한국통신학회 1987년도 춘계학술발표회 논문집(1987. 5. 23)



# RECURSIVE LEAST-SQUARE 알고리즘을 이용한 한국어 음소분류에 관한 연구

°김 회 린 이 황 수 은 종 관

한국과학기술원 통신연구실

# A Study on Korean Phoneme Classification using Recursive Least-Square Algorithm

H.R.Kim H.S.Lee C.K.Un

C.R.L. KAIST

## ABSTRACT

In this paper, a phoneme classification method for Korean speech recognition has been proposed and its performance has been studied. The phoneme classification has been done based on the phonemic features extracted by the prewindowed recursive least-square(PRLS) algorithm that is a kind of adaptive filter algorithms. Appiying the PRLS algorithm to input speech signal, precise detection of phoneme boundaries has been made. Reference patterns of Korean phonemes have been generated by the ordinery vecter quantization(VQ) of feature vectors obtained manualy from prototype regions of each phoneme.

In order to obtain the performance of the proposed phoneme classification method, the method has been tested using spoken names of seven Korean cities which have eleven different consonants and eight different vowels. In the speaker-dependent phoneme classification, the accuracy is about 85% considering simple phonemic rules of Korean language, while the accuracy of the speaker-independent case is far less than that of the speaker-dependent case.

## I. 서론

연속음성 인식과정은 음소나 음절등의 기본단위의 인식에서 출발하여 가정과 검증을 반복하여 단어나 문장등의 인식으로 진행해나가는 bottom-up 인식시스템과 먼저 전체문장을 추정한 후 그 문장에 포함된 구(phrase)나 단어, 그리고 음소등과 같은 기본단위로의 가정과 검증을 반복해 나가는 top-down 인식시스템으로 나눌 수 있다[1]. Bottom-up 인식시스템에서 기본단위로서의 음소의 분할과 분류작업은 전체시스템의 인식율에 결정적 영향을 미치기 때문에 음소인식 과정은 연속음성 인식에 있어 매우 중요한 기본단계이다.

음성신호로부터 음소를 분류해내는 방법은 음소의 분할과 labeling을 분리하여 수행하는 방법과 동시에 수행하는 방법으로 나눌 수 있다. 첫번째 방법으로는 음소의 기준패턴과의 최소 distortion값에 근거하여 음성특징의 stable한 부분과 transition 부분을 구하는 방법[2]과, 시간축상에서의 음성특징의 변화정도에 따라 음소간의 경계를 구하는 방법[3][4]이 있다. 두번째 방법은 이미저장되어 있는 음소단위의 기준패턴과 일정한 구간의 입력음성 부분을 비교하여 labeling을 하고 이 음소열(sequence)에 음성특징에 대한 knowledge를 적용하여 음소의 분할과 labeling을 동시에 수행하게 된다[5][6].

음성특징 추출의 관점에서 볼 때 기존의 linear predictive coding(LPC)에 의해 예측계수(prediction coefficients)를 추출하는 방법[7]이나 filter bank 해석에 의해 주파수 대역 에너지를 이용하는 방법은 모두 음성신호를 일정한 구간으로 분할하여 분석한다. 이와같은 블럭단위의 특징추출 방법 대신에 입력된 음성 신호에 적응 선형예측 알고리즘을 적용하여 각 입력 sample 단위로 특징을 추출할 수도 있다[8]. 이러한 방법은 특징추출시 계산량이 많아지는 단점이 있으나 음성신호의 transition부분과 stable한 부분을 블럭단위의 방법에 비해 정확히 보여줄 뿐만 아니라 음성의 pitch 정보도 정확히 구해낼 수 있다[9].

## II. 음소 특징추출

음소 특징추출을 위해 적응필터 알고리즘의 일종인 prewindowed recursive least-square lattice(PRLSL) 알고리즘에 의하여 sample 단위의 PARCOR 계수(K-coefficient)를 구한다. 이 RLS 알고리즘은 현재의 음성신호를 예측할 때 과거의 모든 음성신호를 이용하는 방식이며 식(1)에서와 같이 과거의 예측오차에 weighting을 가한 총 예측오차를 최소화하도록 K-coefficient를 구하게 된다.

$$e = \sum_{j=-\infty}^{n} w_{n-j}[y(j)-\mathbf{c}^t(n)\mathbf{y}(j)]^2 \qquad (1)$$

여기서 $\{w_{n-j}\}$는 weighting factor이고, $\mathbf{c}(n)=(c_1(n), c_2(n), \ldots, c_p(n))^t$은 시간n에서의 예측계수이며, $\mathbf{y}(j)=(y(j-1), y(j-2), \ldots, y(j-p))^t$은 p개의 입력음성 값으로 이루어진 벡터이다.

이 알고리즘의 결과를 구하기 위해서는 projection operator update formulas를 구해야 하며, 이 과정은 order update와 time update가 포함되는 상당히 복잡한 전개과정을 거친다[12].

다음식은 이 알고리즘의 결과식이다.

$$K_{i+1}(n)=wK_{i+1}(n-1) + e_f(n\mid i)e_b(n-1\mid i)\frac{1}{1-r(n-1\mid i)} \qquad (2.a)$$

$$k^f_{i+1}(n)=\frac{K_{i+1}(n)}{\epsilon_f(n\mid i)}, \quad k^b_{i+1}(n)=\frac{K_{i+1}(n)}{\epsilon_b(n-1\mid i)} \qquad (2.b)$$

$$e_f(n\mid i+1)=e_f(n\mid i)-k^b_{i+1}(n)e_b(n-1\mid i) \qquad (2.c)$$

$$e_b(n\mid i+1)=e_b(n-1\mid i)-k^f_{i+1}(n)e_f(n\mid i) \qquad (2.d)$$

$$\epsilon_f(n\mid i+1)=\epsilon_f(n\mid i)-k^b_{i+1}(n)K_{i+1}(n) \qquad (2.e)$$

$$\epsilon_b(n\mid i+1)=\epsilon_b(n-1\mid i)-k^f_{i+1}(n)K_{i+1}(n) \qquad (2.f)$$

$$r(n\mid i+1)=r(n\mid i)+\frac{e_b^2(n\mid i)}{\epsilon_b(n\mid i)} \qquad (2.g)$$

식(2)에서 $k^f_{i+1}$와 $k^b_{i+1}$는 각각 시간 n에서의 forward K-coefficient와 backward K-coefficient를 나타내며, $e_f(n\mid i+1)$과 $e_b(n\mid i+1)$은 시간 n에서의 (i+1)th-order forward residual과 backward residual을 가르킨다. 또한 $\epsilon_f(n\mid i+1)$과

$\epsilon_b(n \mid i+1)$은 식(3)에서와 같이 이 알고리즘에서 최소화하려는 forward LS cost function과 backward LS cost function을 나타낸다.

$$\epsilon_f(n \mid i+1)=\sum_{j=0}^{n} e_f^2(j \mid i+1) \qquad (3.a)$$

$$\epsilon_b(n \mid i+1)=\sum_{j=0}^{n} e_b^2(j \mid i+1) \qquad (3.b)$$

N이 이 필터의 차수라 할 때, n>N인 각 iteration에서 이 recursion은 i=0 부터 i=N-1 까지 계산된다. 이 알고리즘의 초기조건은 다음식으로 주어진다.

$$r(-1 \mid i)=e_b(-1 \mid i)=K_{i+1}(-1)=0$$

$$\text{for } 0 \le i \le N \qquad (4.a)$$

각 iteration n에서

$$r(n \mid 0)=0 \qquad (4.b)$$

$$e_f(n \mid 0)=e_b(n \mid 0)=y(n) \qquad (4.c)$$

$$\epsilon_b(n \mid 0)=\epsilon_f(n \mid 0)=w\epsilon_f(n-1 \mid 0)+y^2(n) \qquad (4.d)$$

이다.

## III. 음소 기준패턴 구성

앞서 PRLSL 알고리즘에 의해 추출된 K-coefficient는 samplewise adaptation에 의하여 얻어졌기 때문에 그림1에서 볼 수 있듯이 음성신호의 pitch 정보를 상당히 많이 포함하고 있다. 그런데 이 pitch 정보는 speaker identification이나 speaker verification과 같은 일부 분야를 제외하고는 음성인식에서 중요한 정보를 제공하지 않기 때문에 이 알고리즘에 의해 추출된 K-coefficient로부터 pitch 정보를 제거할 필요가 있다. 이를 위해서는 먼저 주어진 K-coefficient를 이용해서 pitch 부분을 검출해 내야하며, 이것은 식(5)로 주어지는 K-coefficient의 squared-difference sum function과 smoothing function을 이용하면 비교적 정확히 구해낼 수 있다.

$$s(n)=\sum_{i=0}^{p}[k_i(n)-k_i(n-1)]^2 \qquad (5.a)$$

$$d(n)=\sum_{j=-1}^{1} s(n+j) \qquad (5.b)$$

여기서 $k_i(n)$은 시간n에서의 i번째 K-coefficient이고 p는 order 이다. d(n)가 어떤 threshold $TH_d$를 넘으면 이곳에 pitch pulse가 있는 것으로 간주한다. 보통 pitch 주기는 유성음의 경우에 160Hz 이하에 있기 때문에 하나의 pitch가 구해지면 다음 pitch는 어느정도 시간이 지난 후에 다시 나타난다. 그러므로 한 pitch를 구하면 그 시간에서부터 $t_d$라는 시간이 지난 후에 다음 pitch를 구하도록 한다. 이와같이 하여 pitch를 구하는 이유는 PRLSL 알고리즘의 수렴(convergence)특성이 pitch로부터 어느정도 시간이 지난 후에 정상상태에 도달하기 때문이다. 무성음 부분에 대해서는 pitch가 존재하지 않기 때문에 유성음 부분에서와 동일한 방법으로 random하게 분할한다.

그림2에 이 알고리즘에 의한 유성음 부분의 pitch단위 segmentation과 무성음 부분의 random segmentation이 그려져 있으며, 특히 유성음 부분에서는 매우 정확하게 pitch가 검출됨을 볼 수 있다.

이제 각 음소의 기준패턴을 구하기 위해 VQ를 이용하려면 여러가지 음소가 포함된 연속음성신호에서 각 음소의 표준패턴(prototype) 을 구해내야 한다. 이것은 앞서 얻어진 분할된 음성신호의 파형을 관찰하고, 여기서 각 음소의 표준이 되는 부분을 결정한 후 각 음소를 대표하는 K-coefficient vector들을 구해낸다. 이 vector들을 구할 때는 사용중인 adaptation 알고리즘의 수렴속도를 고려하여 각 pitch pulse로부터 어느정도 지연시간을 둔 후에 정상상태에 이르면 그중 몇개의 K-coefficient vector를 추출하는 방법을 취한다.

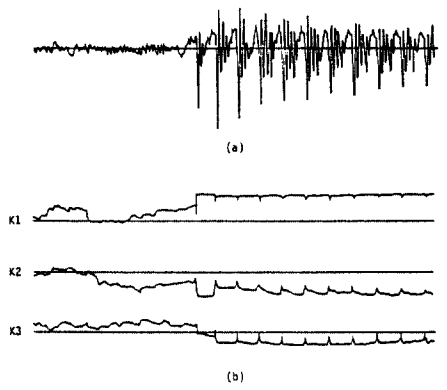


그림 1. 무성음과 유성음에서의 samplewise K-coefficient
(a) 원래의 입력음성 파형 (/삼/)
(b) 반사계수 $k_1$, $k_2$, $k_3$의 변화

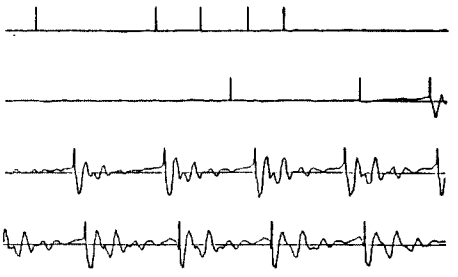

그림 2. 무성음과 유성음 부분의 분할 (/그/)

다음 각 음소에 대한 기준패턴 구성은 지금까지 구한 각 음소별 K-coefficient 벡터를 자기상관계수(autocorrelation coefficient)로 변환시켜 벡터양자화를 행하여 얻을 수 있다.

## IV. 음소인식 과정

지금까지의 음소 기준패턴을 구하는 training과정과 test 음소입력을 인식해내는 과정이 그림3에 그려져 있다. 그림3에 그려져 있는 각 subsystem중에서 test과정을 상세히 설명하면 다음과 같다.

Test하기 위한 음성 data가 입력되면 pre-emphasis를 행한 후 RLS 알고리즘에 의한 K-coefficient vector를 구하는 과정을 거친다. 이로부터 pitch단위로의 분할을 하고 각 frame중에서 정상상태에 도달한 K-coefficient vector 한개를 추출하여 각 음소의 codebook과 distortion을 계산한다. 이때의 distortion measure는 gain-normalized distortion measure를 사용하며, 본 논문에서는 음소에 대한 codebook의 크기를 2개나 4개로 설정했으므로 full search 방법을 이용해서 각 음소와의 distortion을 계산한다.

다음으로 최종적인 decision rule은 모든 기준음소 template와의 distortion 값들 중에서 가장 작은 distortion을 갖는 기준음소 template의 음소를 test data로 인식한다.

## V. Simulation 결과 및 고찰

지금까지 설명한 음소인식 시스템에 대하여 Data General사의 MV/8000 super-mini computer를 사용하여 computer simulation을 하였다. 보통환경에서 발음한 한국어 도시명 7개

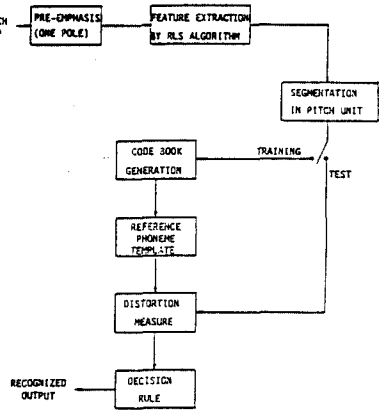


그림 3. 음소인식 시스템의 블럭도

를 음소인식 대상어휘로 선정했으며, 그것은 다음과 같다.

서울, 부산, 대구, 인천, 송탄, 제천, 금성

위와같이 7개의 도시명을 사용한 이유는 음소를 인식대상으로 전제할 때 연속음소로 생각할 수 있고, 이들 도시명내에는 한국어 음소 총 43개[10] 가운데 모음 8개와 자음 11개를 포함하고 있으며, 이들 19개의 음소들은 한국어 음소중에 가장 기본적인 것들을 포함하고 있기 때문이다. 이들 19개의 음소들은 다음과 같다.

모음: 아, 어, 오, 우, 애, 에, 으, 이
자음: ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ, ㅈ, ㅊ, ㅌ

여기서 사용된 음소중에서 모음은 모두 단모음이며, 자음중에는 장애음이 /ㄱ/ㄷ/ㅂ/ㅅ/ㅈ/ㅊ/ㅌ/의 7개이고 비음이 /ㅁ/ㄴ/ㅇ/의 3개, 그리고 유음이 /ㄹ/로 1개이다.

Data base를 만드는 첫 단계로서 임의의 남성화자 6명을 선택하였다. 이중 4명의 화자는 7개의 도시명을 5번씩 반복 발음하고, 2명의 화자는 1번씩 발음하게 하였다.

표2에서 다른 음소가 포함된 경우는 제대로 인식된 음소내에 다른 음소로 결정되는 부분이 포함된 경우로써 올바른 음소는 원으로 구별해 놓았다. 또한 잘못 인식된 음소에서 뒤에 괄호가 있는 것은 그 괄호내의 음소로 인식된 경우이고, 괄호가 없는 것은 일정하게 하나의 음소로 결정하기 힘든 경우를 나타낸 것이다.

화자종속 음소인식 시스템의 기준패턴을 구성하기 위하여 각 도시명 단어에 대해 한 화자가 4번 반복 발음한 음성을 이용하였다. 우선 codebook의 codeword 수가 2개로 된 각 음소에 대한 기준패턴을 만든다. Test 시에는 각 단어에 대해 한 화자가 1번 발음한 음성을 사용하였으며 이 7개의 도시명에는 모두 35개의 음소가 포함되어 있다.

표1은 4명의 화자에 대한 화자종속 음소인식율을 보여주며 표2는 그중 c화자의 인식결과이다. 표1에서 인식의 정확도가 두가지 경우에 대하여 표시되어 있다. 즉 하나는 음성규칙을 전혀 사용하지 않았을 때 맞게 인식된 음소들만의 인식율이고, 다른 하나는 음성발생규칙을 사용하여 보완된 경우까지를 합한 인식율이다.

화자독립 음소인식 시스템의 기준패턴을 구성하기 위하여 각 도시명 단어에 대해 4명의 화자가 4번 반복 발음한 음성을 이용한다. Codebook의 codeword 수가 4개로 된 기준패턴을 만든다음 test시에는 training data 구성에서 제외된 2명의 화자가 1번 발음한 음성을 사용하여 test하였다.

모음의 인식결과를 보면 그림4에서의 모음사각도와 밀접한 관련이 있음을 알 수 있다. 즉, 잘못 인식되거나 다른 모음이 포함되어서 인식된 모음들은 대부분이 모음사각도에서의 인접한 모음으로 인식된 것을 볼 수 있다. 이것은 모음사각도가 혀의 위치에 의해 결정된 것임을 고려할 때 충분히 가능한 결과라 생각된다. 또한 비음(ㄴ, ㅁ, ㅇ)의 경우에는 잘못 인식될 때

표 1. RLS 알고리즘을 이용한 화자종속 음소인식 시스템의 인식율

| 화자종속 | Accuracy (%) (without rule) | Accuracy (%) (rule based) |
|---|---|---|
| A 화 자 | 68.6 | 80.0 |
| B 화 자 | 82.9 | 94.3 |
| C 화 자 | 71.4 | 88.6 |
| D 화 자 | 62.9 | 77.1 |
| 평 균 | 71.4 | 85.0 |

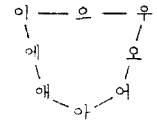


그림 4. 모음사각도

표 2. 화자종속(C) 음소인식 결과

| C 화자 | 맞게 인식된 음 소 | 다른 음소가 포함된 경우 | 잘못 인식된 음 소 |
|---|---|---|---|
| 서 울 | ㅅ,어,우,ㄹ | | |
| 부 산 | ㅂ,우,ㅅ,ㄴ | (아) → 애 → (아) | |
| 대 구 | 애,ㄱ,우 | | ㄷ(ㅌ) |
| 인 천 | 이,ㄴ | ㅁ → (ㄴ)<br>(어) → 아 | ㅊ(ㅈ) |
| 송 탄 | ㅅ,오,ㅇ,아,ㄴ | | ㅌ |
| 제 천 | ㅈ,에,ㄴ | ㅈ → (ㅊ)<br>(어) → 아 | |
| 금 성 | ㄱ,으,ㅁ,ㅅ | (이) → ㄴ → (이) | 어(오) |
| 35 | 25 | 6 | 4 |

다른 비음으로 결정되었는데, 이는 비음을 발음하는 법에 공통점이 있음을 보여준다. 유음(ㄹ)의 경우는 정확히 인식되었고, 장애음의 경우에도 잘못 인식될 때 다른 장애음으로 인식되었다.

이상의 결과를 종합해 볼 때, 잘못 인식된 음소들의 결과가 그 음소의 발음법과 밀접하게 관련되어 있음을 알 수 있으며, 좀더 충분한 음소인식용 data base를 구축하고 한국어 음소에 관한 knowledge를 적용하면 더욱 정확한 결과를 얻을 수 있으리라 예상된다. 이 음소인식 시스템의 보완을 위해서는 한국어 음소에 있어서 각 음소의 발생규칙과 통계적 특성을

연구하고 이로부터 얻어진 knowledge를 이용하여 총 43개의 음소들에 대한 음소인식시스템을 구축해야 할 것이다.

## VI. 결론

본 논문에서는 한국어 음소분류 방식에 대하여 연구하였다. 음소특징 추출을 위해 적응필터 알고리즘의 일종인 prewindowed recursive least-square lattice 알고리즘을 사용하였다. 이 알고리즘을 사용한 이유는 기존의 block processing에 의해 음성특징을 추출할 때 보다 더 정확하게 음성신호의 transition 부분과 stable한 부분을 보여주므로 음소간의 경계검출이 용이하기 때문이다.

음소의 기준패턴 구성을 위해서 vector quantization 방법을 사용하였으며 음소인식 algorithm의 성능시험을 위하여 7개의 도시명을 발음하여 사용하였다. 음소특징 추출을 위하여 PRLSL 알고리즘을 사용한 결과 유성음의 경우 매우 정확한 pitch주기 검출을 할 수 있었고 이를 이용하여 음성에서의 pitch단위의 분할이 가능하였다. 한국어 음소중 자음11개와 모음8개에 대한 기준패턴을 구하기 위하여 음성파형을 관찰하여 각 음소의 표준패턴을 추출해 냈으며 이를 근거로 화자종속 및 화자독립 음소인식 실험을 수행하였다.

그 결과 화자종속인식의 경우 약간의 rule을 고려할 때 약 85%의 음소인식율을 얻었으나, 화자독립인식의 경우 이보다 훨씬 낮은 인식율을 보였다. 그러나 인식된 결과를 볼 때 잘못 인식된 음소는 그 음소의 발음법상 비슷한 다른 음소로 인식되었으며 이는 좀 더 광범위한 음소인식용 data base의 구축과 한국어 음소의 rule을 고려할 때 개선될 수 있을 것으로 예상된다.

표 3. 화자독립(E) 음소인식 결과

| E화자 | 맞게 인식된 음 소 | 다른 음소가 포함된 경우 | 잘못 인식된 음 소 |
|---|---|---|---|
| 서 울 | ㅅ,우,ㄹ | | 어 |
| 부 산 | ㅅ,아 | | ㅂ,우,ㄴ |
| 대 구 | ㄱ | | ㄷ(ㅂ)<br>애(에)<br>우(오) |
| 인 천 | 이,ㄴ,ㅊ,어,<br>ㄴ | | |
| 송 탄 | ㅅ,오,ㄴ | 오 → (아) | ㅌ<br>ㅇ(ㄱ) |
| 제 천 | ㅈ,ㄴ | (어) → 오 | ㅈ(ㅊ)<br>에(이) |
| 금 성 | ㄱ,으 | | ㅁ(ㅇ)<br>ㅅ<br>어(오),ㅇ |
| 35 | 18 | 2 | 15 |

## 참 고 문 헌


1. W.A. Lea, Trends in Speech Recognition. Englewood Cliffs, N.J.: Prentice-Hall, 1980.
2. S. Holtsonen, "Improvement and comparison of three phonemic segmentation methods of speech," in Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 1160-1163, April 1981.
3. S. Holtsonen and P. Ruusunen, "Collection of phoneme samples using time alignment and spectral stationarity of speech signals," in Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 41.5.1-41.5.4, Mar. 1985.
4. Regine ANDRE-OBRECHT, "Automatic segmentation of continuous speech signals," in Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 42.13.1-42.13.4, April 1986.
5. H.C. Leung and V.W. Zue, "A procedure for automatic alignment of phonetic transcriptions with continuous speech," in Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 2.7.1-2.7.4, Mar. 1982.
6. A. Komatsu, A. Ichikawa, K. Nakata, Y. Asakawa, and H. Matsuzaka, "Phoneme recognition in continuous speech," in Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 883-886, May 1982.
7. L.R. Rabiner and S.E. Levinson, "Isolated and connected word recognition -theory and selected applications," IEEE Trans. Commun., vol. COM-29, pp. 621-659, May 1981.
8. T. Fukabayashi and C.K. Chuang, "Speech segmentation and recognition using adaptive linear prediction algorithm," in Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 17.12.1-17.12.4, Mar. 1984.
9. J.M. Turner, "Application of recursive exact least square ladder estimation algorithm for speech recognition," in Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 543-545, May 1982.
10. 하웅, 국어음운학. 정음사, 1984.
11. N. Nocerino, F.K. Soong, L.R. Rabiner, and D.H. Klatt, "Comparative study of several distortion measures for speech recognition," in Proc. Int. Conf. Acoust., Speech, Signal Processing, pp. 1.7.1-1.7.4, Mar. 1985.
12. M.L. Honig and D.G. Messerschmitt, Adaptive Filters. Bell Communications Research, Inc., 1984.
13. F. Itakura, "Minimum Prediction residual principle applied to speech recognition," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 52-72, Feb. 1975.