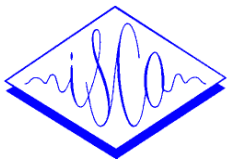

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

MULTISTAGE COARTICULATION MODEL COMBINING ARTICULATORY, FORMANT AND CEPSTRAL FEATURES

*Yuqing Gao, Raimo Bakis, Jing Huang, Bing Xiang**

IBM Thomas J. Watson Research Center, Yorktown Heights, NY

ABSTRACT

We describe a multi-stage speech production model containing a linear, phoneme-independent coarticulation filter, followed by a nonlinear component. The latter generates two cepstra which are then additively combined: one corresponding to a relatively smooth background spectrum, and the other representing three formant-like spectral peaks. A neural net is used for both parts, but the second part also utilizes a hard-coded function that generates exactly three spectral peaks. A unified model of training, adaptation, and decoding is developed, each operation differing only with respect to prior probability distributions. Prior probabilities can be introduced at each stage of the model, providing a flexible framework for utilizing both specific and general prior knowledge. We demonstrate the use of this model for speech synthesis as well as recognition.

1. INTRODUCTION

The coarticulation model for speech recognition proposed by Bakis[1] has been recently revived by several researchers as hidden dynamic models[2, 3, 5]. Such models describe the physical process of speech production, explicitly consider the coarticulations and transitions between neighboring frames and phones, and also introduce additional prior knowledge, thus reducing the amount training data needed.

In a coarticulation model, a sequence of target vectors corresponding to a phone sequence are the input to the system and mapped to "realized vectors" by a set of coarticulation filters. Then the realized target vectors are sent to a nonlinear system to generate synthesized acoustic vectors.

In earlier work, the input target vectors have been formants [2] or vocal tract resonances (VTRs)[3] which are also formant related features. We recognize that linear laws of mechanics, e.g. inertia, operate on mass particles of the articulators, rather than on formant frequencies. We believe it is more appropriate to apply a linear coarticulation filter to articulatory coordinates rather than to formant frequencies. Formants, nevertheless, are a useful and parsimonious way of describing speech spectra. In the work presented in this paper, a sequence of articulatory coordinates corresponding to a time-aligned phonetic transcription is used as the target sequence. Formants are also used as complementary parameters to cepstra to represent the speech spectrum. The objective function we use for system parameter optimization is also different from earlier work in the area. We introduce a number of additive "constraint" terms to the objective function to represent prior probabilities. These are positive definite quadratic functions of variables assumed to have a Gaussian prior. As explained below, these priors can be tailored for training, or adaptation, or decoding. During training, for example, we have only little prior knowledge about the target articulation of a particular phoneme, but during decoding the same target articulation would have a prior with much smaller variances. Thus, our optimization process can be extended to speaker adaptation and utterance adaptation even when the system is already speaker adapted, by adding appropriate terms to the objective function. With this unified treatment, the differences between training, adaptation and decoding disappear, all being examples of the same optimization process.

The rest of the paper is organized as following: The model is reviewed briefly in Section 2, In Section 3, we describe the system structure. In Section 4, we discuss the objective function and the optimization for training, adaptation and decoding. In Section 5, we present the examples of speech recognition and synthesis using the coarticulation model. Section 6 contains additional discussion of the model and underlying concepts.

2. OVERVIEW

Our multi-stage speech production model is applicable to both recognition and synthesis. The model invokes phonetic, articulatory, acoustic, and signal processing concepts in order to connect two speech representations: a pcm waveform and the corresponding time-aligned phonetic transcription. Between these overt representations, the model introduces four "hidden" ones:

- Time-aligned phonetic transcription $\phi(t)$ (overt).
  1. Target articulation $\vec{x}(t)$ (hidden).
  2. Realized articulation $\vec{y}(t)$ (hidden).
  3. Frequencies $\vec{f}(t)$, amplitudes $\vec{a}(t)$ and bandwidths $\vec{b}(t)$ of formants (hidden).
  4. Cepstrum $\vec{C}(t)$ or Log power spectrum $\vec{S}(t)$ (hidden).[1]
- Waveform $W(t)$ (overt).

These hidden speech representations, together with hidden vectors of model parameters constitute the model parameter set $\vec{\theta}$:

$$\vec{\theta} = (\vec{u}(\cdot), \vec{x}(t), \vec{y}(t), \vec{f}(t), \vec{a}(t), \vec{b}(t), \vec{p}, \vec{w}, \vec{E}(t))$$

where, $\vec{u}(\cdot)$ is the table of target articulations discussed in the next section, $\vec{p}$ represents the filter parameters, $\vec{w}$ are the MLP weights in the nonlinear mapping function and $\vec{E}(t)$ is the difference between synthesized and observed

*Also with School of Electrical Engineering, Cornell University, Ithaca, NY

[1] Although an *estimated* power spectrum and cepstrum can be calculated from the overt waveform, the actual speech spectrum and cepstrum are hidden, especially when the waveform is corrupted by noise and distortion.

cepstra. Note that some components of $\vec{\theta}$, such as $\vec{u}(\cdot)$, are utterance-independent, though they may depend on the speaker. Others, such as $\vec{x}(t)$, are utterance-specific.

3. STRUCTURE OF THE SYSTEM

The time-aligned phonetic transcription is represented by the function $\phi(t)$, which maps time frames t into a finite phonetic symbol alphabet Φ.

With each phonetic symbol in that alphabet there is associated a target articulation. In the current work, this is a point $\vec{x} \in \mathbb{R}^5$. Let $\vec{u}(\phi)$ be the table lookup function that maps elements of the phonetic alphabet into target articulations, so that the target at time t is

$$\vec{x}_t = \vec{u}\left(\phi(t)\right). \tag{1}$$

We use five articulatory features as in [4, 6]. They represent lips, tongue blade (TB), tongue dorsum (TD), velum, and larynx, similar to the five vocal tract variables for speech synthesis. They are capable of describing major coarticulatory phenomena. The target values of these features, for each phone, are context independent, and are related to the intended target articulatory gestures. The systematically nonstationary segments are decomposed into relatively stationary subsegments. For example, a diphthong is represented by two target vowels. This table of target values constitutes part of the model parameter vector $\vec{\theta}$. These values are initialized from prior, language-specific knowledge of phonetics but the final values are "learned" during the training of the system and further adjusted in adaptation and decoding.

The target articulations $\vec{x}_t$ are then smoothed by a Kalman smoother to generate "realized articulations" $\vec{y}_t$. This smoothing filter solves Eq. 2.

$$y_t\left(p^{-1}+2\right) = p^{-1}x_t + y_{t+1} + y_{t-1} \tag{2}$$

This can also be written as:

$$y_t = \frac{p^{-1}x_t + y_{t+1} + y_{t-1}}{p^{-1}+2}. \tag{3}$$

In the absence of input, i.e. when $x_t = 0$, the above equation has the solution:

$$y_t = ae^{-\frac{t}{\tau}} \tag{4}$$

where, for $p > 1$, $\tau \approx \pm\sqrt{p}$.

This is a non-causal filter: its impulse response is a two-sided exponential and its phase shift is zero. Its transfer function consists of two poles, one on the positive and one on the negative real axis, equidistant from the origin.

In the present work, such a filter was applied to each component of the $\vec{x}$ vector separately, with independent values of p_i for each. Those values of p_i are also part of the model parameter vector $\vec{\theta}$.

The acoustic signal, and hence the cepstrum, depends on the articulatory configuration, but not in a linear manner. For example, as the tongue rises during the utterance of the word "eat", the second formant at first rises, but then disappears. In our model, the $\vec{y}_t$ vectors are surrogates for articulatory parameters and the synthetic cepstrum at frame t is then a nonlinear function of $\vec{y}_t$.

We first tried to model this nonlinear function by a multilayer perceptron (MLP) with one hidden layer consisting of 100 elements. Because the speech spectrum in many sounds looks like the output of an all-pole filter with definite resonances (formants), we decided to add another optional parallel path which is "hardwired" to generate three spectral peaks. The parameters controlling the amplitudes, center frequencies, and bandwidths of these peaks are also generated by the MLP as nonlinear functions of the $\vec{y}_t$ vectors. The weights $\vec{w}$ in the MLP constitute an additional part of the parameter vector $\vec{\theta}$.

Therefore the MLP has 2 parts of output. One part has 24 output elements to generate the 24 components of the synthetic cepstrum $\hat{C}_{1j}(t)$. The second part has 9 additional output elements in groups of three to generate the frequencies $\vec{f}(t)$, amplitudes $\vec{a}(t)$, and bandwidths $\vec{b}(t)$ of three formants. A formant spectrum (in Mel-scale) $S_j(t)$ is calculated using a bell-shaped formant function from these formant parameters after appropriate dynamic range scaling.

$$S_j(t) = \sum_{k=0}^{2} a_k(t) e^{-\frac{(j-f_k(t))^2}{2b_k(t)^2}} \tag{5}$$

where, $j = 0, 1, ...23$. Typical formant values for American English vowels are given in [7].

$S_j(t)$ then is converted to a cepstrum $\hat{C}_{2j}(t)$ by a DCT (discrete cosine transform) and added to $\hat{C}_{1j}(t)$ to generate the final "synthesized" cepstral parameters $\hat{C}_j(t)$, where $j \in \{0, 1, ..., 23\}$.

The second part of the synthetic cepstrum $\hat{C}_{2j}(t)$, which corresponds to the formant spectrum, is optional and complements the synthetic cepstrum $\hat{C}_{1j}(t)$. We performed experiments with and without this formant synthesis part. The results are presented and discussed in later sections.

The observed cepstral parameters $C_j(t)$ are computed from the original speech waveform $W(t)$. The differences $E_j(t) \equiv \hat{C}_j(t) - C_j(t)$, become additional components of $\vec{\theta}$.

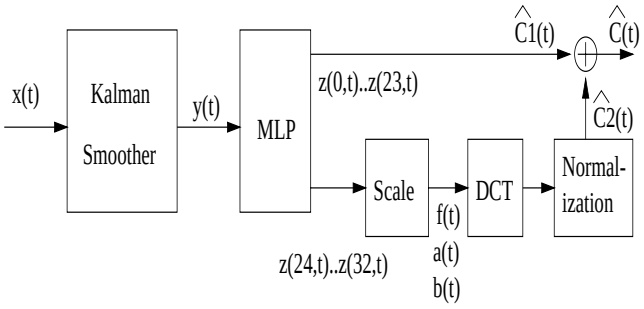

Figure 1: Coarticulation Model diagram

4. OBJECTIVE FUNCTION AND OPTIMIZATION

In the simplest form of the model, the objective function F is just the sum of the squares of the cepstral error terms:

$$F = \sum_t \sum_j E_j^2(t) \tag{6}$$

It is more generally useful, however, to think of the objective function as the negative logarithm of the joint probability of a given transcription and waveform, given the model's

current knowledge. This allows us to interpret the $E_j(t)$ values as normally distributed estimation errors resulting perhaps from acoustic noise and also from the statistical fluctuations inherent in power spectrum estimation. For IID normal variates, the sum of the squares is, indeed, proportional to the negative logarithm of the probability.

If, during training, this knowledge about cepstral estimation errors is the only prior information available, then training consists of minimizing the sum of these quadratic error terms, i.e. all other components of $\vec{\theta}$ can vary freely with no effect on the objective function. In practice, it is useful to also constrain the system from runaway conditions by adding a term $\epsilon \sum_i \theta_i^2$, where ϵ is sufficiently small not to disturb results significantly. This represents prior knowledge that all components of $\vec{\theta}$ must be reasonably scaled. Then

$$F = \sum_t \sum_j E_j^2(t) + \epsilon \sum_i \theta_i^2. \qquad (7)$$

Eq. 7 is the objective function we used during system training.

4.1. Speaker adaptation

After this initial training, the system may assume that values of the utterance-independent components of $\vec{\theta}$ for individual speakers are distributed normally, with means at the speaker-independent $\vec{\theta}$. This assumption generates an additional term for the objective function for each speaker s, proportional to $\sum_i \frac{(\theta_i^s - \theta_i)^2}{2v_i}$ where v_i is the variance of the i-th component of $\vec{\theta}$ among the speaker population. For utterance-specific components of $\vec{\theta}$, this variance would be very large. The objective function for speaker adaptation is adjusted accordingly:

$$F = \sum_t \sum_j E_j^2(t) + \epsilon \sum_i \theta_i^2 + \epsilon_1 \sum_i \frac{(\theta_i^s - \theta_i)^2}{2v_i} \qquad (8)$$

4.2. Utterance adaptation during decoding

Finally, similar reasoning, with speaker dependent variances v_i^s, which have smaller values than v_i, can be applied during decoding, to allow limited adjustment of $\vec{\theta}$ even when the system is already adapted to the current speaker, i.e., assuming $\vec{\theta}$ varying from one test utterance to another given an adapted speaker, the differences of the values of $\vec{\theta}$ for each utterance and for the given speaker vs. the variances v_i^s can be added to the objective function for utterance based adaptation during decoding.

It can be seen from above analysis that, in such a model, the boundaries between training, adaptation, and decoding disappear. All of these operations require minimization of an objective or cost function over the product space of messages (e.g. phonetic transcriptions) and model parameters. Prior probabilities exist for both components. The essential difference is in the shape of the prior. During training the message is well known, i.e. the message-space marginal of the prior is concentrated at or near a single point, but model parameters are not known and their prior distribution is wide. The opposite is true for decoding: model parameters are known from previous training, but there is little prior information about the message. Except for that quantitative difference, however, the operations of training, adaptation, and decoding are mathematically the same.

4.3. Optimization

For training, adaptation and decoding, our computation actually starts from both ends and meets in the middle: The observed pcm waveform goes into a spectrum computation and is then further transformed into the *observed* mel-scale cepstrum $\vec{C}(t)$. Starting from the other end, a known or hypothesized transcription is converted by table lookup to a target articulation function which is then smoothed by the coarticulation filter and transformed nonlinearly into a *synthesized* cepstrum $\hat{\vec{C}}(t)$. The objective function is then computed as discussed above.

For this optimization task, we have chosen the Limited Memory BFGS Algorithm because it imposes fewer limitations on the structure of the model than does the popular estimation-minimization (EM) method, but it nevertheless converges rapidly [8] [9].

During training, for each iteration of the optimization, the input to the BFGS algorithm includes the current value of the objective function F, which is computed over all the training samples, the current values of $\vec{\theta}$ and the derivatives $\frac{\partial F}{\partial \theta_i}$. The algorithm then suggests new values of $\vec{\theta}$ to minimize the objective function.

5. EXPERIMENTAL RESULTS

5.1. Experiments on wide-band, clean speech

The experimental system was trained with 20 minutes of wide-band, clean read speech from one male speaker. Audio output of the synthesized speech and estimated formants from this speech production model will be demonstrated during presentation.

For synthesis, the procedure is simple, at least formally: Starting from a time-aligned transcription $\phi(t)$, proceed to compute $\vec{x}(t)$, $\vec{y}(t)$, etc. to obtain $\vec{C}(t)$. Conversion from the cepstrum to waveform is slightly more difficult, but reasonably accurate methods exist.

From the synthesized spectra, it can be seen that $\hat{C}_{1j}(t)$ (Fig. 4) has good representation for the envelope of the spectrum, while $\hat{C}_{2j}(t)$ (Fig. 5) presents the peaks of the spectrum. These two are complementary to each other. The combined synthetic spectrum (Fig. 3) is very similar to the original spectrum (Fig. 2). Fig. 6 shows the synthetic spectrum when the formant spectrum synthesizer is disabled during training. Although the model with the formant spectrum synthesizer has the potential to achieve a better performance, it is difficult to train because there exist more local optimal values.

The use of the model for speech recognition is presented in the framework of an N-best rescoring mode. Two sets of testing data are tested. One is 10 minutes of wide-band, clean read speech from the same speaker (called speaker-dependent test). The other is 40 minutes of wide-band, clean speech from 4 other male speakers (speaker-independent test). The N-best decoding hypotheses are generated by IBM's large vocabulary continuous speech recognition system (HMM). The rescoring experiments (Table 1) compare the performance between HMM system and our coarticulation model (CAM) system. From the results in Table 1, the CAM model does produce some improvement over HMMs in the speaker dependent trained case, but not for the speaker independent case. This may be because the test data is wide-band, clean speech and doesn't contain enough coarticulation variations.

Figure 2: Spectrum from original speech

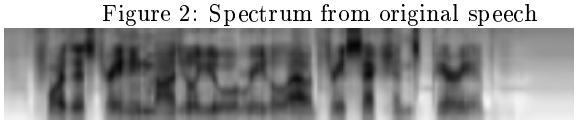

Figure 3: Synthesized spectrum from C1+C2

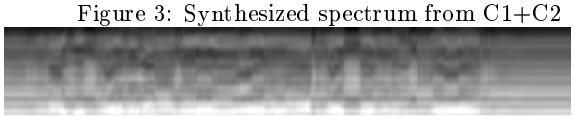

Figure 4: Synthesized spectrum from C1

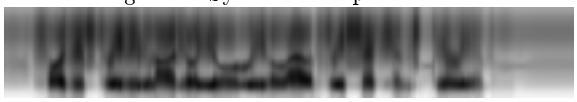

Figure 5: Synthesized spectrum from C2

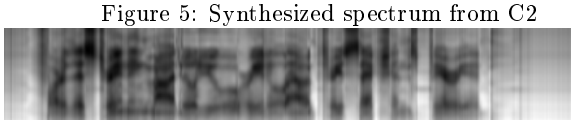

Figure 6: Synthesized spectrum from C1 only

	SD		SI	
	HMM	CAM	HMM	CAM
10-best + Ref	7.6%	7.1%	15.7%	15.8%
10-best	7.8%	7.2%	17.0%	17.9%

Table 1: Wide-band, clean speech recognition results

5.2. Experiments on narrow-band, spontaneous speech

The second experimental system is trained for Switchboard data, which is telephone data, i.e., narrow-band, noisy and spontaneous speech. The training speech is about 10 minutes speech from one male speaker. The test speech is 40 minutes of speech from other 8 male speakers.

The experimental results (Table 2) showed the coarticulation model outperforms the HMM model for rescoring noisy, spontaneous speech even when the model is trained with a very small amount of speech from one different speaker. This is consistent with others' work [2, 3, 5], however, we use only 10 minutes of training speech, which is much less than their training speech.

	HMM	CAM
10-best + Ref	42.6%	41.4%
10-best	45.5%	46.0%

Table 2: Speech recognition for Switchboard data

6. DISCUSSION

Although the multistage coarticulation model has demonstrated good speech synthesis performance, its capability to improve HMM based speech recognizer is not fully illustrated yet, especially when the reference script is included in the N-best hypotheses. One obvious reason is that the coarticulation model is only trained with a very small portion of the training data (20 minutes for CAM vs. 200 hours for HMM for systems in Table 1, and 10 minutes for CAM vs. 270 hours for HMM for systems in Table 2). The number of parameters in the coarticulation model is also very small (4248 parameters in CAM vs. 3.5 million parameters in HMMs). Another possible reason is that CAM is only exposed to the context independent phone alignment of the training data, while the HMM system is trained with context dependent state alignment of the data. Future work will include the utilization of context dependent information of the training data and more complex and detailed models (using more model parameters).

7. REFERENCES

[1] R. Bakis, "Coarticulation Modeling with Continuous-State HMMs", Proc. IEEE Workshop Automatic Speech Recognition, pp 20-21, Arden House, New York, 1991.

[2] H. Richards, J. Bridle, "The HDM: A Segmental Hidden Dynamic Model of Coarticulation", Proc. of ICASSP99, pp 357-360, Phoenix, Arizona, 1999.

[3] L. Deng and J. Ma, "A Statistical Coarticulation Model for the Hidden Vocal-Tract-Resonance Dynamics", Proc. of EuroSpeech'99, pp 1499-1502, Budapest, Hungary, 1999.

[4] L. Deng, "A Statistical Approach to Automatic Speech Recognition Using the Atomic Speech Units Constructed from overlapping Articulatory Features", J. Acoust. Soc. Am., 95 (5), May 1994.

[5] J. Picone, et al, "Initial Evaluation of Hidden Dynamic Models on Conversational Speech", Proc. of ICASSP99, pp 109-112, Phoenix, Arizona, 1999.

[6] C. Browman, L. Goldstein, "Gestural Specification Using Dynamically-Defined Articulatory Structures", J. Phon. 18, 299-320, 1990.

[7] G. Peterson and H. Barney, "Control methods used in a study of the vowels", J. Acoust. Soc. Am., 24, 175-184, 1952.

[8] Jorge Nocedal and S. Nash," A Numerical Study of the Limited Memory BFGS Method and the Truncated-Newton Method for Large Scale Optimization", (1991), SIAM Journal on Optimization, 1, 3, pp. 358-372.

[9] Jorge Nocedal, C. Zhu, R. Byrd and P. Lu, "Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization", (1997), ACM Transactions on Mathematical Software, Vol 23, No. 4, pp. 550 - 560.