PAPER *Special Section on Corpus-Based Speech Technologies*

Fundamental Frequency Modeling for Speech Synthesis Based on a Statistical Learning Technique

Shinsuke SAKAI[†a)], ***Member***

SUMMARY This paper proposes a novel multi-layer approach to fundamental frequency modeling for concatenative speech synthesis based on a statistical learning technique called *additive models*. We define an additive F_0 contour model consisting of long-term, intonational phrase-level, component and short-term, accentual phrase-level, component, along with a least-squares error criterion that includes a regularization term. A *backfitting* algorithm, that is derived from this error criterion, estimates both components simultaneously by iteratively applying cubic spline smoothers. When this method is applied to a 7,000 utterance Japanese speech corpus, it achieves F_0 RMS errors of 28.9 and 29.8 Hz on the training and test data, respectively, with corresponding correlation coefficients of 0.806 and 0.777. The automatically determined intonational and accentual phrase components turn out to behave smoothly, systematically, and intuitively under a variety of prosodic conditions.

key words: *speech synthesis, fundamental frequency, additive models, statistical learning*

1. Introduction

In recent years, corpus-based concatenative methods for speech synthesis have received increasing attention within the research community as well as the speech technology industry, because of their ability to generate natural sounding speech output [1]–[3]. In general, for synthesized speech to be natural and intelligible, it is crucial to have a proper F_0 contour that is compatible with linguistic information such as lexical accent (or stress) and phrasing in the input text. In the corpus-based concatenative speech synthesis setting, target F_0 features (e.g., mean frequency, ending frequency, amount of movement) are generated for each synthesis unit. Distance metrics can then be used to compute a cost between the unit target values, and those available in a speech corpus. Overall cost is minimized during search to find the best matching sequence of synthesis units from the corpus.

In some systems, F_0 target is predicted by an independent rule-based front-end [4], while regression tree-based approaches are popularly used to predict F_0-related measures from a set of linguistic features [3], [5], [6]. A regression tree approach is advantageous in that it is simple to implement yet very powerful. It has a few drawbacks, however. For example, the predicted values do not have a smooth contour, since it essentially represents a piecewise constant function of the input features. It also has a drawback that it cannot capture additive structure when the data does have such a structure.

In this work, we propose a simple yet novel multi-layer additive model approach to F_0 contour prediction, and a method to estimate the component functions through the minimization of a residual sum-of-squares error criterion that includes a regularization (or penalty) term. *Additive Models* [7], [8] are a class of nonlinear regression models, which can be regarded as a generalization of linear models (or multiple linear regression). It and its extension by link function, called *Generalized Additive Models*, are described in detail in the monograph [8], and have been applied to various statistical modeling practices such as weather forecast [9] and public health research [10], among others.

In the next section we define a two-layer additive F_0 model, along with a penalized least-squares criterion from which we derive a backfitting algorithm as the minimizer of the criterion. We then describe experimental results applying the proposed method to a large corpus of Japanese speech in the following section.

2. Additive Model Approach

The basic formulation for the F_0 contour is similar to previous work that models F_0 in a superpositional way, e.g., [11] that models F_0 generation mechanism with second-order linear systems, and [12] that uses multiple linear regression with indicator variables. In our two-layer additive modeling approach, the F_0 contour, Y, is regarded as the output of a statistical model that combines a long-range intonational-phrase level component, g, and a shorter accentual-phrase level component, h:

$$\begin{aligned} Y &= \alpha + g(I, U) + h(A, V) + \epsilon \\ &= \alpha + g_I(U) + h_A(V) + \epsilon, \qquad (1) \end{aligned}$$

where α is a constant, I is a discrete input variable that represents a type of intonational phrase, and indexes the relevant function g_I. U is a continuous variable representing a time point relative to the starting point of the phrase of type I. Similarly, discrete variable A designates a type of accentual phrase, and V represents a time point relative to the starting point of the accentual phrase of type A. The random error term, ϵ, is zero mean. Figure 1 shows how the three terms form the entire F_0 contour function.

2.1 Penalized Least Square and Backfitting algorithm

A unique characteristics of the additive model approach, as

Manuscript received June 30, 2004.
Manuscript revised August 31, 2004.

[†]The author is with MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139 U.S.A.
a) E-mail: sakai@sls.csail.mit.edu

DOI: 10.1093/ietisy/e88–d.3.489

Copyright © 2005 The Institute of Electronics, Information and Communication Engineers

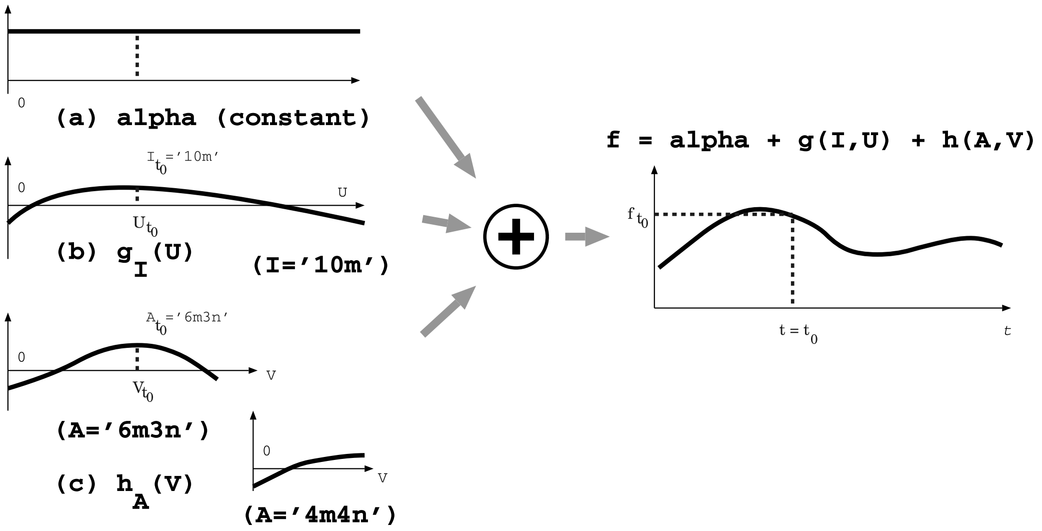


Fig. 1 A schematic diagram of the additive F_0 model $f(I, U, A, V) = \alpha + g_I(U) + h_A(V)$. A constant α and component functions g and h are summed up to form the F_0 contour f. In this example, a 6-mora accentual phrase with the accent nucleus at the third mora ('6m3n', for short), e.g. "monowa'kareni" is followed by an '4m4n' accentual phrase, e.g. "owatta" to form a 10-mora intonational phrase, e.g. "monowa'kareni owatta".

```
(1) Initialize: $\hat{\alpha} = \frac{1}{N}\sum_{n=1}^{N} y_n$,   $\hat{g}_i \equiv 0$, $\hat{h}_a \equiv 0$, $for\ all\ i \in r(I),\ a \in r(A)$.
(2) Cycle: repeat (2g) and (2h) until the functions $\hat{g}_I$ and $\hat{h}_A$ stabilize.
  (2g) Partition the set of training data $\{(i_n, u_n, a_n, v_n, y_n) \mid n = 1, \dots, N\}$, into $|r(I)|$ subsets $\{(i, u_{i,l}, a_{i,l}, v_{i,l}, y_{i,l}) \mid l = 1, \dots, N_i\}$ ($i \in r(I)$), so that each training point has the same value of $i$ if in the same subset. Note that $\sum_{i \in r(I)} N_i = N$.
       For all $i \in r(I)$,
           $\hat{g}_i \leftarrow \mathcal{S}_i \left[ \{y_{i,l} - \hat{\alpha} - \hat{h}_{a_{i,l}}(v_{i,l})\}_{l=1}^{N_i} \right]$.
  (2h) Repartition the training data $\{(i_n, u_n, a_n, v_n, y_n) \mid n = 1, \dots, N\}$ into $|r(A)|$ subsets $\{(i_{a,l}, u_{a,l}, a, v_{a,l}, y_{a,l}) \mid l = 1, \dots, N_a\}$ ($a \in r(A)$), so that each training point has the same value of $a$ if in the same subset. As before, $\sum_{a \in r(A)} N_a = N$.
       For all $a \in r(A)$,
           $\hat{h}_a \leftarrow \mathcal{S}_a \left[ \{y_{a,l} - \hat{\alpha} - \hat{g}_{i_{a,l}}(u_{a,l})\}_{l=1}^{N_a} \right]$.
```

Fig. 2 A backfitting algorithm for the additive F_0 model.

compared to previous work, is that we do not have to assume any parameterized functional form. Instead, we assume a smoothness defined in terms of curvature, and use an estimation scheme derived from a least-squares error criterion with a regularization term, or roughness penalty [7], [8]. We define the penalized residual sum-of-squares (PRSS) error of the model with regard to the overall training data in the following form:

$$\begin{aligned} PRSS &= RSS + \lambda_g J(g) + \lambda_h J(h) \\ &= \sum_{n=1}^{N} \{y_n - \alpha - g_{i_n}(u_n) - h_{a_n}(v_n)\}^2 \\ &\quad + \lambda_g \sum_{s \in r(I)} \int g_s''(w)^2 dw + \lambda_h \sum_{t \in r(A)} \int h_t''(x)^2 dx, \end{aligned} \tag{2}$$

where $(i_n, u_n, a_n, v_n, y_n)$ $(n = 1, \dots, N)$ are a set of training data corresponding to the variables (I, U, A, V, Y), and λ_g, λ_h are fixed smoothing parameters. The number N represents the total number of the available training data points. $r(I)$ and $r(A)$ represents the set of possible values (or *range*) for I and A, respectively. The number of elements in a set will be denoted by vertical bars, e.g., $|r(I)|$ meaning the number of different values for I. The first term in (2) measures the closeness to the data, while the second and third terms penalize the curvatures in the functions, and smoothing parameters λ_g and λ_h establish a tradeoff between them. Large values of λ's yield smoother curves, while smaller values result in more fluctuation.

It can be shown that the minimizer of (2) is an additive cubic spline model, where g_I's and h_A's are *natural cubic splines* in the predictor variables U and V, with *knots*, or break points, at each of the unique values of (i_n, u_n) and (a_n, v_n). To make the solution unique, we assume that $\sum_1^N g(i_n, u_n) = \sum_1^N h(a_n, v_n) = 0$, therefore α will be the overall mean of y_n $(n = 1, \dots, N)$. We can find the solution for (2) with a *backfitting* algorithm [7], a simple iterative procedure depicted in Fig. 2.

In the algorithm, we apply a natural cubic-spline smoother matrix, e.g., $\mathcal{S}_i$, to the vector of partial residual, $\{y_{i,l} - \hat{\alpha} - \hat{h}_{a_{i,l}}(v_{i,l})\}_{l=1}^{N_i}$, which is regarded as a function of $u_{i,l}$, to obtain a new estimate $\hat{g}_i$. Smoothing of partial residual is done for g_i's and h_a's in turn, using the current estimate of the other component function. The iteration is continued until the estimates $\hat{g}_i$'s and $\hat{h}_a$'s stabilize. In the rest of the section, we briefly describe how this backfitting algorithm, with natural cubic spline smoothers, is derived as a iterative procedure equivalent to a blockwise Gauss-Seidel algorithm for solving a system of linear equations emerging from the minimization of the penalized least-square criterion (2).

2.2 Natural Cubic Spline and Its Property

In solving the penalized least square problem, we use a unique property of a family of functions called *natural cubic splines*. A natural cubic spline is a piece-wise cubic function defined in terms of the points called *knots* $x_1, x_2, \ldots, x_K$ on some interval $[a, b]$. It is known that among all twice differentiable functions $g(x)$ defined on $[a, b]$ that passes through the points $(x_1, z_1), (x_2, z_2), \ldots, (x_K, z_K)$ where $a < x_1 < \ldots < x_K < b$, the one minimizing a "roughness" measure defined as an integrated squared second derivative,

$$\int_a^b g''(x)^2 dx,$$

is a natural cubic spline with knots at $x_1, x_2, \ldots, x_K$. We take advantage of this property in solving the penalized least square problem. For convenience of the reader, we review the definition of natural cubic spline and the proof of this property in Appendix A and Appendix B, respectively.

2.3 Derivation of the Backfitting Algorithm

Now, by paying attention to different intonational phrase types, we can partition the entire set of training data into $|r(I)|$ subsets in such a way that the points in a subset have the same value of i_n, i.e., they belong to the same type of intonational phrase. We can then express the entire training data, $\mathcal{D}$, as a union of $|r(I)|$ nonoverlapping subsets:

$$\begin{aligned}\mathcal{D} &= \{(i_n, u_n, a_n, v_n, y_n) \mid n = 1, \ldots, N\}\\ &= \bigcup_{i \in r(I)} \{(i, u_{i,l}, a_{i,l}, v_{i,l}, y_{i,l}) \mid l = 1, \ldots, N_i\},\end{aligned} \tag{3}$$

where $\sum_{i \in r(I)} N_i = N$. Similarly, we can partition the training data based on the identity of the value of a_n:

$$\begin{aligned}\mathcal{D} &= \{(i_n, u_n, a_n, v_n, y_n) \mid n = 1, \ldots, N\}\\ &= \bigcup_{a \in r(A)} \{(i_{a,l}, u_{a,l}, a, v_{a,l}, y_{a,l}) \mid l = 1, \ldots, N_a\},\end{aligned} \tag{4}$$

where $\sum_{a \in r(A)} N_a = N$. By using partitionings in (3) and (4), the expression for the penalized residual sum of squares (2) can now be rewritten in two ways:

$$PRSS = RSS + \lambda_g J(g) + \lambda_h J(h)$$

$$\begin{aligned}&= \sum_{i \in r(I)} \sum_{l=1}^{N_i} \{y_{i,l} - \alpha - g_i(u_{i,l}) - h_{a_{i,l}}(v_{i,l})\}^2\\ &\quad + \lambda_g \sum_{s \in r(I)} \int g_s''(w)^2 dw + \lambda_h \sum_{t \in r(A)} \int h_t''(x)^2 dx\end{aligned} \tag{5}$$

$$\begin{aligned}&= \sum_{a \in r(A)} \sum_{l=1}^{N_a} \{y_{a,l} - \alpha - g_{i_{a,l}}(u_{a,l}) - h_a(v_{a,l})\}^2\\ &\quad + \lambda_g \sum_{s \in r(I)} \int g_s''(w)^2 dw + \lambda_h \sum_{t \in r(A)} \int h_t''(x)^2 dx.\end{aligned} \tag{6}$$

Now, let us consider searching for the optimal function $\hat{g}_{i_0}$ that minimizes the penalized least square criterion (5) for a certain value i_0 of i, when other g_i's ($i \neq i_0$) and h_a's are fixed to certain functions.

Assume we are given any twice continuously differentiable function g that is not a natural cubic spline which passes through the points $(u_{i_0,l}, g(u_{i_0,l}))$ ($l = 1, \ldots, N_{i_0}$). Let $\bar{g}$ be the natural cubic spline that interpolates the same points $(u_{i_0,l}, g(u_{i_0,l}))$ ($l = 1, \ldots, N_{i_0}$). Since $\bar{g}(u_{i_0,l}) = g(u_{i_0,l})$ by definition, it immediately follows that the residual sum of squares error term is exactly the same for these two functions, i.e.,

$$\begin{aligned}&\sum_{l=1}^{N_{i_0}} \{y_{i_0,l} - \alpha - \bar{g}(u_{i_0,l}) - h_{a_{i_0,l}}(v_{i_0,l})\}^2\\ &\quad = \sum_{l=1}^{N_{i_0}} \{y_{i_0,l} - \alpha - g(u_{i_0,l}) - h_{a_{i_0,l}}(v_{i_0,l})\}^2.\end{aligned}$$

On the other hand, due to the property of the natural cubic spline interpolant that we saw in 2.2, it holds that $\int \bar{g}''(t)^2 dt < \int g''(t)^2 dt$. We can therefore conclude that $PRSS(g_{i_0} = \bar{g}) < PRSS(g_{i_0} = g)$. This means that, unless g itself is a natural cubic spline, we can find a natural cubic spline which yields a smaller value of PRSS in (5). It immediately follows that the minimizer $\hat{g}_{i_0}$ of (5) must be a natural cubic spline with knots at each of the unique values of $u_{i_0,l}$ ($l = 1, \ldots, N_{i_0}$). Extending the discussion above to all the instances of g_i in (5) and all instances of h_a in (6), we see that each of g_i's and h_a's has to be a natural cubic spline.

We can now write each of g_i as the linear combination of K_i natural cubic spline basis functions $N_j^{(i)}$ (cf. Appendix A):

$$g_i(u) = \sum_{j=1}^{K_i} N_j^{(i)}(u)\, \theta_j^{(i)}, \quad i \in r(I), \tag{7}$$

where K_i is the number of distinct values for $u_{i,l}$, the time points within the intonational phrase function g_i. Then the vector of the values of g_i at the training data points $u_{i,l}$ ($l = 1, \ldots, N_i$) can be written as

$$\boldsymbol{g}_i = \boldsymbol{N}_i \boldsymbol{\theta}_i \tag{8}$$

where $\boldsymbol{\theta}_i = (\theta_1^{(i)}, \ldots, \theta_{K_i}^{(i)})^T$ and the $N_i \times K_i$ matrix $\boldsymbol{N}_i$ contains K_i cubic spine basis functions evaluated at each of the N_i training data points, i.e. $(\boldsymbol{N}_i)_{l,j} = N_j^{(i)}(u_{i,l})$. Then, by defining a $K_i \times K_i$ matrix $\boldsymbol{\Omega}_{N_i}$ as $(\boldsymbol{\Omega}_{N_i})_{j,k} = \int N_j^{(i)\prime\prime}(x)\, N_k^{(i)\prime\prime}(x)\, dx$, we can write each component roughness penalty for g_i as:

$$\begin{aligned}\int g_i''(x)^2 dx &= \int \left\{ \sum_{j=1}^{K_i} N_j^{(i)\prime\prime}(x)\, \theta_j^{(i)} \right\}^2 dx \\ &= \boldsymbol{\theta}_i^T \boldsymbol{\Omega}_{N_i} \boldsymbol{\theta}_i, \qquad (9)\end{aligned}$$

An accentual phrase function h_a can also be written as a linear combination of natural cubic spline basis functions $N_j^{(a)}$:

$$h_a(v) = \sum_{j=1}^{K_a} N_j^{(a)}(v)\, \theta_j^{(a)}, \quad a \in r(A), \qquad (10)$$

and we can derive the component roughness penalty for h_a in the same way:

$$\int h_a''(x)^2 dx = \boldsymbol{\theta}_a^T \boldsymbol{\Omega}_{N_a} \boldsymbol{\theta}_a, \qquad (11)$$

where $\boldsymbol{\theta}_a$ is a cubic spline coefficient vector $(\theta_1^{(a)}, \ldots, \theta_{K_a}^{(a)})^T$, and $(\boldsymbol{\Omega}_{N_a})_{j,k} = \int N_j^{(a)\prime\prime}(x)\, N_k^{(a)\prime\prime}(x)\, dx$.

PRSS in (5) can now be written in a matrix form:

$$\begin{aligned} PRSS &= \sum_{i \in r(I)} (\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{g}_i - \boldsymbol{h}_i)^T (\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{g}_i - \boldsymbol{h}_i) \\ &\quad + \lambda_g \sum_{i \in r(I)} \boldsymbol{\theta}_i^T \boldsymbol{\Omega}_{N_i} \boldsymbol{\theta}_i + \lambda_h \sum_{a \in r(A)} \boldsymbol{\theta}_a^T \boldsymbol{\Omega}_{N_a} \boldsymbol{\theta}_a \\ &= \sum_{i \in r(I)} (\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{N}_i\boldsymbol{\theta}_i - \boldsymbol{h}_i)^T (\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{N}_i\boldsymbol{\theta}_i - \boldsymbol{h}_i) \\ &\quad + \lambda_g \sum_{i \in r(I)} \boldsymbol{\theta}_i^T \boldsymbol{\Omega}_{N_i} \boldsymbol{\theta}_i + \lambda_h \sum_{a \in r(A)} \boldsymbol{\theta}_a^T \boldsymbol{\Omega}_{N_a} \boldsymbol{\theta}_a, \qquad (12)\end{aligned}$$

where $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,N_i})^T$, $\boldsymbol{\alpha} = (\alpha, \ldots, \alpha)^T$, $\boldsymbol{h}_i = (h_{a_{i,1}}(v_{i,1}), \ldots, h_{a_{i,N_i}}(v_{i,N_i}))^T$. By differentiating (12) with respect to the coefficient vector $\boldsymbol{\theta}_{i_0}$ of one component function $g_{i_0}(u)$ ($i_0 \in r(I)$), and setting the partial derivative to zero, we obtain:

$$\hat{\boldsymbol{\theta}}_{i_0} = (\boldsymbol{N}_{i_0}^T \boldsymbol{N}_{i_0} + \lambda_g \boldsymbol{\Omega}_{N_{i_0}})^{-1} \boldsymbol{N}_{i_0}^T (\boldsymbol{y}_{i_0} - \boldsymbol{\alpha} - \hat{\boldsymbol{h}}_{i_0}). \qquad (13)$$

Similarly, we can derive another matrix form of the penalized least square criterion from (6):

$$\begin{aligned} &PRSS \\ &= \sum_{a \in r(A)} (\boldsymbol{y}_a - \boldsymbol{\alpha} - \boldsymbol{g}_a - \boldsymbol{N}_a\boldsymbol{\theta}_a)^T (\boldsymbol{y}_a - \boldsymbol{\alpha} - \boldsymbol{g}_a - \boldsymbol{N}_a\boldsymbol{\theta}_a) \\ &\quad + \lambda_g \sum_{i \in r(I)} \boldsymbol{\theta}_i^T \boldsymbol{\Omega}_{N_i} \boldsymbol{\theta}_i + \lambda_h \sum_{a \in r(A)} \boldsymbol{\theta}_a^T \boldsymbol{\Omega}_{N_a} \boldsymbol{\theta}_a, \qquad (14)\end{aligned}$$

where $\boldsymbol{y}_a = (y_{a,1}, \ldots, y_{a,N_a})^T$, $\boldsymbol{g}_a = (g_{i_{a,1}}(u_{a,1}), \ldots, g_{i_{a,N_a}}(u_{a,N_a}))^T$, and $(\boldsymbol{N}_a)_{l,j} = N_j^{(a)}(v_{a,l})$. As before, differentiating with respect to the coefficient vector $\boldsymbol{\theta}_{a_0}$ of one component $\boldsymbol{h}_{a_0}$ ($a_0 \in r(A)$), and setting the partial derivative to zero, we obtain

$$\hat{\boldsymbol{\theta}}_{a_0} = (\boldsymbol{N}_{a_0}^T \boldsymbol{N}_{a_0} + \lambda_h \boldsymbol{\Omega}_{N_{a_0}})^{-1} \boldsymbol{N}_{a_0}^T (\boldsymbol{y}_{a_0} - \boldsymbol{\alpha} - \hat{\boldsymbol{g}}_{a_0}). \qquad (15)$$

Repeating the operations above for all $i_0 \in r(I)$ and $a_0 \in r(A)$, we obtain a set of estimating equations:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_i &= (\boldsymbol{N}_i^T \boldsymbol{N}_i + \lambda_g \boldsymbol{\Omega}_{N_i})^{-1} \boldsymbol{N}_i^T (\boldsymbol{y}_i - \boldsymbol{\alpha} - \hat{\boldsymbol{h}}_i) \\ &\quad \textit{for all } i \in r(I) \qquad (16) \\ \hat{\boldsymbol{\theta}}_a &= (\boldsymbol{N}_a^T \boldsymbol{N}_a + \lambda_h \boldsymbol{\Omega}_{N_a})^{-1} \boldsymbol{N}_a^T (\boldsymbol{y}_a - \boldsymbol{\alpha} - \hat{\boldsymbol{g}}_a) \\ &\quad \textit{for all } a \in r(A). \qquad (17)\end{aligned}$$

We note that l-th row of $\hat{\boldsymbol{h}}_i$ in (16) is a linear combination of the elements of $\boldsymbol{\theta}_{a_{i,l}}$ in the form of (10), and that l-th row of $\hat{\boldsymbol{g}}_a$ in (17) is a linear combination of the elements of $\boldsymbol{\theta}_{i_{a,l}}$ in the form of (7). Therefore (16) and (17) together consists a set of $\mathcal{K}$ equations with $\mathcal{K}$ unknowns, where

$$\mathcal{K} = \sum_{i \in r(I)} K_i + \sum_{a \in r(A)} K_a.$$

Multiplying both sides of (16) and (17) by $\boldsymbol{N}_i$ and $\boldsymbol{N}_a$, respectively, from the left, we have

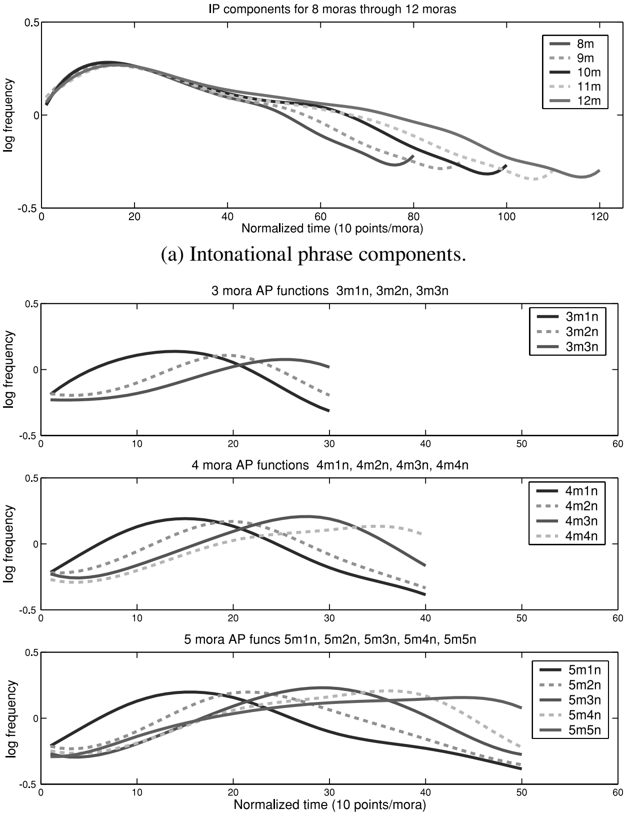
$$\begin{aligned} \hat{\boldsymbol{g}}_i &= \boldsymbol{N}_i \hat{\boldsymbol{\theta}}_i \\ &= \boldsymbol{N}_i (\boldsymbol{N}_i^T \boldsymbol{N}_i + \lambda_g \boldsymbol{\Omega}_{N_i})^{-1} \boldsymbol{N}_i^T (\boldsymbol{y}_i - \boldsymbol{\alpha} - \hat{\boldsymbol{h}}_i) \\ &= \boldsymbol{S}_i (\boldsymbol{y}_i - \boldsymbol{\alpha} - \hat{\boldsymbol{h}}_i) \qquad \textit{for all } i \in r(I) \qquad (18) \\ \hat{\boldsymbol{h}}_a &= \boldsymbol{N}_a \hat{\boldsymbol{\theta}}_a \\ &= \boldsymbol{N}_a (\boldsymbol{N}_a^T \boldsymbol{N}_a + \lambda_h \boldsymbol{\Omega}_{N_a})^{-1} \boldsymbol{N}_a^T (\boldsymbol{y}_a - \boldsymbol{\alpha} - \hat{\boldsymbol{g}}_a) \\ &= \boldsymbol{S}_a (\boldsymbol{y}_a - \boldsymbol{\alpha} - \hat{\boldsymbol{g}}_a) \qquad \textit{for all } a \in r(A). \qquad (19)\end{aligned}$$

Each of $\boldsymbol{S}_i = \boldsymbol{N}_i (\boldsymbol{N}_i^T \boldsymbol{N}_i + \lambda_g \boldsymbol{\Omega}_{N_i})^{-1} \boldsymbol{N}_i^T$ and $\boldsymbol{S}_a = \boldsymbol{N}_a (\boldsymbol{N}_a^T \boldsymbol{N}_a + \lambda_h \boldsymbol{\Omega}_{N_a})^{-1} \boldsymbol{N}_a^T$ in (18) and (19) is called a *smoother matrix* for a cubic smoothing spline. We can obtain the solutions $\hat{\boldsymbol{g}}_i$ for all $i \in r(I)$ and $\hat{\boldsymbol{h}}_a$ for all $a \in r(A)$ using the *backfitting* algorithm, an iterative method depicted in Fig. 2, in which these smoother matrices are applied as smoothing operators in turn until convergence. This backfitting algorithm is equivalent to a block-wise *Gauss-Seidel* method [13] to solve the linear system of (16) and (17).

In our current implementation, we have adopted the arguments in [7] and have used more computationally manageable ($K + 4$) B-spline basis functions, replacing K basis functions of natural cubic spline (Appendix A). It is also suggested that if the number of knots is very large, it is not necessary to use all the knots and some *thinning* strategy will save in computations with negligible effect on the fit [7]. In our current implementation, therefore, we just adopt one every ten time points for the use as knots.

3. Experiments and Results

We have recently been developing a speech synthesizer for Japanese based on our finite-state transducer-based framework [14], [15], and have created a preliminary version for a weather forecast domain [16]. We have therefore attempted to evaluate the use of our F_0 modeling technique for Japanese as well. In our current implementation, we made



(a) Intonational phrase components.

(b) Accentual phrase components.

Fig. 3 Examples of intonational phrase components and accentual phrase components estimated with the proposed method. (a) Intonational phrase components with the length of 8 through 12 moras. (b) 3-, 4- and 5-mora accentual phrase components with all distinct accent nucleus positions.

an simplifying assumption that an intonational phrase component of F_0 is identified by its mora length. The predictor variable I represents the number of moras (or morae) in the intonational phrase. An accentual phrase component is assumed to be identified by the number of moras in it and the position of the nucleus of accent (often called *accent type*). Therefore, the variable A represents a pair (m, n), where m is the number of moras in the accentual phrase and n means that the nucleus is associated with the n-th mora.

We have implemented the algorithm mentioned above in Matlab [17], and estimated component functions g_i's and h_a's in the log frequency domain using a corpus of Japanese utterances read by a female speaker. λ_g and λ_h are both set to be 1.0.

The corpus consisted mostly (around 90%) of general sentences taken from news, novels and other types of general texts. The rest consists of weather (approx. 7%) and stock market report (approx. 3%). The speaker was instructed to read in a fairly neutral manner (but not so neutral as to be completely unexpressive), in other words to give a delivery typical of that of a newscaster.

In the transcription of this corpus, the intonational phrase boundaries were defined simply in terms of pauses the speaker made, and assigned in the labels by human transcriptionists, who checked and marked the phrase boundaries for all of the recorded material. It was attempted to check carefully the location of pauses during recordings so that they did not occur in unnatural spots, although the speaker was not given specific instructions about where to pause in the text and where not to.

The corpus comprised 7,282 utterances, which in turn consist of 16,181 intonational phrases and 44,717 accentual phrases. The number of distinct types of intonational phrases (or distinct mora lengths) was 49, and there were 130 unique accentual phrase types. Utterances in the corpus are annotated with accentual phrase and intonational phrase boundary information as well as phone labels. F_0 values were extracted from the corpus every 10 ms using the Snack Sound Toolkit, a public domain toolkit developed at KTH [18], [19]. These F_0 data were used as is, and no particular postprocessing such as elimination of "microprosody" was performed. The mean and the standard deviation of the corpus F_0 were 207 Hz and 48.2 Hz, respectively.

Before the estimation, the original pitch samples were normalized to have the same number of samples per mora by uniformly interpolating or decimating each accentual phrase. The data instances for which no pitch was extracted for more than half of the mora interval at the beginning or end of all the instances of an accentual phrase type were discarded before estimation, which resulted in accentual phrase types available for training reduced to 116, i.e. 89% of the number of distinct types before discarding. As a side effect, the number of unique intonational phrase types was reduced to 46, which is 94% of the number before discarding.

The backfitting iteration (Fig. 2, (2)) converged well when sixth loop was over. As a result, estimates for 46 distinct intonational phrases, and 116 types of accentual phrases were obtained.

Figure 3 shows examples of extracted intonational and accentual phrase components. Figure 4 illustrates an example of the estimated F_0 contour plotted with the actual F_0 data in the training corpus. As an objective evaluation, we measured the goodness of fit in terms of root mean square error (RMSE) and correlation coefficient (Corr) in the voiced portions of the data, which are often used in the evaluation of F_0 modeling [5], [20]. On the training data, RMSE was 28.9 Hz, and the Corr was 0.806. Measured on 85 intonational phrases set aside from the training data, RMSE and Corr were 29.8 Hz, and 0.777, respectively (Table 1). Measured in the log frequency domain to the base 2, the RMSE for training and test set were 0.195 (octave) and 0.203 (octave), respectively.

Although it can be difficult to compare performance across different speech corpora and languages, we believe these results are quite promising. For example, state-of-the-art results of 33–34 Hz RMSE, and 0.6–0.72 Corr have been reported on a female-speaker English radio news corpus [5], [20] with the standard deviation reported as e.g. 53 Hz in [20].

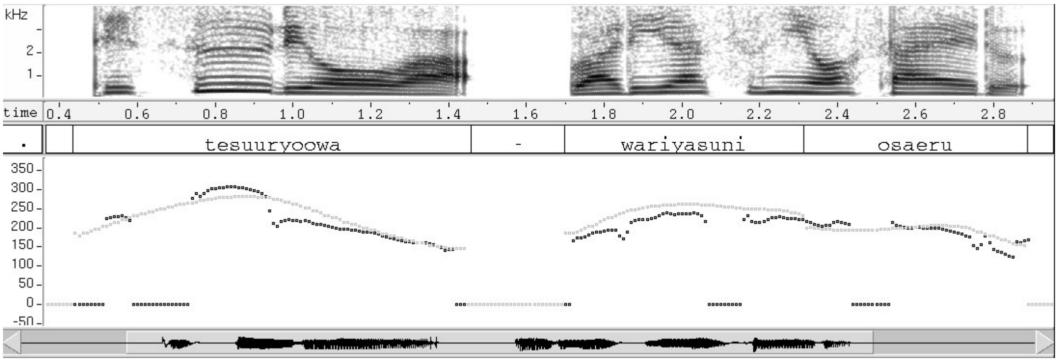


Fig. 4 F_0 contour from the trained model, displayed with the actual F_0 contour. The dark dots are the F_0 data in the training corpus, and light dots are the F_0 contour derived from the additive model trained on the entire training corpus.

Table 1 Experimental results for the additive F_0 model. "RMSE" stands for the root mean square error and "Corr" for the correlation coefficient.

	RMSE	Corr
training	28.9	0.806
test	29.8	0.777

4. Conclusions and Future Work

In this paper, we have proposed a novel two-layer approach to F_0 modeling that uses a statistical learning technique for nonparametric regression called *Additive Models*. We confirmed by experiment that intonational and accentual phrase components that shows a quite regular patterns can be successfully estimated from a large Japanese speech corpus with the proposed method.

The fundamental frequency predicted by the model can be used as the reference for deriving a substitution (target) cost for unit selection in a corpus-based speech synthesizer. It may also be used in part of a post-processor to modify the waveform units to have pitch contour closer to the target.

Although current paper has only examined a two-layer modeling with the proposed additive framework, there is no theoretical limitation to the number of layers. It is expected that we may be able to add more layers as far as additivity of the component effects holds and those components are linearly independent from each other.

We plan to incorporate the F_0 measures predicted by the model, as one of the target measures to evaluate the goodness of the corpus units, into our next generation speech synthesis system we are currently developing. We also plan to apply this framework to F_0 modeling for English speech synthesis.

Acknowledgments

The author would like to thank Jim Glass for the insightful comments, and Tony Ezzat for helpful comments on implementing smoothing spline. The author is grateful to Bill Ham, Michael Phillips, Dan Faulkner, and Yun-Sun Kang at Scansoft for providing the annotated Japanese speech corpus as well as helpful information. The author is also grateful to the two anonymous reviewers whose helpful comments led to the improvements in the manuscript.

This research was supported in part by an industrial consortium supporting the MIT Oxygen Alliance.

References

[1] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," Proc. ICASSP'96, pp.373–376, Atlanta, GA, May 1996.

[2] M. Chu, H. Peng, Y. Zhao, Z. Niu, and E. Chang, "Microsoft Mulan – A bilingual TTS system," Proc. ICASSP 2003, pp.I-264–I-267.

[3] E. Eide, A. Aaron, R. Bakis, P. Cohen, R. Donovan, W. Hamza, T. Mathes, M. Picheny, M. Polkosky, M. Smith, and M. Viswanathan, "Recent improvements to the IBM trainable speech synthesis system," Proc. ICASSP-03, pp.I-708–I-711, 2003.

[4] R.E. Donovan, A. Ittycheriah, M. Franz, B. Ramabhadran, E. Eide, M. Viswanathan, R. Bakis, W. Hamza, M. Picheny, P. Gleason, T. Rutherfoord, P. Cox, D. Green, E. Janke, S. Revelin, C. Waast, B. Zeller, C. Guenther, and J. Kunzmann, "Current status of the IBM trainable speech synthesis system," Proc. 4th ESCA Tutorial and Research Workshop on Speech Synthesis, Perthshire, Scotland, Sept. 2001.

[5] X. Sun, "F0 generation for speech synthesis using a multi-tier approach," Proc. ICSLP 2002, pp.2077–2080, Denver, CO, 2002.

[6] M. Chu, H. Peng, H. Yang, and E. Chang, "Non-uniform units from a very large corpus for concatenative speech synthesizer," Proc. ICASSP 2001, Salt Lake City, UT, May 2001.

[7] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer, 2001.

[8] T. Hastie and R. Tibshirani, Generalized Additive Models, Chapman and Hall, 1990.

[9] R.L. Vislocky and J.M. Fritsch, "Generalized additive models versus linear regression in generating probabilistic mos forecasts of aviation weather parameters," Weather and Forecasting, vol.10, no.4, pp.669–680, 1995.

[10] F. Dominici, A. McDermott, S. Zeger, and J. Samet, "On the use of generalized additive models in time-series studies of air pollution and health," Am. J. Epidemiol., vol.56, no.3, pp.193–203, 2002.

[11] H. Fujisaki and K. Hirose, "Analysis of voice fundamental frequency contours for declarative sentences of Japanese," J. Acoust. Soc. Jpn. (E), vol.5, no.4, pp.233–241, 1984.

[12] M. Abe and H. Sato, "Two-stage F0 control model using syllable based F0 units," Proc. ICASSP'92, pp.53–56, San Francisco, 1992.

[13] G. Strang, Introduction to Linear Algebra, 3rd ed., Wellesley Cambridge Press, 2003.

[14] J. Yi, J. Glass, and L. Hetherington, "A flexible, scalable finite-state transducer architecture for corpus-based concatenative speech synthesis," Proc. Intl. Conf. on Spoken Language Processing, pp.322–325, Beijing, Oct. 2000.

[15] J. Yi and J. Glass, "Information-theoretic criteria for unit selection synthesis," Proc. ICSLP 2002, pp.2617–2620, Denver, Sept. 2002.
[16] M. Nakano, T. Minami, S. Seneff, T.J. Hazen, D.S. Cyphers, J. Glass, J. Polifroni, and V. Zue, "Mokusei: A telephone-based Japanese conversational system in the weather domain," Proc. European Conf. on Speech Communication and Technology, Aalborg, Denmark, Sept. 2001.
[17] http://www.mathworks.com.
[18] http://www.speech.kth.se/snack/.
[19] K. Sjölander and J. Beskow, "Wavesurfer — An open source speech tool," Proc. Intl. Conf. on Spoken Language Processing, pp.464–467, Beijing, Oct. 2000.
[20] K.E. Dusterhoff, A.W. Black, and P. Taylor, "Using decision trees within the tilt intonation model to predict F0 contours," Proc. European Conf. on Speech Communication and Technology, 1999.
[21] P. Green and B. Silverman, Nonparametric Regression and Generalized Linear Models, Chapman and Hall, 1994.

Appendix A: Definition of Natural Cubic Spline [7], [21]

Suppose we have real numbers $\xi_1, \ldots, \xi_K$ on some interval $[a, b]$, satisfying $a < \xi_1 < \xi_2 < \ldots < \xi_K < b$. A function g defined on $[a, b]$ is a *cubic spline* if two conditions are satisfied. First, on each of the intervals $(a, \xi_1), (\xi_1, \xi_2), \ldots, (\xi_K, b)$, g is a cubic polynomial. Second, the polynomial pieces fit together at the points ξ_i in such a way that g itself and its first and second derivatives are continuous at each ξ_i, hence on the whole of $[a, b]$. The points ξ_i are called *knots*.

A *natural cubic spline* has additional constraints, namely that its second and third derivatives are zero at a and b. These constraints incurs that the function is linear beyond the two boundary knots, ξ_1 and ξ_K. It is known that a natural cubic spline with K knots can be represented in the form of a linear combination of K basis functions

$$g(x) = \sum_{j=1}^{K} \theta_j N_j(x), \tag{A$\cdot$1}$$

where each of the basis functions N_j is a some polynomial with an order up to three. See, for example, [7] (pp.120–122) for the detailed discussion.

Appendix B: A Property of Natural Cubic Spline Interpolant (Green and Silverman [21])

Suppose that $N \geq 2$ and that g is the natural cubic spline interpolant to the pairs (x_i, z_i) $(i = 1, \ldots, N)$ with $a < x_1 < \ldots < x_N < b$. This is a natural cubic spline with knots at x_i $(i = 1, \ldots, N)$. Let $\tilde{g}$ be any other twice continuously differentiable function on $[a, b]$ that also interpolates the N pairs, i.e. $\tilde{g}(x_i) = z_i \; for \; i = 1, \ldots, N$. Then, it holds that

$$\int_a^b \tilde{g}''(x)^2 dx \geq \int_a^b g''(x)^2 dx,$$

with equality only if $\tilde{g}$ and g are identical.

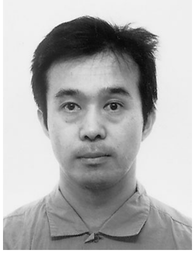
Proof Let $h(x) = \tilde{g}(x) - g(x)$. Since $\tilde{g}$ and g both interpolates the pairs (x_i, y_i), h is zero at all x_i $(i = 1, \ldots, N)$. Using the boundary conditions that g'' is zero at a and b, integration by parts yields

$$\begin{aligned}
&\int_a^b g''(x)h''(x)dx \\
&\quad = g''(b)h'(b) - g''(a)h'(a) - \int_a^b g'''(x)h'(x)dx \\
&\quad = -\int_a^{x_1} g'''(x)h'(x)dx - \sum_{j=1}^{N-1} \int_{x_j}^{x_{j+1}} g'''(x)h'(x)dx \\
&\qquad - \int_{x_N}^{b} g'''(x)h'(x)dx \\
&\quad = -\sum_{j=1}^{N-1} g'''(x_j^+) \int_{x_j}^{x_{j+1}} h'(x)dx \\
&\quad = -\sum_{j=1}^{N-1} g'''(x_j^+)\{h(x_{j+1}) - h(x_j)\} = 0.
\end{aligned} \tag{A$\cdot$2}$$

We have used the fact that g''' is zero on the intervals (a, x_1) and (x_N, b) and is constant on each of the intervals (x_j, x_{j+1}) with the value $g'''(x_j^+)$. Using (A$\cdot$2), it follows that

$$\begin{aligned}
&\int_a^b \tilde{g}''(x)^2 dx \\
&\quad = \int_a^b \{g''(t) + h''(t)\}^2 dx \\
&\quad = \int_a^b g''(x)^2 dx + 2\int_a^b g''(x)h''(x)dx + \int_a^b h''(x)^2 dx \\
&\quad = \int_a^b g''(x)^2 dx + \int_a^b h''(x)^2 dx \\
&\quad \geq \int_a^b g''(x)^2 dx,
\end{aligned} \tag{A$\cdot$3}$$

and equality will hold only if $\int_a^b h''(x)^2 dx$ is zero, so that h is linear on $[a, b]$. But since h is zero at $x_1, \ldots, x_N$, and since $N \geq 2$, this can only happen if h is identically zero, which means that g and $\tilde{g}$ are the same function. □



Shinsuke Sakai received his B.E. and M.E. degrees from Kyoto University in 1982 and 1984, respectively. He joined NEC Corp. in 1984, where he worked on various aspects of speech and natural language processing for machine translation and speech recognition. Between 1991–1993, he was a visiting scientist at MIT. In 2002, he moved to the Laboratory for Computer Science at MIT, and is currently a Research Engineer at Computer Science and Artificial Intelligence Laboratory. His research interests include exploring speech technology that can be used in various daily life situations, such as in the office, living room and on the street. He is a member of the Information Processing Society of Japan and the Acoustical Society of Japan.