Intonational Speaker Verification: A Study on Parameters and Performance Under Noisy Conditions

Sadjad Siddiq[1], *Tomi Kinnunen*[1], *Martti Vainio*[2], *Stefan Werner*[1]

[1]University of Eastern Finland, Joensuu, Finland
[2]University of Helsinki, Helsinki, Finland

ssiddiq@cs.joensuu.fi, tkinnu@cs.joensuu.fi

Abstract

Prosody-based speaker verification using fundamental frequency (f_0) is considered. Our study consists of two phases. First, we do extensive optimization of parameters to establish a baseline system before dealing with noisy conditions. This includes a study of f_0 extractor parameters, choice of features (discrete cosine transform, discrete Fourier transform, Legendre polynomials, linear prediction), f_0 track interpolation (none, linear, Hermite), framing parameters and windowing (none, Hamming), f_0 representation domain (linear, log), number of transformation coefficients and, finally, use of higher-level delta coefficients. Using the optimized parameters, we then explore the robustness of prosody features under white noise and factory noise degradations. Using a GMM-UBM system on the NIST 2006 SRE corpus, we reach an EER of 28.4 % and 27.6 % for the intonational and MFCC features respectively at -20 dB SNR white noise contamination; fusion of the two yields an EER of 24.38 %.

Index Terms: speaker recognition, prosodic features, fundamental frequency

1. Introduction

Speaker verification is the task of deciding whether two utterances were spoken by the same speaker [1]. For a long time, the dominant approach has been based on stochastic Gaussian mixture modeling of spectral features [2, 3]. While the spectrum contains rich information about the speaker's identity, it is subject to environment and channel variations [4]. Since human beings tend to pay attention to prosody [5], many authors have considered prosodic features, most notably the fundamental frequency or f_0, for speaker recognition [6, 7, 8, 9, 10]

In early studies, f_0 contours were used in text-dependent speaker recognition using time registration [11]. In text-independent recognition, in turn, long-term distribution modeling of f_0 is common [6]. But such a model discards the local f_0 contour shape at the word and syllable levels. The use of f_0 contour stylization and tokenization (based largely on intonational phonology research tradition) is commonly used to model the temporal properties of f_0 [8, 9, 10]. In these methods, one segments the f_0 contour into syllable-like segments and represents each segment using either discrete (e.g. rising and falling pitch accents) or continuous features (e.g. max/min values and slopes of stylization segments).

In this paper, we consider a computationally efficient and straightforward modeling of local prosody for speaker recognition. We adopt a few common techniques from spectral feature extraction to modeling of temporal and spectral content of the f_0 track. To this end, we chunk the f_0 track into fixed-length frames which are then transformed into a sequence of feature vectors (Fig. 1) modeled using a standard Gaussian mixture model approach [2]. Our goal is to answer the following design questions:

1. Should the f_0 extractor be configured to produce less (but more reliable) f_0 values or more intonation data (but with possible tracking errors)?
2. Should gaps in the f_0 contour be interpolated?
3. How to choose the frame size and frame rate? Should data be windowed?
4. Which f_0 domain should be used (linear or log)?
5. Which basis function best suits prosody modeling? How many features are needed?
6. Are local dynamic (delta) features useful?
7. How are f_0 features affected by additive noise? At what SNR level do we have a break-down point?
8. How do f_0 features compare to spectral MFCC features?

While some of these questions are independently addressed in literature [9, 10, 12] our goal is to provide conclusive recommendations on these design considerations on a common set of data (chosen to be the telephone quality NIST 2006 SRE corpus). Moreover, due to our recent efforts in recognition under noisy conditions [4], we pay special attention to robustness of f_0 features under additive noise degradation.

2. Computing Intonational Features

2.1. f_0 Tracking and Pre-Processing

Figure 1 summarizes the feature extraction of the f_0 features. For the first step, f_0 tracking, we utilize the autocorrelation based `getf0s` method from the *Snack Sound Toolkit* [13], also distributed in the *WaveSurfer* software. In an early phase of the study, we also considered the autocorrelation method in the

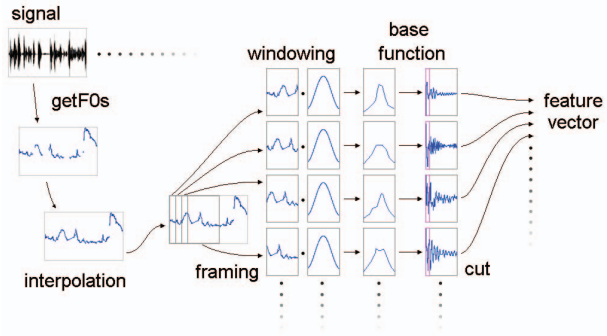

Figure 1: The feature extraction setup

The work of T. Kinnunen was supported by the Academy of Finland (project no. 132129).

popular *Praat* software [14] but ended up using `getf0s`. The two methods yielded generally similar f_0 tracks, but `getf0s` as computationally more feasible was chosen.

We interpolate in short gaps (less than 200 milliseconds) of the f_0 contour caused by unvoiced consonants or short non-speech segments. In addition to standard linear interpolation (e.g. [10]), we were curious to try if higher order polynomial interpolation would be useful; to this end, we also consider *Hermite* interpolation, where values are interpolated with the help of Hermite basis functions. If $y(x)$ is a curve for which we want to calculate values between $x = 0$ and $x = 1$, we can do so using the formula $y_{ip}(x) = \sum_{n=0}^{3} p_n h_n(x)$, where $p_0 = y(0)$; $p_1 = y'(0)$; $p_2 = y'(1)$; $p_3 = y(1)$ and h_n are the Hermite basis functions $h_0 = 2x^3 - 3x^2 + 1$; $h_1 = x^3 - 2x^2 + x$; $h_2 = x^3 - x^2$ and $h_3 = -2x^3 + 3x^2$.

The interpolated f_0 curve is then segmented into overlapping f_0 *frames* of N samples. We also wanted to see if data windowing would have any benefit. For this, we apply a standard Hamming window $w(n) = 0.54 - 0.46\cos(2\pi n/N)$ where $0 \leq n \leq N-1$ indices the samples within an f_0 frame. Windowing in DFT and autocorrelation-based LP for reducing spectral leakage and boundary effects is standard.

2.2. Intonational Feature Extraction

Four techniques are considered for local f_0 contour parametrization. The first technique, **discrete cosine transform (DCT)**, sometimes known as DCT-II, over N-sample frame $x(n)$ is defined as $D(k) = A(k)\sum_{n=0}^{N-1} x(n)\cos[(\pi/N)(n + 1/2)k]$ where $A(k) = 1/\sqrt{N}$ for $k = 0$ and $A(k) = 2/\sqrt{N}$ for $0 \leq k \leq N-1$. The DCT is effective in de-correlating the features and a standard tool in data compression. It has also been used for representing intonational features in both recognition [10] and voice conversion [15] applications.

The second technique, **discrete Fourier transform (DFT)**, is computed using the fast Fourier transform (FFT) and defined as $X(k) = \sum_{n=0}^{N-1} x(n)e^{-2\pi i \frac{nk}{N}}$, where $i \triangleq \sqrt{-1}$ is the imaginary unit and k denotes the discrete frequency index. We are not aware of other works using the DFT for intonation parametrization. In this paper, to mimic typical process for spectral feature extraction, we consider only the log-spectral magnitude $\log_{10}|X(k)|$. Cutting the signal down into segments distorts the phase which is discarded by keeping the magnitude information only, and logarithmic representation helps to balance the magnitudes which would otherwise be dominated by the lowest frequencies only due to the lowpass nature of the f_0 contour.

Another popular technique uses **Legendre polynomials (Leg.)** to represent local intonation [9]. Here, we use MATLAB's built-in function `legendre` to generate fully normalized associated Legendre functions. The Legendre features are then generated by projecting the f_0 frame on these basis functions.

The last technique, **linear predictive cepstral coefficient (LPCC)** features, are based on the well-known linear prediction model [16]. LP is commonly used for modeling short-term spectrum in both recognition and synthesis applications, but we are not aware of it being studied for intonation representation. In LP, one assumes that a signal sample can be predicted as linear combination of p previous samples as $\hat{x}(n) = \sum_{k=1}^{p} a_k x(n-k)$. We fix the predictor order to $p = 16$ in this study. The predictor coefficients $\{a_k\}$ are optimized by minimizing the residual energy $E = \sum_n (x(n) - \sum_{k=1}^{p} a_k x(n-k))^2$ over each analysis frame and then converted into cepstral coefficients using the standard recursive formula (e.g. [17]).

2.3. Further Considerations

Comparing DCT and DFT, DCT can be seen as a contour approximation that captures both the magnitude (range of local f_0) and shape (e.g. locations of peaks and valleys), whereas DFT captures the magnitude only. Note also that it is important to keep in the DC coefficient in both DCT and DFT ($D(0)$ and $|X(0)|$, respectively) as this represents the average f_0 information of the segment, which is known to discriminate speakers (e.g. [18, 6]). Similarly, the LP model is insensitive to signal scaling, that is, the same predictor coefficients are obtained for an f_0 frame multiplied by a constant. To include f_0 scale information, we include the average f_0 of the f_0 frame to the LPCC feature vectors.

Delta and double delta coefficients of spectral features are used in nearly all speech processing front-ends to incorporate local spectral dynamics to the short-term frames. Thus, we were curious to see if they are helpful for intonation modeling as well. We first compute the base coefficients and then append deltas and double deltas calculated from these coefficients. The delta coefficients are computed using $\Delta\mathbf{c}(t) = \mathbf{c}(t+1) - \mathbf{c}(t-1)$ where $\mathbf{c}(t)$ denotes the DCT, DFT, Legendre or LPCC coefficients at the tth f_0 frame. Similarly, double deltas are obtained as $\Delta^2\mathbf{c}(t) = \Delta\mathbf{c}(t+1) - \Delta\mathbf{c}(t-1)$. Careful handling at the voiced/unvoiced boundaries is required. Here, we simply discard those feature vectors whose delta or double delta computation extends over a voiced/unvoiced boundary. This approach of modeling intonational dynamics is *not* the same as appending f_0 with its deltas (e.g. [7]) because the deltas here are computed using the basis function coefficients rather than raw f_0 values. Since f_0 frames already contain information of local f_0 dynamics, the delta features in this study span over longer temporal contexts.

3. Experimental Setup

We have selected the core condition in the NIST 2006 speaker recognition evaluation (SRE) corpus for the experiments[1]. The corpus consists of telephony speech with 816 target speakers (354 males, 462 females), 5077 genuine trials and 48,889 impostor trials that are all gender-matched. For feature modeling and classification, we utilize a standard Gaussian mixture model – universal background model (GMM-UBM) [2]. We use 64 and 512 Gaussians for the f_0 and MFCC features (12 MFCCs + RASTA + Δ/Δ^2 + CMVN), respectively. Gender-dependent UBMs are trained using the NIST SRE 2004 corpus. To assess recognition accuracy, we report the equal error rate (EER) which corresponds to the operating point with equal number of misses and false alarms.

Table 1: Parameters of prosodic feature extraction based on f_0-values

Basis function independent	Range
`getf0s` configuration	Config. 1, 2, 3 or 4
Interpolation	None, linear or Hermite
Frame duration	Approx. 70 ms to 300 ms
Frame shift	($\frac{1}{20}$ to $\frac{1}{2}$) x frame duration
Basis function dependent	**Range**
Windowing	Rectangular or Hamming
Processing domain of f_0 values	Linear or logarithmic
Number of coefficients	2 to 20

[1] `http://www.itl.nist.gov/iad/mig/tests/sre/2006/index.html`

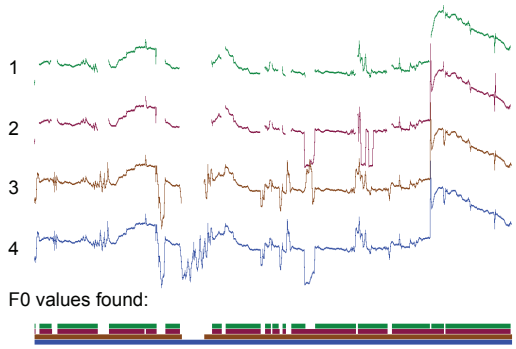

Figure 2: Different f_0 curves extracted from the same segment of speech using all four configurations in Table 2

Table 1 summarizes the most important parameters of feature extraction. Four different f_0 tracker configurations, as visualized in Fig. 2, with different amount and quality of extracted f_0 values are considered. The detailed `getf0s`-settings for each configurations are shown in Table 2. Configurations 1 and 4 lead to smallest and highest number of f_0 values, respectively, with the other two falling in between these two.

Table 2: Settings of the four configurations; Config. 2 is `getf0s`'s default configuration, differing values are printed bold

	1	2	3	4
Cost for octave f_0 jumps	**0.5**	0.35	**0.7**	**0.5**
Weighting given to f_0 trajectory smoothness	**0.0225**	0.02	**0.02**	**0.0225**
Correlation peak threshold	**0.05**	0.3	**0.05**	**0.05**
Weighting of shorter lags	**0.7**	0.3	**0.7**	**0.7**
Amplitude-change-modulated VUV transition cost	0.5	0.5	**1.4**	**1.4**
Spectral-change-modulated VUV transition cost	0.5	0.5	**0.42**	**0.42**
Bias towards voiced hypothesis	0	0	**0.42**	**0.91**

4. Results

4.1. Choosing f_0 Tracker and Interpolation Parameters

We first optimize the f_0 tracker and interpolation parameters. For these experiments, we use 6 coefficients extracted using the DCT on Hamming windowed frames of linear f_0 values with a length of 200 ms. The results for all 12 combinations of the four extractor configurations and three interpolation techniques are shown in Fig. 3. Hermite interpolation performs poorly whereas the two other techniques are close to each other. In general, accuracy improves by extracting less but more reliable f_0 frames (configurations 1 and 2), as was also shown in previous work [10]. For the following experiments, we will use `getf0s`-configuration 2, no interpolation, a frame size of 200 ms and a frame shift of 20 ms.

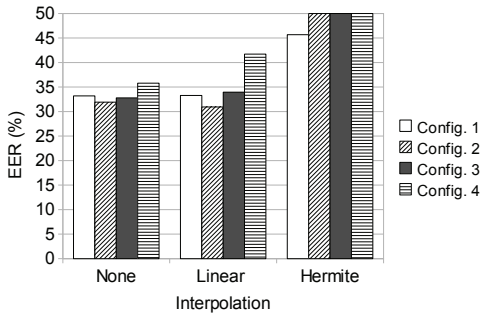

Figure 3: The performance of the different interpolations. Refer to Table 2 for the four f_0 configurations

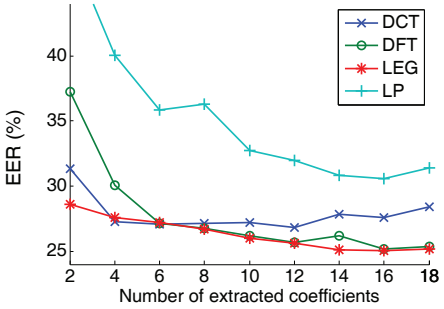

Figure 4: Varying number of extracted coefficients for all basis functions

4.2. Comparing the Basis Functions

Table 3 shows the results for all four basis functions using different settings for the processing domain of the f_0 values (linear or log) and the windowing of the frames (none or Hamming). The optimum setting clearly depends on the basis function. For DCT, DFT and LPCC, windowing has mostly positive effects and will be used from now on but for Legendre features, windowing degrades accuracy and is not applied. Logarithmic f_0-values will be used for the DCT, Legendre or LPCC features. For DFT, linear f_0 is better (note that the DFT magnitudes, however, are always represented in log-domain).

Table 3: Windowing and processing domain

Windowing	Domain	DCT	DFT	Leg.	LPCC
None	linear f_0	29.63	28.23	27.28	39.56
	log f_0	27.81	29.23	26.02	31.06
Hamming	linear f_0	28.49	27.13	27.98	41.74
	log f_0	27.06	29.27	26.69	30.34

Figure 4 further compares the four basis functions, configured with the best settings as determined in the preceeding experiment, by varying the number of feature coefficients. It shows that DFT and Legendre features improve by increasing the number of coefficients to 16; for DCT, good values are between 6 to 12 coefficients. The LPCC method yields generally high error rates and is not considered further in this paper. We hypothesize the reason to be that the autocorrelation method treats values outside of the frame as zeros. Unlike a speech waveform, which generally has positive and negative sample values, the f_0 track contains strictly positive values – for speakers with high pitch range, the boundary effects will be more dramatic. Further study of the LP is required.

Table 4: Added Delta features

		Frame length + shift (ms)				
		200+20	200+10	150+10	100+10	70+10
DCT	Base	26.49	27.28	28.52	30.57	32.48
	$+\Delta$	26.57	25.54	25.38	24.76	24.41
	$+\Delta^2$	26.95	26.24	25.72	25.17	24.19
DFT	Base	25.18	25.33	24.72	24.96	26.55
	$+\Delta$	25.59	26.24	24.98	24.47	25.04
	$+\Delta^2$	27.43	29.03	27.95	26.72	26.32
Leg.	Base	25.82	25.83	25.41	24.82	26.04
	$+\Delta$	23.56	23.89	22.81	22.99	22.24
	$+\Delta^2$	24.13	24.15	23.7	22.97	22.28

Table 4 shows the effect of including the delta and double delta coefficients. Since delta computation leads to a smaller number of feature vectors due to boundary handling, we also re-consider the framing parameters (frame duration and frame shift). The results in Table 4 indicate, firstly, that smaller frame duration improves accuracy. But the more interesting observa-

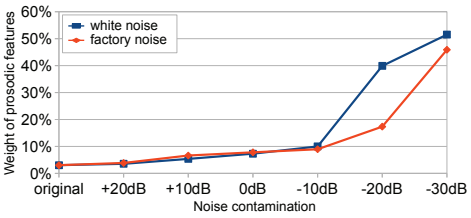

Figure 5: Trained weighting of prosodic vs. spectral classifiers

tion is that the higher-order dynamic information is very useful – for instance, DCT accuracy improves from 32.48 % to 24.19 %, a relative reduction of more than 25 %. Finally, most of the improvement comes from the first order deltas – accuracy degrades in most cases when double deltas are included.

4.3. Evaluation in Additive Noise and Fusion With MFCCs

We next evaluate f_0 feature robustness under additive noise conditions. Based on Table 4, we use a 70 ms long f_0 frame with 10 ms shift, with base and first order delta coefficients. The results under white and factory noise corruptions are shown in Tables 5 and 6, respectively, along with a MFCC baseline reference (both without and with spectral subtraction [19]) and fusion of the two. f_0 features remain almost intact until -10 dB for both noise types (sometimes they even slightly *improve* when more noise is added which confirms the general claim of robustness of intonation). For the MFCC features, spectral subtraction appears critical but the f_0 features do not require any additional pre-processing. In fact, spectral subtraction was found to be detrimental for the intonational features since it introduces artefacts recognized as voicing by the `getf0s`-algorithm, corrupting the extracted f_0 curve since f_0 values are also found in unvoiced regions and even regions without speech content.

Table 5: Performance under white noise contamination

	Intonational features			MFCC features		Fusion
SNR (dB)	(1) DCT	(2) DFT	(3) Leg.	(4) w/o SS	(5) with SS	(3)+(5)
original	24.41	25.04	22.24	9.87	9.99	9.37
20	24.25	24.92	22.11	9.98	10.12	9.36
10	23.95	24.72	21.96	10.56	10.28	9.40
0	24.23	24.92	21.98	14.81	11.49	10.23
-10	25.73	26.77	23.15	31.67	14.95	13.04
-20	30.52	31.76	28.42	39.33	27.61	24.38
-30	42.29	41.90	40.49	45.92	46.03	41.48

Table 6: Performance under factory noise contamination

	Intonational features			MFCC features		Fusion
SNR (dB)	(1) DCT	(2) DFT	(3) Leg.	(4) w/o SS	(5) with SS	(3)+(5)
original	24.41	25.04	22.24	9.87	9.99	9.37
20	24.74	24.96	22.61	10.28	10.16	9.51
10	24.62	24.77	22.49	10.70	10.68	9.84
0	24.56	24.78	22.40	11.65	11.46	10.26
-10	26.23	26.47	23.81	22.51	13.25	11.73
-20	31.80	32.98	30.04	29.78	21.12	19.13
-30	42.45	42.50	41.34	39.11	36.85	36.34

Fusion of prosodic and spectral classifiers yields the best results. Tables 5 and 6 show the EERs for the fusion of the best prosodic (Leg.) and the best spectral (MFCC with spectral subtraction) classifiers. Fusion is realized as linear weighted fusion $f = \beta + w_{\text{Leg}}\text{LLR}_{\text{Leg}} + w_{\text{MFCC}}\text{LLR}_{\text{MFCC}}$, where the bias β and the weighting for the log-likelihood ratios of the MFCC and Legendre classifiers are optimized using logistic regression[2]. Even though the oversimplified approach of training and testing fusion on the same data set and fixed SNR rates hardly match real-world conditions, the classifier weights clearly indicate increasing importance of prosodic features with decreasing SNR; Fig. 5 shows $|w_{\text{Leg}}|/(|w_{\text{Leg}}| + |w_{\text{MFCC}}|)$.

[2] `http://www.dsp.sun.ac.za/~nbrummer/focal/`

5. Conclusion

Coming back to the questions posed in the introduction, we recommend to use the default `getf0s` configuration for the f_0 tracker, producing fewer but more reliable f_0 values. Data interpolation is *not* recommended. A window size of about 70 ms and a very small frame shift of 10 ms with logarithmic f_0 values seem to work best, as suggested by Sönmez *et al.* ([18]). Regarding the basis functions, Legendre polynomials are recommended – after optimizing the parameters of each method, the Legendre method yielded systematically the lowest error rates under all considered SNR levels and for both white and factory noise. Interestingly, the first order delta coefficients of the base features yield significant boost to the features. As for the accuracy in noisy conditions, f_0 features are almost intact until -10 dB SNR level. Finally, MFCC features yield systematically higher accuracy but additional spectral subtraction processing is necessary; the intonational features, in turn, require no additional data cleaning. Fusion experiments show that prosodic features can especially improve the recognition rate of noisy signals.

6. References

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, January 2010.

[2] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41, January 2000.

[3] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Trans. Audio, Speech and Language Processing*, vol. 16, no. 5, pp. 980–988, July 2008.

[4] R. Saeidi, J. Pohjalainen, T. Kinnunen, and P. Alku, "Temporally weighted linear prediction features for tackling additive noise in speaker verification," *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 599–602, 2010.

[5] J. Fletcher, *The Handbook of Phonetic Sciences*. Birkhauser, 2010, ch. The Prosody of Speech: Timing and Rhythm.

[6] T. Kinnunen and R. González-Hautamäki, "Long-term F_0 modeling for text-independent speaker recognition," in *Proc. 10th International Conf. Speech and Computer (SPECOM'2005)*, Patras, Greece, October 2005, pp. 567–570.

[7] A. Adami, "Modeling prosodic differences for speaker recognition," *Speech Communication*, vol. 49, no. 4, pp. 277–291, April 2007.

[8] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke, "Modeling prosodic feature sequences for speaker recognition," *Speech Communication*, vol. 46, no. 3-4, pp. 455–472, July 2005.

[9] N. Dehak, P. Kenny, and P. Dumouchel, "Modeling prosodic features with joint factor analysis for speaker verification," *IEEE Trans. Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 2095–2103, September 2007.

[10] M. Kockmann, L. Burget, and J. Černocký, "Investigations into prosodic syllable contour features for speaker recognition," in *Proc. ICASSP 2010*, 2010, pp. 4418–4421.

[11] B. Atal, "Automatic speaker recognition based on pitch contours," *Journal of the Acoustic Society of America*, vol. 52, no. 6, pp. 1687–1697, 1972.

[12] K. Iwano, T. Asami, and S. Sadaoki, "Noise-robust speaker verification using f_0 features," in *Proc. Interspeech 2004*, Jeju Island, Korea, October 2004, pp. 1417–1420.

[13] "The snack sound toolkit," April 2010, http://www.speech.kth.se/snack/. [Online]. Available: http://www.speech.kth.se/snack/

[14] P. Boersma and D. Weenink, "Praat: doing phonetics by computer [computer program]," WWW page, February 2011, http://www.praat.org/.

[15] E. Helander and J. Nurminen, "A novel method for prosody prediction in voice conversion," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2007)*, vol. 4, Honolulu, Hawaii, USA, April 2007, pp. 509–512.

[16] J. Makhoul, "Linear prediction: a tutorial review," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 561–580, April 1975.

[17] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: a Guide to Theory, Algorithm, and System Development*. New Jersey: Prentice-Hall, 2001.

[18] M. Sönmez, L. Heck, M. Weintraub, and E. Shriberg, "A lognormal tied mixture model of pitch for prosody-based speaker recognition," in *Proc. 5th European Conf. on Speech Communication and Technology (Eurospeech 1997)*, Rhodos, Greece, September 1997, pp. 1391–1394.

[19] P. Loizou, *Speech Enhancement: Theory and Practice*. CRC Press, 2007.