EURASIP Journal on Applied Signal Processing 2005:7, 1127–1136

Adaptive Peak Frequency Estimation Using a Database of PARCOR Coefficients

Tetsuya Watanabe

Department of Communications Engineering, Graduate School of Engineering, Osaka University,
Suita 565-0871, Japan
Email: tetsuya@comm.eng.osaka-u.ac.jp

Youji Iiguni

Department of Systems Innovation, Graduate School of Engineering Science, Osaka University,
Toyonaka 560-8531, Japan
Email: iiguni@sys.es.osaka-u.ac.jp

Received 10 March 2004; Revised 23 July 2004; Recommended for Publication by Chin-Hui Lee

This paper presents an adaptive peak frequency estimation method using a database that stores PARCOR coefficients as key attributes and the corresponding peak frequencies as nonkey attributes. The least-square lattice algorithm is used to recursively estimate the PARCOR coefficients to adapt to changing circumstances. The nearest neighbor to the current PARCOR coefficient is retrieved from the database, and the corresponding peak frequency is regarded as the estimation. A simultaneous execution of database construction and peak estimation with database update is performed to accelerate the processing time and to improve the estimation accuracy.

Keywords and phrases: peak frequency, database retrieval, AR spectrum, PARCOR coefficient, adaptive estimation.

1. INTRODUCTION

Estimation of peak frequency of the power spectrum plays an important role in direction-of-arrival estimation, communication system, fault diagnosis, and speech processing. Direction-of-arrival of multiple wave sources arriving on a linear array antenna can be estimated by finding local maxima of the space spectrum obtained from the array inputs [1]. Also in biological signals such as electroencephalogram, blood pressure, heart rate, and laser Doppler flow, peak frequencies provide important information on the diagnosis of diseases [2, 3, 4, 5]. Peak frequencies of speech signals are an important cue in the characterization of speech sounds since they have a close relation to the phonetic content. Therefore, extensive studies on peak frequency estimation have been conducted for many years [6, 7, 8, 9, 10, 11, 12, 13].

There are two approaches to power spectrum estimation: nonparametric estimation methods and parametric estimation methods. The FFT (fast Fourier transform) method is a representative one of the nonparametric methods, and the AR (autoregressive) method is a representative one of the nonparametric methods. The FFT method estimates the power spectrum directly from the data. The FFT method is very fast, however, the estimation error variance increases as the number of observations decreases. Moreover, the computationally expensive peak picking is necessary for extracting peak frequencies from the FFT spectrum. The AR method estimates the AR parameters and characterizes the power spectrum by the AR parameters. The AR method provides high-resolution spectrum even with a small number of observations [14]. However, polynomial root finding or peak picking is necessary for extracting peak frequencies from the AR spectrum.

When signal statistics change with time as often happen in real applications, adaptive estimation of peak frequencies is required to adapt to the change. The short-time FFT uses a sliding data window with a constant duration to accommodate nonstationary data. The sliding window technique can also be employed in the AR method. However, the iterative computation of FFT spectrum or AR spectrum is computationally expensive.

In the AR spectrum method, the peak frequency is usually estimated by tracking roots of the AR polynomial rather than AR parameters. Several root-finding algorithms have been derived based on the LMS (least mean squares) algorithm [15, 16, 17] and the RLS (recursive least squares) algorithm [18]. The AR model constructed as a cascade of 2nd-order sections has also been designed to estimate the roots of the AR polynomial, because the roots can be easily obtained by solving 2nd-order polynomial equations [19].

Several adaptive algorithms for estimating coefficients of the cascade AR model have been derived [20, 21]. However, the adaptive algorithms require much computation time. Moreover, the cost function to be minimized is unimodal, and thus the adaptive algorithms may get stuck at local minima.

There is a one-to-one correspondence between AR coefficients and partial autocorrelation (PARCOR) coefficients, and therefore we can characterize the AR spectrum by the PARCOR coefficients instead of the AR coefficients. The LSL (least-square lattice) algorithm can recursively estimate the PARCOR coefficient in a computationally efficient manner [22]. The LSL algorithm is derived based on the least-square-criterion, nevertheless, the computation time is very fast. The PARCOR coefficients are well suited for fixed-point arithmetic, because they are robust against rounding errors and the absolute value is assured to be less than unity.

This paper proposes a fast and recursive peak frequency estimation method using a database of PARCOR coefficients. In database construction process, we estimate the PARCOR coefficients from real speech signals by the LSL algorithm, and compute the corresponding peak frequencies by using either root-finding or peak-picking technique. We put the quantized PARCOR coefficients as key attributes and the corresponding peak frequencies as nonkey attributes, and then enter a pair of the quantized PARCOR coefficients and the peak frequencies into a database. In estimation process, we estimate the PARCOR coefficients from new observations by the LSL algorithm, retrieve a record with a key nearest to the current key from the database, and use the corresponding peak frequency as the estimation. This estimation method requires neither root finding nor peak picking.

The estimation accuracy strongly depends on the database contents. The database size would become intractably large if we would store a large number of records obtained under many different circumstances. Such a large database requires large computation time and memory. We thus perform database construction and peak frequency estimation simultaneously. More precisely, we estimate the current PARCOR coefficient and enter it into the database only when the database does not contain a record with a key close to the current one. The simultaneous execution of database construction and estimation is useful for decreasing processing time and increasing estimation accuracy. Moreover, we update database contents according to timestamp so that the number of records does not exceed the predetermined threshold. The database update procedure prevents a monotonous increase of database size and thus resulting in reduction of processing time. We apply the proposed estimation method to real speech signals to evaluate the estimation accuracy, the processing speed, and the storage space.

2. AR SPECTRUM ESTIMATION

2.1. AR model and AR spectrum

We define a signal at time n as s_n, a white noise with mean zero and variance σ_e^2 as e_n, and the ith AR coefficient as a_i. The Pth AR model is then represented by

$$s_n = -\sum_{i=1}^{P} a_i s_{n-i} + e_n. \tag{1}$$

The AR model can be regarded as the system that inputs e_n, outputs s_n, and has the transfer function

$$H(z) = \frac{1}{A(z)} \tag{2}$$

with

$$A(z) = 1 + \sum_{i=1}^{P} a_i z^{-i}. \tag{3}$$

The AR spectrum of the signal s_n is described by

$$P(\omega) = \frac{\sigma_e^2}{\left|A\left(e^{j\omega}\right)\right|^2}, \tag{4}$$

where the sampling period is assumed to be unity for simplicity. It is evident that the peak frequencies depend only on the values of the AR coefficients $\{a_i\}_{i=1}^{P}$. We here consider the problem of finding peak frequencies that provide local maxima of $P(\omega)$.

2.2. Conventional peak frequency estimation method

There are two methods for finding peak frequencies from AR coefficients: peak picking and root finding. The peak-picking method finds local maxima of $P(\omega)$ by regularly sampling it with a constant sampling interval. Therefore, it is computationally expensive. The root-finding method computes peak frequencies from roots of the AR polynomial equation, defined by $A(z) = 0$. Putting the roots as z_p $(p = 1, 2, \ldots, P)$, we can compute the resonance frequency and the bandwidth by [14]

$$f_p = \frac{\angle z_p}{2\pi T}\,[\text{Hz}], \tag{5}$$

$$B_p = -\frac{\log\left|z_p\right|}{\pi T}\,[\text{Hz}], \tag{6}$$

respectively, where T is the sampling period. The root-finding method is also computationally expensive and is not suited for real-time processing.

Using 2nd-order polynomials, we can express the Pth-order AR polynomial as

$$A(z) = \prod_{i=1}^{P/2}\left(1 + a_1^{(i)} z^{-1} + a_2^{(i)} z^{-2}\right), \tag{7}$$

where $a_1^{(i)}$ and $a_2^{(i)}$ are the 2nd-order coefficients in the ith section. Using the AR model constructed as a cascade of 2nd-order sections, we can easily get the roots of $A(z) = 0$ by solving 2nd-order polynomial equations $1 + a_1^{(i)} z^{-1} + a_2^{(i)} z^{-2} = 0$ [19]. Several adaptive algorithms for estimating the 2nd-order coefficients have been proposed [20, 21], but they require much more computation time than the standard AR parameter estimation algorithms. Moreover, the cost function to be minimized is unimodal, and thus they may get stuck at local minima.

2.3. *Recursive estimation of model coefficients*

There are batch-type and recursive-type algorithms for estimating the AR coefficients $\{a_i\}_{i=1}^{P}$. Recursive-type algorithms are suited for processing nonstationary signals. The representative ones are the LMS and RLS algorithms. The LMS algorithm updates the AR coefficients based on the gradient-descent technique. Unfortunately, it has a drawback that the convergence performance is very sensitive to the eigenvalue spread of the input data covariance matrix [23]. The RLS algorithm updates them based on the least-square method, and therefore it converges faster than the LMS algorithm. However, the RLS algorithm requires $O(P^2)$ operations per iteration, while the LMS algorithm requires only $O(P)$ operations per iteration.

There is a one-to-one correspondence between AR coefficients $(a_1, a_2, \ldots, a_P)$ and PARCOR coefficients $(\rho^1, \rho^2, \ldots, \rho^P)$. Therefore, the peak frequencies can be uniquely characterized by the PARCOR coefficients $\{\rho^p\}_{p=1}^{P}$. The LSL algorithm recursively computes the optimal PARCOR coefficients, in least-squares sense, in $O(P)$ operations. The convergence speeds of the LSL and RLS algorithms are exactly the same since they are derived based on the same least-square criterion. Nevertheless, the computational complexity of the LSL algorithm is almost equal to that of the LMS algorithm. This is an advantage of updating the PARCOR coefficients with the LSL algorithm.

Algorithm 1 summarizes the LSL algorithm, where λ is the exponential forgetting parameter such that $1 - \lambda \ll 1$. The influence of past signal values decays exponentially faster as the size of λ is smaller. The LSL algorithm recursively updates the PARCOR coefficient ρ_n^{p+1} from the observation x_n. The absolute value of the PARCOR coefficient is assured to be less than unity. The PARCOR coefficients can be converted into the AR coefficients by using the following recursion [23]:

$$a_i^{(p+1)} = a_i^{(p)} + \rho^{p+1} a_{p+1-i}^{(p)} \quad (i = 1, 2, \ldots, p), \tag{8}$$

where $a_i^{(p)}$ denotes the ith AR coefficient of order p. In the following, we simply denote $a_i^{(P)}$ as a_i.

3. PEAK FREQUENCY ESTIMATION USING A DATABASE

We will explain a peak frequency estimation method that uses a database of PARCOR coefficients. We first describe how to construct the database from observations, and then show how to estimate peak frequencies by database searching.

$$\begin{aligned}
R_{-1} &= 0, \\
n &= 0, 1, \ldots, \\
R_n &= \lambda R_{n-1} + x_n^2, \\
f_n^0 &= r_n^0 = \frac{x_n}{\sqrt{R_n}}, \\
p &= 0, 1, \ldots, \min(n, P) - 1, \\
\rho_n^{p+1} &= \sqrt{1 - \left(f_n^p\right)^2}\sqrt{1 - \left(r_{n-1}^p\right)^2}\rho_{n-1}^{p+1} - f_n^p r_{n-1}^p, \\
f_n^{p+1} &= \frac{f_n^p + \rho_n^{p+1} r_{n-1}^p}{\sqrt{1 - \left(\rho_n^{p+1}\right)^2}\sqrt{1 - \left(r_{n-1}^p\right)^2}}, \\
r_n^{p+1} &= \frac{r_{n-1}^p + \rho_n^{p+1} f_n^p}{\sqrt{1 - \left(\rho_n^{p+1}\right)^2}\sqrt{1 - \left(f_n^p\right)^2}}.
\end{aligned}$$

Algorithm 1: LSL algorithm.

3.1. *Database construction using training data*

3.1.1. *Quantization of PARCOR coefficients*

The LSL algorithm is used to recursively estimate the PARCOR coefficient ρ_n^p from observation x_n. Since the PARCOR coefficients are assured to be less than unity in magnitude, we can efficiently quantize them with a uniform quantizer. It is known that the AR spectrum becomes more sensitive to changes in PARCOR coefficient as $|\rho^p|$ approaches to unity [24, 25]. Moreover, the first and second PARCOR coefficients of speech signals are relatively larger in magnitude than the others. We thus transform (ρ^1, ρ^2) into (g^1, g^2) by

$$g^p = \log \frac{1 + \rho^p}{1 - \rho^p} \quad (p = 1, 2), \tag{9}$$

and quantize g^p uniformly, so that the quantization interval of ρ^p becomes smaller as $|\rho^p|$ approaches to unity. On the other hand, we quantize $\rho^3, \rho^4, \ldots, \rho^P$ uniformly. We denote the quantized value of ρ^p as $\tilde{\rho}^p$, and then define a vector of the quantized PARCOR coefficients as $\tilde{\boldsymbol{\rho}} = (\tilde{\rho}^1, \tilde{\rho}^2, \ldots, \tilde{\rho}^P)$. We further define the number of bits allocated to the pth PARCOR coefficient as α_p. It is known that spectrum distortion caused by rounding errors of high-order coefficients is relatively small [24, 25]. We thus put $P = 10$, and choose α_p as

$$\begin{aligned}
\{\alpha_p\}_{p=1}^{10} = \{&\alpha, \alpha - 1, \alpha - 1, \alpha - 1, \alpha - 1, \alpha - 1, \alpha - 1, \\
&\alpha - 2, \alpha - 2, \alpha - 2\},
\end{aligned} \tag{10}$$

so that higher-order coefficients are more coarsely quantized.

3.1.2. *Computation of peak frequencies*

We transform the quantized PARCOR coefficients $\{\tilde{\rho}^p\}_{p=1}^{P}$ into the AR coefficients $\{\tilde{a}_i^{(P)}\}_{i=1}^{P}$ by using (8), and then define a vector of the AR coefficients as $\tilde{\boldsymbol{a}} = (\tilde{a}_1^{(P)}, \tilde{a}_2^{(P)}, \ldots, \tilde{a}_P^{(P)})$. We put the peak frequency of the AR spectrum $P(\omega)$ as f_i,

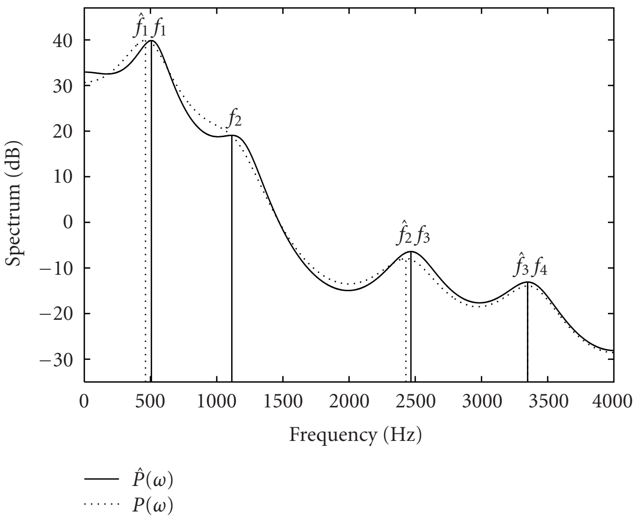


Figure 1: AR specturms with and without quantization.

and then define a vector of the peak frequencies as $\boldsymbol{f} = (f_1, f_2, \ldots)$. We here use the peak-picking (PP) method or the Jenkins-Traub (JT) method to compute $\boldsymbol{f}$ from $\tilde{\boldsymbol{a}}$.

The PP method finds local maxima of $P(\omega)$ by sampling it with a constant sampling interval. We applied the PP method with a sampling interval of 1 [Hz] to a set of real speech signals. Then we have $\boldsymbol{f} = (507, 1114, 2467, 3349)$ (Hz). Next we quantized the AR coefficient by using a quantizer with $\alpha = 6$. Then we have $\boldsymbol{f} = (462, 2429, 3349)$ (Hz). Figure 1 shows the AR spectrums, where the solid line denotes the true AR spectrum and the dashed line denotes the AR spectrum obtained with the quantized AR coefficients. We see that the second peak labeled "f_2" has disappeared due to rounding errors of the AR coefficients. This phenomenon often occurs in a dull peak such as f_2.

The JT method finds the roots of the AR polynomial equation $A(z) = 0$. The resonance frequency f_p and the bandwidth B_p are then computed from the roots by using (5) and (6), respectively. As B_p is smaller, the corresponding peak at frequency f_p becomes sharper. Therefore, in the JT method, we can easily find sharp peaks according to the bandwidth. Therefore, the peak selection method is effective in overcoming the problem of peak disappearance.

3.1.3. Database construction

We put $\tilde{\boldsymbol{\rho}}$ and $\boldsymbol{f}$ as key and data (nonkey), respectively, and enter a record consisting of key and data items into a database at each time step. The kd trie (k-dimensional digital search tree) is used as the data structure to allow efficient range searching [26, 27]. The average computation time for range searching is only $O(\log N)$, where N is the number of records contained in the database. When the database has contained a record with a key equal to the current key, the current record is discarded without being entered into the database.

3.2. Peak frequency estimation using test data

When evaluating the estimation performance, we used a different set of speech signals from the ones used for database construction. We recursively compute ρ_n^p from x_n by the LSL algorithm to adapt to changing statistics of signals. We then obtain a coefficient vector $\tilde{\boldsymbol{\rho}}$ by quantizing ρ_n^p in the same way as in Section 3.1. We choose $\tilde{\boldsymbol{\rho}}$ as a query, and retrieve a coefficient vector nearest to $\tilde{\boldsymbol{\rho}}$, defined by $\hat{\boldsymbol{\rho}}$, from the database by range search. More concretely, we retrieve records with keys lying in the space $\{(\rho_1, \rho_2, \ldots, \rho_p) \mid |\rho_p - \tilde{\rho}_p| \le d,\ \forall p\}$ from the database, where d is the range size, and increase the value of d from 0 by one until more than or equal to one record is retrieved. When more than one record is retrieved, the nearest neighbor to the query $\tilde{\boldsymbol{\rho}}$ is selected out of them and the corresponding data value (peak frequency) is used as the estimation. In this way, we can recursively estimate peak frequencies without solving $A(z) = 0$ nor sampling $P(\omega)$.

3.3. Experimental results

A speech signal of L seconds from a male is sampled at 8 [kHz] to obtain $8000L$ records. We then constructed the database consisting of $8000L$ records. Throughout the simulations, we put $\lambda = 0.995$ in the LSL algorithm so that good performance is achieved. We evaluated the estimation performance by using ten sets of speech signals of one second.

We denote the peak frequency estimation method that directly computes $\boldsymbol{f}$ from $\tilde{\boldsymbol{a}}$ by using the PP and JT methods as PP1 and JT1, respectively. Algorithm 2 summarizes the JT1 and PP1 methods without using a database. We denote the proposed estimation method that uses the PP and JT methods to compute $\boldsymbol{f}$ from $\tilde{\boldsymbol{a}}$ as PP2 and JT2, respectively. Algorithm 3 summarizes the JT2 and PP2 methods with using a database. We can estimate degradation of estimation accuracy due to rounding errors of PARCOR coefficients by comparing the results of the PP1 (JT1) method and PP2 (JT2) method.

We generated several databases by putting $\alpha = 5, 6, 7$ [bit] and changing L from 10 [s] to 500 [s], and measured the estimation accuracies of the PP2 and JT2 methods. Figures 2 and 3 show the results of the PP2 and JT2 methods, respectively. We can measure the estimation accuracy of the PP2 and JT2 methods by the difference between $\hat{f}_k$ and f_k, where f_k is the peak frequency estimated by the PP1 (JT1) method and $\hat{f}_k$ is the one by the PP2 (JT2) method. Spectrum distortion caused by rounding errors of high order coefficients is relatively small. We thus evaluated the estimation accuracy by checking whether the relative error $|\hat{f}_k - f_k|/f_k < 0.1$ is satisfied or not. When using the JT2 method, we selected sharp peaks of bandwidth less than 500 [Hz]. In both methods, better results are obtained by putting $\alpha = 6$ and $L = 500$, and the PP2 and JT2 methods achieve accuracies of 78.8% and 76.8%, respectively. The estimation accuracy seems to be improved as L is larger.

One may think that the use of a finer quantizer improves the estimation accuracy, but Figures 2 and 3 show that the estimation accuracies for $\alpha = 6$ and $\alpha = 7$ are almost the

```
Peak frequency estimation
    (a) Estimate $\boldsymbol{\rho}$ from $x_n$ by the LSL algorithm.
    (b) Transform $\boldsymbol{\rho}$ into $\boldsymbol{a}$.
    (c) Compute $\boldsymbol{f}$ corresponding to $\boldsymbol{a}$ with
        the PP (JT) method.
    (d) Go to step (a).
```

Algorithm 2: JT1 (PP1) method.

```
Database construction using training data
    (a) Estimate $\boldsymbol{\rho}$ from $x_n$ by the LSL algorithm.
    (b) Quantize $\boldsymbol{\rho}$ into $\tilde{\boldsymbol{\rho}}$, and then transform
        $\tilde{\boldsymbol{\rho}}$ into $\tilde{\boldsymbol{a}}$.
    (c) Compute $\boldsymbol{f}$ corresponding to $\tilde{\boldsymbol{a}}$ with the
        PP (JT) method.
    (d) Enter $(\tilde{\boldsymbol{\rho}}, \boldsymbol{f})$ into a database.
    (e) Go to step (a).
Peak frequency estimation using test data
    (a) Estimate $\boldsymbol{\rho}$ from $x_n$ by the LSL algorithm.
    (b) Quantize $\boldsymbol{\rho}$ into $\tilde{\boldsymbol{\rho}}$.
    (c) Find $\hat{\boldsymbol{\rho}}$ by range search, where the range
        size is increased by one until more than
        or equal to one record is retrieved.
    (d) Choose $\boldsymbol{f}$ corresponding to $\hat{\boldsymbol{\rho}}$ as the
        estimation.
    (e) Go to step (a).
```

Algorithm 3: JT2 (PP2) method.

same. We can explain the reason as follows: if a record of $(\tilde{\boldsymbol{\rho}}, \boldsymbol{f})$ is stored for each bin distributed in $\tilde{\boldsymbol{\rho}}$-space, a finer quantization of PARCOR coefficients results in a higher estimation accuracy. However, the database size would become extremely large, because the number of bins is increased by a factor of 2^P as α is increased by one. The estimation accuracy improves with an increase of the number of records, only under the assumption that we store a record for each bin. When the assumption is not satisfied, as is the current case, the improvement is not expected.

As mentioned earlier, the current record is discarded when a record with a key equal to the current key has been stored in the database. We thus evaluated the number of records in the database, the rate to the total number of records, and the storage space. Table 1 summarizes the results for $L = 500$. We see that the database size for $\alpha = 7$ is large enough, and the number of records stored in the database increases as α is larger.

Figures 4a and 4b plot peak frequencies for every 50 time steps estimated by the JT1 and JT2 methods. We put $L = 500$ and $\alpha = 6$ in the JT2 method, because this choice gave good result as shown in Figures 2 and 3. We see that the results of the JT1 and JT2 methods are very close to each other, but differ in some snapshots.

Finally we measured the computation time of the PP1, PP2, JT1, and JT2 methods on a personal computer with an Intel Pentium III 1 GHz. Most of the computation time is spent in range searching in the PP2 and JT2 methods, and peak finding in the PP1 and JT1 methods. In other words, the computational complexities of the LSL algorithm and (8) are negligibly small as compared to range searching and peak finding. The computation times of the PP1, JT1, and JT2 (PP2) methods required for processing signals of one second are 84.0 seconds, 5.3 seconds, and 5.0 seconds, respectively. The JT2 (PP2) method is much faster than the PP1 method, but the computation time is almost the same as that of the JT1 method, and real-time processing is not possible in either case.

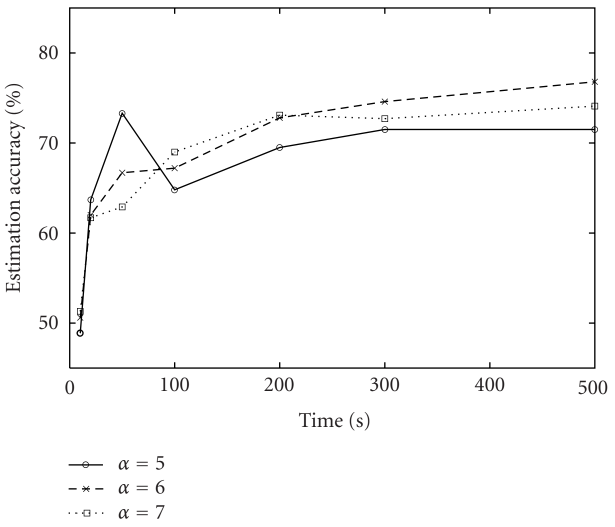


Figure 2: Estimation accuracy of PP2 for different values of L and α.

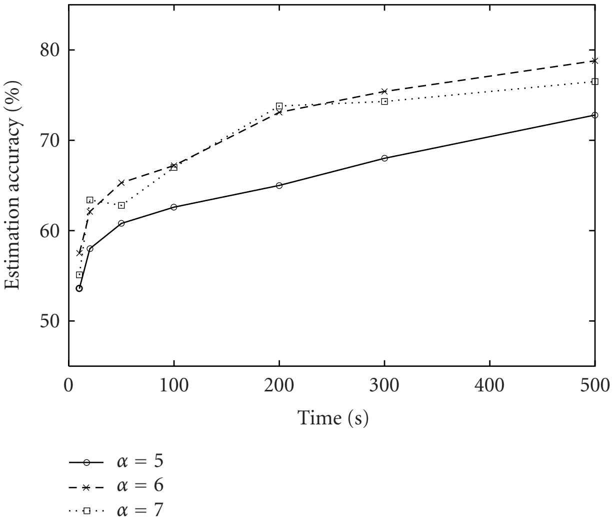


Figure 3: Estimation accuracy of JT2 for different values of L and α.

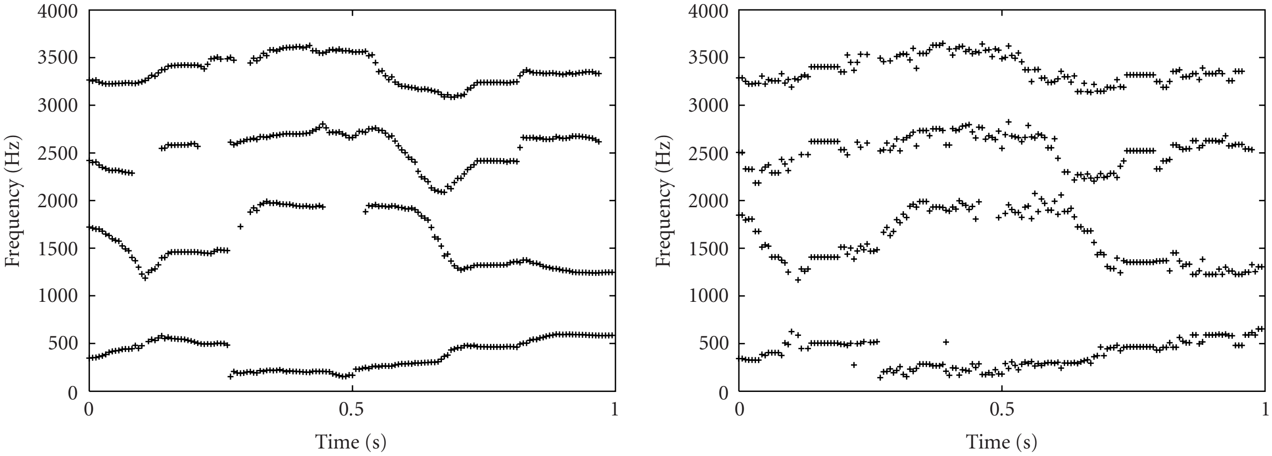
4. IMPROVED ESTIMATION METHOD

4.1. *Simultaneous execution of database construction and estimation*

Both of the PP2 and JT2 methods are not satisfactory in terms of processing speed and estimation accuracy. This is

Table 1: Number of records in the database for different values of α.

α (bit)	5	6	7
Number of records	1.9×10^5 (4.7%)	5.8×10^5 (14.5%)	1.2×10^6 (29.2%)
Storage space (MB)	61	108	170



(a)

(b)

Figure 4: Plots of peak frequencies: (a) JT1 and (b) JT2.

due to the fact that searching in the large database of size 108 (MB) is computationally expensive, and that the range size is repeatedly increased by one until more than or equal to one record is retrieved. The repeated range search increases the processing time, and degrades the estimation accuracy with an increase of range size, because records with keys far apart from a query are extracted from the database.

The estimation accuracies of the PP2 and JT2 methods strongly depend on the database contents, for example, whether training and test data are from the same speaker or not, whether acoustic environments in training and testing phases are the same or not. We have to build a large database if we would store a large number of records obtained from different speakers in different acoustic environments. However, searching in the large database requires large computation time and memory requirements. As can be seen from Table 1, the database size is large enough even when we store speech signals from only one speaker. A further data insertion into the database should be avoided from the view point of processing speed and storage space.

For the solution, we simultaneously execute database construction and peak frequency estimation. We first set the maximum range size as $d_{\max}$ so that records with keys far apart from a query are not retrieved. We then compute the peak frequency vector $\boldsymbol{f}$ associated with the current PARCOR coefficient $\tilde{\boldsymbol{\rho}}$ by using the JT method, only when no record is retrieved by range search with $d = d_{\max}$, we choose $\boldsymbol{f}$ as the estimation, and then enter a record of $(\boldsymbol{f}, \tilde{\boldsymbol{\rho}})$ into the database. The simultaneous execution technique decreases processing time and increases estimation accuracy by appropriately choosing $d_{\max}$. We here denote the improved estimation method using the simultaneous execution as JT3, and summarize it in Algorithm 4. In JT3, the JT method needs to be performed when there is no record close to the current query in the database. Fortunately, the JT method is seldom performed, because statistical properties of speech signals slowly vary with time, and the current key is usually very close to the previous ones.

```
Database construction & peak frequency estimation
  (a) Estimate $\boldsymbol{\rho}$ from $x_n$ by the LSL algorithm.
  (b) Quantize $\boldsymbol{\rho}$ into $\tilde{\boldsymbol{\rho}}$.
  (c) Find $\hat{\boldsymbol{\rho}}$ by range search, where the range size
      is increased from 0 to $d_{\max}$ by one until more
      than or equal to one record is retrieved.
  (d) When more than or equal to one record is
      retrieved, choose $\boldsymbol{f}$ corresponding to $\hat{\boldsymbol{\rho}}$ as the
      estimation, and go to step (a).
  (e) When no record is retrieved by range search
      with $d = d_{\max}$,
        (1) quantize $\boldsymbol{\rho}$ into $\tilde{\boldsymbol{\rho}}$, and then transform
            $\tilde{\boldsymbol{\rho}}$ into $\tilde{\boldsymbol{a}}$,
        (2) compute $\boldsymbol{f}$ from $\tilde{\boldsymbol{\rho}}$ with the JT method,
        (3) enter $(\tilde{\boldsymbol{\rho}}, \boldsymbol{f})$ into the database,
        (4) go to step (a).
```

Algorithm 4: JT3 method.

Table 2: Estimation accuracy and processing time for $\alpha = 6$.

$d_{\max}$	0	1	2	3
Number of records	11655 (14.5%)	9665 (12.1%)	2261 (2.8%)	2212 (2.8%)
Accuracy	91.2%	91.0%	88.0%	88.0%
Time (s)	1.18	0.97	0.55	0.55

Table 3: Estimation accuracy and processing time for $\alpha = 7$.

$d_{\max}$	0	2	4	6
Number of records	22705 (28.4%)	6211 (7.8%)	1516 (1.9%)	1477 (1.8%)
Accuracy	95.6%	94.6%	89.5%	87.4%
Time (s)	1.81	0.79	0.53	0.53

Table 4: Estimation accuracy and processing time for $\alpha = 8$.

$d_{\max}$	0	4	6	8	12
Number of records	36060 (45.0%)	4398 (5.5%)	3200 (4.0%)	1826 (2.3%)	1004 (1.3%)
Accuracy	98.0%	95.4%	95.2%	92.2%	89.1%
Time (s)	2.68	0.70	0.64	0.61	0.63

Table 5: Estimation accuracy and processing time for $\alpha = 9$.

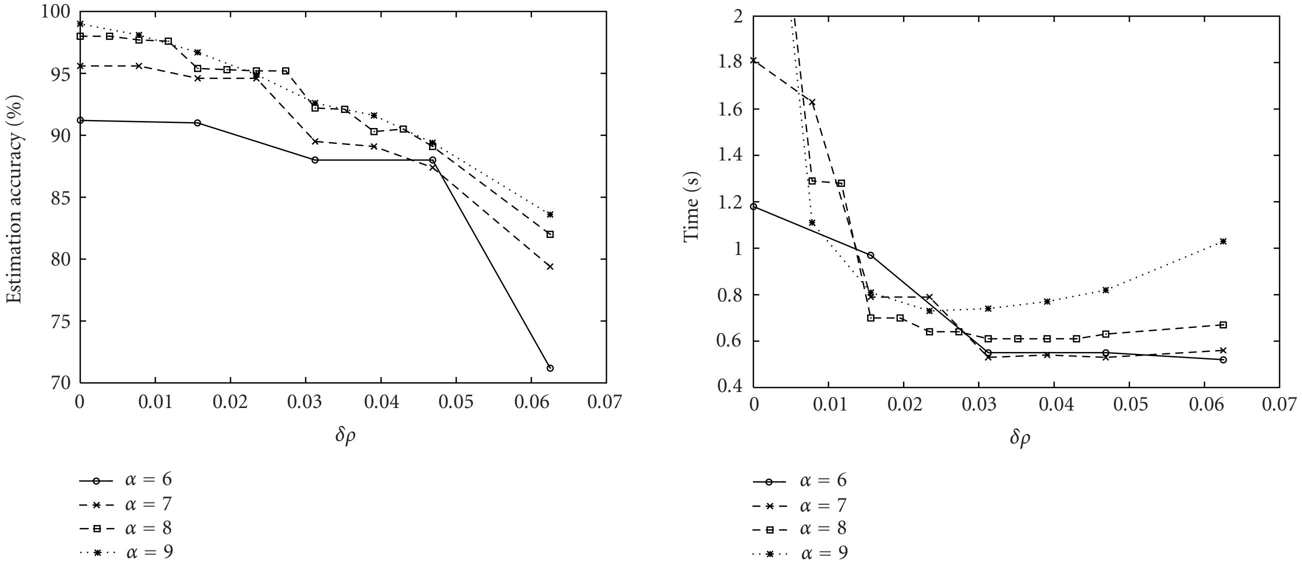
$d_{\max}$	0	8	16	24
Number of records	49793 (62.2%)	5168 (6.5%)	2048 (2.6%)	1095 (1.4%)
Accuracy	99.0%	96.7%	92.6%	89.4%
Time (s)	3.58	0.81	0.74	0.82

4.2. Experimental results

We have applied the JT3 method to ten sets of speech signals of one second, and have measured the estimation accuracy of four peak frequencies (f_1, f_2, f_3, f_4), the number of records in database, and the processing time per 8000 samples (one second) for different values of α and $d_{\max}$. Tables 2, 3, 4, and 5 summarize the results for $\alpha = 6, 7, 8, 9$, respectively. Since the range size in $\widetilde{\boldsymbol{\rho}}$-space is represented by $\delta\rho = d_{\max}/2^{\alpha}$, the actual range sizes for $(\alpha = a, d_{\max} = b)$ and $(\alpha = a+1, d_{\max} = 2b)$ are the same. Figure 5 shows the estimation accuracy and the processing time for different values of $\delta\rho$. We see that the database size and the processing time decrease as $d_{\max}$ is larger, while the estimation accuracy decreases. If we put $d_{\max} = 0$ and $\alpha = \infty$, a perfect accuracy of 100% is achieved. However, instead of achieving perfect accuracy, the time-consuming polynomial rooting is frequently done. A choice of $\alpha = 9$ and $d_{\max} = 0$ achieves the best accuracy of 99.0%, but requires a large processing time of 3.58 seconds. A choice of $\alpha = 6$ and $d_{\max} = 4$ achieves the fastest processing speed of 0.52 second, but provides poor accuracy of 71.2%. We have to make a tradeoff between processing time and estimation accuracy in choosing α and $d_{\max}$. When the sampling interval is large, we can make α large and $d_{\max}$ small to improve estimation accuracy. When the sampling interval is small, we should make α small and $d_{\max}$ small to reduce processing time. The optimal choice depends on the application. In the current application, 8000 sample data must be processed within one second. Therefore, we should put $\alpha = 8$ and $d_{\max} = 6$, because this choice gave a satisfactory accuracy of 95.2% with a processing time of 0.64 second per 8000 samples. Figure 6 plots the estimated peak frequencies. It is evident that the estimation result is better than the previous one in Figure 4b.

4.3. Database update procedure

In the JT3 method, the current record is entered into the database when no record is retrieved by range search with $d = d_{\max}$. Therefore, the database size monotonously increases as time passes, and therefore the processing speed gets slower. This property is not suitable for real-time processing.



(a)

(b)

Figure 5: Results of the improved estimation method: (a) estimation accuracy and (b) processing time.

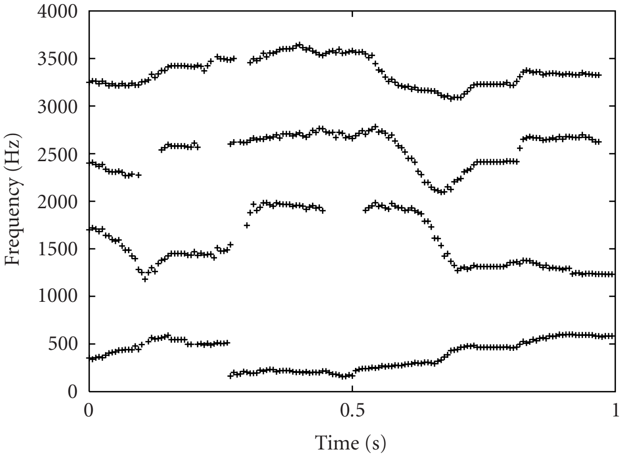


Figure 6: Plots of peak frequencies obtained by JT3.

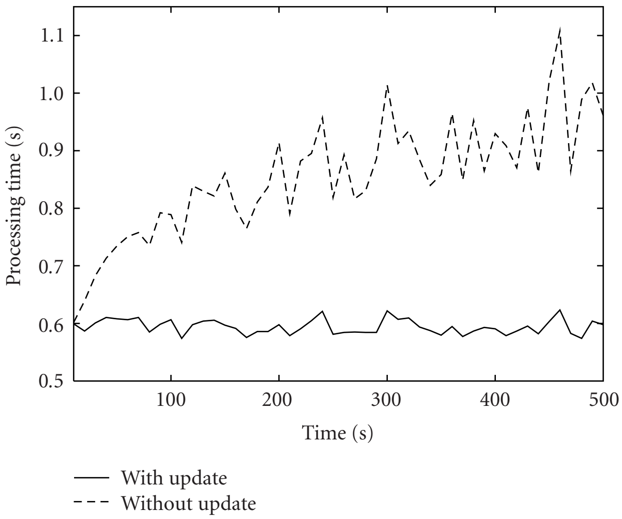


Figure 7: Processing time with and without database update (single speaker).

We thus update the database contents according to timestamp. More precisely, we determined the maximum number of records in the database as $N_{\max}$, and deleted the oldest record from the database if the number of records in the database exceeds $N_{\max}$. The oldest record in the kd trie can be easily deleted without adding the time index into the key field [28].

Table 4 shows that the number of records is about 3200 when we put $\alpha = 8$ and $d_{\max} = 6$. We thus put $N_{\max} = 4000$, and measured the processing time per 8000 samples. Figure 7 plots the result using speech signals of 500 seconds, where the solid and dashed lines denote the results with and without database update, respectively. We see that the processing time of the JT3 method without database update becomes larger as time passes, while the processing time with database update is almost constant, because the maximum number of records in the database is fixed at 4000. The estimation accuracies with and without database update are 92.7% and 92.3%, respectively. They are found to be almost the same. We see that the database update procedure can make the processing time almost constant with maintaining high estimation accuracy.

Finally, we combined several set of speech signals, recorded from a radio, into a single file of speech signals of 500 seconds. Figure 8 shows the result. The estimation accuracies with and without database update are 92.9% and 92.4%, respectively, which are almost the same as the previous ones. We see that the performance of the proposed method with database update is fairly robust against changing statistics of signals and environments.

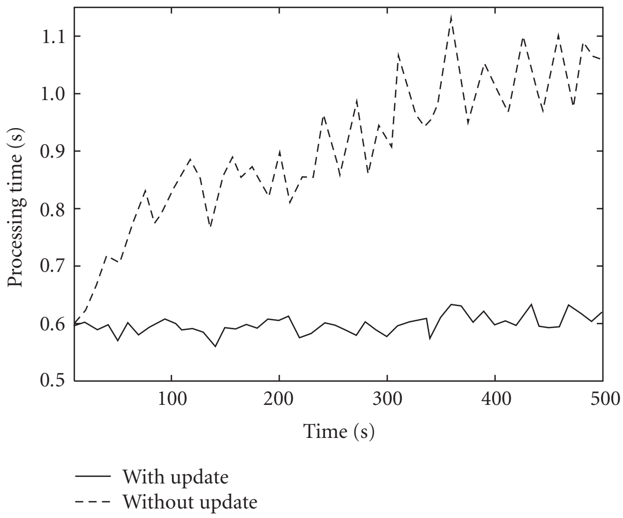


FIGURE 8: Processing time with and without database update (multiple speakers).

4.4. Application to LSP estimation

Also LSP (line spectrum pair) can be uniquely characterized by the PARCOR coefficients, and polynomial root finding is required to compute LSPs from PARCOR coefficients. We thus replaced peak frequencies by LSPs in the key field, and estimated LSPs by using the JT3 method with database update. Figure 9b plots the estimated LSPs for every 50 time steps, where we put $p = 10$ and $\alpha = 6$. The result of the JT1 method is shown in Figure 9a for comparison purpose. The processing times of the JT1 and JT3 methods per 8000 samples are 9.2 seconds and 0.96 second, respectively, and the JT3 method achieves a very good accuracy of 98.4%.

5. CONCLUSION

We have developed the fast and recursive peak frequency estimation method using a database of PARCOR coefficients. We have investigated searching range size and quantization interval of PARCOR coefficients so that a good tradeoff between estimation accuracy and processing speed is achieved. We have simultaneously executed database construction and peak frequency estimation for decreasing processing time and increasing estimation accuracy. Moreover, we have updated the database contents according to timestamp so that processing time is not monotonously increased. We have also applied the database-based method to LSP estimation, and have shown the effectiveness.

The concept of the database-based method originally comes from an intelligent landing system designed previously by one of authors [29]. It is very difficult to model a human skill with a simple mathematical equation. We thus built a database that stores aircraft states as key and control commands provided by a human expert as nonkey, and succeeded to generate a control command close to human operation by database searching. The PARCOR coefficient and the peak frequency presented in this paper correspond to the aircraft state and the control command, respectively.

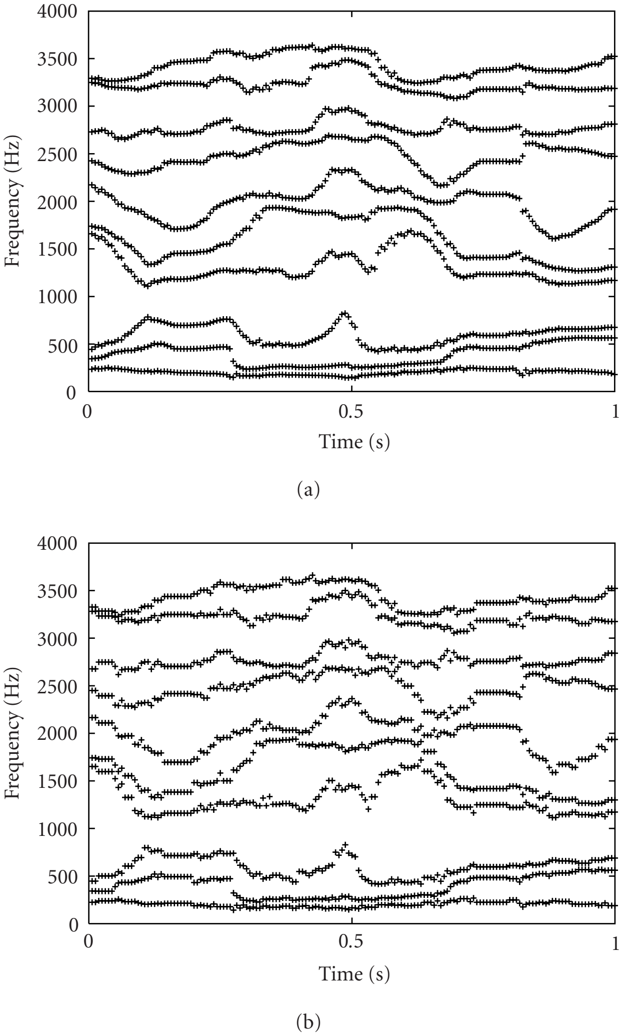


FIGURE 9: Plots of LSPs: (a) JT1 and (b) JT3.

REFERENCES

[1] D. H. Johnson, "The application of spectral estimation methods to bearing estimation problems," *Proc. IEEE*, vol. 70, no. 9, pp. 1018–1028, 1982.

[2] G. Parati, J. P. Saul, D. M. Rienzo, and G. Mancia, "Spectral analysis of blood pressure and heart rate variability in evaluating cardiovascular regulation: A critical appraisal.," *Hypertension*, vol. 25, no. 6, pp. 1276–1286, 1995.

[3] D. W. Rickey and A. Fenster, "Evaluation of an automated real-time spectral analysis technique," *Ultrasound in Medicine and Biology*, vol. 22, no. 1, pp. 61–73, 1996.

[4] M. Arnold, H. Witte, P. Leger, H. Boccalon, S. Bertuglia, and A. Colantuoni, "Time-variant spectral analysis of LDF signals on the basis of multivariate autoregressive modelling," *Technology and Health Care*, vol. 7, no. 2–3, pp. 103–112, 1999.

[5] I. Gülera, M. K. Kiymikb, M. Akinc, and A. Alkan, "AR spectral analysis of EEG signals by using maximum likelihood estimation," *Computers in Biology and Medicine*, vol. 31, pp. 441–450, 2001.

[6] B. S. Atal and S. L. Hanaver, "Speech analysis and synthesis by linear prediction of the speech wave.," *Journal of the Acoustic Society of America*, vol. 50, no. 2, pp. 637–655, 1971.

[7] J. Markel, "Digital inverse filtering-a new tool for formant trajectory estimation," *IEEE Transactions on Audio and Electroacoustics*, vol. 20, no. 2, pp. 129–137, 1972.

[8] S. S. McCandless, "An algorithm for automatic formant extraction using linear prediction spectra," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 22, no. 2, pp. 135–141, 1974.

[9] N. Ouaaline and L. Radouane, "Pole zero estimation of speech signal based on zero tracking algorithm," *Advances in Modelling & Analysis B*, vol. 30, no. 4, pp. 35–48, 1994.

[10] A. Alwan, "Modeling speech production and perception mechanisms and their applications to synthesis, recognition, and coding," in *Proc. IEEE 5th International Symposium on Signal Processing and Its Applications (ISSPA '99)*, vol. 1, p. 7, Brisbane, Queensland, Australia, 1999.

[11] D. W. Guillaume, "A comparison of peak frequency-time plots produced with Hilbert and wavelet transforms," *Review of Scientific Instruments*, vol. 73, no. 1, pp. 98–101, 2002.

[12] P. F. Assmann and W. F. Katz, "Time-varying spectral change in the vowels of children and adults," *Journal of the Acoustical Society of America*, vol. 108, no. 4, pp. 1856–1866, 2000.

[13] I. Bazzi, A. Acero, and L. Deng, "An expectation-maximization approach for formant tracking using a parameter-free non-linear predictor," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, vol. 1, pp. 464–467, Hong Kong, 2003.

[14] S. M. Kay and S. L. Marple Jr., "Spectrum analysis: A modern perspective," *Proc. IEEE*, vol. 69, no. 11, pp. 1380–1419, 1981.

[15] D. Hush, N. Ahmed, and R. David, "Instantaneous frequency estimation using adaptive linear predictor weights," *IEEE Trans. on Aerospace and Electronics Systems*, vol. AES-22, no. 4, pp. 442–431, 1986.

[16] T. J. Shan and T. Kailath, "Directional signal separation by adaptive arrays with a root-tracking algorithm," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP '87)*, pp. 2288–2291, Dallas, TX, USA, 1987.

[17] S. Orfanidis and L. Vail, "Zero-tracking adaptive filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 6, pp. 1566–1572, 1986.

[18] A. Nehorai and D. Starer, "Adaptive pole estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 5, pp. 825–838, 1990.

[19] L. Jackson and S. Wood, "Linear prediction in cascade form," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 6, pp. 518–528, 1978.

[20] P. C. Chin, C. C. Goodyear, and D. Phil, "Adaptive cascade filter for speech analysis," *IEE Proc.*, vol. 130, no. 1, pp. 11–18, 1983.

[21] G. Rigoll, "A new algorithm for estimation of formant trajectories directly from the speech signal based on an extended Kalman-filter," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '86)*, pp. 1229–1232, Tokyo, 1986.

[22] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[23] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*, Wiley & Sons, New York, NY, USA, 1998.

[24] N. Kitawaki and F. Itakura, "Efficient coding of speech by nonlinear quantization and nonuniform sampling of PARCOR coefficients," *Transactions of the Institute of Electronics and Communication Engineers of Japan (IEICE)*, vol. J61-A, no. 6, pp. 543–550, 1978.

[25] N. Kitawaki, F. Itakura, and S. Saito, "Optimum coding of transmission parameters in PARCOR speech analysis synthesis system,," *Transactions of the Institute of Electronics and Communication Engineers of Japan (IEICE)*, vol. J61-A, no. 2, pp. 119–126, 1978.

[26] E. Fredkin, "Trie memory," *Comm. ACM*, vol. 3, no. 9, pp. 490–499, 1960.

[27] J. A. Orenstein, "Multidimensional tries used for associative searching," *Inf. Process. Lett.*, vol. 14, no. 4, pp. 150–157, 1982.

[28] Y. Iiguni, I. Kawamoto, and N. Adachi, "A nonlinear adaptive estimation method based on local approximation," *IEEE Trans. Signal Processing*, vol. 45, no. 7, pp. 1831–1841, 1997.

[29] Y. Iiguni, H. Akiyoshi, and N. Adachi, "An intelligent landing system based on a human skill model," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 3, pp. 877–882, 1998.

Tetsuya Watanabe received his B.E. and M.E. degrees in communications engineering from Osaka University, Osaka, Japan, in 2001 and 2003, respectively. Since 2003, he has been working in Frontier Systems Group Network Division, NEC TOSHIBA Space Systems, Ltd., Yokohama, Japan.

Youji Iiguni received his B.E. and M.E. degrees in applied mathematics and physics from Kyoto, University, Kyoto, Japan, in 1982 and 1984, respectively, and the D.E. degree from Kyoto University in 1990. He was an Assistant Professor at Kyoto University from 1984 to 1995, and an Associate Professor at Osaka University from 1995 to 2003. Since 2003, he has been a Professor at Osaka University. His research interests include signal and image processing.