal{X}(v)$  into subspaces  $\mathcal{X}_l = \mathcal{X}(v) \cap \lambda_l$  and  $\mathcal{X}_r = \mathcal{X}(v) \cap \lambda_r$ . With subspace split  $s_s = (\mathcal{X}, \mathcal{X}_l, \mathcal{X}_r)$ , the parameters are given to the spaces  $\mathcal{X}_l$  and  $\mathcal{X}_r$ , denoted by  $\xi(\mathcal{X}_l)$  and  $\xi(\mathcal{X}_r)$ , respectively. Then space split  $s_s$  invokes splits of all terminal nodes in the subtree  $T_v^{(k)}$  rooted by node  $v$ . The set of terminal nodes in  $T_v^{(k)}$  is denoted by  $\bar{T}_v^{(k)}$ . The set,  $S_n(s_s)$ , of the node splits invoked by subspace split  $s_s$  on tree  $T^{(k)}$  is as follows:

$$S_n(s_s) = \{(v_t, L(v_t), R(v_t)) \mid v_t \in \bar{T}_v^{(k)}\}, \quad (4.15)$$

where  $L(v_t)$  and  $R(v_t)$  are the derived nodes from node  $v_t$ . The corresponding spaces and parameters are given for all  $L(v_t)$  and  $R(v_t)$  such that  $v_t \in \bar{T}_v^{(k)}$  by

$$\mathcal{X}(L(v_t)) = \mathcal{X}(v_t) \cap \mathcal{X}_l, \quad (4.16)$$

$$\mathcal{X}(R(v_t)) = \mathcal{X}(v_t) \cap \mathcal{X}_r, \quad (4.17)$$

and

$$\gamma(L(v_t)) = \xi(\mathcal{X}_l), \quad (4.18)$$

$$\gamma(R(v_t)) = \xi(\mathcal{X}_r). \quad (4.19)$$

Equations (4.18) and (4.19) imply that parameters  $\xi(\mathcal{X}_l)$  and  $\xi(\mathcal{X}_r)$  are shared by all nodes  $L(v_t)$  and  $R(v_t)$  such that  $v_t \in \bar{T}_v^{(k)}$ , respectively.

For instance, the CTR tree shown in Fig. 4.2 is obtained from the CTR tree shown in Fig. 4.3 by splitting the subspace in node 3,

$$\mathcal{X}(3) = \{c(1, 3) \cup c(1, 4)\} \cap \{c(2, 1) \cup c(2, 2)\},$$

into subspaces:

$$\mathcal{X}_l = \{c(1, 3) \cup c(1, 4)\} \cap c(2, 1),$$

$$\mathcal{X}_r = \{c(1, 3) \cup c(1, 4)\} \cap c(2, 2).$$

This split results in new terminal nodes 10, 11, 12, 13, which respectively correspond to the following subspaces:

$$\mathcal{X}(10) = \mathcal{X}(6) \cap \mathcal{X}_l = c(1, 3) \cap c(2, 1),$$

$$\mathcal{X}(11) = \mathcal{X}(6) \cap \mathcal{X}_r = c(1, 3) \cap c(2, 2),$$

$$\mathcal{X}(12) = \mathcal{X}(7) \cap \mathcal{X}_l = c(1, 4) \cap c(2, 1),$$

$$\mathcal{X}(13) = \mathcal{X}(7) \cap \mathcal{X}_r = c(1, 4) \cap c(2, 2).$$

The values given to these derived nodes are:

$$\begin{aligned}\gamma(10) &= \gamma(12) = \xi(\mathcal{X}_l), \\ \gamma(11) &= \gamma(13) = \xi(\mathcal{X}_r).\end{aligned}$$

A CTR tree is constructed by sequence  $S_s$  of  $p$  successive splits of subspace from the whole predictor variable space  $\mathbf{X}$ , which corresponds to root node  $v_0$  with parameters  $\gamma(v_0)$  and  $\xi(\mathbf{X})$ , where  $\gamma(v_0) = \xi(\mathbf{X})$ .  $S_s$  is defined as follows:

$$\begin{aligned}S_s &= (s_s(1), \dots, s_s(p)) \\ &= ((\mathcal{X}_0 = \mathbf{X}, \mathcal{X}_1, \mathcal{X}_2), (\mathcal{X}_{a(2)}, \mathcal{X}_3, \mathcal{X}_4), \\ &\quad \dots, (\mathcal{X}_{a(p)}, \mathcal{Z}_{2p-1}, \mathcal{Z}_{2p})),\end{aligned}\tag{4.20}$$

where  $s_s(i)$  is  $i$ th split;  $\mathcal{X}_{a(k)}$  is the subspace split in  $s_s(k)$ . The suitable split sequence is determined by *best-first* search strategy, although more complicated search strategies can be applied. In each step of growth by splitting subspaces, the best split  $s_s^*(k)$ , is selected so as to minimize an average squared error,

$$g(T^{(k)}, s_s^*(k)) = \min_{s_s(k)} g(T^{(k)}, s_s(k)),\tag{4.21}$$

$$k = 1, 2, \dots, p,$$

where  $g(T^{(k)}, s_s)$  represents an average squared error by the tree obtained by carrying out split  $s_s$  on tree  $T^{(k)}$ . An efficient search method for the best split is described in Appendix B. Each split derives new terminal nodes and their parameter sharing. Splits are successively carried out to construct a suitable size tree.

It is easily understandable that CTR is equivalent to linear regression when only the root node is allowed to be split, and to tree regression when only terminal nodes are allowed to be split.

### Parameter estimates

The values of parameters for nodes in tree  $T$  are estimated so as to minimize an averaged squared error  $G(T)$  obtained by tree  $T$ . Suppose that CTR tree  $T$  is constructed from root node  $v_0$  by the split sequence in Eq. (4.20). Allocated are parameters  $\xi(\mathcal{X}_0)$ ,  $\xi(\mathcal{X}_1)$ ,  $\xi(\mathcal{X}_2)$ ,  $\dots$ ,  $\xi(\mathcal{X}_{2p-1})$ ,  $\xi(\mathcal{X}_{2p})$ . One free parameter is added in each split, while two new subspaces, to which corresponding parameters are given, are derived. This allocation of free parameters is based on the fact that for split  $(\mathcal{X}, \mathcal{X}_l, \mathcal{X}_r)$  there exists the following linear constraint:

$$\delta(\mathcal{X}, \mathbf{x}) = \delta(\mathcal{X}_l, \mathbf{x}) + \delta(\mathcal{X}_r, \mathbf{x}).\tag{4.22}$$

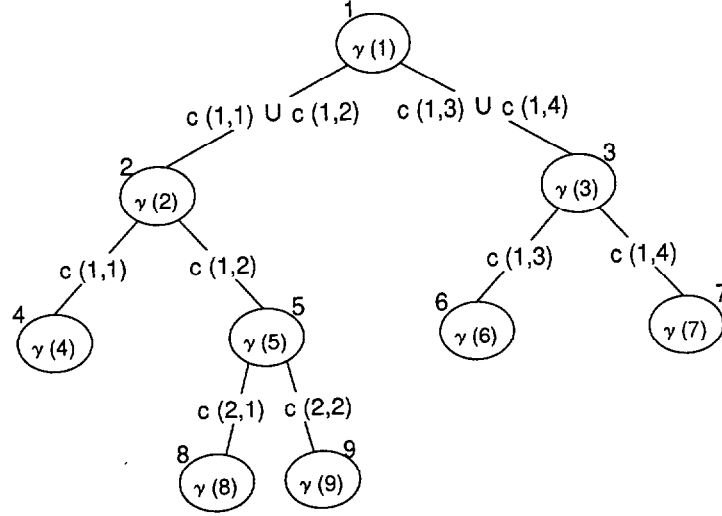


Figure 4.3: Before split on the tree in Fig. 4.2

In the experiments described in this chapter, parameters  $\xi(\mathcal{X}_0), \xi(\mathcal{X}_2), \dots, \xi(\mathcal{X}_{2p})$  are used for free parameters, and  $\xi(\mathcal{X}_1), \xi(\mathcal{X}_3), \dots, \xi(\mathcal{X}_{2p-1})$  are all fixed to 0, although other ways for the allocation of free parameters is possible.

The least squared error estimates  $\hat{\xi}(\mathcal{X}_0), \hat{\xi}(\mathcal{X}_2), \dots, \hat{\xi}(\mathcal{X}_{2p})$  can be calculated as the solution of the following equations, by setting  $\frac{\partial G(T)}{\partial \xi(\mathcal{X}_{2i})} = 0, i = 0, 1, \dots, p;$

$$\sum_{j=0}^p f(\mathcal{X}_{2u}, \mathcal{X}_{2j}) \hat{\xi}(\mathcal{X}_{2j}) = \sum_{i=1}^n y_i \delta(\mathcal{X}_{2u}, \mathbf{x}_i), \quad (4.23)$$

$$u = 0, 1, \dots, p.$$

Here,  $n$  is the number of samples, and coefficients  $f(\mathcal{X}_u, \mathcal{X}_v)$  of the left side of the equation represent the number of samples which respond to both subspaces  $\mathcal{X}_u$  and  $\mathcal{X}_v$  simultaneously:

$$f(\mathcal{X}_u, \mathcal{X}_v) = \sum_{i=1}^n \delta(\mathcal{X}_u, \mathbf{x}_i) \delta(\mathcal{X}_v, \mathbf{x}_i). \quad (4.24)$$

## 4.4 Hierarchical Representation in Regression

In this section, a new method is proposed for handling the hierarchical structure of observed data. Let the observed data with two level hierarchical structure consist of  $n$  cases in a high level,  $(\mathbf{x}_1^{(H)}, y_1^{(H)}), (\mathbf{x}_2^{(H)}, y_2^{(H)}), \dots, (\mathbf{x}_n^{(H)}, y_n^{(H)})$ . To each high level case  $(\mathbf{x}_i^{(H)}, y_i^{(H)})$ , associated is the low level samples,  $(\mathbf{x}_{i,1}^{(L)}, y_{i,1}^{(L)}), (\mathbf{x}_{i,2}^{(L)}, y_{i,2}^{(L)}), \dots, (\mathbf{x}_{i,n'(i)}^{(L)}, y_{i,n'(i)}^{(L)})$ , where  $n'(i)$  is the number of the low level samples associated

with the  $i$ th high level case. Let function  $F$  represent the relationship between data in two hierarchical levels:

$$y_i^{(H)} = F(y_{i,1}^{(L)}, \dots, y_{i,n'(i)}^{(L)}). \quad (4.25)$$

$F$  is given depending on the problem. For speech segmental duration, syllable and phoneme duration correspond to high and low level data, respectively, and syllable duration is the sum of all phoneme durations in the syllable.

Clearly, the least squared error estimate in the low level data does not necessarily minimize the squared error in the high level data. To minimize predictive errors for both levels, a hierarchical error function  $Q$  is defined as:

$$\begin{aligned} Q \equiv & w \sum_{i=1}^n \sum_{j=1}^{n'(i)} (y_{i,j}^{(L)} - \hat{y}_{i,j}^{(L)})^2 \\ & + (1 - w) \sum_{i=1}^n (y_i^{(H)} - \hat{y}_i^{(H)})^2 \end{aligned} \quad (4.26)$$

where  $\hat{y}_{i,j}^{(L)}$  is a response value from a predictor for the  $j$ th low level sample in the  $i$ th high level case, and  $\hat{y}_i^{(H)}$  is a response value for the  $i$ th high level case, which can be calculated by  $F(\hat{y}_{i,1}^{(L)}, \dots, \hat{y}_{i,n'(i)}^{(L)})$ .  $w$  is a weighting coefficient.

If function  $F$  gives the sum of all input elements, estimate of CTR model parameters, which minimize the hierarchical error function, can be got by the solution of the following equations:

$$\begin{aligned} & \sum_{k=0}^p \hat{\xi}(\mathcal{X}_{2k}) [w f_L(\mathcal{X}_{2u}, \mathcal{X}_{2k}) + (1 - w) f_H(\mathcal{X}_{2u}, \mathcal{X}_{2k})] \\ & = w \sum_{i=1}^n \sum_{j=1}^{n'(i)} y_{i,j}^{(L)} \delta_L(\mathcal{X}_{2u}, \mathbf{x}_{i,j}^{(L)}) \\ & \quad + (1 - w) \sum_{i=1}^n y_i^{(H)} \delta_H(\mathcal{X}_{2u}, \mathbf{x}_i^{(H)}), \end{aligned} \quad (4.27)$$

$u = 0, 1, \dots, p.$

Here,

$$\begin{aligned} f_L(\mathcal{X}_{2u}, \mathcal{X}_{2k}) &= \sum_{i=1}^n \sum_{j=1}^{n'(i)} \delta_L(\mathcal{X}_{2u}, \mathbf{x}_{i,j}^{(L)}) \delta_L(\mathcal{X}_{2k}, \mathbf{x}_{i,j}^{(L)}) \\ f_H(\mathcal{X}_{2u}, \mathcal{X}_{2k}) &= \sum_{i=1}^n \delta_H(\mathcal{X}_{2u}, \mathbf{x}_i^{(H)}) \delta_H(\mathcal{X}_{2k}, \mathbf{x}_i^{(H)}) \end{aligned}$$

$\delta_L(\mathcal{X}_k, \mathbf{x}_{i,j}^{(L)})$  is 1 if the  $j$ th low level sample in the  $i$ th high level case is in subspace  $\mathcal{X}_k$ , otherwise 0.  $\delta_H(\mathcal{X}_k, \mathbf{x}_i^{(H)})$  is the number of the low level samples, that is in  $\mathcal{X}_k$ ,

corresponding to the  $i$ th high level data.  $\mathbf{x}_i^{(H)}$  is just an identity of the  $i$ th high level sample. The hierarchical error function is also applicable to linear and tree regression, and parameter is estimated by similar manners.

## 4.5 Experiments on Speech Segmental Duration Modelling

In this section, the prediction of speech segment duration is modelled using CTR. The effect of employing of a hierarchical error function is also studied.

### 4.5.1 Speech Corpus and Control Factors

Two hundred sentences uttered by an English male speaker were used as duration data [Laber et al. 88]. Phoneme and syllable counts were 8715 and 3317, respectively. Mean durations of phoneme and syllable were 71 ms and 187 ms, respectively. The control factors considered as predictor variables are listed in Table 4.1. Two factor sets,  $F_S$  and  $F_{S+P}$ , from these factors were prepared to investigate interactions between control factors in different hierarchical levels: syllable and phoneme levels. Set  $F_S$  (10 factors, 39 categories) consists of all factors except the identity of the phoneme; this is the syllable level factor set. Set  $F_{S+P}$  (11 factors, 85 categories) consists of all factors in Table 4.1; this contains both syllable level factors and a phoneme level factor (the identity of the phoneme). The contribution of each of these factors to the control of segment duration has been investigated in detail in [Campbell 92].

### 4.5.2 Obtained Statistical Models

Using the above data, statistical models for phoneme and syllable durations were obtained by CTR. For comparison, models were also obtained by linear and tree regressions. Linear regression was carried out by CTR in which only the root node was allowed to be split. Tree regression was carried out by CTR in which only leaf nodes were allowed to be split.

For phoneme duration models, factor set  $F_{S+P}$  was used as predictor variables, and the number of free parameters was 75 in all three types of regression. For syllable duration models, set  $F_S$  and 30 free parameters are used. Multiple correlations are shown as 'Linear', 'Tree' and 'CTR' for each regression model in Fig. 4.4 and Fig. 4.5. In these results, we can see that CTR got the highest value of multiple correlation

Table 4.1: *Predictor variables for speech segmental duration*

| factors                                    | categories           |
|--------------------------------------------|----------------------|
| the number of phonemes in the syllable     | 1, 2, 3, 4, 5, 6, 7  |
| the strength of the phrase break           | four classes         |
| the nature of the rhyme                    | open, close          |
| the nature of the syllabic peak            | four classes         |
| the strength of the stress                 | four classes         |
| the position in the foot                   | head, others         |
| the number of syllables in the foot        | 1, 2, 3, 4, 5, 6     |
| the position in the word                   | 1, 2, 3, 4           |
| the position of the phrase in the sentence | four classes         |
| the part-of-speech distinction             | function, content    |
| the identity of the phoneme                | forty-six categories |

with the same number of free parameters in both phoneme and syllable duration modelling. In addition, the values of the standard deviation of predictive errors in these models are shown in Table 4.2 and Table 4.3.

The relationship between the number of free parameters and the value of multiple correlations for phoneme duration modelling is shown in Fig. 4.6. We can see that with any number of free parameters CTR got the highest multiple correlation value compared with linear and tree regressions.

A simplified structure of an obtained CTR tree for phoneme duration is shown in Fig. 4.7. The nodes at which space splits were carried out and the factors used in the corresponding category splits are also shown. We can see that all splits except the sixth and seventh splits were substantially split of the whole factor space. The sixth and seventh splits were carried out not on the whole factor space but on its subspaces. The sixth split captured the dependence of three factors: the phoneme identity, the position of phrase in the sentence and the number of phonemes in the syllable. The seventh one captured the dependence of the phoneme identity and stress index. In addition, the third split also seems to be carried out on a subspace superficially, but it was substantially the one on the whole factor space, because it was re-split of the subspace derived by the first split, using the same factor that the first split used.

Moreover, for the parameter estimate of a phoneme duration model using CTR, the hierarchical error function was used where the value of  $w$  is set to 0 so as to

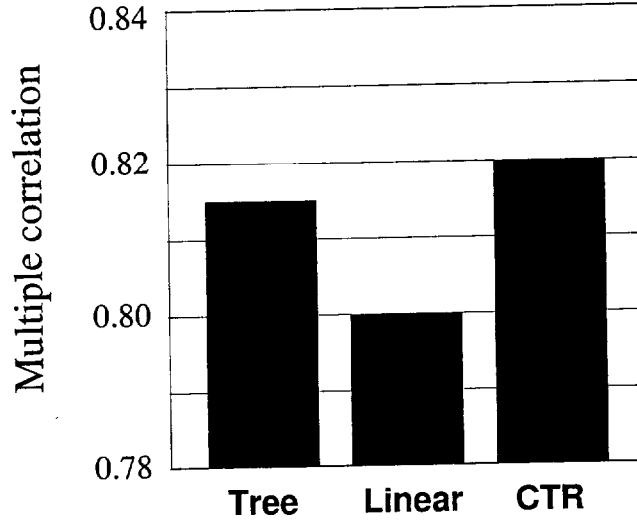


Figure 4.4: *Multiple correlation in models for phoneme duration*

Table 4.2: *Standard deviation of the predictive errors in the models for phoneme duration*

|                | Tree | Linear | CTR  |
|----------------|------|--------|------|
| Std. dev. (ms) | 19.6 | 20.3   | 19.4 |

optimize only syllable duration. The model was trained using factor set  $F_{S+P}$  as predictor variables with 30 free parameters. Needless to say, the predicted value of the syllable duration can be obtained from the sum of phoneme durations. Multiple correlation on syllable duration by this model, is shown as ‘CTR+’ in Fig. 4.5. We can see that this value is higher than value by ‘CTR’ in which factor set  $F_S$  was used with the same number of free parameters.

Figure 4.8 shows the effect of  $w$  in the hierarchical error function:  $w$  was set to 1 or 0 to optimize phoneme/syllable duration. We can see the accuracy trade-off between phoneme and syllable durations was adjusted by changing the value of  $w$ .

### 4.5.3 Discussion

From the results of the modelling experiments, we can see that CTR could model speech segmental duration more efficiently than linear and tree regressions with the same number of free parameters. Figure 4.7 shows the CTR’s capability both in

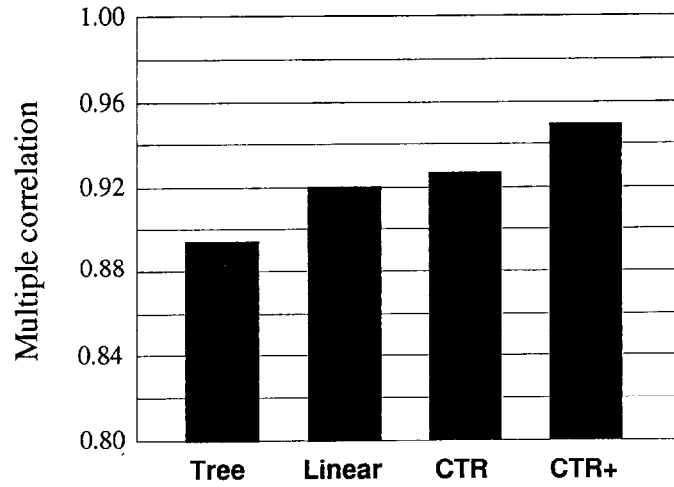


Figure 4.5: *Multiple correlation in models for syllable duration*

Table 4.3: *Standard deviation of the predictive errors in the models for syllable duration*

|                | Tree | Linear | CTR  | CTR+ |
|----------------|------|--------|------|------|
| Std. dev. (ms) | 43.7 | 38.1   | 36.6 | 30.7 |

representing dependence among effects of the control factors and in putting the constraints of linear independence. These results shows that CTR has the merits of both linear and tree regression methods. This characteristics of CTR makes it possible to model complex effects of control factors suitably according to given learning samples. In the case when the number of the learning samples related to certain factors is small, CTR imposes the constraints of the linear independence between such factors and the others in the model for reliable inference using small number of free parameters efficiently as the way of linear regression. While there are a large amount of the learning samples related to certain factors, CTR represents the dependence among such factors for precise model using larger number of parameters as the way of tree regression. Particularly, CTR has advantage in modelling the prosodic characteristics by considering the complex effects of control factors, such as speech rate, prominence, intention, emotion and etc., because both the suitable representation of complex effects and the reliable inference using sparse data are necessary in the modelling.

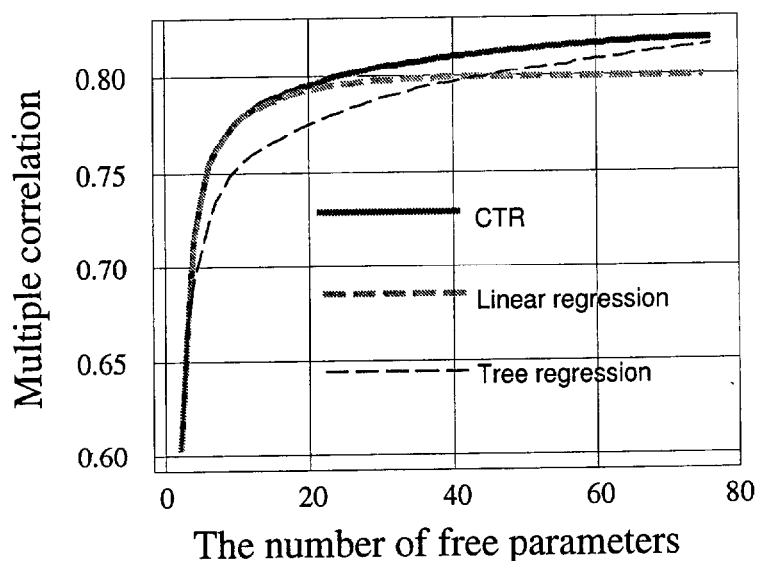


Figure 4.6: *Relationship between the number of free parameters and multiple correlation*

In Fig. 4.6, difference between tree regression and CTR is small with the large number of free parameters. This tendency comes from the essential nature of CTR. The increase of the number of parameters in CTR implies the decrease of the number of shared parameters. Thus, as the number of free parameters increases, a CTR tree becomes closer to a conventional regression tree. In addition, we should note that only the first twenty-four nodes were allowed to be split to reduce computational time in the experiments. By loosening this limitation, better results might be obtained.

Comparison of the results of 'CTR' and 'CTR+' in Fig. 4.5 shows that syllable duration could be modelled more adequately by both syllable and phoneme level factors with the hierarchical error function than by only syllable level factors. This gives evidence of interaction of phoneme and syllable level control factors for syllable duration. In the 'CTR+' model, all factors (one phoneme level factor and ten syllable level factors) were used to predict the response. Eight parameters were allocated to the phoneme level factors, and others were allocated to the syllable level factors. These results may show that syllable duration is controlled by both syllable level constraints and the effect of phoneme characteristics. It is interesting that the model from which response was phoneme duration gave more precise syllable duration than the model from which response was directly syllable duration, although considered categories were different.

When a phoneme level factor would be used for the prediction of syllable dura-

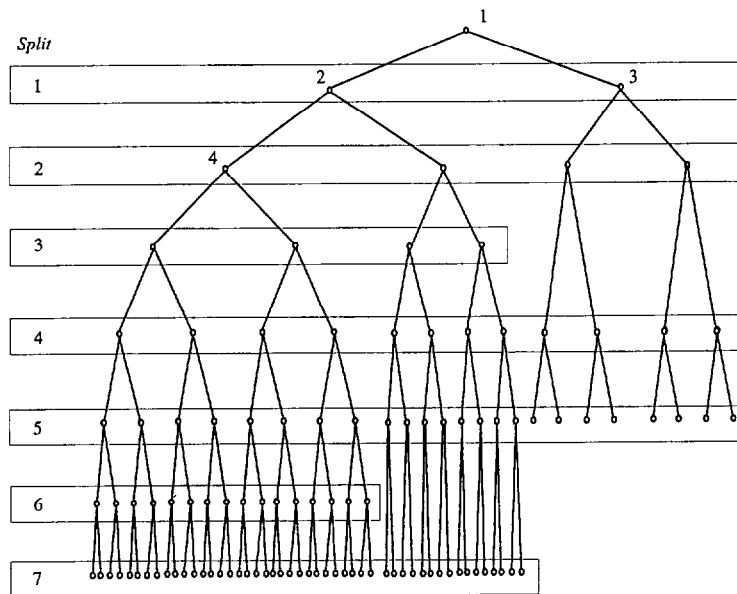


Figure 4.7: *Simplified structure of a CTR tree for phoneme duration model*

tion, unless the hierarchical error function is used, phoneme level factors attributed to all phonemes in each syllable must be attached to the syllable observation as predictor variables. This leads to an undesirable increase of the complexity in inferring model. The hierarchical error function makes the model structure very simple because a phoneme duration predictor can be used to get suitable syllable duration. It can be said that the hierarchical error function provided the framework for the efficient use of multi level factors. Actually, the experimental results show that CTR with the hierarchical error function could represent adequate syllable duration by efficient use of both syllable and phoneme level factors.

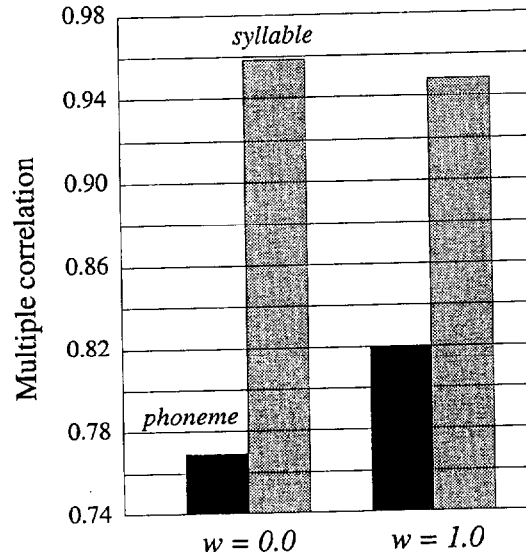


Figure 4.8: *Effect of weight  $w$  in hierarchical predictive error function*

## 4.6 Conclusion

We have described the new regression methods: Constrained Tree Regression (CTR) and the hierarchical error function, for statistical modelling of prosody control in speech synthesis. CTR has desirable characteristics to represent dependence among effects of prosody control factors. The hierarchical error function makes it possible to utilize hierarchical characteristics efficiently in prosody control. By using these methods, smaller prediction errors in speech segmental durations could be achieved than by using conventional methods with same number of free parameters.

CTR has advantage in representing the prosodic characteristics when incorporating more complex utterance conditions : speech rate, prominence, intention, emotion and etc. because CTR can handle both complex effects (dependence) of factors and sparseness of data in such cases properly. For example, analysis of fundamental frequency control by using CTR has been successfully carried out [Hirai et al. 96], where some important characteristics of the control could be found.

Whilst CTR has been described mainly from the view point of tree structures in this paper, CTR can also be described as non-linear regression also. In a sense, CTR can be taken as a special case of non-linear regression with categorical predictor variables [Kobayashi et al. 91], which was applied to the analysis of contextual effects in speech spectrum. In fact, a set of all spaces  $\mathcal{X}_j$ ,  $j = 0, 2, \dots, 2p$ , in Eq. (4.23) corresponds to a polynomial form of categorical variables in non-linear regression. We point out, however, that CTR assigns shared parameters for coefficients in a

polynomial form, and the space split strategy makes it possible to find appropriate parameter sharing. This scheme of the space split plays an important role in generalizing linear and tree regressions.

Furthermore, deciding the size of a CTR tree is a problem to get a suitable statistical model. That is, a stopping rule for the split process should be considered as in decision tree [Breiman et al. 84, Quinlan 88]. A generality check procedure such as cross validation, would be applicable. The problem of the stopping rule is related to the problem of how do we search the best CTR structure. This is another general problem in finding the best tree. The growth process for a CTR tree described in Sect. 4.3 is based on a *best-first* search strategy. For improvement, more powerful search methods, such as tree pruning technique [Chou et al. 89], branch-and-bound method, beam search, and relaxation techniques, would be applicable.

## Chapter 5

# Adaptation of Speaker Characteristics

### 5.1 Introduction

Control of speech individuality in speech synthesis is of interest for the construction of user-friendly man-machine interfaces. The control of voice quality of an individual speaker has been studied through many control factors [Price 89, Childer 89, Klatt 90] both for source-resonant spectral characteristics and prosodic characteristics. In this chapter, we will focus on controlling spectral envelope characteristics in speech synthesis. Spectral envelope information has an important role on speech individuality [Kuwabara 87] as other factors. In the spectral control method, it is desired that the speech spectrum is controlled over a wide range, and it is necessary to keep speech quality high with such control. To satisfy these conditions both for flexibility and stability of control, it is important to represent both the intrinsic characteristics and intra-speaker consistency suitably for the control. That is, it is necessary that the characteristics of the speech spectral structure and dynamics are modelled suitably and that the transformation is carried out under constraints based on such a spectral model.

So far, emphasizing/suppressing the dynamics of the formant trajectory [Takagi 86] and manipulating the pole frequency of the spectrum for male-female conversion [Hakoda 87] have been proposed as speech transformation methods. In these transformation methods, although stable control of the speech spectrum is possible, little variation in speech individuality is obtained. This is due to the constraints used being too strong, resulting in a low degree of freedom for the control. To get a more flexible transformation, articulatory modelling on vocal tract shape and its dynamics can be considered. However, it is difficult to synthesize high quality speech by

such articulatory modelling at present, except for the manually tuned formant-based synthesizer [Klatt 80, 90].

One application of the spectral transformation is adaptation for a target speaker. Several previous approaches have been studied for voice conversion based on spectral mapping techniques, such as VQ codebook mapping [Abe 88, 91], linear multivariate regression and dynamic frequency warping [Valbret 92]. In these mapping methods, no constraints on speech spectrum consistency are used. In these approaches, therefore, in order to obtain a suitable mapping into the target speaker's spectrum, a large amount of spectrum data from the target speaker is needed. In the case when it is not possible to extract such large amounts of data, these previous methods do not work well. A speaker adaptation method using only a small amount of spectrum data to model the target speaker characteristics is the preferred way of achieving this.

On the other hand, in automatic speech recognition, it has been reported [Kosaka 93] that adaptation of only probabilistic weights on multiple speakers' HMM models is effective when a small amount of training data is available. It can be taken that this depends on both the use of intra-speaker consistency and adaptation by a small number of adapted parameters. It can be expected that a similar method is applicable to the spectral adaptation method in speech synthesis.

We propose a spectral transformation method by speaker interpolation. A new spectrum sequence is generated by interpolating multiple speakers' spectral information. This method is based on non-parametric modelling which consists of multiple speakers' spectral sequences for speech spectrum transformation. Speech spectrum sequences from each speaker represent speaker consistency directly, and work as constraints of control.

Moreover, to get a more precise transformation into the target speaker's spectrum when a large amount of training data is available, use of multiple interpolating functions can be considered. We propose a multi-functional representation with weighting by Radial Basis Function networks for spectral transformation. This multi-functional transformation has more freedom than the single interpolation function for the adaptation.

In the following, firstly the spectrum transformation by the speaker interpolating function will be formulated. Subsequently the speaker adaptation method based on the speaker interpolating function will be described. Finally, extending the interpolating scheme, the multi-functional representation with weighting by Radial Basis Function networks for the spectral transformation will be described.

## 5.2 Spectrum Transformation by Speaker Interpolation

For spectrum transformation by speaker interpolation, the spectrum data of utterances spoken by a moderate number of speakers is pre-stored in the synthesis system. The block diagram of the speaker interpolation is shown in Fig. 5.1. After the spectrum sequences of the same utterance by multiple speakers are time-aligned by DTW, the interpolation is carried out between these DTW-ed spectrum sequences by the following transformation:

$$\mathbf{Y}_i = F_r(\mathbf{X}_i) \quad (5.1)$$

$$= \mathbf{X}_i \cdot \mathbf{a} \quad (5.2)$$

where

$$\mathbf{X}_i = \begin{bmatrix} x_{i11} & \cdot & \cdot & x_{i1K} \\ \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot \\ x_{iJ1} & \cdot & \cdot & x_{iJK} \end{bmatrix} \quad (5.3)$$

$$\mathbf{Y}_i^T = [Y_{i1}, Y_{i2}, \dots, Y_{iJ}] \quad (5.4)$$

$$\mathbf{a}^T = [a_1, \dots, a_K] \quad (5.5)$$

under the constraint

$$\sum_{k=1}^K a_k = 1. \quad (5.6)$$

Here,  $x_{ijk}$  represents the  $j$ th spectral parameter ( *e.g.* cepstrum ) of the  $i$ th frame in a DTW-ed utterance of the  $k$ th speaker.  $J$  is the order of the spectral parameters.  $K$  is the number of pre-stored speakers, and  $a_k$  is the interpolation coefficient for the  $k$ th speaker.  $Y_{ij}$  represents the  $j$ th spectral parameter of the  $i$ th frame of the spectrum generated by interpolation.

In this transformation, the following characteristics are desired for the interpolated parameters:

1. Interpolation is always stable: By using parameters obtained by interpolation, the waveform should always be stably synthesized. For example, cepstrum parameter is stable, but LPC not.
2. The spectral sensitivity is flat: Large differences of sensitivity to spectrum in parameter space make the optimization procedure difficult.

3. A formant structure is interpolated naturally by interpolation: To generate high quality speech, it is desirable to keep the formant structure as natural as possible.

While spectral parameters which satisfy these conditions completely do not seem to be found at present, cepstrum and log area ratio parameters have comparatively good characteristics [Viswanathan 75, Atal 83], and we choose them as spectral parameters.

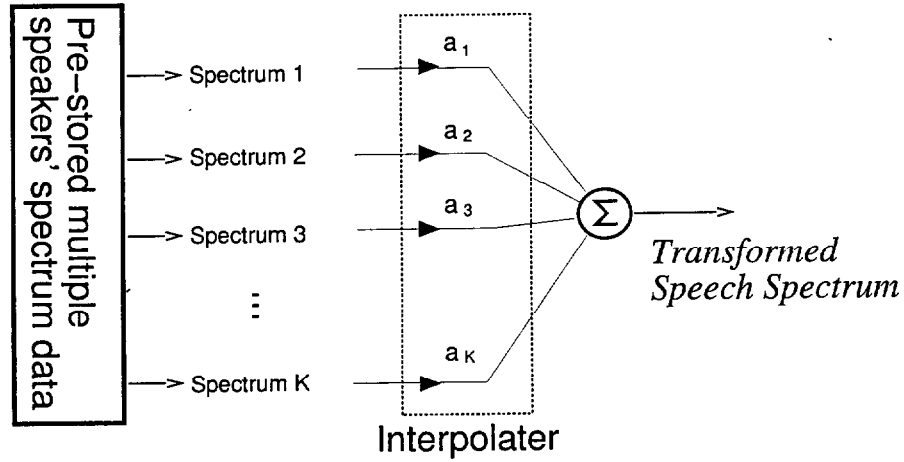


Figure 5.1: Block diagram of speaker interpolation

### 5.3 Speaker Adaptation by Speaker Interpolation

For adaptation to a target speaker, the optimal interpolation ratio is determined so as to minimize the distance between the target speaker's spectrum and the spectrum generated by interpolation.

The pre-stored data of the multiple speakers directly provide the information on speech spectral consistency, characteristics of spectral structure and its dynamics in natural speech. In the speaker interpolation, such consistency works as a reasonable constraint for the adaptation, although the interpolation might not always generate enough natural speech spectra due to characteristics of the interpolation parameters and other possible factors. Therefore, adaptation by a small amount of spectral data of a target speaker is possible. The adaptation procedure is as follows:

1. Get the waveform data for one word uttered by a target speaker.
2. Calculate sequences of spectral parameters of the word.

3. Time-align the target speaker's utterance and pre-stored speaker's utterance by DTW with the cepstrum distance measure.
4. Determine the optimal interpolation coefficients  $a_1, a_2, \dots, a_K$  between pre-stored speakers by minimization of the function

$$Q_r(\mathbf{a}) = \sum_{i=1}^N \|\mathbf{Y}_i - \mathbf{y}_i\| \quad (5.7)$$

$$= \sum_{i=1}^N \|\mathbf{X}_i \cdot \mathbf{a} - \mathbf{y}_i\| \quad (5.8)$$

where

$$\mathbf{y}_i^T = [y_{i1}, y_{i2}, \dots, y_{iJ}]. \quad (5.9)$$

Here,  $y_{ij}$  represents the  $j$ th spectral parameter of the  $i$ th frame of the spectrum of the target speaker.  $N$  is the number of the training samples. This minimization is done by solving normal equations for linear regressive analysis.  $\mathbf{Y}_i$  is the new spectral parameter vector, calculated by Equation 5.2.

5. Generate spectral parameter sequences for any other speech utterances by interpolation with the determined interpolation coefficients between the pre-stored speakers' utterances.
6. Synthesize speech with the new spectral sequences.

## Increasing the adaptation freedom

For more precise adaptation to the target, larger flexibility is desired in the adaptation scheme. Increasing the number of adapted parameters should be pursued for this purpose. Constant terms are adopted with the interpolation coefficients in the case of using cepstrum parameters as interpolated parameters. In this case, the adaptation is carried out by minimizing the following function:

$$Q_a(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^N \|\mathbf{Y}_i - \mathbf{y}_i\| \quad (5.10)$$

where

$$\mathbf{Y}_i = F_a(\mathbf{X}_i) \quad (5.11)$$

$$= \mathbf{X}_i \cdot \mathbf{a} + \mathbf{b} \quad (5.12)$$

$$\mathbf{b}^T = [b_1, \dots, b_J] \quad (5.13)$$

Here,  $b_j$  is the  $j$ th cepstrum parameter.  $F_a(\cdot)$  is a linear function, whose parameters are decided by solving normal equation in the adaptation procedure as well as the interpolation coefficients.

## 5.4 Multi-functional Representation with Weighting by Radial Basis Function Networks

### 5.4.1 Advantages of using multiple linear functions

A single linear function in Eq. (5.12) might be not enough to precisely represent the mapping characteristics from multiple speakers' spectra to those of target speaker, because localities of mapping characteristics may remain even though speech spectrum modelling by multiple speakers' spectra would be effective. Therefore, use of multiple linear functions can be considered, where each linear function is applied in spectral subdivisions of the entire spectral space. This subdivided space scheme can be extended to give weight vectors for each linear function at any point in the spectral space. This can be regarded as a fuzzy subdividing of the space. The formulation of the mapping transformation by weighted multiple linear functions is

$$\mathbf{Y}_i = [F_{a1}(\mathbf{X}_i), \dots, F_{aL}(\mathbf{X}_i)] \cdot \mathbf{g}_i \quad (5.14)$$

$$= (\mathbf{X}_i \cdot \mathbf{A} + \mathbf{B}) \cdot \mathbf{g}_i \quad (5.15)$$

where

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdot & \cdot & a_{1L} \\ \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot \\ a_{K1} & \cdot & \cdot & a_{KL} \end{bmatrix} \quad (5.16)$$

$$\mathbf{B} = \begin{bmatrix} b_{11} & \cdot & \cdot & b_{1L} \\ \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot \\ b_{J1} & \cdot & \cdot & b_{JL} \end{bmatrix} \quad (5.17)$$

$$\mathbf{g}_i^T = [g_1(\mathbf{X}_i), \dots, g_L(\mathbf{X}_i)] \quad (5.18)$$

with a constraint

$$\sum_{l=1}^L g_l(\mathbf{X}_i) = 1 \quad (\forall i). \quad (5.19)$$

Here,  $L$  is the number of the linear functions,  $F_{al}$  is the  $l$ th linear function,  $a_{kl}$  is the weighting value for the  $k$ th speaker in the  $l$ th function,  $b_{jl}$  is an additive constant of  $j$ th order in the  $l$ th linear function,  $g_l$  is a weighting function in which the input is the multiple speakers' spectral vector and the output is a weighting value for the  $l$ th linear function, and  $a_{lk}$  is a weighting value for the  $k$ th speaker in the  $l$ th linear function. A structure of the multiple linear functional representation with weighting function is shown in Fig. 5.2. This is similar to the modular connectionist architecture [Jacobs 91].

The weighting function fuzzy-subdivides whole spectral space to catch the local characteristics on the spectral transformation. Considering the representation of this locality, Radial Basis Function networks [Broomhead 88] are attractive to use, as RBF networks can produce a localized response and are suitable for localizing each linear function's coverage in spectral space.

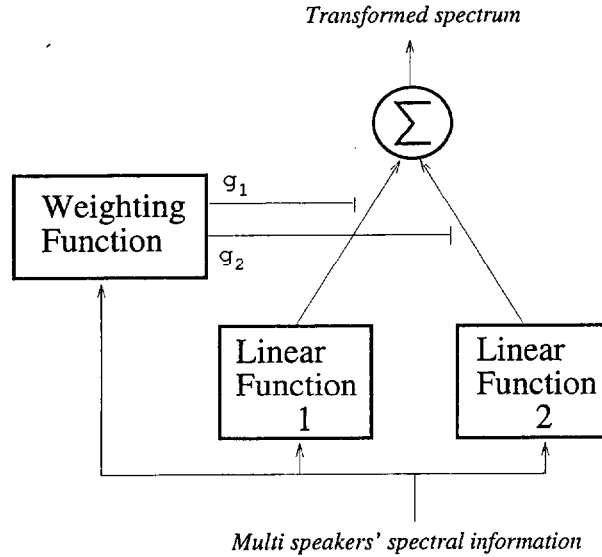


Figure 5.2: Block diagram of multi-functional representation with weighting function

#### 5.4.2 Weighting by Radial Basis Function networks

A structure of the weighting function by using RBF networks is shown in Fig. 5.3. The radial basis functions in the hidden layer are gaussian kernel functions,

$$o_q = \exp \left\{ -\frac{\|z - c_q\|^2}{2\sigma_q^2} \right\} \quad (5.20)$$

$$= \exp \left\{ -\frac{1}{2\sigma_q^2} \sum_{m=1}^M (z_m - c_{qm})^2 \right\}. \quad (5.21)$$

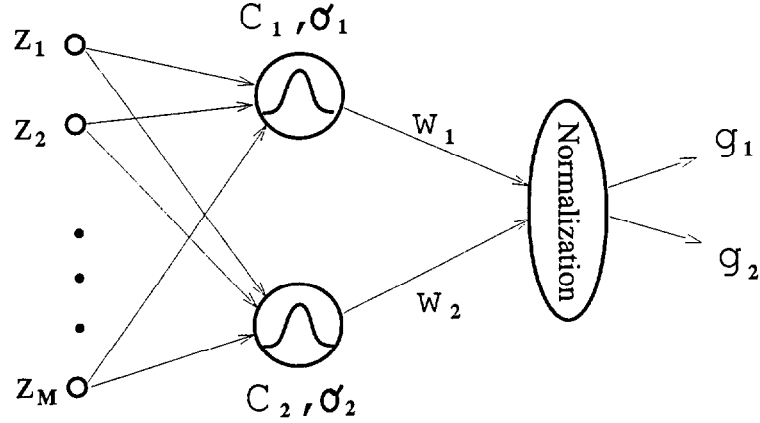


Figure 5.3: Structure of the weighting function by radial basis function networks with output normalization

Here,  $z_m$  is the input parameter to the weighting function where the input dimension is  $M$ .  $c_q$  and  $\sigma_q$  are the center vector and the normalizing factor in the  $q$ th gaussian kernel function.  $o_q$  is the output of the  $q$ th gaussian kernel function. To satisfy the constraint for the output of the weighting function in Eq. (5.19), the output node is modified to produce normalized values for multiple input values. The output node works as the function,

$$g_p = \frac{w_p o_p}{\sum_{l=1}^L w_l o_l} \quad (p = 1, \dots, L) \quad (5.22)$$

where

$$\sum_{l=1}^L w_l = 1. \quad (5.23)$$

Here,  $w_l$  is a link weight.  $g_p$  is the  $p$ th output value of the weighting function. This output node can be regarded as a modification of the multipartitioning unit [Tan 89].

### 5.4.3 Optimization procedure

Parameters of the above-mentioned multi-functional model are optimized to minimize  $Q_m$ , the sum of squared errors between the responses of this model and the values of the target speaker's spectrum parameters in the learning samples, as shown in following:

$$Q_m \equiv \sum_{i=1}^N d_i \quad (5.24)$$

$$= \sum_{i=1}^N \|y_i - Y_i\| \quad (5.25)$$

$$= \sum_{i=1}^N \left\{ \sum_{j=1}^J (y_{ij} - \sum_{l=1}^L g_{il} (\sum_{k=1}^K a_{kl} x_{ijk} + b_{jl}))^2 \right\} \quad (5.26)$$

Here,  $g_{il}$  is the  $l$ th output value of the weighting function for the  $i$ th learning sample.  $N$  is the number of training samples.

The learning of this multi-functional model can be carried out by breaking the problem down into two processes. One is an optimization process of the linear functions, and the other is an update process of the nonlinear weighting function. These two processes are done alternately in an iterative fashion.

Assuming that weighting values  $g_{il}$  ( $i = 1, \dots, N, l = 1, \dots, L$ ) are fixed, the values of the parameters  $a_{kl}, b_{jl}$  in the linear functions are determined uniquely by solving the following linear equations obtained from the partial derivatives of  $Q_m$ , providing that the coefficient vectors in the left hand side of these equations are linearly independent of each other.

$$\sum_{i=1}^N \sum_{j=1}^J \left\{ g_{iq} x_{ijp} \sum_{l=1}^L g_{il} \left( \sum_{k=1}^K x_{ijk} a_{kl} + b_{jl} \right) \right\} = \sum_{i=1}^N \sum_{j=1}^J y_{ij} g_{iq} x_{ijp} \quad (5.27)$$

$$\sum_{i=1}^N \left\{ g_{is} \sum_{l=1}^L g_{il} \left( \sum_{k=1}^K x_{irk} a_{kl} + b_{rl} \right) \right\} = \sum_{i=1}^N y_{ir} g_{is} \quad (5.28)$$

$$(p = 1, \dots, K \quad q = 1, \dots, L \quad r = 1, \dots, J \quad s = 1, \dots, L)$$

Parameters in the weighting function shown in Fig. 5.3 are updated by the gradient descent method. As for the parameter  $c_{rs}$ , the  $s$ th element of the center vector for the  $r$ th gaussian kernel function, it is determined iteratively according to

$$c_{rs}(t+1) = c_{rs}(t) - \mu \left. \frac{\partial Q_m}{\partial c_{rs}} \right|_{\Phi(t)} \quad (5.29)$$

where  $\mu$  is a positive constant, learning rate.  $\Phi(t)$  represents the values of all parameters in the multi-functional model in Fig. 5.2 at the  $t$ th iteration. The partial derivative of  $Q_m$  with respect to  $c_{rs}$  can be written using the Chain Rule

$$\begin{aligned} \frac{\partial Q_m}{\partial c_{rs}} &= \sum_{i=1}^N \frac{\partial d_i}{\partial c_{rs}} \\ &= \sum_{i=1}^N \sum_{p=1}^L \frac{\partial d_i}{\partial g_{ip}} \frac{\partial g_{ip}}{\partial c_{rs}} \\ &= \sum_{i=1}^N \sum_{p=1}^L \frac{\partial d_i}{\partial g_{ip}} \frac{\partial g_{ip}}{\partial o_{ir}} \frac{\partial o_{ir}}{\partial c_{rs}} \end{aligned} \quad (5.30)$$

where

$$\begin{aligned} \frac{\partial d_i}{\partial g_{ip}} = & \sum_{j=1}^J \left\{ 2 \left( \sum_{k=1}^K x_{ijk} a_{kp} \right) \left( \sum_{k=1}^K \sum_{l=1}^L g_{il} x_{ijk} a_{kl} \right) + 2b_{jp} \sum_{l=1}^L (g_{il} b_{jl}) \right. \\ & - 2y_{ij} \sum_{k=1}^K (x_{ijk} a_{kp}) - 2y_{ij} b_{jp} + 2b_{jp} \sum_{k=1}^K \sum_{l=1}^L g_{il} x_{ijk} a_{kl} \\ & \left. + 2 \sum_{k=1}^K (x_{ijk} a_{kp}) \sum_{l=1}^L (g_{il} b_{jl}) \right\} \end{aligned} \quad (5.31)$$

$$\frac{\partial g_{ip}}{\partial o_{ir}} = w_p \frac{1}{\sum_{l=1}^L w_l o_{il}} - \frac{w_p^2 o_{ip}}{\left( \sum_{l=1}^L w_l o_{il} \right)^2} \quad (p = r) \quad (5.32)$$

$$= - \frac{w_p w_r o_{ip}}{\left( \sum_{l=1}^L w_l o_{il} \right)^2} \quad (p \neq r) \quad (5.33)$$

$$\frac{\partial o_{ir}}{\partial c_{rs}} = - \frac{1}{\sigma_r^2} (-z_{is} + c_{rs}) \exp \left( - \frac{1}{2\sigma_r^2} \sum_{m=1}^M (z_{im} - c_{rm})^2 \right). \quad (5.34)$$

Here,  $z_{im}$  is the  $m$ th input value for the  $i$ th training sample.  $o_{ir}$  is the output value of the  $r$ th gaussian kernel function of the  $i$ th training sample.

The other parameters, such as  $\sigma_l$  and  $w_l$ , are similary updated.

## 5.5 Evaluation and Discussion

### 5.5.1 Characteristics of speaker interpolation

So far, the characteristics of cepstrum and log area ratio on interpolation have been known mainly for spectrum sequences in a single speaker. We checked how the interpolation works between speakers. The analysis conditions are shown in Table 5.1. Output speech waveforms were synthesized by cepstrum parameters [Imai 80].

The results are almost the same for both the cepstrum and log area ratio cases. By listening to speech generated by changing the interpolation coefficients step by step between two speakers, it was found that speaker individuality changed smoothly from one speaker to the other. For example, Fig. 5.4 shows spectrograms generated by speaker interpolation between two speakers. We can see that while the band width of formants in the transformed spectrum becomes a little larger than in the original, the spectrum pattern maintains smooth spectral dynamics when interpolated between two speakers. We can select the preferred speech spectrum characteristics by changing the interpolation coefficients.

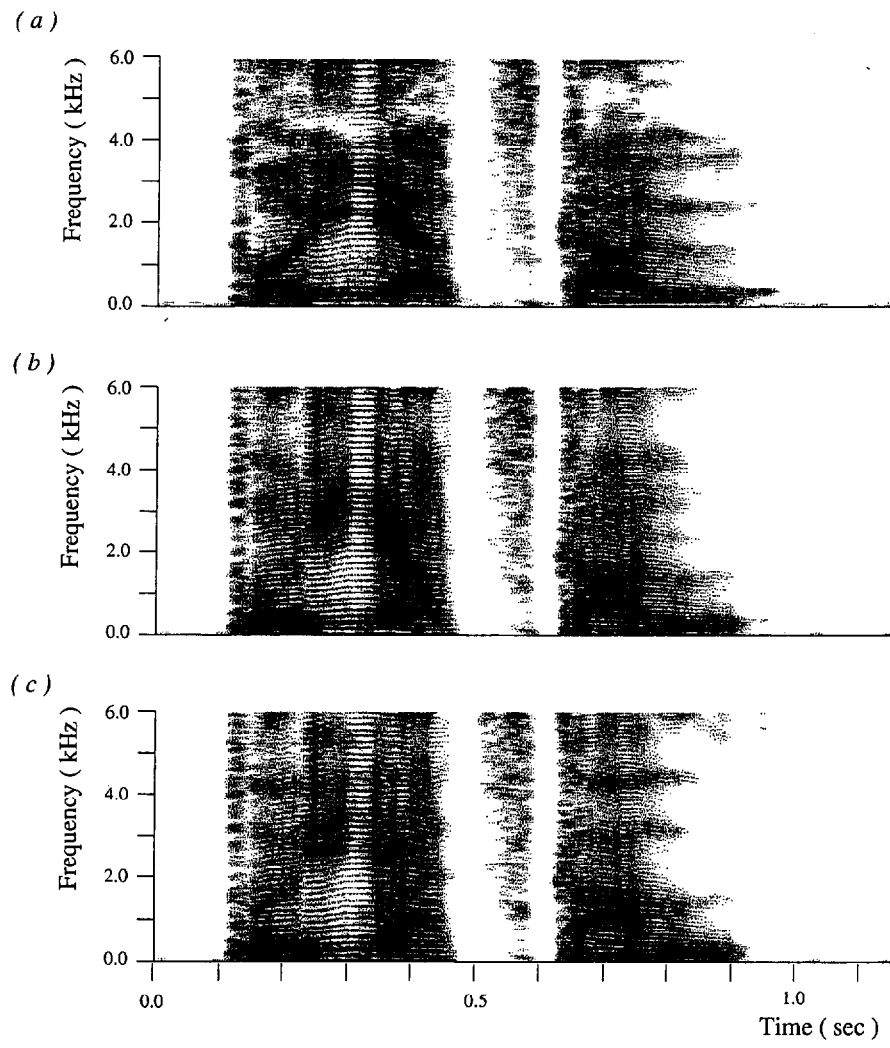


Figure 5.4: Spectrum generated by speaker interpolation (a) male speaker, (b) interpolated spectrum between (a) and (c) with ratio 0.5 : 0.5 , (c) female speaker (utterance : 'toriatsukau')

Table 5.1: Analysis conditions

|                    |                                           |
|--------------------|-------------------------------------------|
| Sampling frequency | 12 kHz                                    |
| Frame period       | 5 ms                                      |
| Window length      | 21.3 ms ( Hamming window )                |
| Log area ratio     | 12 order                                  |
| Cepstrum           | 30 order LPC-cepstrum<br>( 12 order LPC ) |

### 5.5.2 Experiments of adaptation by a single interpolating function

First, the adaptation procedure in Section 5.2 with the interpolating function in Eq. (5.2) was tested. The number of pre-stored speakers was four - two males and two females. The interpolation coefficients were determined by one utterance (training data) each of the Japanese words “urayamashii” and “toriatsukau”, spoken by the target speakers. In the adaptation procedure, DTW was carried out by considering manually labelled phonemic boundaries, with allowance for some range of time-divergence of corresponding phonemic boundaries in utterances from two different speakers. Cepstrum parameter and log area ratio parameter are used as interpolated spectral parameters. When log area ratio is used as a spectral parameter, at Step 4 in the adaptation procedure, a non-linear optimization process is subsequently carried out to minimize log-spectral distance between the target and the generated spectra, as interpolation coefficients which minimize the difference in log area ratio do not always minimize the corresponding log-spectral difference. The distance between the spectrum of the target speaker and that generated by the interpolation was calculated for sixty-four words (test data). Eight males and eight females were used as target speakers. Figure 5.5 shows the mean cepstrum distance values per frame for all target speakers using log area ratio and cepstrum (LAR, CEP). For comparison, the mean cepstrum distance values between target speakers and each pre-stored speaker are shown in Fig. 5.5, where the mean values (A, B, C, D) were calculated after the pre-stored speakers were sorted according to the distance to the target speaker in each target speaker’s case. It is shown that the distance between the spectrum of the target speaker and the spectrum generated by the interpolation is smaller by about 25% than the distance between the spectrum of the target speaker

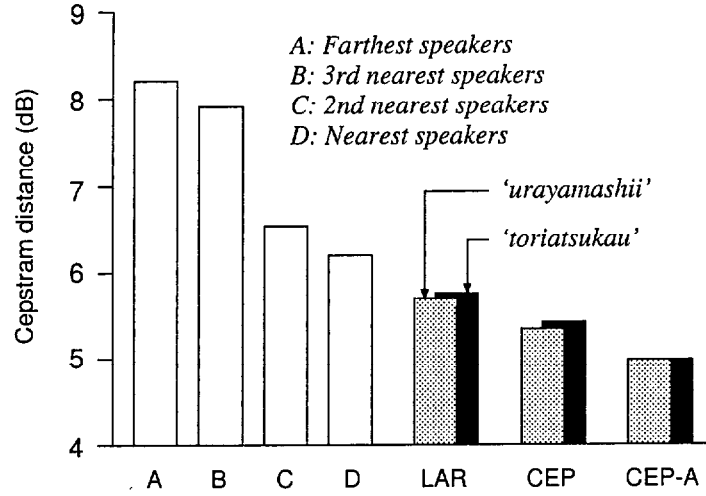


Figure 5.5: Reduction of the cepstrum distance to target spectrum by the interpolation between pre-stored speakers

and the spectrum of the pre-stored speaker closest to the target speaker. This distance reduction rate is larger when interpolating cepstrum rather than log area ratio. Table 5.2 shows the determined interpolation coefficients and mean values of cepstrum distance between each target speaker's spectrum and the generated spectrum in the case of using cepstra as the interpolated parameters. We can see that gender dependency of coefficients is quite consistent (*i.e.* Target speaker's gender decides high ratios of speech data of the same gender). The tendency of obtained interpolation coefficients was almost the same for two different training utterances in all tested target speakers. This shows that interpolation coefficients can stably characterize the target speaker's spectrum by using one utterance as training data. Also, we can see a large variance in the log-spectral distance for the different target speakers. This seems to be dependent upon the relationship between the characteristics of pre-stored speakers and that of the target speaker. A suitable selection of pre-stored speakers should be considered.

Next, the adaptation by the function with the cepstral constant terms in Eq. (5.12) was tested. By using this adaptation method (CEP-A), the distance reduction rate increased to 36% (Fig. 5.5). Table 5.3 shows a comparison of distance reduction rates at training and testing. Differences between rates at training and testing are larger in the CEP-A method than in the method of the adapting interpolation ratio only (LAR, CEP). This shows that over-adaptation occurred in the CEP-A method. Use of a larger amount of training data might be able to improve the performance for test data in the CEP-A method.

Table 5.2: Interpolation coefficients determined by the adaptation and mean log-spectral distance between target speaker's spectrum and generated spectrum ( Upper row for each target speaker shows result by using 'toriatsukau' as training utterance , and lower row 'urayamashii'. )

$PM_i$  : Pre-stored  $i$ -th Male Speaker

$PF_i$  : Pre-stored  $i$ -th Female Speaker

$TM_i$  : Target  $i$ -th Male Speaker

$TF_i$  : Target  $i$ -th Female Speaker

(a) Male target speakers

| Target speaker | Interpolation coef.<br>( $PM_1 : PM_2 : PF_1 : PF_2$ ) | Cepstrum distance (dB) |
|----------------|--------------------------------------------------------|------------------------|
| $TM_1$         | 0.42 : 0.18 : 0.26 : 0.15                              | 5.5                    |
|                | 0.39 : 0.27 : 0.31 : 0.02                              | 5.5                    |
| $TM_2$         | 0.17 : 0.52 : 0.20 : 0.11                              | 5.1                    |
|                | 0.37 : 0.46 : 0.08 : 0.09                              | 5.4                    |
| $TM_3$         | 0.23 : 0.52 : -0.01 : 0.27                             | 5.6                    |
|                | 0.27 : 0.36 : 0.13 : 0.24                              | 4.8                    |
| $TM_4$         | 0.46 : 0.33 : 0.04 : 0.17                              | 4.9                    |
|                | 0.44 : 0.43 : -0.01 : 0.15                             | 5.1                    |
| $TM_5$         | 0.22 : 0.71 : -0.10 : 0.17                             | 4.9                    |
|                | 0.37 : 0.56 : -0.21 : 0.28                             | 6.3                    |
| $TM_6$         | 0.15 : 0.59 : -0.13 : 0.39                             | 4.9                    |
|                | 0.16 : 0.51 : -0.09 : 0.41                             | 6.3                    |
| $TM_7$         | 0.41 : 0.20 : 0.02 : 0.37                              | 4.9                    |
|                | 0.35 : 0.40 : 0.14 : 0.11                              | 5.2                    |
| $TM_8$         | 0.34 : 0.41 : 0.20 : 0.05                              | 4.9                    |
|                | 0.43 : 0.36 : 0.17 : 0.04                              | 6.1                    |

(b) Female target speakers

| Target speaker | Interpolation coef.<br>( $PM_1 : PM_2 : PF_1 : PF_2$ ) | Cepstrum distance (dB) |
|----------------|--------------------------------------------------------|------------------------|
| $TF_1$         | 0.10 : 0.04 : 0.59 : 0.27                              | 5.5                    |
|                | 0.13 : 0.01 : 0.62 : 0.24                              | 5.5                    |
| $TF_2$         | 0.09 : 0.18 : 0.37 : 0.36                              | 5.5                    |
|                | 0.08 : 0.10 : 0.52 : 0.31                              | 5.1                    |
| $TF_3$         | -0.07 : 0.18 : 0.42 : 0.47                             | 4.9                    |
|                | 0.17 : 0.00 : 0.47 : 0.36                              | 4.9                    |
| $TF_4$         | -0.02 : 0.05 : 0.59 : 0.39                             | 5.0                    |
|                | 0.16 : -0.07 : 0.57 : 0.34                             | 5.1                    |
| $TF_5$         | 0.10 : 0.14 : 0.36 : 0.41                              | 6.7                    |
|                | 0.12 : 0.10 : 0.30 : 0.49                              | 5.5                    |
| $TF_6$         | 0.11 : 0.01 : 0.26 : 0.62                              | 6.5                    |
|                | 0.10 : 0.08 : 0.30 : 0.52                              | 4.9                    |
| $TF_7$         | 0.11 : 0.10 : 0.55 : 0.24                              | 5.3                    |
|                | 0.16 : 0.02 : 0.57 : 0.25                              | 5.0                    |
| $TF_8$         | 0.03 : 0.13 : 0.37 : 0.47                              | 6.1                    |
|                | 0.06 : 0.19 : 0.36 : 0.39                              | 4.9                    |

Table 5.3: Comparison of distance reduction rates (%) at training and test in each case of using ‘urayamashii’ and ‘toriatsukau’ as training utterance

| Method | ‘urayamashii’ |         | ‘toriatsukau’ |         |
|--------|---------------|---------|---------------|---------|
|        | Training      | Testing | Training      | Testing |
| LAR    | 22            | 16      | 11            | 15      |
| CEP    | 25            | 26      | 27            | 24      |
| CEP-A  | 47            | 35      | 45            | 36      |

### 5.5.3 Experiments of adaptation by multiple linear functions

The adaptation carried out by using the multiple linear functions weighted by RBF networks was tested. In order to check the basic ability of the multi-functional model, two linear functions were used, and only the center vectors of the gaussian kernel functions in the weighting function were adapted. The initial center vectors were given as vectors which were generated by adding small perturbations to the centroid vector in the training data.  $\sigma_l$  and  $w_l$  for  $l = 1, 2$  in the weighting function were fixed to 0.5 and 0.5 respectively. The learning rate  $\mu$  in Eq. (5.29) was set to 0.001. No attempt was made to optimize the choice of this value. One male speaker was used as the target speaker in the experiments. Because the number of adapted parameters in this scheme was larger than in the case of using the single function, ten words were used as training data. The cepstrum parameters of one pre-stored speaker which obtained the largest weight, by using the single linear function, were used as input to the weighting function. Other conditions for the learning were the same as in the previous subsection. For comparison, the single linear function represented in Eq. (5.12) was also adapted using the same training data.

The results are shown in Table 5.4. We can see that whilst the distance reduction rate from using the single linear function ( CEP-A ) was 42 %, the rate from the multiple linear functions ( MLF ) increased to 48 %. This also shows that over-adaptation occurred to some extent when using multi-functions, because there were larger differences between the reduction rates at training and testing in the MLF case in comparison with the CEP-A case. More training data might improve the performance. We should select the number of the linear functions according to the amount of training data by an open data generality check method, such as cross validation. Table 5.5 shows obtained interpolation coefficients in MLF and CEP-A. In the two linear functions of MLF, in the interpolation coefficients for female pre-stored speakers, particular differences can be founded. Figure 5.6 shows the spectral characteristics of cepstral constant terms in each of the multiple linear functions. Notice that addition in the cepstral domain corresponds to multiplication in spectral domain. It can be found clearly that characteristics in the single linear functional case ( CEP-A ) split into two different characteristics in the multiple linear functional case ( MLF ). Figure 5.7 shows surfaces of the obtained weighting function for the first one of two linear functions. Cross sections at a point where output is 0.5 from the weighting function are drawn. This shows that different states of surface of the weighting function were obtained for different orders of input cepstrum parameter. In addition, Fig. 5.8 shows the decreasing of the distance by the iterative adaptation

Table 5.4: Comparison of distance reduction rates (%) between the single linear function case ( CEP-A) and the multiple linear functions case ( MLF) with using ten training words

| Method | Training | Testing |
|--------|----------|---------|
| CEP-A  | 44       | 42      |
| MLF    | 52       | 48      |

Table 5.5: Interpolation coefficients in the two linear functional method ( MLF ) and the single linear functional method ( CEP-A )

| Linear func.        | Interpolation coef.       |
|---------------------|---------------------------|
| Fisrt func. in MLF  | 0.42 : 0.28 : 0.09 : 0.04 |
| Second func. in MLF | 0.32 : 0.24 : 0.09 : 0.19 |
| Func. in CEP-A      | 0.33 : 0.35 : 0.10 : 0.07 |

procedure for the multiple functional model described in Section 5.4. These results confirmed that the proposed non-linear optimization method worked well.

## 5.6 Conclusion

A new speech spectrum transformation method for controlling individuality has been described in this paper, based on interpolating multiple speakers' spectral patterns and a multi-functional representation with weighting by Radial Basis Function networks. The interpolation is carried out in the cepstrum or log area ratio domains so that the resultant speech preserves natural spectral dynamics. We confirmed that the individuality of the generated speech is changed smoothly as the interpolation coefficients are changed.

A speaker adaptation method based on this interpolation has been proposed for speech synthesis. This adaptation method does not need a large amount of data from the target speaker. Adaptation experiments were carried out using four pre-stored speakers' spectral data. Using only one word spoken by the target speaker for training, it was shown that the distance between the target speaker's spectrum and the spectrum generated by the proposed method was reduced by about 35 % compared with the distance between the target speaker's spectrum and the spectrum

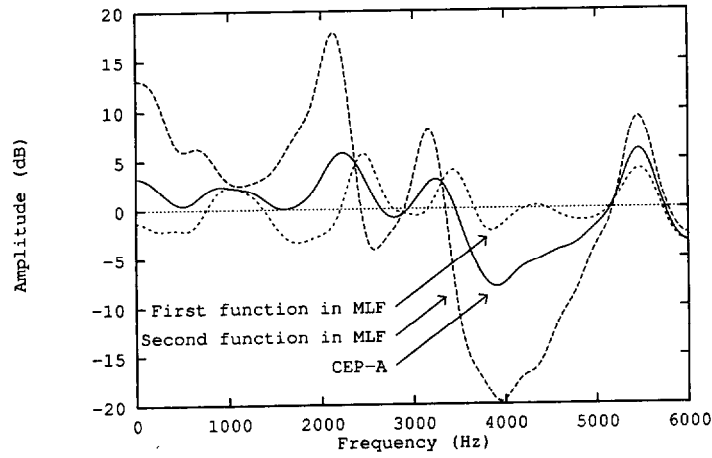
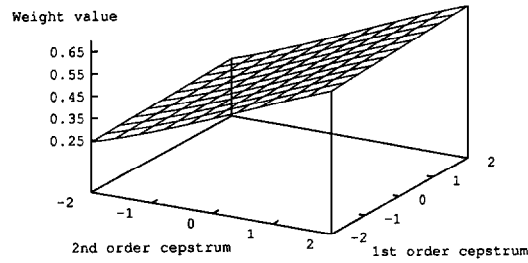


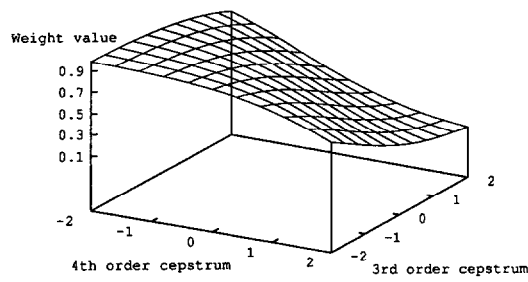
Figure 5.6: Spectral characteristics of cepstral constant terms in multi-functional transformation

of the speaker closest to the target among pre-stored ones. Moreover, a multiple linear functional transformation with RBF networks has been proposed to get a more precise mapping to a target speaker when comparatively large amounts of training data could be used. By using ten words as training data, whilst the distance reduction rate in the case of using the single linear function was 42 %, the rate by the two linear functions increased to 48 %. We can select the number of linear functions for adaptation according to the amount of training data.

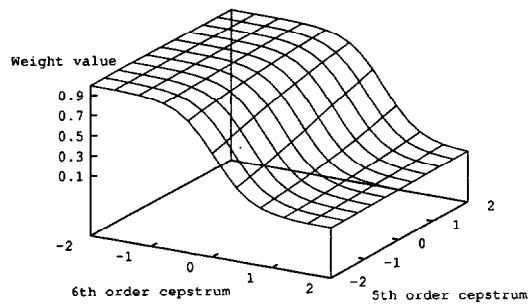
These proposed methods can be used in a rule-based concatenative synthesis system by preparing segmental units for a moderate number of stored speakers. The synthetic speech spectrum can be transformed and adapted to the target speaker's spectrum by the proposed methods. These methods are available in the case of other control parameters such as formant frequency, bandwidth and spectral tilt. In future work, studies on control parameters and how to select pre-stored speakers would be needed.



(a) Plane of 1st and 2nd order cepstrums



(b) Plane of 3th and 4th order cepstrums



(c) Plane of 5th and 6th order cepstrums

Figure 5.7: Surface components of the weighting function for one of two linear transformations

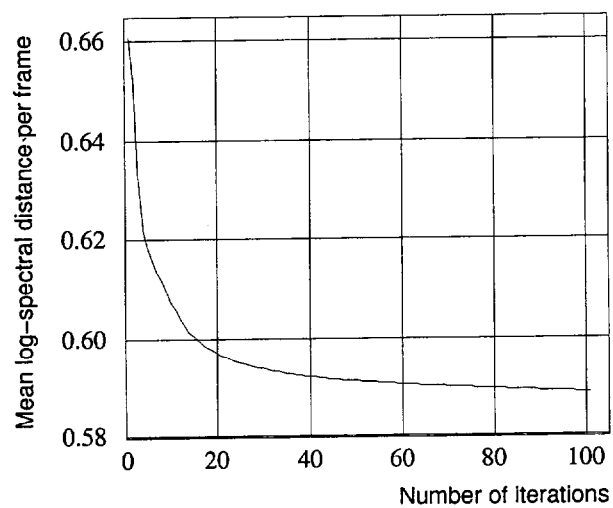


Figure 5.8: Decrease of log-spectral distance in the non-linear optimization process for the multi-functional model

# Chapter 6

## Classifier Training

### 6.1 Introduction

It is desirable that pattern classifiers should maintain suitable recognition performance in spite of changes in input conditions which affect patterns to be classified. Such robustness is often a very important factor in practical recognition systems. In speech recognition, changes in input conditions are caused by different speakers, room acoustics, background noise, channel noise and so on. To achieve robustness against these changes, several approaches to improving classifiers have been studied [Junqua and Haton 96]. A conventional way to obtain robust classifiers is to use a large amount of training data under as many input conditions as possible [Lee 90, Ohkura et al. 93]. In this approach, although the average performance in changes of input conditions can be improved, the performance in terms of individual input conditions is not satisfactory: the performance can be very low under certain input conditions. To find a way of achieving higher recognition performance the adaptation of the pattern classifier to changes in input conditions has been intensively studied [Junqua and Haton 96]. Several methods have been proposed in terms of parameters to be adapted, the adaptation methods of the parameters, and optimization criteria for the adaptation. Of these methods, adaptation methods based on the transformation of classifier parameters have been investigated to pursue efficient adaptation with a small amount of adaptation data [Leggetter and Woodland 95, Minami and Furui 95, Matsui and Furui 95]. Previously, however, classifiers on which this type of adaptation is carried out have been trained independently of the adaptation procedures used. Such classifiers are not always optimal in the overall recognition system including the adaptation process. A framework for training classifiers suited to the adaptation procedures used on them is therefore needed.

In this paper, we show a framework for optimizing classifiers so as to achieve high performance after adaptation has been carried out, and describe a training method based on this framework [Iwahashi 95, Iwahashi 96]. In the proposed training method, the classifier is trained while adaptation is being carried out. The objective function for the training is given based on the recognition performance obtained by the adapted classifier. The proposed method is applicable to any type of adaptation methods, and makes any optimization criteria, such as minimum squared error (MSE), maximum likelihood (ML), minimum classification errors (MCE), or maximum mutual information (MMI), available for classifier training and the adaptation process.

The remainder of this chapter is organized as follows. In Sect. 6.2, we discuss the formulation for classifier optimization. In Sect. 6.3, we present the objective functions for training classifier using sample data. In Sect. 6.4, we discuss the algorithm for optimizing the objective functions. Then we apply the objective function to the training of classifiers for a Japanese five vowel recognition task. The experimental arrangements and the recognition results are given in Sect. 6.5. The generality of the proposed method is discussed in Sect. 6.6. Finally, we summarize our findings in Sect. 6.7.

## 6.2 Formulation for Minimum Error Classification

We consider the classification of a  $d$ -dimensional vector  $\mathbf{O}$  in observation space into one of the  $K$  classes  $\{C_k\}_{k=1}^K$ , with adaptation of the classifier to changes in the input condition represented by the set of parameters  $V$ . This classification uses the decision rule:

$$\begin{aligned} C(\mathbf{O}) = C_i \quad & \text{if} \quad g_i(\mathbf{O}; F(\Lambda, \Gamma)) \\ & = \max_j g_j(\mathbf{O}; F(\Lambda, \Gamma)), \end{aligned} \quad (6.1)$$

where  $C(\cdot)$  denotes a classification operation,  $g_i$  is the discriminant function for class  $C_i$ ,  $\Lambda$  is the set of classifier parameters,  $\Gamma$  is the set of parameters of adaptation transformation, and  $F(\Lambda, \Gamma)$  represents the parameter set of the classifier obtained through the transformation of  $\Lambda$  by  $\Gamma$ . The block diagram of this pattern recognizer is shown in Fig. 1. The process of the recognizer is composed of two stages: adaptation and classification. When the input condition is changed, the adaptation is performed preceding the classification. At the adaptation stage, the classifier is

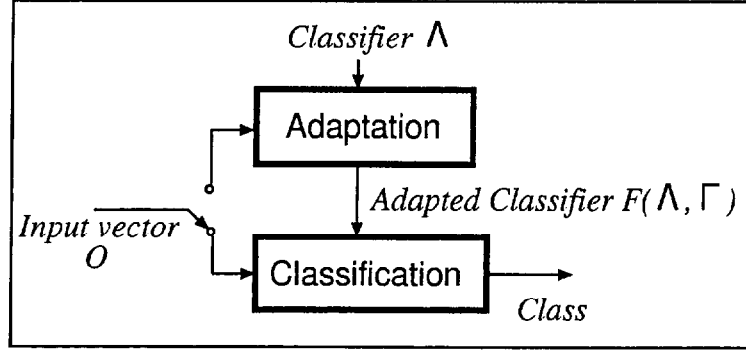


Figure 6.1: Pattern recognizer with adaptation of classifier

adapted to the input condition, the values of adaptation-transformation parameter set  $\Gamma$  are determined using the samples extracted under the input condition, and the adapted classifier  $F(\Lambda, \Gamma)$  is obtained. Then at the classification stage, the following each input observation sample is classified into one of  $K$  classes using the adapted classifier  $F(\Lambda, \Gamma)$ .

The ultimate goal here is to achieve classifiers that minimize the probability of classification error, given that an adaptation procedure is performed on them. The values of classifier parameter set  $\Lambda$  are determined in the training process using a training sample set.

The problem of optimizing a conventional pattern classifier, which is designed independently of the adaptation procedure, can be formulated as follows:

$$\sum_{k=1}^K P(C_k) \int 1(g_k(\mathbf{O}; \Lambda) \neq \max_i g_i(\mathbf{O}; \Lambda)) p(\mathbf{O}|C_k) d\mathbf{O} \rightarrow \min, \quad (6.2)$$

where  $P(C_k)$  is the class probability,  $p(\mathbf{O}|C_k)$  is the conditional probability density function, and  $1()$  is an indicator function:

$$1(\alpha) = \begin{cases} 1, & \text{if } \alpha \text{ is true} \\ 0, & \text{otherwise.} \end{cases} \quad (6.3)$$

The integration is over the entire observation space. Note that this formulation is for minimizing the expected error rate when the classifier has not been adapted to the input conditions.

Now, we consider the problem of optimizing the classifier used in rule 6.1. The classifier should be optimized so as to achieve high performance after adaptation has been carried out. This problem is formulated as follows:

$$\int p(V) \int \sum_{k=1}^K P(C_k|V)$$

$$\int 1(g_k(\mathbf{O}; F(\Lambda, \Gamma)) \neq \max_i g_i(\mathbf{O}; F(\Lambda, \Gamma))) p(\mathbf{O}|C_k, V) p(\Gamma|\Lambda, V) d\mathbf{O} d\Gamma dV \rightarrow \min. \quad (6.4)$$

Note that because the value of  $\Gamma$  is determined in the adaptation process using adaptation samples extracted,  $\Gamma$  can be taken as random variables. The conditional probability density  $p(\Gamma|\Lambda, V)$  depends on the adaptation procedure used. Each integration is over the entire space of the corresponding variable.

## 6.3 Training Objective

We will now consider the training objective of classifiers based on Eq. (6.4). We assume that the value of the input condition parameter  $V$  takes discrete values ( $V_b, b = 1, 2, \dots$ ). The general scheme for the adaptation method based on the transformation of classifier parameters is described first, and then the objective function for the training of classifiers is shown.

### 6.3.1 Adaptation Criterion

In the process of adaptation to input condition  $V_b$ , the values of parameters  $\Gamma_b$  of adaptation transformation are determined using the adaptation sample set (denoted by  $\mathcal{A}_b$ ) extracted under input condition  $V_b$ . The optimum value  $\hat{\Gamma}_b$  of  $\Gamma_b$  is obtained as follows:

$$\hat{\Gamma}_b = \underset{\Gamma}{\operatorname{argmin}} \sum_{k=1}^K \sum_{i=1}^{m_{b,k}} \ell_a(\mathbf{O}_{b,k,i}^{(\mathcal{A}_b)}, C_k, F(\Lambda, \Gamma)), \quad (6.5)$$

where  $m_{b,k}$  is the number of samples in class  $C_k$  in  $\mathcal{A}_b$ ,  $\mathbf{O}_{b,k,i}^{(\mathcal{A}_b)}$  denotes the  $i$ th feature vector in them, and  $\ell_a$  is the loss function for individual samples, which is used to determine the value of transformation parameters for the adaptation. By changing the loss function, (6.5) can represent any optimization criteria, such as MSE, ML, MCE, MMI and so on. Note that minimization in (6.5) is often carried out with some constraints among the parameters of the transformation [Hattori and Sagayama 92, Gales 98], such as smoothing, to reduce small-sample-size effects [Raudy and Jain 91].

### 6.3.2 Empirical Average Loss for Classifier Training

We consider the objective function to be minimized for training a classifier. As with the conventional training method, in order to emulate the classification operation

and to approximate the performance evaluation in a smooth function form, the loss function for individual samples is introduced as

$$\begin{aligned} \ell(\mathbf{O}, C, F(\Lambda, \Gamma)) \\ \approx 1(g_k(\mathbf{O}; F(\Lambda, \Gamma)) \neq \max_i g_i(\mathbf{O}; F(\Lambda, \Gamma))). \end{aligned} \quad (6.6)$$

Given a training sample set which is composed of samples extracted under  $B$  input conditions, we consider the empirical average loss in the training samples as the objective function. Let each sample have labels for the class which it belongs to and the input condition which it is extracted under. Using the loss function, the objective function is derived from (6.4) by substituting the summations for the integrations with regard to the sample-observation parameter space and input-condition parameter space as

$$L_1(\Lambda) = \frac{1}{N} \sum_{b=1}^B \int \sum_{k=1}^K \sum_{i=1}^{n_{b,k}} \ell(\mathbf{O}_{b,k,i}, C_k, F(\Lambda, \Gamma)) p(\Gamma|\Lambda, V_b) d\Gamma, \quad (6.7)$$

where  $n_{b,k}$  is the number of samples in class  $C_k$ , which is extracted under input condition  $V_b$ ,  $\mathbf{O}_{b,k,i}$  denotes the  $i$ th feature vector in them, and  $N$  is the number of the training samples. The distribution  $p(\Gamma|\Lambda, V_b)$  here is unknown generally. Taking into account the simultaneous training of distribution  $p(\Gamma|\Lambda, V_b)$ , sample sets  $\mathcal{A}_{b,h}$ ,  $h = 1, \dots, H_b$  are prepared as the adaptation sample sets extracted for each input condition  $V_b$ ,  $b = 1, \dots, B$ . The set of the values of adaptation parameters  $\hat{\Gamma}_{b,h}$  decided using  $\mathcal{A}_{b,h}$ ,  $h = 1, \dots, H_b$ , are substituted for the distribution  $p(\Gamma|\Lambda, V_b)$ . Then the values of loss function for individual samples extracted under input condition  $V_b$  are calculated using each of  $\hat{\Gamma}_{b,h}$ ,  $h = 1, \dots, H_b$ , and are averaged. The objective function is given by substituting summation for the integration with regard to the adaptation parameter space as

$$L_2(\Lambda) = \frac{1}{MN} \sum_{b=1}^B \sum_{h=1}^{H_b} \sum_{k=1}^K \sum_{i=1}^{n_{b,k}} \ell(\mathbf{O}_{b,k,i}, C_k, F(\Lambda, \hat{\Gamma}_{b,h})), \quad (6.8)$$

where

$$\hat{\Gamma}_{b,h} = \underset{\Gamma}{\operatorname{argmin}} \sum_{k=1}^K \sum_{i=1}^{n_{b,k}^{(h)}} \ell_a(\mathbf{O}_{b,k,i}^{(\mathcal{A}_{b,h})}, C_k, F(\Lambda, \Gamma)). \quad (6.9)$$

$\mathbf{O}_{b,k,i}^{(\mathcal{A}_{b,h})}$  is the  $i$ th feature vector in class  $C_k$  in the adaptation sample set  $\mathcal{A}_{b,h}$ , and  $M$  is the number of the adaptation sample sets.

In addition, when the size of the given training sample set for the adaptation is large enough with regard to the degree of freedom in  $\Gamma$ , distribution  $p(\Gamma|\Lambda, V_b)$  becomes very sharp. In this case, we assume that distribution  $p(\Gamma|\Lambda, V_b)$  is approximately a delta function, where  $p(\Gamma|\Lambda, V_b) \neq 0$  at  $\Gamma = \hat{\Gamma}_b$ . Based on this assumption, the value of  $\hat{\Gamma}_b$  is determined by using all samples extracted under the input condition  $V_b$  in the training sample set as an adaptation sample set. Then the objective function is given as

$$L_3(\Lambda) = \frac{1}{N} \sum_{b=1}^B \sum_{k=1}^K \sum_{i=1}^{n_{b,k}} \ell(\mathbf{O}_{b,k,i}, C_k, F(\Lambda, \hat{\Gamma}_b)), \quad (6.10)$$

where

$$\hat{\Gamma}_b = \underset{\Gamma_b}{\operatorname{argmin}} \sum_{k=1}^K \sum_{i=1}^{n_{b,k}} \ell_a(\mathbf{O}_{b,k,i}, C_k, F(\Lambda, \Gamma_b)). \quad (6.11)$$

Furthermore, when  $\ell_a(\mathbf{O}, C_k, F(\Lambda, \Gamma_b)) = \ell(\mathbf{O}, C_k, F(\Lambda, \Gamma_b))$ , by combining (6.10) and (6.11) we can obtain the objective function as

$$L'_3(\Lambda, \Gamma) = \frac{1}{N} \sum_{b=1}^B \sum_{k=1}^K \sum_{i=1}^{n_{b,k}} \ell(\mathbf{O}_{b,k,i}, C_k, F(\Lambda, \Gamma_b)), \quad (6.12)$$

where  $\Gamma = \{\Gamma_1, \dots, \Gamma_B\}$ .

We note that all objective functions defined here represent empirical average loss and that the minimization is only for the loss incurred in classifying the training samples. Although generalization for the expected loss, *i.e.*, the average loss in a population, should also be considered, we have used these objective functions with large amount of training data as the first step in confirming the validity of the proposed approach. By changing the loss function, these objective functions can represent any optimization criteria, such as MSE, ML, MCE, MMI and so on. In addition, we should note that the values of adaptation parameters  $\Gamma$  calculated in the process of classifier training are not used in the process of recognizer; instead, the values of  $\Gamma$  decided at the adaptation stage are used at the classification stage.

## 6.4 Optimization Method

For the objective functions given in Section 6.3, a local or global optimum solution can be obtained by iterative neighborhood search strategies, such as the gradient

descent method. In addition, for the objective function  $L'_3$ , if optimization processes with respect to  $\Lambda$  and  $\Gamma$  both exhibit global convergence, carrying out both optimization processes alternately in an iterative fashion would also exhibit global convergence. Particularly, whenever

$$L'_3(\Lambda, \Gamma) \geq L'_3(\Lambda, \hat{\Gamma}) \geq L'_3(\hat{\Lambda}, \hat{\Gamma}), \quad (6.13)$$

where  $\hat{\Lambda}$  and  $\hat{\Gamma}$  are renewed values in each process, then the global convergence holds. This property becomes effective particularly in cases where a global solution can easily be found at each alternately carried out process. In the experiments described in the following section, we utilized this property. Note that this convergence property does not necessarily hold for the objective functions  $L_2$  and  $L_3$ , because the sample set used in Eq. (6.8) can be different from the sample set used in Eq. (6.9), and because loss function  $\ell_a$  can be different from  $\ell$ .

## 6.5 Experiments

### 6.5.1 Arrangements

To evaluate the proposed training method, we conducted simple experiments in five-class ( $K = 5$ ), fixed-dimensional Japanese vowel pattern recognition with speaker adaptation. The objective function  $L'_3$  was used in the training of classifiers, where  $\ell = \ell_a$ . Differences between speakers were taken into account here as changes in input conditions. Vowel tokens were extracted from 216 isolated words spoken by 40 speakers (20 males, 20 females) and were digitized (fs 12 KHz). The center fragment of each vowel was selected using a 32 msec Hamming window and was converted into a feature vector consisting of sixteen-order LPC cepstral coefficients. The number of tokens in the whole token set was 26,667. The classifier consisted of normal distribution functions with diagonal covariances as discriminant functions. The classifier parameter set  $\Lambda$  is represented as

$$\Lambda = \{\mu_1, \Sigma_1, \dots, \mu_K, \Sigma_K\}, \quad (6.14)$$

where  $\mu_k$  and  $\Sigma_k$  respectively denote the mean vector and the diagonal covariance matrix in the discriminant function for class  $C_k$ .

$$\mu_k = [\mu_{k,1}, \dots, \mu_{k,J}]^T, \quad (6.15)$$

$$\Sigma_k = \text{diag}(\sigma_{k,1}^2, \dots, \sigma_{k,J}^2), \quad (6.16)$$

where  $J(= 16)$  is the number of components in the feature vector. The superscript  $T$  denotes the matrix transposition. The classifier adaptation to the speaker represented by  $V_b$  is carried out by moving all mean vectors through the addition of the

single vector  $\Gamma_b = [\gamma_{b,1}, \dots, \gamma_{b,J}]^T$  : parameters of the adaptation transformation. In this case,

$$F(\Lambda, \Gamma_b) = \{\mu_1 + \Gamma_b, \Sigma_1, \dots, \mu_K + \Gamma_b, \Sigma_K\}. \quad (6.17)$$

As training criteria, ML and MCE were used in the experiments. In the case of ML, Eq. 6.12 is written as

$$\begin{aligned} L'_3(\Lambda, \Gamma) = & \frac{1}{N} \sum_{b=1}^B \sum_{k=1}^K \sum_{i=1}^{n_{b,k}} \left[ \frac{J}{2} \log(2\pi) + \frac{1}{2} \log(|\Sigma_k|) \right. \\ & \left. + \frac{1}{2} (\mathbf{O}_{b,k,i} - \mu_k - \Gamma_b)^T \Sigma_k^{-1} (\mathbf{O}_{b,k,i} - \mu_k - \Gamma_b) \right]. \end{aligned} \quad (6.18)$$

In the case of MCE, the loss function was defined to approximate the classification error more directly [Juang and Katagiri 92, Rainton 91] as follows:

$$\ell(\mathbf{O}, C_k, F(\Lambda, \Gamma)) = \frac{1}{1 + e^{-\xi d}}, \quad \xi > 0, \quad (6.19)$$

where

$$d = -g_k(\mathbf{O}, F(\Lambda, \Gamma)) + \max_j g_j(\mathbf{O}, F(\Lambda, \Gamma)). \quad (6.20)$$

The value  $\xi = 0.5$  was used in the experiments.

The use of objective function  $L'_3(\Lambda, \Gamma)$  allows efficient procedure for the optimization with respect to both  $\Lambda$  and  $\Gamma$  by alternate optimization and gradient methods as mentioned in Section 6.4. In optimizing the ML criterion, alternate optimization procedure with respect to both  $\Lambda$  and  $\Gamma (= \{\Gamma_1, \dots, \Gamma_B\})$  was carried out. In each process, the solutions can be obtained using simple equations:  $\Gamma$  in particular was calculated using the following equation, which can be obtained by setting  $\frac{\partial L'_3}{\partial \gamma_{b,j}} = 0$ :

$$\begin{aligned} \gamma_{b,j} = & \frac{\sum_{k=1}^K \frac{1}{\sigma_{k,j}^2} \sum_{i=1}^{n_{b,k}} (\mathbf{o}_{b,k,i,j} - \mu_{k,j})}{\sum_{k=1}^K \frac{n_{b,k}}{\sigma_{k,j}^2}}, \\ & b = 1, \dots, B, \quad j = 1, \dots, J \end{aligned} \quad (6.21)$$

where  $\mathbf{o}_{b,k,i,j}$  represents the  $j$ th component in the feature vector  $\mathbf{O}_{b,k,i}$ .

In optimizing the MCE criterion, the objective function was minimized by the gradient method with respect to all parameters using the following adjustment rule:

$$\{\Lambda, \Gamma\}_{t+1} = \{\Lambda, \Gamma\}_t - \epsilon \nabla L'_3(\Lambda, \Gamma), \quad (6.22)$$

where  $\{\Lambda, \Gamma\}_t$  denotes the parameter set at the  $t$ th iteration. The value  $\epsilon = 0.05$  was used. The values of parameters  $\{\Lambda, \Gamma\}$  were adjusted after the values of the loss function for all training samples were calculated. The parameter values obtained

in the ML training were used as the initial values; MCE training was carried out additionally to ML training.

As the solution is infinity for this optimization problem, after all processes were completed, the adaptation vectors  $\Gamma_b, b = 1, \dots, B$  were modified by adding vector  $\mathbf{r}$  such that the following equation was satisfied:

$$\sum_{b=1}^B \Gamma_b = \mathbf{0}. \quad (6.23)$$

Inversely, the mean vectors in classifiers were modified by deleting the same vector  $\mathbf{r}$ .

In the experiments, as a comparison with speaker adaptive classifiers (SAC) that were obtained by the proposed training methods, speaker independent classifiers (SIC) were obtained by the conventional training method based on Eq. (6.2), in which adaptation parameter set  $\mathbf{\Gamma}$  was not used.

For each classifier, the performance after adaptation was evaluated by *leave-one-out* testing: each classifier was trained using data from thirty-nine speakers ( $B = 39$ ), recognition was carried out for the other speaker  $V_i$  as a test speaker, and the average recognition rate was calculated in the forty experiments conducted changing the test speaker in the speakers  $V_1, \dots, V_{40}$ . In the process of the recognizer, the adaptation was carried out on each test speaker  $V_i$  using the randomly selected thirty-two feature vectors for each class from the samples of the speaker  $V_i$ . The value of adaptation vector  $\Gamma_i$  was obtained based on the same criterion as that by which the classifier  $\Lambda$  had been trained. Then the classification error rate was calculated for the remaining samples from the speaker  $V_i$  using adapted classifier  $F(\Lambda, \Gamma_i)$ . In addition, the classification error rate before the adaptation was carried out was calculated using the same samples for comparison.

## 6.5.2 Results

First, in the case of using ML criteria, the average values of error rates obtained in the forty experiments by changing the adaptation target speaker are shown in Fig. 2. Here, let Adapted SIC and Adapted SAC represent the SIC and SAC on which the speaker adaptation was carried out. The error rate of 12.1% obtained with Adapted SAC was lower than that of 13.4% obtained with Adapted SIC: comparison by the statistical test based on the difference of the means of two binomial distributions [Nakagawa and Takagi 94] showed that the difference between these results was significant ( $p < 0.0001$ ). The decrease of 25% ( $(1 - \frac{12.1}{13.4}) \times 100$ ) in the value of the error rate from SAC to Adapted SAC was much larger compared with that of 12% from

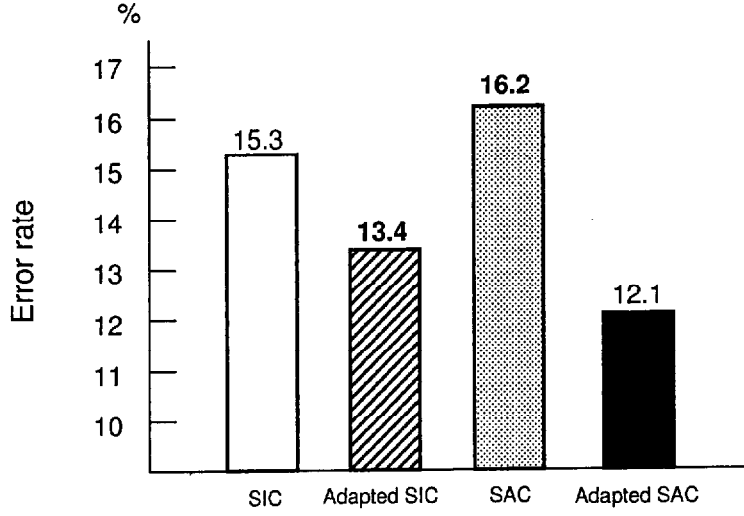


Figure 6.2: Average error rate in the case of using maximum likelihood criteria. SIC and SAC respectively denote the speaker-independent and speaker-adaptive classifiers on which speaker adaptation was not carried out. Adapted SIC and Adapted SAC respectively denote the speaker-independent and speaker-adaptive classifiers after the speaker adaptation was carried out on them.

SIC to adapted SIC. We can also see that, as expected, SIC achieved better performance than did SAC when adaptation is not carried out. This is understandable because SIC was trained so as to get high performance without adaptation, whereas SAC was trained so as to get high performance with adaptation.

To investigate the details of the effect of the proposed training method, we looked at the average log likelihood for the test samples (Fig. 6.3). We can see that in the average log likelihood Adapted SAC obtained a higher value than Adapted SIC did. These results confirm that the maximum likelihood training worked as expected. In addition, the comparison between SAC and SIC in terms of the values of variance parameters  $\sigma_{k,1}^2, \dots, \sigma_{k,16}^2$  are shown in Fig. 6.4. We can see that the values of variance parameters were consistently reduced by the proposed training method. This result shows that the adaptation vectors used in the classifier training process effectively absorb the dispersion due to the difference of speaker characteristics.

The average error rates after the additional training using MCE criteria are shown in Fig. 6.5. Lower values of error rate were obtained by additional MCE training compared with the ML case. The error rate of 9.2% obtained with Adapted SAC was lower than that of 10.7% obtained with Adapted SIC: comparison by the statistical test showed that the difference between these results was significant ( $p < 0.000001$ ).

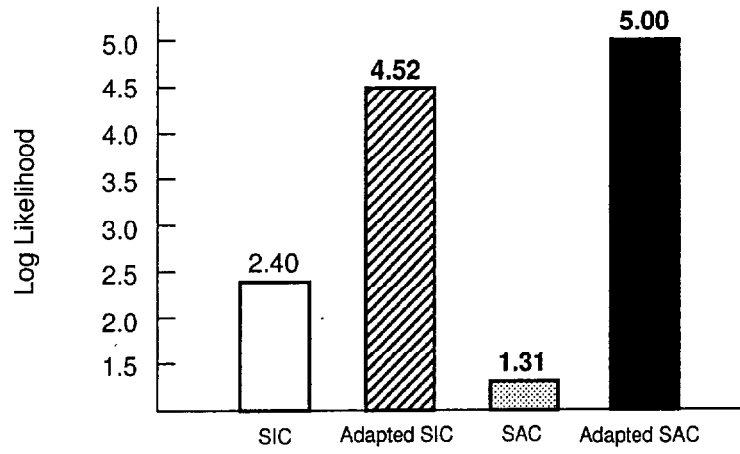


Figure 6.3: Average log likelihood for the test samples

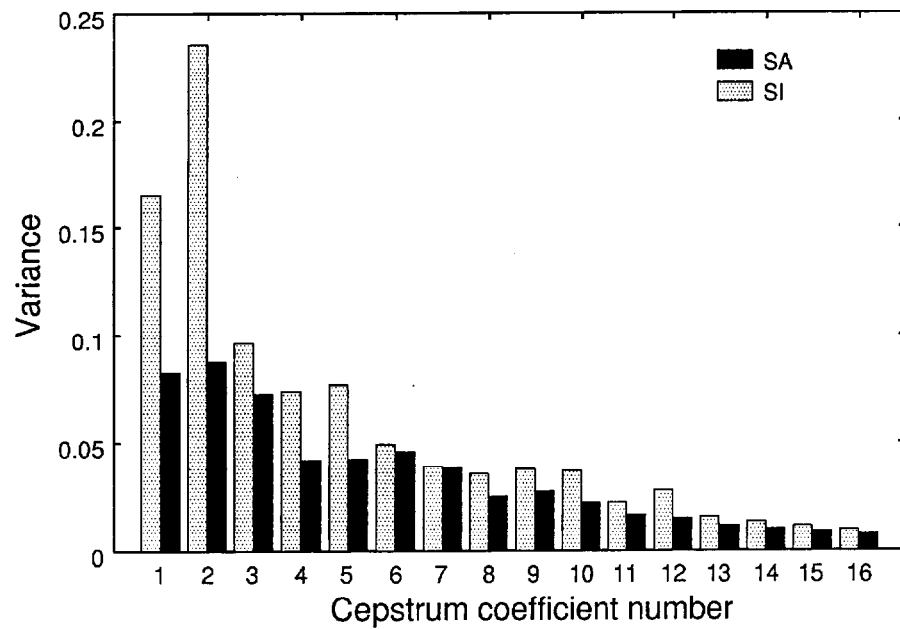


Figure 6.4: Differences in variance parameters of classifiers between speaker independent (SI) classifiers and speaker adaptive (SA) classifiers

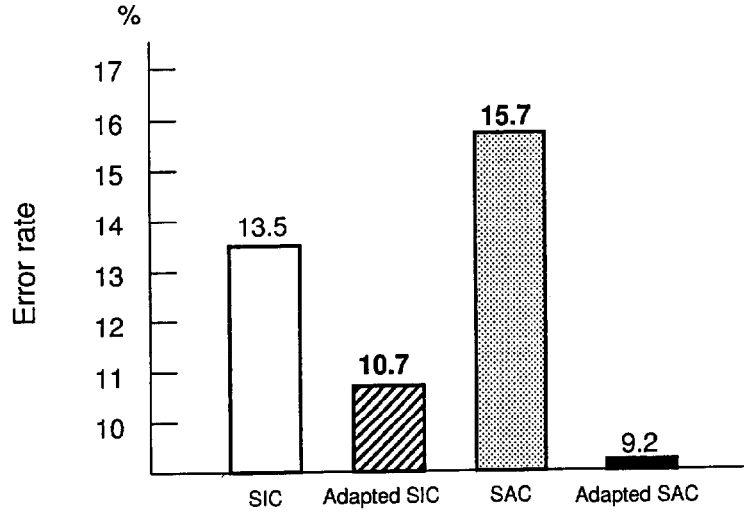


Figure 6.5: Average error rate in experiments in the case of using minimum classification error (MCE) criteria

The decrease of 41% in the value of the error rate from SAC to Adapted SAC was much larger compared with that of 21% from SIC to Adapted SIC.

## 6.6 Discussion

The experimental results confirmed the validity of the proposed method under the adopted experimental condition in that the normal density function was used as each discriminant function and all mean vectors in the discriminant functions were adapted by adding a single adaptation vector. While the classifier training was based on the empirical average loss criteria, the calculation of the expected loss might have to be considered for further improvement.

It appears that the small values of variances in the discriminant functions of the SACs shown in Fig. 6.4 contributed to the improved performance. In the adaptation of SIC, if variance vectors as well as mean vectors were adapted, the performance of the adapted SIC may be improved. Generally, however, the larger the degree of freedom of the adaptation transform function, the more adaptation data is needed to accurately infer the parameters [Gales 98, Hattori and Sagayama 92, Raudy and Jain 91].

The proposed training method, including the formulation of the training objectives and the optimization method for it, is independent of parametric forms of discriminant functions and adaptation transformation, although the conducted experiments were very limited. Thus it might be able to be applied to other kinds

of classifiers, such as artificial neural networks, linear and polynomial discriminant functions, radial-basis function networks, hidden Markov models and so on. In fact, after I originally proposed this method [Iwahashi 95, Iwahashi 96], its application to hidden Markov models as a classifier in speech recognition has been intensively studied (e.g., [Anastasakos et al. 96]), and it has been shown that the proposed method improves the performance in large vocabulary continuous speech recognition task.

This paper has described the case in which classifier parameters are adapted, but the proposed method is also applicable to the case in which the feature extractor is adapted. Moreover, for application to the adaptation methods that change the degree of freedom of the adaptation transform function according to the size of adaptation data (e.g., [Hattori and Sagayama 92]), it might be effective to prepare multiple classifiers dependent on the degree of freedom.

## 6.7 Conclusion

This paper has proposed a new method for training a pattern classifier that achieves high performance after transformation-based adaptation has been carried out on it. It is based on the integration of both classification and adaptation processes in the classifier training. The formulation of training objectives and their optimization method were presented.

The validity of the method was clearly confirmed in vowel recognition experiments in that normal density functions were used as discriminant functions and all mean vectors in the normal density functions were adapted to speakers by adding a single adaptation vector. It should be noted that it is necessary to conduct experiments under a wide variety of conditions in terms of classifier, adaptation, and recognition task in order to confirm the generality of the methods. Simultaneous training of a classifier and a feature extractor [Biem and Katagiri 93] in the framework of the proposed method, *i.e.*, overall optimization of the recognition system composed of feature extraction, classification and adaptation processes, is future work.

# Chapter 7

## Conclusion

This thesis has presented research on the following topics related to improving the speech quality and the flexibility of TTS synthesis.

**Unit selection algorithm based on acoustic measures:**

A new unit selection method based on objective optimization and taking advantage of a large corpus was reported. Four types of distortion were proposed as acoustic quantities of objective measures, and perceptually reasonable and computationally efficient methods for searching for the optimal unit sequence in a large corpus were presented. The method generated natural-sounding speech and led to a new paradigm of TTS synthesis, which made it possible to put many applications of TTS systems to practical use currently.

**Unit set construction algorithm based on segment network optimization:**

To construct a synthesis unit set to minimize segmental and/or intersegmental distortions, a new approach which takes unit set construction as a network optimization problem was reported and its promise was shown by experimental results.

**Statistical modeling for duration control:**

A new statistical modeling which can make, with a moderate amount of learning data, a suitable representation of the complex effects of factors controlling prosody was reported. Also presented was a hierarchical error function that enables hierarchical structure, which is inherent in many linguistic aspects, to be taken into account in the modeling. Experimental results showed that hierarchical characteristics of segmental duration were able to be represented in inferred models.

**Adaptation of speaker characteristics:**

A new method for the adaptation of speaker characteristics was presented, as were experimental results showing that speech close to target speech could be produced using only a small amount of speech spoken by the target speaker.

### **Classifier training method based on Performance-After-Adaptation criteria**

A new method for training pattern classifiers that improve the adaptation performance was also presented. Experimental results showed the validity of the method and provide a framework for integrating the training and the adaptation of a classifier.

The presented work was very promising and contributed to improving the speech quality and the flexibility of TTS system. Higher naturality of synthetic speech, however, should be pursued for sophisticated and intelligent applications. The work described in this thesis indicates several directions we can explore. The author would like to conclude the thesis by listing his major concerns.

- The perceptual measure for the objective evaluation of the speech quality of TTS synthesis should be developed instead of mean-squared errors in the parameters in traditional speech parameterization, and be affectively used in TTS synthesis process.
- In unit selection, the nonlinear effects of perception in multiple distortion measures in terms of spectral characteristics of units, pitch contour, duration, and intensity, should be considered instead of simple summing up such measures.
- The optimization of speech corpus should be studied. Most work on corpus-based synthesis has not addressed the problem of what corpus is appropriate for TTS synthesis, though it should be critical issue.
- The challenge of controlling speaker characteristics in TTS system is attractive and appears to make TTS synthesis flexible in practical use. Only the perceptual and acoustic similarity, however, is evaluated, which does not tell us what we want to know. The methods should be evaluated at a higher level, such as in terms of the effectiveness in a whole human-machine interactive system.
- Most work on automatic phoneme segmentation of speech database for TTS synthesis has merely been the diversion from speech recognition technology

without any particular specification to TTS synthesis. The incorporation of the criteria used in unit selection appears to be valid.

# Appendix A

## Computational Complexity in Speech Segment Network Optimization Problem

We consider the SSN optimization problem from the view point of NP-completeness [Garey and Johnston 80]. At first, we define the following decision problem, which we call MINIMIZING SUM OF ALL DISTANCES; it corresponds to the SSN optimization problem.

### MINIMIZING SUM OF ALL DISTANCES

**Instance:** Collection  $A$  of disjoint subsets of a finite set  $S$ , distance  $h(s_i, s_j) \in Z^+$  for each pair of  $s_i, s_j \in S$  and a bound  $B \in Z^+$  (where  $Z^+$  denotes the positive integers).

**Question:** Is there a subset  $S' \subseteq S$  which contains exactly one element from each subset in  $A$ , such that

$$\sum_{s_i, s_j \in S'} h(s_i, s_j) \leq B \quad ?$$

In this decision problem, we can give a value on intra-segmental and inter-segmental distortions to each distance  $h(s_i, s_j)$  without any loss of generality; other correspondences to the SSN optimization problem are straightforward. Clearly, if we could find an optimal solution to the SSN optimization problem, then we could also solve MINIMIZING SUM OF ALL DISTANCES in polynomial time. All we have to do is find the minimal value of cost in the SSN optimization problem and

compare that value to the given bound  $B$ . Thus, if we could demonstrate that MINIMIZING SUM OF ALL DISTANCES is NP-complete, we would know that the SSN optimization problem is at least as hard. We show computational complexity in the SSN optimization problem by proving the following theorem.

*Theorem 1.* MINIMIZING SUM OF ALL DISTANCES is NP-complete.

*Proof:* It is easy to see that MINIMIZING SUM OF ALL DISTANCES  $\in$  NP, since a nondeterministic algorithm need only guess a subset  $S'$ , which contains exactly one element from each subset in  $A$ , and check in polynomial time whether sum of the distances for all pairs of elements in  $S'$  is no more than  $B$ .

We transform 3-SATISFIABILITY problem, which is NP-complete, to MINIMIZING SUM OF ALL DISTANCES. Let collection  $C = \{c_1, c_2, \dots, c_m\}$  of three-literal clauses on a set  $U = \{u_1, u_2, \dots, u_n\}$  of Boolean variables be arbitrary instance of 3-SATISFIABILITY. We must construct a collection  $A$  of disjoint subsets of a finite set  $S$  and a distance  $h(s_i, s_j) \in \mathbb{Z}^+$  for each pair of  $s_i, s_j \in S$  and a bound  $B \in \mathbb{Z}^+$ , such that there is a subset  $S' \subset S$  which contains exactly one element from each subset in  $A$ , and satisfies

$$\sum_{s_i, s_j \in S'} h(s_i, s_j) \leq B,$$

if and only if  $U$  is satisfiable.

We construct such instance through a graph  $G(V, E)$  corresponding to the instance of 3-SATISFIABILITY. For each variable  $u_i \in U$ , given a subgraph  $T_i(V_i, E_i)$  with  $V_i = \{u_i, \bar{u}_i\}$  and  $E_i = \{\{u_i, \bar{u}_i\}\}$ , that is, two vertices joined by a single edge. For each clause  $c_j \in C$ , given a subgraph  $Q_j(V'_j, E'_j)$ , consisting of three vertices and three edges joining them:

$$\begin{aligned} V'_j &= \{r_1[j], r_2[j], r_3[j]\} \\ E'_j &= \{\{r_1[j], r_2[j]\}, \{r_1[j], r_3[j]\}, \{r_2[j], r_3[j]\}\} \end{aligned}$$

In addition, for each clause  $c_j \in C$ , let the three literals in  $c_j$  be denoted by  $x_1[j]$ ,  $x_2[j]$ , and  $x_3[j]$ . Edges emanating from  $Q_j$  are given by:

$$E''_j = \{\{r_1[j], x_1[j]\}, \{r_2[j], x_2[j]\}, \{r_3[j], x_3[j]\}\}$$

The graph  $G(V, E)$  with  $|V| = 2n + 3m$  is constructed as follows:

$$\begin{aligned} V &= \left( \bigcup_{i=1}^n V_i \right) \cup \left( \bigcup_{j=1}^m V'_j \right), \\ E &= \left( \bigcup_{i=1}^n E_i \right) \cup \left( \bigcup_{j=1}^m E'_j \right) \cup \left( \bigcup_{j=1}^m E''_j \right). \end{aligned}$$

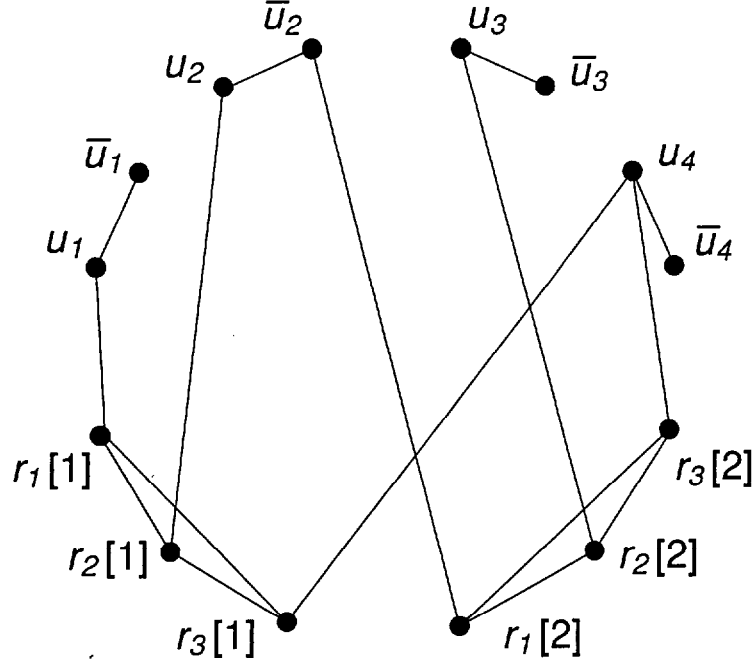


Figure A.1: Graph corresponding to 3-SATISFIABILITY instance:  $U = \{u_1, u_2, u_3, u_4\}$  and  $C = \{\{u_1, u_2, u_4\}, \{\bar{u}_2, u_3, u_4\}\}$

Figure A.1 shows an example of the graph corresponding to  $U = \{u_1, u_2, u_3, u_4\}$  and  $C = \{\{u_1, u_2, u_4\}, \{\bar{u}_2, u_3, u_4\}\}$ .

Moreover, we construct the instance of MINIMIZING SUM OF ALL DISTANCES corresponding to the graph  $G(V, E)$ . The corresponding instance of MINIMIZING SUM OF ALL DISTANCES has a set  $S$  that is identical to  $V$  with  $|S| = 2n + 3m$ . Each subset in  $A$  is identical to each of subsets  $V_i$  and  $V'_j$  with  $|A| = n + m$ . For any two elements  $s_i, s_j \in S$  which are identical to  $v_i, v_j \in V$ , distance  $h(s_i, s_j)$  is defined to be 2 if  $\{v_i, v_j\} \in E$  and 1 otherwise. The bound  $B$  is set equal to  $(n + m)(n + m - 1)/2$ : the number of all pairs of subsets in  $A$ .

It is easy to see how the construction can be accomplished in polynomial time from 3-SATISFIABILITY to MINIMIZING SUM OF ALL DISTANCES, through the graph  $G(V, E)$ . All that remains to be shown is that  $C$  is satisfiable if and only if there is a subset  $S'$  which contains exactly one element from each subset in  $A$ , such that the sum of distances in all pairs of vertices in  $S'$  is no more than  $B$ .

First, suppose that  $S' \subseteq S$  is a subset which contains exactly one element from

each subset in  $A$  with  $|S'| = |A| = n + m$ , such that the sum of distances in all pairs of elements in  $S'$  is no more than  $B$ . Since any two elements are either distance 1 or 2 apart, and since exactly  $(n + m)(n + m - 1)/2$  such distances are summed in checking the boundary condition, the fact that  $B = (n + m)(n + m - 1)/2$  implies that each pair of elements must be exactly distance 1 apart. Thus, for all  $v_i, v_j$  corresponding to  $s_i, s_j \in S'$ , the edge  $\{v_i, v_j\}$  is not in  $E$ . That is, a subset  $W \subset V$  of vertices corresponding to  $S'$  is an independent set for  $G$  with  $|W| = n + m$ , where  $W$  contains exactly one element from each of  $V_i$  and  $V'_j$ . By graph theory, it follows that  $V - W$  is a vertex cover for  $G$  with  $|V - W| = |V| - (n + m) = n + 2m$ .  $V - W$  contains one vertex from each  $T_i$  and two vertices from each  $Q_j$ . Among the three edges in  $E''_j$  emanating from  $Q_j$ , only two of those edges can be covered by vertices in  $V'_j \cap (V - W)$ , so one of them must be covered by a vertex in some  $V_i$  that belongs to  $V - W$ . That implies that clause  $c_j$  is satisfied if the corresponding literal, either  $u_i$  or  $\bar{u}_i$ , is true. This holds for every  $c_j \in C$ . Therefore, a satisfying truth assignment for  $C$  is given as a function  $t : U \rightarrow \{True, False\}$  for each variable in  $U$ , setting  $t(u_i) = True$  if  $u_i \in V - W$  and  $t(u_i) = False$  if  $\bar{u}_i \in V - W$ .

Conversely, suppose that  $t : U \rightarrow \{True, False\}$  is a satisfying truth assignment for  $C$ . The corresponding vertex cover  $Z \subseteq V$  for  $G$  includes one vertex from each  $T_i$  and two vertices from each  $Q_j$  with  $|Z| = n + 2m$ . The vertex from  $T_i$  is  $u_i$  if  $t(u_i) = True$  and is  $\bar{u}_i$  if  $t(u_i) = False$ . This ensures that at least one of the three edges from each set  $E''_j$  is covered, because  $t$  satisfies each clause  $c_j$ . Two vertices from each  $Q_j$  is chosen so as to cover other two edges in  $E''_j$ . Since  $Z$  is a vertex cover for  $G$ , it follows that  $V - Z$  is an independent set for  $G$ , where  $V - Z$  includes exactly one vertex from each of  $V_i$  and  $V'_j$  with  $|V - Z| = n + m$ . Therefore, a subset  $S'$  corresponding to  $V - Z$  contains exactly one element from each subset in  $A$  with  $|S'| = m + n$ , such that the sum of distances in all pairs of elements in  $S'$  is equal to  $B = (n + m) * (n + m - 1)/2$ , and this gives the desired subset.

## Appendix B

# Efficient Search in Growing a CTR Tree

In each step of choosing the best split  $s_s^*$ , all possible splits are represented by the combination of node  $v$ , category set  $\lambda = B_m(\mathcal{X}(v))$ , and its split  $s_c$ . If the finite category set  $\lambda = \{c_1, c_2, \dots, c_A\}$  is given, then the number of all possible splits is  $2^{A-1}$ . Thus, the run time of the category split procedure is exponential in the number of the categories in the set. For example, the duration control model described in Sec. 4.5 adopts the identity of the phoneme as a predictor variable. This factor contains 46 categories, and thus the number of permissible splits in the category set becomes  $2^{45}$ : if the exhaustive search is carried out, the run time of the split procedure becomes impossibly large. Efficient search is necessary for practical implementation.

For an regression tree, some efficient split methods have already been considered, whose principles can be applicable to not only regression tree but classification tree also. Fisher [Fisher 58] showed that the least squares splitting is contiguous. That is, if the categories  $c_i$ ,  $c_j$  and  $c_k$  have the order  $m(c_i) < m(c_j) < m(c_k)$ , and if categories  $c_i$  and  $c_k$  are assigned to the same subset, then category  $c_j$  must be assigned to that same categories, where  $m(c)$  is the mean over all samples which respond the category  $c$ . Thus, for binary split, it is enough to consider  $(A - 1)$  splits, and choose the best split. Breiman *et al.* [Breiman et al. 84] extended Fisher's result to the case response is binary (two class classification), and to arbitrary convex impurity measures. Moreover, this result was extended by Chou [Chou 91] to  $K$ -ary response (in the classification case) and vectors of arbitrary length as response (in the regression case). In the method proposed by Chou, sub-optimal split was searched in linear time (in  $A$ ) per iteration by the  $K$ -means algorithm (e.g. see [Duda and Hart 73]) or the generalized Lloyd algorithm [Linde et al. 80].

Unfortunately, these methods directly can not be applied to CTR. They need the condition that values of parameters for all nodes except the node at that split is carried out are fixed. This condition is not always satisfied in CTR. Thus, different search method for CTR must be developed.

To achieve efficient search, the split procedure is broken down into two processes. One is the process of selecting the best category split on the condition that values of parameters, that are decided by the preceding split sequence, are fixed. The other is the process of the estimation of overall parameters under the given category split. For the estimation, the method described in Sec. 4.3.3 can be used. These two processes are done alternately in an iterative fashion. In each process, over all squared error decreases or does not vary. Thus, locally optimal solution can be obtained by this iterative procedure.

In the process of selecting category split among above two processes, the previously mentioned split methods for ordinary regression tree can be used with small modification. In Fisher's method, categories are ordered according to mean values in corresponding sample data. In the case of CTR, however, mean in sample data is not suitable for optimal split, because some predictions have already been given at the nodes in the subtree rooted by the node at which the category split is carried out. The mean in errors for sample data, obtained by the CTR tree which has been constructed by the preceding split sequence, is used instead. Clearly, Fisher's result holds for the category split at internal node in CTR by this substitution.

Now consider the  $k$ th split. The mean values  $\epsilon(c)$  in errors for sample data, that respond each category  $c$  in the split node, are ordered as follows:

$$\epsilon(c_{l(1)}) < \epsilon(c_{l(2)}) < \cdots < \epsilon(c_{l(A)}), \quad (\text{B.1})$$

where  $c_l(i)$  is the  $i$ th ordered category. All possible sets of categories divided from  $\lambda$  are given as

$$\begin{aligned} \lambda'_h &= \{c_{l(1)}, \dots, c_{l(h)}\} \\ h &= 1, \dots, A - 1. \end{aligned} \quad (\text{B.2})$$

Moreover, let the over all squared error in the sample for the split node, obtained by selecting the set  $\lambda'$ , be  $\epsilon(\lambda')$ .  $\epsilon(\lambda'_h)$  is unimodal in  $h$ . In the search among the all possible sets, thus, the linear search methods [Scale 85], such as Fibonacci search, can also be applicable for seeking the best split, resulting more efficiency in some extent.

When the order of categories is given in advance and fixed, the ordering process is omitted. The parameter estimation is carried out for each possible split, and the optimal split can be selected among these without the iterative process. This

strategy for the fixed-order case is also applicable to non-ordered category case; the order is fixed by first ordering process in each category split process, and the overall parameter estimation is carried out for each possible split, and the best split is selected among them.

In addition, to pursue the more reduction of computational cost, nodes for split are limited; splits are allowed only at nodes near the root node. Since the farther the node is from the root, the smaller the space corresponding to it is, this restriction means that large spaces are allowed to be split.

# Bibliography

- [Abe 91] Abe, M., "A segment-based approach to voice conversion," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 765-768, 1991.
- [Abe and Imai 79] Abe Y., Imai S., "Speech synthesis from the C-V Syllable Parameters" (in Japanese), *Trans. the Committee on Speech Research, The Acoustical Society of Japan*, S79-41, pp.309-314, 1979.
- [Abe and Sato 92] Abe, M and Sato, H., "Two-stage F0 control model using syllable based F0 units," *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Vol.II, pp.53-56, 1992.
- [Abe et al. 88] Abe, M., Nakamura, S., Shikano, K. and Kuwabara, H., "Voice conversion through vector quantization," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1988, pp.655-658, 1988.
- [Abe et al. 89] Abe, K., Takeda, K. and Sagisaka, Y., "On the concatenation of speech synthesis units according to unit extraction context" (in Japanese), *Institute of Electronics, Information and Communication Engineers Japan Tech. Rep.*, SP89-66, pp.17-22, 1989.
- [Anastasakos et al. 96] Anastasakos, T., McDonough, J., Schwartz, R. and Makhoul, J. 1996. A compact model for speaker-adaptive training, *Proc. Int. Conf. on Spoken Language Processing*, Vol.2, 1137-1140.
- [Atal 83] Atal, B., "Efficient coding of LPC parameters by temporal decomposition", *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pp.81-84, 1983.
- [Bartkova and Sorin 87] Bartkova, K. and Sorin, C., "A model of segmental duration for speech synthesis in French," *Speech Communication*, Vol.6, pp.245-260, 1987.

- [Biem and Katagiri 93] Biem, A. and Katagiri, S, "Feature extraction based on minimum classification error/generalized probabilistic descent method," *Proc. Int. Conf. on Audio, Speech and Signal Processing*, Vol.II, pp.275-278, 1993.
- [Breiman et al. 84] Breiman, L., Friedman, J. H. and Stone, C. J., *CLASSIFICATION AND REGRESSION TREES*, The Wadsworth statistisc/probability series, 1984.
- [Broomhead and Lowe 88] Broomhead, D. S. and Lowe, D., "Radial Basis Functions, Multi-Variable Function Interpolation and Adaptive Networks," *Royal Signals and Radar Establishment Memorandum 4148*, 1988.
- [Campbell 91] Campbell, W. N., "Segment durations in a syllable frame," *Journal of Phonetics*, Vol.19, No.1, pp.37-48, 1991.
- [Campbell 92] Campbell, W. N., *Multi-level Timing in Speech*, PhD thesis, Sussex University UK, 1992.
- [Campbell 93] Campbell, W. N., "Predicting segmental durations for accommodation within a syllable-level timing framework," *Proc.EUROSPEECH*, 1993.
- [Childers et al. 89] Childers, D. G., Wu, K., Hicks, D.,M. and Yegnanarayana, B., "Voice Conversion," *Speech Communication*, Vol.8, pp.147-158, 1989.
- [Chou 91] Chou, P. A., "Optimal Partitioning for Classification and Regression Trees," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.13, No.4, pp.340-354, 1991.
- [Chou et al. 89] Chou, P. A., Lookbaugh, T. and Gray, R. M., "Optimal pruning with applications to tree structured source coding and modelling," *IEEE Trans. on Information Theory*, Vol.35, No.2, 1989.
- [Duda and Hart 73] Duda, R. O. and Hart, P. E., *Pattern classification and scene analysis*, New York: Wiley, 1973.
- [Dutoit 96] Dutoit, H., *An Introduction to Text-to-Speech Synthesis*, Academic Publishers, 1996.
- [Fisher 58] Fisher, W. D., "On grouping for maximum homogeneity," *Journal of American Statistics Association*, Vol.53, pp.789-798, 1958.
- [Gales 98] Gales, M. J. F., "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language* Vol.12, No.2, pp.75-98, 1998.

- [Garey and Johnston 80] Garey, M. R. and Johnston, D. S., *Computers and intractability*, W. H. Freeman and Company, 1980.
- [Hakoda 87] Hakoda, K., "Methods of converting voice characteristics between male and female by modifying pole parameters" (in Japanese), *Proc. Fall Meeting of Acoustical Society of Japan*, pp.213-214, 1987.
- [Hattori and Sagayama 92] Hattori, H. and Sagayama, S., "Vector field smoothing principle for speaker adaptation," *Proc. Int. Conf. on Audio, Speech and Signal Processing*, Vol.1, pp.381-384, 1992.
- [Hayashi 50] Hayashi, C., "On the quantification of qualitative data from the mathematico-statistical point of view," *Ann. Inst. Math*, 2, 1950.
- [Hirai et al. 96] Hirai, T., Iwahashi, N., Higuchi, N. and Sagisaka, Y., "Automatic extraction of  $F_0$  control rules using statistical analysis," *Speech Synthesis*, Springer-Verlag, 1996.
- [Hirokawa and Hakoda 90] Hirokawa, T. and Hakoda, K., "Segment selection and pitchmodification for high quality speech synthesis using waveform segments," *Proc. of Int. Conf. on Spoken Language Processing*, pp.337-340, 1990.
- [Nomura et al. 90] Nomura, T., Mizuno, H. and Sato, H., "Speech synthesis by Optimum Concatenation of Phoneme Segments," *Proc. of the ESCA Workshop on Speech Synthesis, Autrans, France*, pp.39-42, 1990.
- [Huang et al. 96] Huang, X., Acero, A., Adcock, J., Hon, H.-W., Goldsmith, J., Liu, J. and Plumpe, M., "WHISTLER: A trainable text-to-speech system," *Proc of Int. Conf. on Spoken Language Processing*, pp.2387-2390, 2000.
- [Hunt and Black 96] Hunt, A. and Black, A., "Unit selection in a concatenative speech synthesis system using a large speech database," *Proc of Int. Conf. on Acoustics, Speech and Signal Processing*, pp.373-376, 1996.
- [Huggins 68] Huggins, A. W. F., "The perception of timing in natural speech I: Compensation within the syllable," *Language and Speech*, 11, pp.1-11, 1968.
- [Ichikawa et al.] Ichikawa, M., Iwata, M., Mitome, Y. and Fushikida, K., "Phoneme Sequence Unit for Speech Synthesis by Rule" (in Japanese), *Trans. Committee on Speech Research, The Acoustical Society of Japan*, SP-87-6, 1987.

- [Imai 80] Imai, S., "Log Magnitude Approximation LMA Filter" (in Japanese), *Trans. The Institute of Electronics, Information and Communication Engineers*, Vol.J63-A, No.12, pp.886-893, 1980.
- [Imai and Abe 79] Imai, S. and Abe, Y., "Spectral Envelope Extraction by Improved cepstral method" (in Japanese), *Trans. The Institute of Electronics, Information and Communication Engineers* Vol.J62-A, No.4, pp.217-223, 1979.
- [Iwahashi 95] Iwahashi, N., "Design of pattern classifier considering adaptation" (in Japanese), *Proc. of the autumn meeting of the acoustical society of Japan*, 1995.
- [Iwahashi 96] Iwahashi, N., "Design method for a pattern classifier suited to adaptation," *Proc. of the IEEE workshop on neural networks for signal processing*, pp.263-272, 1996.
- [Iwahashi 00] Iwahashi, N., "Training method for pattern classifier based on the performance after adaptation," *Trans. of Institute of Electronics, Information and Communication Engineers*, Vol.E83-D, pp.1560-1566, 1996.
- [Iwahashi et al. 91] Iwahashi, N., Fujiwara, S., Komori, Y., Sugiyama, M. and Sagisaka, Y., "Generation of Speech Synthesis Units by Automatic Labelling" (in Japanese), *Proc. of Acoustic Society of Japan Fall Meeting*, 1-6-21, pp.231-232, 1991.
- [Iwahashi et al. 92] Iwahashi, N., Kaiki, N and Sagisaka, Y., "Concatenative speech synthesis by minimum distortion criteria," *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol.II, pp.65-68, 1992.
- [Iwahashi et al. 93] Iwahashi, N., Kaiki, N. and Sagisaka, Y., "Speech Segment Selection for Concatenative Synthesis Based on Spectral Distortion Minimization," *Trans. of Institute of Electronics, Information and Communication Engineers*, Vol.E76-A, pp.1942-1948, 1993.
- [Iwahashi and Sagisaka 95a] Iwahashi, N., Sagisaka, Y., "Speech spectrum conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks," *Speech Communication*, Vol.16, pp.139-151, 1995.
- [Iwahashi and Sagisaka 95b] Iwahashi, N., Sagisaka, Y., "Speech segment network approach for optimal synthesis unit set," *Computer Speech and Language*, pp.335-352, 1995.

- [Iwahashi and Sagisaka 00] Iwahashi, N., Sagisaka, Y., "Statistical modelling of speech segment duration by constrained tree regression," *Trans. of Institute of Electronics, Information and Communication Engineers*, Vol.E83-D, pp.1550-1559, 2000.
- [Jacobs et al. 91] Jacobs, R. A., Jordan, M. I., Nowlan, S. J. and Hinton, G. E., "Adaptive mixtures of local experts," *Neural Computation*, Vol.3, No.1, pp.79-87, 1991.
- [Juang and Katagiri 92] Juang, B.-H. and Katagiri, S., "Discriminative Learning for Minimum Error Classification," *IEEE Trans. Signal Processing*, Vol.40, No.12, pp.3043-3054, 1992.
- [Junqua and Haton 96] Junqua, J.-C. and Haton, J.-P., *Robustness in Automatic Speech Recognition*, Kluwer Academic Publishers, 1996.
- [Kaiki et al. 91] Kaiki, N., Takeda, K. and Sagisaka, Y., "The control of segmental duration in speech synthesis using linguistic properties," *TALKING MACHINES: Theories Models, and Applications*, Elsevier Science, 1991.
- [Kaiki et al. 92] Kaiki, N., Takeda, K. and Sagisaka, Y., "Linguistic properties in the control of segmental duration for speech synthesis," *Talking Machines: Theories, Models, and Designs*, Elsevier Science Publishers, pp.255-263, 1992.
- [Kirkpatrick et al. 83] Kirkpatrick, S. Gelatt, C. D. and Jr. and Vecchi, M. P., "Optimization by simulated annealing," *Science*, Vol. 220, pp.671-680, 1983.
- [Klatt 80] Klatt, D. H., "Software for a cascade/parallel formant synthesizer," *J. Acoustical Society of America*, Vol.67, No.3, 1980.
- [Klatt 87] Klatt, D. H., "Review of text-to-speech conversion for English," *Journal of the Acoustical Society of America*, Vol.82, No.3, pp.737-793, 1987.
- [Klatt 90] Klatt, D. H. and Klatt, L. C., "Analysis, synthesis, and perception of voice quality variations among female and male talkers," *J. Acoustical Society of America*, Vol.87, No.2, pp.820-857, 1990.
- [Kobayashi et al. 91] Kobayashi, T, Matsuda, T. and Watanabe, K., "Quantification theory dealing with categorical dependency and its application for the modeling of spectral difference of phoneme" (in Japanese), *Trans. The Institute of Electronics, Information and Communication Engineers*, Vol.J74-A, No.3, pp.345-352, 1991.

- [Kosaka et al. 93] Kosaka, T., Takami, J., Sagayama, S., "Rapid speaker adaptation using speaker-mixture allophone models applied to speaker-independent speech recognition," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol.II, pp.570-573, 1993.
- [Kuwabara and Takagi 87] Kuwabara, H., Takagi, T., "Quality Control of Speech by Modifying Formant Frequencies and Bandwidth," *Proc. 11th Int. Congress of Phonetic Science*, pp.281-284, 1987.
- [Laber et al. 88] Laver, J. et al., *ATR/CSTR Speech Database Project Status Report #1*, Edinburgh University, 1988.
- [Lee 90] Lee, K.-F., "Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition," *IEEE Trans. Signal Processing*, Vol.38, No.4, pp.599-609, 1990.
- [Leggetter and Woodland 95] Leggetter, C.H. and Woodland, P.C., "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, Vol.9, No.2, pp.171-185, 1995.
- [Linde et al. 80] Linde, Y., Buzo, A. and Gray, R. M., "An algorithm for vector quantizer design," *IEEE Trans. Communication*, Vol.COM-28, pp.84-95, 1980.
- [Makashay et al. 00] Makashay, M., Wightman, C., Syrdal, K. and Conkie, A., "Perceptual evaluation of automatic segmentation in text-to-speech synthesis," *Proc. Int. Conf. on Spoken Language Processing*, Oct. 2000.
- [Matsui and Furui 95] Matsui, T. and Furui, S., "A study on speaker adaptation based on minimum classification error training," *Proc. of EUROSPEECH*, 1995.
- [Mimura et al.91] Mimura, K., Kaiki, N. and Sagisaka, Y., "Statistically derived rules for amplitude and duration control in Japanese speech synthesis," *Proc. Korea-Japan Joint Workshop on Advanced Technology of Speech Recognition and Synthesis*, 1991.
- [Minami and Furui 95] Minami, Y. and Furui, S., "A maximum likelihood procedure for a universal adaptation method based on HMM composition," *Proc. Int. Conf. on Audio, Speech and Signal Processing*, Vol.1, pp.129-132, 1995.
- [Moulines and Charpentier 90] Moulines, E., Charpentier, F., "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, Vol.9, pp.453-467, 1990.

- [Nakagawa and Takagi 94] Nakagawa, S. and Takagi, H., "Statistical methods for comparing pattern recognition algorithms and comments on evaluating speech recognition performance" (in Japanese), *The Journal of Acoustical Society of Japan*, Vol.50, No.10, pp.849-854, 1994.
- [Nakajima and Hamada 88] Nakajima, S. and Hamada, H., "Automatic Generation of Synthesis Units Based on Context Oriented Clustering," *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing*, pp.659-662, 1988.
- [Nomura et al. 90] Nomura, T., Mizuno, H. and Sato, H., "Speech synthesis by Optimum Concatenation of Phoneme Segments," *Proc. the ESCA Workshop on Speech Synthesis, AuTRANS France*, pp.39-42, 1990.
- [Ohkura et al. 93] Ohkura, K., Rainton, D. and Sugiyama, M., "Noise Robust HMMs Based on Minimum Error Classification," *Proc. Int. Conf. on Audio, Speech and Signal Processing*, Vol.II, pp.75-78, 1993.
- [Olive 90] Olive, J. P., "A New Algorithm for a Concatenative Speech Synthesis System Using an Augmented Acoustic Inventory of Speech Sounds," *Proc. of the ESCA Workshop on Speech Synthesis, AuTRANS, France* pp.25-29, 1990.
- [Port 81] Port, R., "Linguistic timing factors in combination," *Journal of ASA*, Vol.69, No.1, pp.262-274, 1981.
- [Price 89] Price, P., "Male and female voice source characteristics: Inverse filtering results," *Speech Communication*, Vol.8, pp.261-277, 1989.
- [Quinlan 88] Quinlan, J. R., "Discovering rules by induction from large collections of examples," In *Michie, D., Ed. Expert Systems in the Micro Electronic Age*. Edinburgh: University Press, 1988.
- [Rainton 91] Rainton, D. and Sagayama, S., "Minimum Error Classification Training of HMMs - Implementation Details and Experimental Results," *Institute of Electronics, Information and Communication Engineers Technical Report*, SP91, pp.39-46, 1991.
- [Raudy and Jain 91] Raudy, S.J. and Jain, A.K., "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.13, No.3, pp.252-264, 1991.

- [Riley 92] Riley, M., "Tree-based modelling of segmental durations," *Talking Machines: Theories, Models, and Designs*, Elsevier Science Publishers, pp.265-273, 1992.
- [Sagisaka 88] Sagisaka, Y., "Speech synthesis by rule using an optimal selection of non-uniform synthesis units," *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing*, pp.679-682, 1988.
- [Sagisaka and Kaiki 92] Sagisaka, Y. and Kaiki, N., "Optimization of intonation control using statistical F0 resetting characteristics," *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Vol.II pp.49-52, 1992.
- [Sagisaka and Sato 86] Sagisaka, Y. and Sato, H., "Composite Phoneme Units for the Speech Synthesis of Japanese", *Speech Communication*, Vol.5, pp.217-223, 1986.
- [Sagisaka 88] Sagisaka, Y., "Speech synthesis by rule using an optimal selection of non-uniform synthesis units," *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, pp.449-452, 1988.
- [Sakayori 87] Sakayori, T., Sasabe, S. and Kitagawa, H., "Prosody Control using Categorical Multiple Regression in Spoken Sentences" (in Japanese), *Proc. Acoustic Society of Japan*, 3-6-5, pp.195-196, 1987.
- [Salza et al. 87] Salza, P. L., Sandri, S. and Forti, E., "Evaluation of Experimental Diphones for Text-to-Speech Synthesis", *Proc. of European Conf. on Speech Thechnology*, Vol.1, pp.63-66, 1987.
- [Sato 84] Satoh, H., "Speech synthesis using CVC concatenation units and excitation waveform elements" (in Japanese), *Trans. the Committee on Speech Research, The Acoustical society of Japan*, S83-69, 1984.
- [Scale 85] Scale, L. E., *Introduction to non-linear optimization Macmillan Publishers*, 1985.
- [Syrdal et al. 00] Syrdal, A. K., Wightman, C. W., Conkie, A., Stylianou, Y., Beutnagel, M., Schroeter, J., Storm, V, Lee, K.-S. and Makashay, M., "Corpus-based technologies in the AT&T NEXTGEN synthesis system," *Proc of Int. Conf. on Spoken Language Processing*, 2000.
- [Takagi and Kuwabara 86] Takagi, T. and Kuwabara, H., "Speech-quality control by manipulating formant frequency dynamics in Japanese," *Proc. Fall Meeting of Acoustical Society of Japan*, pp.145-146, 1986.

- [Takeda et al. 87] Takeda, K., Sagisaka, Y. and Katagiri, S., "Acoustic-phonetic labels in a Japanese speech database," *Proc. the European Conf. on Speech Technology*, Vol.2, pp.195-198, 1987.
- [Takeda et al. 92] Takeda, K., Abe, K. and Sagisaka, Y., "On the basic scheme and algorithms in non-uniform unit speech synthesis," *Talking Machines: Theories, Models, and Designs*, Elsevier Science Publishers, 1992.
- [Tan and Ejima 89] Tan, Y. and Ejima, Y., "A network with multipartitioning units," *Proc. Int. Joint Conf. on Neural Networks*, Washington, D.C., Vol.II, pp.439-442, 1989.
- [Tohkura and Sagisaka 80] Tohkura, Y. and Sagisaka, Y., "Synthesis by rule using CV-syllables," *Proc. Acoustical Society of Japan*, 3-4-3, pp.623-624, May. 1980.
- [Valbret et al. 92] Valbret, H., Moulines E. and Tubach J. P., "Voice transformation using PSOLA technique," *Speech Communication*, Vol.11, pp.175-187, 1992.
- [Viswanathan and Makhoul 75] Viswanathan, R. and Makhoul, J., "Quantization Properties of Transmission Parameters in Linear Predictive Systems," *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP-23, No.3, pp.309-321, 1975.
- [Yamashita et al. 93] Yamashita, Y., Tanaka, M., Amako, Y., Nomura, Y., Ohta, Y., Kitoh, A., Kakusho, A. and Mizoguchi, R., "Tree-based approaches to automatic generation of speech synthesis rules for prosodic parameters," *Trans. of The Institute of Electronics, Information and Communication Engineers*, Vol.E76-A, No.11, pp.1934-1941, 1993.
- [van Laarhoven and Aarts 87] van Laarhoven, P. J .M. and Aarts, E. H. L., *Simulated Annealing: Theory and Applications*, D. Reidel Publishing Company, 1987.
- [van Santen and Olive 90] van Santen, J. and Olive J., "The analysis of contextual effects on segmental duration," *Computer Speech and Language*, Vol.4, pp.359-390, 1990.

# Achievements

## Publications related to this thesis

### Journal

1. Iwahashi, N., Kaiki, N., Sagisaka, Y., "Speech Segment Selection for Concatenative Synthesis Based on Spectral Distortion Minimization," *Trans. of Institute of Electronics, Information and Communication Engineers*, Vol.E76-A, pp.1942-1948, Nov. 1993.
2. Iwahashi, N., Sagisaka, Y., "Speech spectrum conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks," *Speech Communication*, Vol.16, pp139-151, Feb. 1995.
3. Iwahashi, N., Sagisaka, Y., "Speech segment network approach for optimal synthesis unit set," *Computer Speech and Language*, pp.335-352, Oct. 1995.
4. Iwahashi, N., Sagisaka, Y., "Statistical modelling of speech segment duration by constrained tree regression," *Trans. of Institute of Electronics, Information and Communication Engineers*, Vol.E83-D, pp.1550-1559, Jul. 2000.
5. Iwahashi, N., "Training method for pattern classifier based on the performance after adaptation," *Trans. of Institute of Electronics, Information and Communication Engineers*, Vol.E83-D, pp.1560-1566, Jul. 2000.

## International Conference and Workshop

1. Iwahashi, N., Kaiki, N., Sagisaka, Y., "Concatenative Speech Synthesis by Minimum Distortion Criteria," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. II, pp.65-68, Mar. 1992.
2. Iwahashi, N., Sagisaka, Y., "Speech Segment Network Approach for an Optimal Synthesis Unit Set," *Proc. Int. Conf. on Spoken Language Processing*, pp.479-482, Oct. 1992.
3. Iwahashi, N., Sagisaka, Y., "Duration Modeling with Multiple Split Regression," *Proc. Eurospeech*, pp.329-323, 1993.
4. Iwahashi, N., Sagisaka, Y., "Speech spectrum transformation by speaker interpolation," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Mar. 1994.
5. Iwahashi, N., Sagisaka, Y., "Voice Adaptation Using Multi-Functional Transformation with Weighting by Radial Basis Function Networks," *Proc. Int. Conf. on Spoken Language Processing*, pp.1599-1603, 1994.
6. Iwahashi, N., "Design Method for a Classifier Suited to Adaptation," *IEEE Workshop on Neural Networks for Signal Processing*, Sep. 1996.
7. Iwahashi, N., "Novel training method for classifiers used in speaker adaptation," *Proc. Int. Conf. on Spoken Language*, pp.2119-2122 Oct. 1996.

## Japanese Conference and Workshop

1. Iwahashi, N, Kaiki, N., Sagayama, S, Sagisaka, Y., "Unit Selection with Acoustic Measure for Speech Synthesis" (in Japanese), *Proc. Acoustic Society of Japan Spring Meeting*, 1-6-1, pp.207-208, Mar.1991.
2. Iwahashi, N, Kaiki, N., Sagisaka, Y., "Non-uniform Unit Selection based on Several Acoustic Measures for Speech Synthesis" (in Japanese), *IECIE Technical Report*, SP91-5, pp.33-40, May 1991.
3. Iwahashi, N, Fujiwara, S., Komori, Y., Sugiyama, M. and Sagisaka, Y., "Generation of Speech Synthesis Units by Automatic Labelling," (in Japanese) *Proc. Acoustic Society of Japan Fall Meeting*, 1-6-21, pp.231-232, Oct. 1991.

4. Iwahashi, N and Sagisaka, Y., "Organization Method of Speech Synthesis Segment Set Minimizing Concatenative Distortion" (in Japanese), *Proc. Acoustic Society of Japan Spring Meeting*, 1-2-6, pp.217-218, Mar. 1992.
5. Iwahashi, N. and Sagisaka, Y., "Objective and Subjective Evaluation of Segment Selection Method by Minimization of Acoustic Distortion" (in Japanese), *Proc. Acoustic Society of Japan Spring Meeting*, 2-2-15, pp.281-282, Mar. 1992.
6. Iwahashi, N, Sagisaka, Y., "Speech Segment Network Approach for an Optimal Synthesis Unit Set," *Institute of Electronics, Technical Report*, SP92-51, pp.9-16, Sep. 1992.
7. Iwahashi, N, Sagisaka, Y., "Statistical Modeling of Speech by Space Partition Regression" (in Japanese), *Proc. Acoustic Society of Japan Fall Meeting*, 1-5-11, pp.237-238, Oct. 1992.
8. Iwahashi, N, Sagisaka, Y., "Construction of DIPHONE Segment Set by Speech Segment Network Approach," (in Japanese) *Proc. Acoustic Society of Japan Fall Meeting*, 1-P-18, pp.335-336, Oct. 1992.
9. Iwahashi, N., Sagisaka, Y., "English duration control modeling by Multiple Split Regression," (in Japanese) *Proc. Acoustic Society of Japan Spring Meeting*, Mar. 1993.
10. Iwahashi, N. and Sagisaka, Y., "Speech spectral conversion based on speaker interpolation," (in Japanese) *Proc. Acoustic Society of Japan Fall Meeting*, pp.601-602, Oct. 1993.
11. Iwahashi, N. and Sagisaka, Y., "Speech Spectral Conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks" (in Japanese) *Proc. Acoustic Society of Japan Fall Meeting*, pp.331-332, Oct. 1994.
12. Iwahashi, N., "Design method of classifier considering its adaptation" (in Japanese) *Proc. Acoustic Society of Japan Fall Meeting*, pp.137-138, Oct. 1995.

## Other publications

### Book chapter

1. Sagisaka, Y. and Iwahashi, N., "Objective Optimization in Algorithms for Text-to-Speech Synthesis," In *Speech Coding and Synthesis*, Edited by W.B. Kleijn and K.K. Paliwal, ELSEVIER SCIENCE B.V., 1996.
2. Hirai, T., Iwahashi, N., Higuchi, N. and Sagisaka, Y., "Automatic extraction of F0 control rules using statistical analysis," In *Progress in Speech Synthesis*, Academic Press, 1996.

### Journal

1. Hirai, T., Iwahashi, N., Higuchi, N., and Sagisaka, Y., "Automatic Extraction of Fundamental Frequency Control Rules Using Statistical Analysis" (in Japanese), *Trans. of Institute of Electronics, Information and Communication Engineers*, Vol.E78-D-II, pp.1572-1580, Nov. 1995.

### International Conference and Workshop

1. Sagisaka, Y., Kaiki, N., Iwahashi, N. and Mimura, K., "ATR  $\nu$ -talk speech synthesis system," *Proc. Int. Conf. on Spoken Language Processing*, pp.479-482, Oct. 1992.
2. Hirai, T., Iwahashi, N., Higuchi, N. and Sagisaka, Y., "Automatic extraction of F0 control parameters using statistic analysis," *The second ESCA/IEEE workshop on speech synthesis*, pp.57-60, Sep. 1994.
3. Wang, W. J., Campbell, W. N., Iwahashi, N. and Sagisaka, Y., "Tree-based unit selection for English speech synthesis," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pp.191-194, 1993.
4. Iwahashi, N., Hongchang, P., Honda, H., Minamino, K. and Omote, M., "Stochastic features for noise robust speech recognition," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1998.

5. Iwahashi, N., "Spoken language acquisition algorithm with the conceptual analysis of nonlinguistic perceptual information," *Proc. Int Joint Conf. on Information Science*, pp.852-857, Feb. 2000.
6. Iwahashi, N., "Statistical Model Based Approach to Spoken Language Acquisition," In Riichiro Mizoguchi and John Slaney (Eds.) *PRICAI2000 Topics in Artificial Intelligence*, Springer, 2000.
7. Iwahashi, N., "Language acquisition through a human-robot interface," *Proc. on Spoken Language Processing*, Oct. 2000.
8. Iwahashi, N., "Speaker adaptation in noisy environments based on parameter estimation using uncertain data," *Proc. on Spoken Language Processing*, Oct. 2000

## Japanese Conference and Workshop

1. Iwahashi, N "Identification of Isolated Word Accent Type using Neural Networks" (in Japanese), *Proc. Acoustic Society of Japan*, 2-P-18, pp.165-166, Sep. 1990.
2. Shingo Fujiwara, Iwahashi, N., Komori, Y., Sugiyama, M., "A Hybrid Automatic Phoneme Labelling System Based on HMM and Spectrogram Reading Knowledge" (in Japanese), *Proc. Acoustic Society of Japan Fall Meeting*, 2-5-20, pp.87-88, Oct. 1991.
3. Wang, W. J., Campbell, W. N., Iwahashi, N., Sagisaka, Y., "Unit Selection for English Speech Synthesis Using Regression Trees" (in Japanese), *Proc. Acoustic Society of Japan Fall Meeting*, 1-5-10, pp.235-236, Oct. 1992.
4. Mimura, K., Iwahashi, N., Sagisaka, Y., "The effect of discontinuity on the clearness of synthetic speech," *Proc. Acoustic Society of Japan Fall Meeting*, Mar. 1993.
5. Hirai, T., Iwahashi, N., Higuchi, N. and Sagiska, Y., "Auto Classification of F0 control commands using statistical analysis," *Technical report of Institute of Electronics*, SP94-12, 1994.

6. Hirai, T., Iwahashi, N., Higuchi, N. and Sagisaka, Y. "Differences in F0 control rules extracted from multiple speech corpora," *Proc. Acoustic Society of Japan Fall Meeting*, 2-5-3, pp.271-272, Oct. 1994.
7. Hirai, T., Iwahashi, N., Higuchi, N., and Sagisaka, Y., "Realization of variability in speech synthesis system," *Proc. Society of Electronics Meeting*, S40, 1994.
8. T. Hirai, Iwahashi, N., Higuchi, N., Sagisaka, Y., "Automatic Extraction of F0 Control Parameter using statistic analysis", *Proc. Acoustic Society of Japan Spring Meeting*, Mar. 1994.
9. Iwahashi, N. and Tamura, M., "Spoken language acquisition based on the conceptual pattern analysis in perceptual information," *Technical report of Information Processing Society of Japan*, SIG-SLP, Oct. 1999.
10. Kim, K. and Iwahashi, N., "The algorithms for the acquisition of linguistic speech units based on integrating perceptual information," *Technical report of Institute of Electronics*, TL2000-21, Oct. 2000.
11. Haoka, T. and Iwahashi, N., "Learning of the reference-point-dependent concepts on movement for language acquisition," Technical report of Institute of Electronics, PRMU2000-105, Nov. 2000.
12. Kim, K. and Iwahashi, N., "The acquisition of linguistic speech units with hierarchical structure based on the integration of perceptual information," (in Japanese) *Proc. Acoustic Society of Japan Spring Meeting*, Vol.I, pp.99-100, Mar. 2001.
13. Haoka, T. and Iwahashi, N., "Speech understanding based on cognitive language knowledge," (in Japanese) *Proc. Acoustic Society of Japan Spring Meeting*, Vol.I, pp.159-160, Mar. 2001.

## Other

1. Hirai, T., Iwahashi, N., Higuchi, N. and Sagisaka, Y., "Automatic Extraction of Fundamental Frequency Control Rules by Statistical Analysis", *Systems and Computers in Japan*, Vol.28, No.3, 1997.