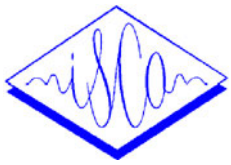
ISCA Archive
http://www.isca-speech.org/archive



7th ISCA Workshop on
Speech Synthesis (SSW-7)
Kyoto, Japan
September 22-24, 2010

KLATTSTAT: Knowledge-based Parametric Speech Synthesis

Gopala Krishna Anumanchipalli[1], *Ying-Chang Cheng*[2],*Joseph Fernandez*[2]
Xiaohan Huang[2], *Qi Mao*[2], *Alan W Black*[1]

[1]Language Technologies Institute, [2]Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA, USA
{gopalakr, awb}@cs.cmu.edu

Abstract

This paper is an initial investigation into using knowledge-based parameters in the field of statistical parametric speech synthesis (SPSS). Utilizing the types of speech parameters used in the Klatt Formant Synthesizer we present automatic techniques for deriving such parameters from a speech database and building a statistical parametric speech synthesizer from these derived parameters. Although the work is exploratory, it shows promise in using more speech production inspired parameterizations for statistical speech synthesis.

Index Terms: statistical speech synthesis, Klatt formant synthesizer.

1. Introduction

Over the last thirty years we have seen the advancement of speech synthesis from hand crafted rule-driven formant synthesis techniques [1]; controlled inventory concatenative synthesis [2], (e.g. diphones), large inventory unit selection synthesis [3], and the latest technology investigates statistical parametric generation based techniques [4]. We can view this progression as benefiting from improved machine learning modeling techniques which have in turn been aided by the advancement in computation power and increasing database sizes. One advantage is that synthesis is now feasible in languages where little phonetic or linguistic knowledge is available. Modeling techniques are often sufficient to capture language properties such that adequate synthesis is possible with sometimes only orthography and audio of a reasonably small database [5].

However it is notable that the selection of parameterizations for SPSS is still a hot research topic. There is substantial active work on finding improved excitation modeling techniques [6, 7, 8]. Although alternative spectral parameterization is also being studied (MFCCs vs LSF [9]) these are currently mostly addressed at derived functions from FFTs. We wish to expand that search to investigate parameterizations that are more targeted to human speech. For our initial study we returned to the earlier speech synthesis work of Dennis Klatt.

2. Klatt Formant Synthesis

Klatt Formant Synthesis [10] is a synthesis technique where a set of parameters are generated from text by rule from which a waveform file is constructed from a cascade of modules to give a resulting signal. The choice of parameters is based on established theories of speech production and perception. They include source features (like glottal sampling; pitch; measurements of aspiration and frication) and vocal tract features (like resonant nasal and formant frequencies, bandwidths and amplitudes). Though ground breaking at the time, the technique required experts to construct such suitable values for such parameters by hand in order to optimally produce human sounding speech. With the advent improved computational resources, both speed and space, techniques that automatically train from recorded natural speech have prevailed as they can offer both more natural synthesis, and can require less phonetic knowledge of the language and speaker to create. However in requestioning the optimal parameterization for modern statistical parametric speech synthesis we decided to re-visit the original selection of Klatt Formant Parameters to see how they perform in today's statistical synthesis framework. In addition to using Klatt-like parameters in a statistically synthesizer, we must also address the novel issue of automatically deriving these parameters for a large database of natural speech. We do not have the expertise to do develop these parameters by hand or access to the original MITalk to get expert aid. We therefore have developed our own initial techniques to derive Klatt-like features directly from speech signals.

Broadly, Klatt parameters as described in [1] fall into three categories – i) F0 and Formant parameters (amplitudes, frequencies and bandwidths of the first 6 formants and the nasal formant), ii) quantified measures of articulatory features (amplitudes of aspiration, frication and nasality), and iii) Voicing amplitude, Overall gain etc., A complete description of Klatt features is presented in Appendix A. The following sections describe the techniques used for extraction of these parameters.

2.1. Formant Parameters

We use the *formant* package from the ESPS toolkit [11] to extract the formant parameters. For each 50 second analysis window with a 5 millisecond shift, we get the frequency and the bandwidth. We use the *FFT* program to compute the magnitude spectrum. The amplitude at the formant frequencies are noted as the formant amplitudes. It is to be noted that there are several practical considerations here like the kind of smoothing window, the number of points in the FFT, window size/shift etc. For the experiments here, we manually chose the parameters that best approximate the peaks on the spectra with the extracted formant frequencies.

Figure 1 marks the formants on the FFT magnitude spectrum for a voiced segment of speech. The decision to extract 6 formants was merely practical, as the Klatt synthesizer software we use expects 6 formants. Also, human speech is fairly well represented within the range of frequencies spanned by 6 formants.

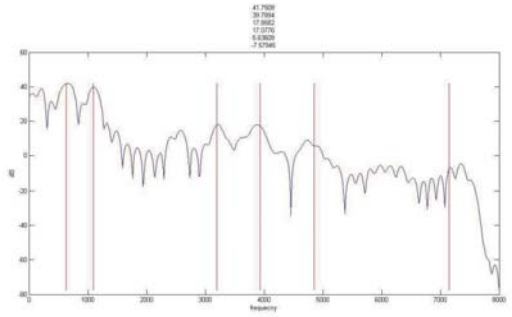


Figure 1: First 6 formants marked on the FFT spectrum

2.2. Nasality, Aspiration and Frication

Klatt's original synthesizer proposes use of coefficients of nasality, aspiration and frication. In this work, we use a discriminative approach as described in [12] to find these features in a signal. In [12], Gaussian mixture models (GMMs) are used to find the likelihoods of these articulatory phenomena. We build both positive and negative models for each phenomena using features (e.g MFCC) from the training set. As the data is already labeled with standard phonemes with a three state HMM labeller we can make of this information to training the models. We use only the frames labeled with the middle states of the relevant phonemes, making the assumption that the first and last states may cover transitions between the phonemes. A positive state refers to when a characteristic is present (nasality, aspiration, and frication) and negative states refer to when a characteristic is not present (non-nasality, non-aspiration, and non-frication). When training the positive state GMMs, the following phonemes are used: nasality, *n, m, ng*; frication, *f, hh, s, sh, th, v, z, zh*; and aspiration, *hh*. When training the negative states, all other phonemes not in these sets are used. In addition, phonemes that bordered a positive state phoneme are excluded from training the negative state GMM, so as to avoid transitional effects (for example, a non-nasal phoneme may be colored with some nasality if it is next to a nasal phoneme, and as such should not be used for training the non-nasal GMM). Once these GMMs are trained, they are used to aid in both scoring (i.e., how nasal a segment of speech is) and classification (i.e., if a segment of speech is nasal or non-nasal). These "detectors" can be used to output non-binary scores of these acoustic events.

During the testing phase, each utterance is processed by testing each short time feature sequence for each of the three detectors. Each detector assigns a score of 0 to a segment that is detected to be in the negative state. A non-zero score is assigned for positive classification of the events. The dynamic ranges of these scores may scaled to correspond to the decibel ranges as specified in original Klatt implementation. Each detector addresses two tasks: assigning the actual score and thresholding all negative state features. Some detectors investigated are described briefly below.

2.2.1. Maximum likelihood detector

This naïve detector assigns the class with the higher likelihood on the speech segment under consideration. Where the L_+ and L_- are the likelihoods for the positive and negative states from the respective GMM PDFs, the score is calculated as Eqn 1. Note that a score of zero is assigned whenever the negative state likelihood is greater than the positive state score, by thresholding any scores that are negative or imaginary.

$$S = log_{10}(L_+ - L_- + 1) \tag{1}$$

2.2.2. Bayes detector

The Bayes detector attempts to take into consideration the prior probabilities for the positive or negative states (obtained from the training data). The detector scores each test speech segment such that

$$S = \{ \begin{array}{cc} \log(L_+) & P(+|x) > P(-|x) \\ 0 & P(+|x) < P(-|x) \end{array} \tag{2}$$

Note that the likelihood from the GMM i is in the form $P(x|i)$. To get $P(i|x)$, Bayes rule may be used to transform the comparison between $P(x|i)$ and $P(x|j)$ to a comparison between $P(x|i)P(i)$ and $P(x|j)P(j)$, where i and j could assume positive and negative states. Note that the prior probabilities weight our decision.

2.2.3. Linear Discriminant Analysis

An LDA detector is developed to separate the GMM outputs of the positive and negative states. Here, the score output is the same as is found in Eqn 2, but the decision criteria are based on whether LDA determines a test MCEP to be either from the positive or negative state. To determine a threshold to use for this comparison, cross validation is done on the training data to determine the optimal threshold to use to separate the two classes. Two versions of this detector are tested ; one that uses all training data and one that used equal amounts of training data for the positive and negative states (the latter made the detector's decision bias more fair.

Another detector is used that discounts the GMM scores. Instead, LDA is applied to the features themselves. This essentially projects a high dimensional vector into a single dimension. Cross validation is performed to obtain optimal thresholds to distinguish the positive state from the negative state. Once projected onto one dimension, instances falling in the negative state are assigned a score of 0 and positive state instances were given a score based on a scoring function. Here, a Gaussian scoring function was used, but this could easily be extended with the use of different scoring functions and dimension of the projected space.

These methods are tested on a development and training set both in terms of error rates and cepstral distortion between the reference features and the resynthesized utterance's features. All other parameters in the Klatt synthesizer remain constant during resynthesis. The naïve detector performed the best in terms of error rates, but the LDA detector (equal training size) based on MCEPs performed the best in terms of cepstral distortion, and is our chosen method for our final implementation. These results are shown in Table 1. It should be noted that perceptually, it is difficult to notice a difference between these different methods, and as such all may be considered as good detectors for current purposes.

2.3. Other Parameters

Parameters like the *gain*, *skew* and *aturb* have been set empirically. The resynthesis is perceptually checked to sound as close to the original speech as possible. The default values of the

TrueState	Naïve	Bayes	GMM LDA	GMM LDAE	MCEP LDA	MCEP LDAE
Nasal	1.43%	3.40%	7.26%	19.90%	15.95%	4.79%
Non-Nasal	0.75%	0.40%	0.23%	0.40%	0.37%	1.41%
Fricative	3.78%	6.60%	20.38%	6.33%	12.62%	7.79%
Non-Fricative	5.00%	3.82%	6.80%	12.36%	7.05%	9.22%
Aspiration	3.01%	13.28%	63.66%	8.52%	89.47%	10.03%
Non-Aspiration	1.32%	0.21%	0.07%	4.00%	0.09%	15.89%
MCD mean	11.88	11.95	11.96	11.95	11.79	11.67
MCD variance	0.36	0.37	0.38	0.38	0.26	0.26

Table 1: Error rates on positive and negative examples.

program are used for the rest of the parameters. Wherever appropriate, silence and unvoiced segments are set to zeros or defaults. In all default values were used for 5 of the 40 parameters suggested by Klatt.

3. Synthesis Experiments

We used the Arctic *rms* database [13] for our experiments as it offers one of clearest spoken standard American voice. We extracted the 40 parameters for each 5 ms frame in the databases using the techniques described in the previous section. The parameters were then used within our Clustergen Statistical Parameter Speech Synthesizer [14]. We effectively replaced the MFCC features that we normally use with the Klatt Parameters. Although Clustergen offers various options for which features are used for clustering, and the option to build multiple models for different subsets of the parameter vectors, we used the simplest option and clustered with all the parameters.

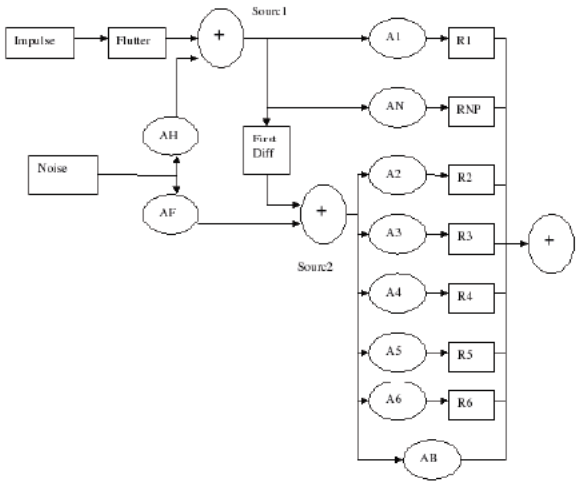


Figure 2: Schematic representation of Klatt-based synthesis [1]

In the following sections, we describe two experiments – 1) resynthesis of speech from extracted Klatt parameters and 2) Synthesis of speech from unseen text based on Klatt parameter based statistical speech synthesis.

3.1. Resynthesis

For resynthesis of these generated parameters we used [15], a C implementation of Klatt's original Fortran code. The Fig. 2 illustrates the schematic representation of the Klatt synthesizer. The input excitation is either an impulse train for voiced sounds and noise for unvoiced sounds. This is input to aspiration, frication and resonators (corresponding to the formant resonances) as illustrated in the Fig. 2.

The extracted Klatt features are used as input to the Klatt synthesizer to reconstruct the speech signal. The results are encouraging with perceptually almost perfect resynthesis. Fig. 3 compares the spectra of original and synthesized portions of a voiced segment. Evidently, the peaks align precisely in the lower frequency regions. We are still investigating the attenuation effect that is affecting the higher frequency ranges.

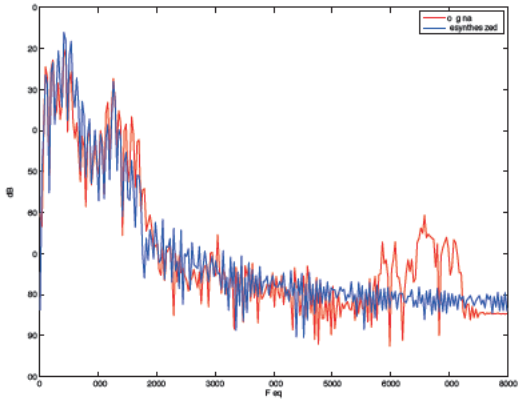


Figure 3: Resynthesis vs original spectrum of 5ms speech segment of the phoneme *eh*

3.2. Text-to-Speech

To investigate the Klatt-parameter based Clustergen framework for Text-to speech, we built CART trees of Klatt parameters extracted for the speech database. The trees are clustered using the same contextual questions that are commonly used in MCEP based voice building. A Klatt parameter tree is trained

for each of the three HMM states within a phoneme. At runtime, the parameter vectors are generated using the duration, F0 and the Klatt parameter trees. For synthesizing speech from the predicted parameter file, we use the same C code used for resynthesis. The models for duration and F0 are the same that are built for the default voice (using MCEPs).

We compared the two voices built using MCEPs and Klatt parameters. Since the same duration model is used for the two voices, outputs are time-aligned. Appendix B shows the spectrograms for a synthesized utterance of an unseen test sentence using the two parametrizations. As evident from the spectograms, Klatt parameters sufficiently model the spectral aspects of speech. Perceptually, the speech is completely intelligible and listeners transcribed all the words in the sentence. There is, however, the 'processed' quality to the synthesis that is quite distinct from MCEP based synthesis.

Predicted parameters can be post-processed based on the identity of the underlying phoneme by merely increasing or decreasing its value as appropriate for the task (e.g, to make output speech sound more 'nasal' or 'bursty'). This flexibility is unique to knowledge-based parametrizations, like the one we presented in this work. For the example reported, we did not do any post-processing on the predicted vectors except for smoothing. We are currently working on objective comparisons of the two parametrizations.

4. Discussion

The synthesis quality is fully understandable but has a "processed" quality to it. Interestingly although the output speech clearly contains the speaker identity of rms, the quality is also sounds like "DECtalk". Thus it is clear that the Klatt parameters introduce a particular type of speech distortion due to the parametric and resynthesis techniques.

We are aware that expertly highly-tuned Klatt parameters can produce synthesis quality far beyond the quality that raw text to speech can give, and hoped that our techniques might help improve text-to-speech quality for Klatt-like formant synthesis. But even our resynthesis quality is closer to TTS output quality than we hoped. The resynthesis quality is not as good as we hoped, suggesting there is still more work in improving both the extraction of parameters and the method of resynthesis. Ultimately in statistical speech synthesis there are three constraints on the appropriateness of a set of parameters. First they must be automatically derivable from data bases of natural speech; second the parameters must give rise to high quality resynthesis; and finally the parameters must be predictable from text.

5. Conclusion

In this paper, we revisit the classical knowledge based parametrization of speech for use within the framework of statistical parametric speech synthesis. We present techniques for extraction of these parameters directly from speech data. Analytical results are presented for resynthesis and text based modeling/prediction of Klatt parameters. We intend to further improve our parameter extraction and vocoding algorithms. We are also investigating the use of Klatt-style parameters in a range of speech applications like speech recognition, speaker identification and voice conversion.

6. Acknowledgments

The first author is supported by the Fundação de Ciência e Tecnologia through the CMU/Portugal Program, a joint program between the Portuguese Government and Carnegie Mellon University.

7. References

[1] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," *Journal of the Acoustical Society of America*, 1980.

[2] J. Olive, A. Greenwood, and J. Coleman, *Acoustics of American English Speech: A Dynamic Approach*. Springer Verlag, 1993.

[3] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *ICASSP-96*, vol. 1, Atlanta, Georgia, 1996, pp. 373–376.

[4] H. Zen, K. Tokuda, and A. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1059–1064, 2009.

[5] J. Kominek, "TTS from zero: Building synthetic voices for new languages," Ph.D. dissertation, Carnegie Mellon University, 2009.

[6] T. Drugman, G. Wilfart, and T. Dutoit, "A deterministic plus stochastic model of the residual signal for improved parametric speech synthesis," in *Interspeech 09*, Brighton, UK, 2009.

[7] R. Maia, T. Toda, H. Zen, Y. Nankaku, and K. Tokuda, "Mixed excitation for HMM-based speech synthesis using residual modeling." in *ISCA Speech Synthesis Workshop 7*, Bonn, Germany, 2007.

[8] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigne, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f0 extraction: possible role of a repetitive structure in sounds," *Speech Communications*, vol. 27, pp. 187–207, 1999.

[9] Y.-J. Wu and K. Tokuda, "Minimum generation error training by using original spectrum as reference for log spectral distortion measure," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 0, pp. 4013–4016, 2009.

[10] J. Allen, S. Hunnicut, and D. Klatt, *Text-to-speech: The MITalk system*. Cambridge, UK.: Cambridge University Press, 1987.

[11] E. R. Laboratory, "Entropic signal processing system (esps)." [Online]. Available: //http://www.entropic.com/esps.html

[12] F. Metze, "Discriminative speaker adaptation using articulatory features," *Speech Communication*, vol. 49, no. 5, 2007.

[13] J. Kominek and A. Black, "The CMU ARCTIC speech databases for speech synthesis research," Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-LTI-03-177 `http://festvox.org/cmu_arctic/`, 2003.

[14] A. Black, "CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling," in *Interspeech 2006*, Pittsburgh, PA., 2006.

[15] N. Iles, J & Ing-Simmons, "Klatt: A Klatt-style speech synthesizer implemented in c," CMU Artificial Intelligence Repository, 1995.

A. A detailed list of Klatt Parameters

No.	Parameter	Description
1	f0	fundamental frequency (pitch) of the segment
2	av	Amplitude of voicing for the cascade branch in dB, Range 0-70
3	f1	First formant frequency in the range 200-1300 Hz
4	b1	Cascade branch bandwidth of first formant in the range 40-1000 Hz
5	f2	Second formant frequency in the range 550 - 3000 Hz
6	b2	Cascade branch bandwidth of second formant in the range 40-1000 Hz
7	f3	Third formant frequency in the range 1200-4999 Hz
8	b3	Cascade branch bandwidth of third formant in the range 40-1000 Hz
9	f4	Fourth formant frequency in 1200-4999 Hz
10	b4	Cascade branch bandwidth of fourth formant in the range 40-1000 Hz
11	f5	Fifth formant frequency in the range 1200-4999 Hz
12	b5	Cascade branch bandwidth of fifth formant in the range 40-1000 Hz
13	f6	Sixth formant frequency in the range 1200-4999 Hz
14	b6	Cascade branch bandwidth of sixth formant in the range 40-2000 Hz
15	fnz	Frequency of the nasal zero in the range 248-528 Hz (cascade branch only)
16	bnz	Bandwidth of the nasal zero in the range 40-1000 Hz (cascade branch only)
17	Fnp	(default 200) Frequency of the nasal pole in the range 248-528 Hz (constant)
18	Bnp	(default 30) Bandwidth of the nasal pole in the range 40-1000 Hz (constant)
19	asp	Amplitude of aspiration 0-70 dB
20	Kopen	(default 40) Open quotient of voicing waveform, range 0-60
21	Aturb	(default 0) Amplitude of turbulence 0-80 dB, simulates breathy quality
22	tilt	(default 0) Voicing spectral tilt in dB, range 0-24
23	af	Amplitude of frication in dB, range 0-80 (parallel branch)
24	Skew	(default 0) Spectral Skew - skewness of alternate periods, range 0-40
25	a1	Amplitude of first formant in the parallel branch, in 0-80 dB
26	b1p	Bandwidth of the first formant in the parallel branch, in Hz
27	a2	Amplitude of parallel branch second formant
28	b2p	Bandwidth of parallel branch second formant
29	a3	Amplitude of parallel branch third formant
30	b3p	Bandwidth of parallel branch third formant
31	a4	Amplitude of parallel branch fourth formant
32	b4p	Bandwidth of parallel branch fourth formant
33	a5	Amplitude of parallel branch fifth formant
34	b5p	Bandwidth of parallel branch fifth formant
35	a6	Amplitude of parallel branch sixth formant
36	b6p	Bandwidth of parallel branch sixth formant
37	anp	Amplitude of the parallel branch nasal formant
38	ab	Amplitude of bypass frication in dB, 0-80.
39	avp	Amplitude of voicing for the parallel branch, 0-70 dB.
40	Gain	(default 80) Overall gain in dB range 0-80.

B. A comparison of Klatt/MCEP parameter based TTS

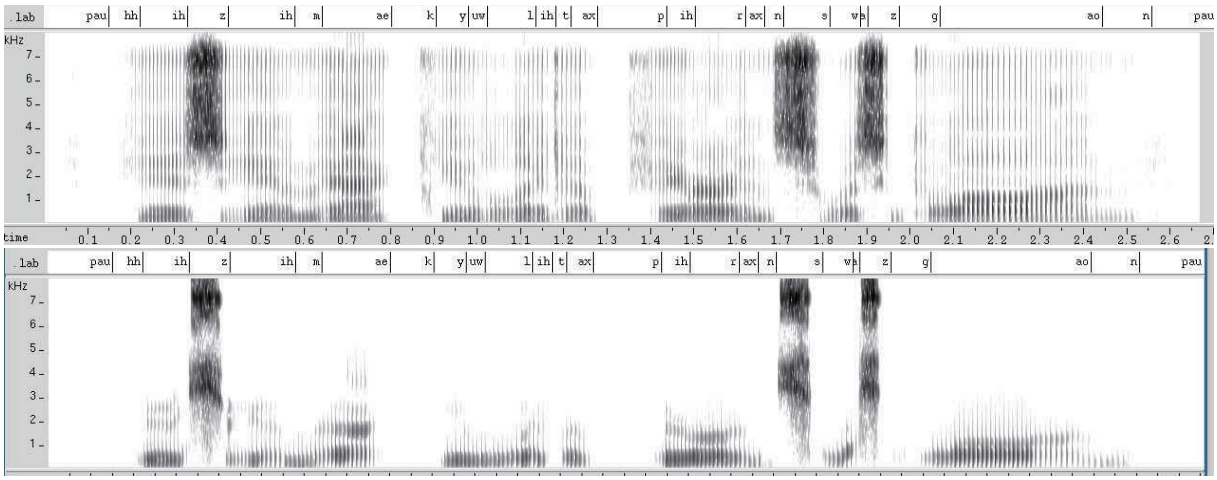


Figure 4: Synthesized example from MCEP(above) and Klatt(below) parameters for sentence *"His immaculate appearance was gone."*