# FUNDAMENTAL FREQUENCY MODELING FOR CORPUS-BASED SPEECH SYNTHESIS BASED ON A STATISTICAL LEARNING TECHNIQUE

*Shinsuke Sakai and James Glass*

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{sakai,glass}@mit.edu

## ABSTRACT

This paper proposes a novel two-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 0.81 and 0.77. The automatically determined intonational and accentual phrase components behave smoothly, systematically, and intuitively under a variety of prosodic conditions.

## 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, 2]. 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, dynamic range) 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 [3], while regression tree-based approaches are often used to predict $F_0$-related measures from a set of linguistic features [4, 5]. A regression tree approach is advantageous in that it is simple to implement yet 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.

In this work, we propose a simple yet novel two-layer *additive model* [6, 7] 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 term. In the following section we define the additive $F_0$ model, along with the penalized least-squares criterion from which a backfitting algorithm is derived as the minimizer of the criterion. We then describe experimental results applying the proposed method to a large corpus of Japanese speech.

## 2. ADDITIVE MODEL APPROACH

The basic formulation for the $F_0$ contour is similar to previous work, e.g., [8, 9]. In this approach, the $F_0$ contour, $Y$, is the output of a statistical model that combines a long-range intonational-phrase level component, $g$, and a shorter accentual-phrase level component, $h$:

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

where $\alpha$ is a constant, $I$ is a discrete-valued (i.e., symbolic) 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, $\epsilon$, is zero mean. Figure 1 shows how the three terms form the entire $F_0$ contour function.

This research was supported in part by NTT.

In *Proceedings of IEEE ASRU 2003*, Nov. 30-Dec. 4, 2003, St. Thomas, U.S. Virgin Islands, pp.712–717 

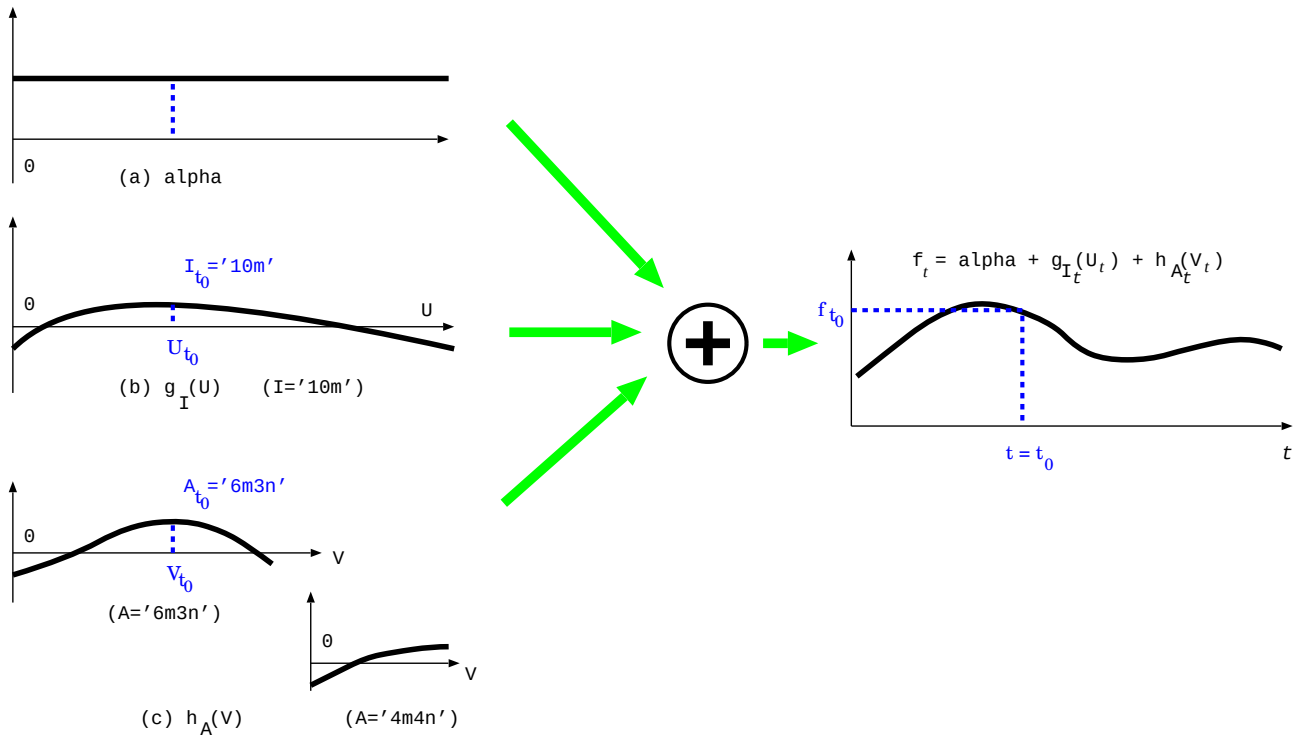



**Fig. 1**. A schematic diagram of the additive $F_0$ model $f(I, U, A, V) = \alpha + g_I(U) + h_A(V)$. A constant $\alpha$ and component functions $g$ and $h$ are summed up to form the $F_0$ contour $f$.

A unique characteristic of our approach, as compared to previous work, is that we do not 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 [6, 7]. We define the penalized residual sum-of-squares (PRSS) error in the following form:

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

where $(i_n, u_n, a_n, v_n, y_n)$ $(n = 1, ..., N)$ are a set of training data corresponding to the variables $(I, U, A, V, Y)$, and $\lambda_g$, $\lambda_h$ are fixed smoothing parameters. $r(I)$ represents the set of possible values (or *range*) for I, and $r(A)$ is defined in the same way. The number of elements in a set, for example $r(I)$, will be denoted as $|r(I)|$, hereafter. The first term measures the closeness to the data, while the second and third terms penalize the curvatures in the functions, and smoothing parameters $\lambda_g$ and $\lambda_h$ establish a tradeoff between them. Large values of $\lambda$'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 $\alpha$ will be the overall mean of $y_n$ $(n = 1, ..., N)$. We can find the solution for (2) with a *backfitting* [6] algorithm, a simple iterative procedure depicted in Figure 2.

In the algorithm, we apply a natural cubic-spline smoother, e.g., $\mathcal{S}_i$, to the 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 $\hat{g}_i$. Partial residual smoothing is done, for $g$'s and $h$'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 blockwise Gauss-Seidel algorithm for solving a system of linear equations emerging from the minimization of the penalized least-square criterion (2).

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 in-

(1) Initialize: $\hat{\alpha} = \frac{1}{N}\sum_{n=1}^{N} y_n, \quad \hat{g}_i \equiv 0, \hat{h}_a \equiv 0 \; for \; all \; i \in r(I), a \in r(A)$

(2) Cycle: repeat (2g) and (2h) until the functions $\hat{g}_I$ and $\hat{h}_A$ change less than a prespecified threshold.

(2g) Partition the set of training data $\{(i_n, u_n, a_n, v_n, y_n) \mid n = 1, ..., N\}$, into $|r(I)|$ subsets $\{(i, u_{i,l}, a_{i,l}, v_{i,l}, y_{i,l}) \mid l = 1, ..., N_i\}$ $(i \in r(I))$, 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 $i \in r(I)$,

$$\hat{g}_i \leftarrow \mathcal{S}_i[\{y_{i,l} - \hat{\alpha} - \hat{h}_{a_{i,l}}(v_{i,l})\}_{l=1}^{N_i}].$$

(2h) Repartition the training data $\{(i_n, u_n, a_n, v_n, y_n) \mid n = 1, ..., N\}$ into $|r(A)|$ subsets $\{(i_{a,l}, u_{a,l}, a, v_{a,l}, y_{a,l}) \mid l = 1, ..., N_a\}$ $(a \in r(A))$, 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 $a \in r(A)$,

$$\hat{h}_a \leftarrow \mathcal{S}_a[\{y_{a,l} - \hat{\alpha} - \hat{g}_{i_{a,l}}(u_{a,l})\}_{l=1}^{N_a}].$$

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

tonational phrase. We can then express the entire training data, $\mathcal{D}$, as a union of $|r(I)|$ subsets:

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

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

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

where $\sum_{a \in r(A)} N_a = N$. By using (3) and (4), the expression for PRSS can then be rewritten in two ways:

$$\begin{aligned}&PRSS(\alpha, g, h) \\ &= \sum_{i \in r(I)} \sum_{l=1}^{N_i} \{y_{i,l} - \alpha - g_i(u_{i,l}) - h_{a_{i,l}}(v_{i,l})\}^2 + \\ &\quad \lambda_g \sum_{s \in r(I)} \int g_s''(w)^2 dw + \lambda_h \sum_{t \in r(A)} \int h_t''(x)^2 dx \quad (5) \\ &= \sum_{a \in r(A)} \sum_{l=1}^{N_a} \{y_{a,l} - \alpha - g_{i_{a,l}}(u_{a,l}) - h_a(v_{a,l})\}^2 + \\ &\quad \lambda_g \sum_{s \in r(I)} \int g_s''(w)^2 dw + \lambda_h \sum_{t \in r(A)} \int h_t''(x)^2 dx. \quad (6)\end{aligned}$$

Now, we can consider searching for the optimal function $\hat{g}_{i_0}$ that minimizes the penalized least square criterion (5) for a certain value 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 $(u_{i_0,l}, g(u_{i_0,l}))$ $(l = 1, ..., N_{i_0})$. Let $\bar{g}$ be the natural cubic spline that interpolates the points $(u_{i_0,l}, g(u_{i_0,l}))$. Since $\bar{g}(u_{i_0,l}) = g(u_{i_0,l})$ by definition, it immediately follows that

$$\sum_{l=1}^{N_i} \{y_{i,l} - \alpha - \bar{g}(u_{i,l}) - h_{a_{i,l}}(v_{i,l})\}^2 = \sum_{l=1}^{N_i} \{y_{i,l} - \alpha - g(u_{i,l}) - h_{a_{i,l}}(v_{i,l})\}^2.$$

Due to the optimality property of the natural cubic spline interpolant (cf. Appendix B), $\int \bar{g}''(t)^2 dt < \int g''(t)^2 dt$. 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 $u_{i_0,l}$ $(l = 1, ..., N_{i_0})$. Extending the discussion above to all $g_i$'s and the other form of PRSS in (6) and all $h_a$'s, 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 natural cubic spline basis functions $N_j^{(i)}$ (cf. Appendix A):

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

Then the vector of the values of $g_i$ at the training data points $u_{i,l}$ $(l = 1, ..., N_i)$ can be written as

$$\boldsymbol{g}_i = \boldsymbol{N}_i \boldsymbol{\theta}_i \quad (8)$$

where $\boldsymbol{\theta}_i = (\theta_1^{(i)}, ..., \theta_{N_i}^{(i)})^T$ and $(\boldsymbol{N}_i)_{l,j} = N_j^{(i)}(u_{i,l})$. Then, by defining a matrix $\boldsymbol{\Omega}_{N_i}$ as $(\boldsymbol{\Omega}_{N_i})_{j,k} = \int N_j^{(i)''}(x)\ N_k^{(i)''}(x)\ dx$, we can write each component roughness penalty for $g_i$ as:

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

where $\boldsymbol{K}_i = (\boldsymbol{N}_i^{-1})^T \boldsymbol{\Omega}_{N_i} \boldsymbol{N}_i^{-1}$. We can derive the same form of component roughness penalty for $h_a$ in the same way. From (5), PRSS can now be written in a matrix form:

$$PRSS = \sum_{i \in r(I)} (\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{g}_i - \boldsymbol{h}_i)^T (\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{g}_i - \boldsymbol{h}_i) + \lambda_g \sum_{i \in r(I)} \boldsymbol{g}_i^T \boldsymbol{K}_i \boldsymbol{g}_i + \lambda_h \sum_{a \in r(A)} \boldsymbol{h}_a^T \boldsymbol{K}_a \boldsymbol{h}_a, \quad (10)$$

where $\boldsymbol{y}_i = (y_{i,1},\ ...,\ y_{i,N_i})^T$, $\boldsymbol{\alpha} = (\alpha,\ ...,\ \alpha)^T$, $\boldsymbol{h}_i = (h_{a_{i,1}}(v_{i,1}), ..., h_{a_{i,N_i}}(v_{i,N_i}))^T$. $\boldsymbol{h}_a$'s and $\boldsymbol{K}_a$'s are defined with regard to $h_a$'s in the same manner as $\boldsymbol{g}_i$ and $\boldsymbol{K}_i$ in (8), (9). By differentiating (10) with respect to one component $\boldsymbol{g}_{i_0}$ $(i_0 \in r(I))$, and setting the partial derivative to zero, we obtain:

$$\hat{\boldsymbol{g}}_{i_0} = (I + \lambda_g \boldsymbol{K}_{i_0})^{-1} (\boldsymbol{y}_{i_0} - \boldsymbol{\alpha} - \hat{\boldsymbol{h}}_{i_0}). \quad (11)$$

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

$$PRSS = \sum_{a \in r(A)} (\boldsymbol{y}_a - \boldsymbol{\alpha} - \boldsymbol{g}_a - \boldsymbol{h}_a)^T (\boldsymbol{y}_a - \boldsymbol{\alpha} - \boldsymbol{g}_a - \boldsymbol{h}_a) + \lambda_g \sum_{i \in r(I)} \boldsymbol{g}_i^T \boldsymbol{K}_i \boldsymbol{g}_i + \lambda_h \sum_{a \in r(A)} \boldsymbol{h}_a^T \boldsymbol{K}_a \boldsymbol{h}_a, \quad (12)$$

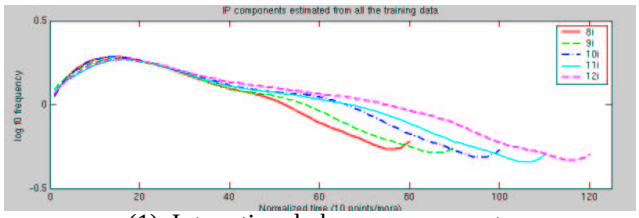
where $\boldsymbol{y}_a = (y_{a,1},\ ...,\ y_{a,N_a})^T$, $\boldsymbol{g}_a = (g_{i_{a,1}}(u_{a,1}),\ ...,\ g_{i_{a,N_a}}(u_{a,N_a}))^T$, and $\boldsymbol{h}_a = (h_a(v_{a,1}),\ ...,\ h_a(v_{a,N_a}))^T$. As before, differentiating with respect to one component $\boldsymbol{h}_{a_0}$ $(a_0 \in r(A))$, we obtain

$$\hat{\boldsymbol{h}}_{a_0} = (I + \lambda_h \boldsymbol{K}_{a_0})^{-1} (\boldsymbol{y}_{a_0} - \boldsymbol{\alpha} - \hat{\boldsymbol{g}}_{a_0}). \quad (13)$$

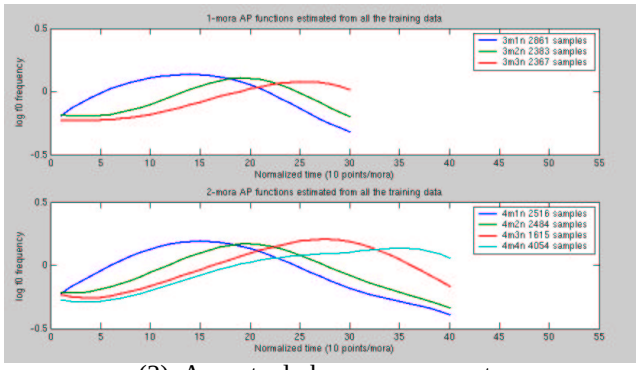
Repeating the above discussion for all $i_0 \in r(I)$ and $a_0 \in r(A)$, we obtain a set of estimating equations:

$$\begin{aligned}\hat{\boldsymbol{g}}_i &= (I + \lambda_g \boldsymbol{K}_i)^{-1} (\boldsymbol{y}_i - \boldsymbol{\alpha} - \hat{\boldsymbol{h}}_i) \quad for\ all\ i \in r(I) \\ \hat{\boldsymbol{h}}_a &= (I + \lambda_h \boldsymbol{K}_a)^{-1} (\boldsymbol{y}_a - \boldsymbol{\alpha} - \hat{\boldsymbol{g}}_a) \quad for\ all\ a \in r(A), \end{aligned} \quad (14)$$

which is a system of $2 \times N$ (i.e. the sum of the length of all $\hat{\boldsymbol{g}}_i$'s and $\hat{\boldsymbol{h}}_a$'s) linear equations with $2 \times N$ unknowns.




(1) Intonational phrase components




(2) Accentual phrase components

**Fig. 3**. Examples of intonational phrase components and accentual phrase components estimated with the proposed method. (1) Intonational phrase components with the length of 8 through 12 moras. (2) 3- and 4-mora accentual phrase components with all distinct accent nucleus positions.

Here, each of $\boldsymbol{S}_i = (I + \lambda_g \boldsymbol{K}_i)^{-1}$, and $\boldsymbol{S}_a = (I + \lambda_h \boldsymbol{K}_a)^{-1}$ in (14) is a smoother matrix for a cubic smoothing spline, and is used as a smoothing operator in the backfitting algorithm of Figure 2. We can solve this $2N \times 2N$ system using a block-wise Gauss-Seidel procedure [10], which is the *backfitting* algorithm depicted in Figure 2.

In our current implementation, we have adopted the arguments in [7] and have used more computationally manageable $(N+4)$ B-spline basis functions, replacing $N$ basis functions mentioned in Appendix A.

## 3. EXPERIMENTS AND RESULTS

We have recently been developing a speech synthesizer for Japanese based on our finite-state transducer-based framework [11, 12], and have created a preliminary version for a weather forecast domain [13]. We have evaluated the use of our $F_0$ modelling technique for Japanese as well. In our initial implementation, we made an assumption that an intonational phrase component of $F_0$ is identified by its mora[1] length. The predictor variable, $I$, represents the number of

[1] A *mora* is a temporal unit that typically corresponds to one hiragana (phonetic alphabet) character. It consists of either one vowel(V), a consonant followed by a vowel(C+V), C+/y/+V, N(moraic nasal), or Q(roughly described as a moraic pause).

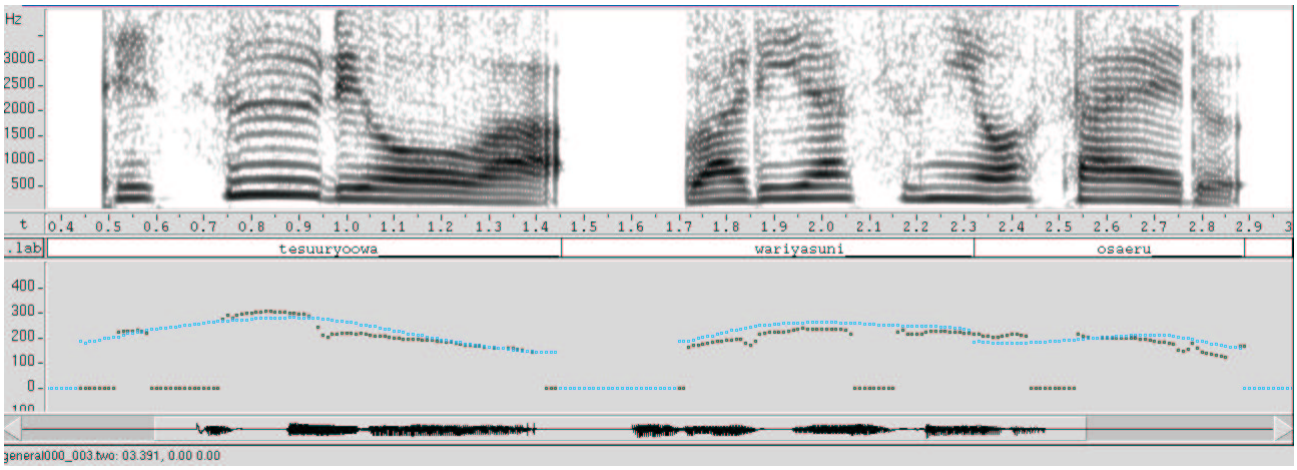



**Fig. 4**. $F_0$ contour from the trained model, displayed with the actual $F_0$ contour. The dark dots are the $F_0$ data in the training corpus, and light dots are the $F_0$ contour derived from the additive model trained on the entire training corpus.

moras 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, 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. 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. 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. The backfitting iteration (Figure 2, (2)) converged after six loops. 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 a preliminary evaluation, we measured the goodness of fit in terms of root mean square error (RMSE) and correlation coefficient (CORR), which are often used in the evaluation of $F_0$ modeling [14, 4]. On the training data, RMSE was 28.9 Hz, and the CORR was 0.81. Measured on 85 intonational phrases set aside from the training data, RMSE and CORR were 29.8 Hz, and 0.77, 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 [14, 4]. We are currently investigating comparative evaluation with different approaches such as regression tree-based methods.

## 4. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a novel two-layer approach to $F_0$ modeling, and have estimated intonational and accentual phrase components from a 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 as part of a post-processor to modify the waveform units to have pitch contour closer to the target. We plan to incorporate the $F_0$ measures predicted by the model, as one of the target measures to derive the costs, into our speech synthesis system. We also plan to apply this framework for $F_0$ modeling of English, for more general purpose concatenative speech synthesis.

## 5. ACKNOWLEDGEMENT

The authors are grateful to Michael Phillips, Dan Faulkner, Bill Ham, and Yun-Sun Kan at SpeechWorks International for providing the annotated Japanese speech corpus as well as helpful information.

## 6. REFERENCES


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

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

[3] R.E. Donovan et al., "Current status of the IBM trainable speech synthesis system," in *Proc. 4th ESCA Tutorial and Research Workshop on Speech Synthesis*, Scotland, Sept. 2001.

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

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

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

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

[8] H. Fujisaki and K. Hirose, "Analysis of voice fundamental frequency contours for declarative sentences of Japanese," *Journal of the Acoustical Society of Japan(E)*, vol. 5, no. 4, pp. 233–241, 1984.

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

[10] G. Strang, *Introduction to Linear Algebra*, Wellesley Cambridge Press, 1998.

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

[12] J. Yi and J. Glass, "Information-theoretic criteria for unit selection synthesis," in *Proc. ICSLP 2002*, Denver, Sept. 2002, pp. 2617–2620.

[13] M. Nakano, T. Minami, S. Seneff, T. J. Hazen, D. Scott Cyphers, J. Glass, J. Polifroni, and V. Zue, "Mokusei: A telephone-based Japanese conversational system in the weather domain," in *Proc. European Conf. on Speech Communication and Technology*, Aalborg, Denmark, Sept. 2001.

[14] K. E. Dusterhoff, A. W. Black, and P. Taylor, "Using decision trees within the tilt intonation model to predict F0 contours," in *Proc. European Conf. on Speech Communication and Technology*, 1999.

[15] P. Green and B. Silverman, *Nonparametric Regression and Generalized Linear Models*, Chapman and Hall, 1994.


## APPENDIX

### A. DEFINITION OF NATURAL CUBIC SPLINE [6, 15]

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

A *natural cubic spline* has additional constraints, namely that the function is linear beyond the two boundary knots, $\xi_1$ and $\xi_K$. 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

$$g(x) = \sum_{j=1}^{K} \theta_j N_j(x), \tag{15}$$

where each of the basis functions $N_j$ is a some polynomial with an order up to three. See, for example, [6] (pp.120–122) for detail.

### B. A PROPERTY OF NATURAL CUBIC SPLINE INTERPOLANT

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

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

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

***[outline of the proof]*** If we define $h(x) = \tilde{g}(x) - g(x)$ and calculate $\int_a^b g''(x)h''(x)dx$ using integration by part, we confirm that it turns out to be zero. Then it follows that

$$\begin{aligned}\int_a^b \tilde{g}''(x)^2 dx &= \int_a^b \{g''(t) + h''(t)\}^2 dx \\ &= \int_a^b g''(x)^2 dx + \int_a^b h''(x)^2 dx \geq \int_a^b g''(x)^2 dx.\end{aligned}$$

See [15] (pp.16–17) for the detailed demonstration.