Proceedings of the International Joint Conference IBERAMIA/SBIA/SBRN 2006 - 4th Workshop in Information and Human Language Technology (TIL'2006), Ribeirão Preto, Brazil, October 23–28, 2006. CD-ROM. ISBN 85-87837-11-7

Automatically Estimating the Input Parameters of Formant-Based Speech Synthesizers

Aline Figueiredo[1], Tales Imbiriba[1], Edward Bruckert[2] and Aldebaro Klautau[1]

[1] Signal Processing Laboratory (LaPS) - Universidade Federal do Pará
DEEC, UFPA, 66075-110, Belem, Para, Brazil. http://www.laps.ufpa.br
[2] Fonix Corporation
180 W. Election Road, Suite 200, Draper, UT 84020, USA. http://www.fonix.com

Abstract. The paper presents preliminary results of a new framework for automatically extracting the input parameters of a class of synthesizers. The framework allows to speed up the process of utterance copy (or speech imitation), where one has to find the model parameters that lead to a synthesized speech sounding close enough to the natural target speech. The results confirm that the error surface is non-convex with many local minimal, making the task a hard-to-solve inverse problem. Therefore, the framework is based on genetic algorithms, which is a robust non-linear optimization technique. The work also discusses the use of speech analysis toolkits, such as Praat and Snack, to improve convergence.

1 Introduction

This paper presents preliminary experimental results of a framework based on evolutionary computing for estimating the input parameters of the so-called *formant*-based speech synthesizers. More especifically, two synthesizers are studied: **Klatt's** [1, 2] and **HLsyn** [3, 4].

The framework aims to speedup the process of *utterance copy*, where one has to find the model parameters that lead to a synthesized speech sounding close enough to the natural target (or reference) speech. The proposed solution is centered in evolutionary computing; more specifically, in genetic algorithms (GA) [5]. The task can be cast as a hard *inverse problem*, because it is not an easy task to extract the desired parameters automatically (see, e.g., [6]). Because of that, in spite of recent efforts [7–9], most studies using parametric synthesizers adopt a relatively time-consuming process (see, e.g., [2]) for utterance copy and end up using short speech segments (words or short sentences). The possibility of automatically analyzing speech corpora is very important to increase the knowledge about phonetic and phonological aspects of specific dialects, endangered languages, spontaneous speech, etc.

The work is organized as follows. Section 2 gives information about the two adopted synthesizers. In Section 3 we describe the proposed approach while Section 4 presents preliminary simulation results, followed by the conclusions in Section 5.

2 The Synthesizers

The speech synthesizer is the back end of *text-to-speech* (TTS) systems [10–13]. For example, in [14, 15], Klatt-based synthesizers were used in TTS systems for the Romanian and Brazilian Portuguese languages, respectively. Synthesizers are also useful in speech analysis, such as in experiments about perception and production [16]. For example, in [17], an *analysis-by-synthesis* approach using Klatt's was adopted to accurately synthesize electrolarynx speech.

The Klatt's synthesizer is called a formant synthesizer because some of its most important parameters are the formant frequencies (the resonance frequencies of the vocal tract, see, e.g., [18]). HLsyn is another parametric synthesizer, which runs on top of Klatt's, and is an hybrid of the formant and articulatory approaches. It should be noticed that these two were chosen due to their widespread use, but the framework is valid for other parametric synthesizers as well (e.g., the ones described in [19, 12]).

Parametric synthesis has been adopted in commercial products such as the DECTalk (see [20] for demos). However, nowadays, most commercial TTS systems adopt *concatenative* synthesis [13], which relies on algorithms such as PSOLA [21] and MBROLA [22] that modify and concatenate previously stored segments of speech to generate the synthesized speech.

When speech technologists shifted from formant-based to concatenative synthesis, a gap was created in some areas of speech sciences, especially linguistics. In spite of not being competitive with concatenative techniques for developing some commercial applications, formant-based synthesis is very important in many speech studies. Most parameters of a formant synthesizer are closely related to physical parameters and have a high degree of interpretability. This is essential, for example, in studies of the acoustic correlates of voice quality, such as male/female voice conversion, and the simulation of breathiness, roughness, and vocal fry [19]. Unfortunately, after loosing most of the economical appeal to concatenative techniques, the number of research efforts on developing automatic tools for dealing with formant synthesizers is currently very limited. This paper aims to reduce this gap.

2.1 Klatt's

There are many versions of the Klatt's synthesizer. The one called KLSYN [2] had its source code published in [1] (in Fortran). Later [2], Dennis Klatt and his daughter presented an improved version called KLSYN88, which is currently commercialized by Sensimetrics (www.sens.com). In the early Nineties, a C version of KLSYN was posted in the comp.speech USENET group. Jon Iles rewrote it in C++ and called it Object Formant Synthesizer (OFS). Some of the differences between KLSYN and KLSYN88 are discussed in [2]. With respect to source-filter modeling of speech production, KLSYN88 has three "sources": impulsive, "Klatt's natural" and Liljencrants-Fant.

Basically, the Klatt synthesizer works as follows. For each frame (its duration is set by the user, often in the range from 5 to 10 milliseconds), a new set of parameters drives the synthesizer. Some parameters are used to generate the excitation signal (mimicking the influence of the air flow), while others are used to set the filters that shape the speech spectrum (mimicking the action of the vocal tract organs).

Table 1 lists the KLSYN88 parameters used in this work. For their complete description, the reader is referred to [1,2]). The parameters that do not vary over time are not listed here. There are three filter-banks in KLSYN88: one in which the resonators are in series (*cascade*) and two in which they are in parallel. As conventionally done, the voicing-excited parallel bank was not used (the amplitude parameters ANV, ATV, A1V, A2V,. . ., A6V were assumed to be zero). The number of cascaded resonators was NF=5. The glottal source was the KLGOTT88 (SQ was not used).

Table 1. KLSYN88 parameters used in this work.

Description	ID	Range
Fundamental frequency (tenths of Hz)	F0	[0, 5000]
Amplitude of voicing (dB)	AV	[0, 80]
Open quotient	OQ, KOPEN	[10, 99]
Extra tilt of voicing spectrum	TL, TILT	[0, 41]
Flutter - random fluctuation in F0 (%)	FL	[0, 100]
Diplophonia (%)	DI	[0, 100]
Amplitude of aspiration	AH, ASP	[0, 80]
Amplitude of frication	AF	[0, 80]
Formant F1	F1	[180, 1300]
F1 bandwidth	B1	[30, 1000]
F1 change - open portion of a period	DF1, df	[0, 100]
B1 change - open portion of a period	DB1, db	[0, 400]
Formant F2	F2	[550, 3000]
F2 bandwidth	B2	[40, 1000]
Formant F3	F3	[1200, 4800]
F3 bandwidth	B3	[60, 1000]
Formant F4	F4	[2400, 4990]
F4 bandwidth	B4	[100, 1000]
Formant F5	F5	[3000, 4990]
F5 bandwidth	B5	[100,1500]
Formant F6	F6	[3000, 4990]
F6 bandwidth	B6	[100,4000]
Nasal pole frequency	FNP, fp	[180, 500]
Nasal pole bandwidth	BNP, bp	[40, 1000]
Nasal zero frequency	FNZ, fz	[180, 800]
Nasal zero bandwidth	BNZ, bz	[40, 1000]
Frequency of the tracheal pole	FTP	[300, 3000]
Bandwidth of the tracheal pole	BTP	[40, 1000]
Frequency of the tracheal zero	FTZ	[300, 3000]
Bandwidth of the tracheal zero	BTZ	[40, 2000]
Amplitude of frication-excited parallel formants (x=2 to 6)	AxF	[0, 80]
Bypass path amplitude	AB	[0, 80]
Bandwidth of frication-excited parallel formants	B2F	[40, 1000]
Bandwidth of frication-excited parallel formants	B3F	[60, 1000]
Bandwidth of frication-excited parallel formants	B4F	[100, 1000]
Bandwidth of frication-excited parallel formants	B5F	[100, 1500]
Bandwidth of frication-excited parallel formants	B6F	[100, 4000]

2.2 HLsyn

HLsyn incorporates specializard knowledge about acoustic and articulatory phonetics. Its purpose is to work as a *wrapper* (or an upper layer) to the Klatt synthesizer and achieve a reduction on the number of input parameters. Hence, for synthesizing each *frame* of speech, HLsyn requires 13 parameters, which are then converted to the 48 parameters used by Klatt's. The Klatt's synthesizer is invoked and actually generates the speech samples. Besides reducing the number of parameters, HLsyn imposes restrictions that help avoiding non-feasible solutions. That is, a given set of 48 Klatt's parameters can eventually match the desired sound, but they may be physically unfeasible [3]. Table 2 lists the HLsyn parameters used in this work and their range.

Table 2. List of HLsyn parameters and the range adopted in the simulations.

ID	Description	Unity	Range
f1	First natural (formant) frequency	Hz	[180, 1300]
f2	Second natural (formant) frequency	Hz	[550, 3000]
f3	Third natural (formant) frequency	Hz	[1200, 4800]
f4	Fourth natural (formant) frequency	Hz	[2400, 4990]
f0	Fundamental frequency (known as "pitch")	Hz	[0, 500]
ag	Average area of glottal (membranous portion)	mm^2	0.01 × [0, 4000]
ap	Area of the posterior glottal opening	mm^2	0.01 × [0, 1000]
ps	Subglottal pressure	cm H$_2$O	0.01 × [0, 1000]
al	Cross-sectional area of constriction at the lips	mm^2	0.1 × [0, 1000]
ab	Cross-sectional area of tongue-blade constriction	mm^2	0.1 × [0, 1000]
an	Cross-sectional area of velopharyngeal port	mm^2	0.1 × [0, 1000]
ue	Rate of increase of vocal-tract volume	cm^3/s	[0,1000]
dc	Change in vocal-fold or wall compliances	%	[-150, 150]

3 Automatically Learning the Input Parameters

The approach described in this section tries to solve the following problem: given an utterance to be synthesized, find for each frame a sensible set of parameters to drive the synthesizers. The number of parameters and their dynamic range make an exhaustive search unfeasible. GA [5] was adopted as the main learning strategy. One key point when posing a new optimization problem is to choose the objective (or fitness) function. The following subsections discuss the options considered in this work.

3.1 Figures of merit - Fitness functions

One of the fitness functions used in this work is the *mean square error* (MSE), calculated between the target $h(n)$ and synthesized $s(n)$ waveforms. The main reason for considering MSE is its simplicity. On the other hand, it is well-known that MSE does not reflect the perceived quality of speech signals.

Another adopted fitness function was the *spectral distortion* (SD) between the target spectrum $H(f)$ and the synthesized spectrum $S(f)$, which is given by

$$SD = \sqrt{\frac{1}{f_2 - f_1} \int_{f_1}^{f_2} \left[20 \log_{10} \frac{|H(f)|}{|S(f)|} \right]^2 df}$$

and calculated through a fast Fourier transform (FFT) routine.

The spectral distortion is widely adopted in speech coding for quantizing the filter in the source-filter model (also known as LPC filter, after the linear prediction algorithm used for its estimation). The related concept of transparent coding [23], which means that $H(f)$ and $S(f)$ are close enough to be perceptually undistinguishable, is based on the following statistics of SD: a) the average SD (among all frames) is less or equal than 1 dB, b) there should be no frames for which $SD > 4$ dB, and c) the number of frames for which $2 \leq SD \leq 4$ dB is less than 2%. This empirical performance goal was modified (to be less restringing) and used as part of the GA termination procedure.

3.2 Architecture

Among many possible ways of using GA for solving the posed problem, three architectures were studied:

- *Intraframe*: for each speech frame, we setup a conventional GA problem. For example, if the target utterance is 1 second long and the frame duration is 10 milliseconds (ms), 100 GA problems are solved in a completely independent way from each other, with the populations randomly initialized for each frame. The random numbers are uniformly distributed on the range specified for each parameter.
- *Interframe*: a fraction F of the initial population for frame t is obtained by copying some of the best individuals from the previous frame $t - 1$. The remaining $1 - F$ fraction is randomly (and uniformly) initialized. This architecture takes in account that speech varies smoothly, specially for stationary sounds such as vowels.
- *Knowledge-based*: the initial population for frame t depends not only on previous frames, but also on parameter values estimated through speech analysis algorithms such as the ones adopted in Praat and Snack.

Note the knowledge-based architecture has many degrees of freedom. Some possibilities, not implemented yet, are briefly discussed here. For example, one can take in account the kind of sound to be synthesized in each frame. Such information can be obtained from an algorithm for phonetic segmentation (see, e.g., [24]) and used both when doing the speech analysis and GA optimization. While it is quite difficult to get good accuracy with a large vocabulary continuous speech recognition as the one used in [24], much better phonetic segmentations can be obtained when the orthographic transcription is known a priori (by using the so-called *forced alignement*). It is fair to say that this is the case with utterance copy experiments and, consequentely, one can assume a reasonably accurate phonetic transcription could be obtained (assuming the speech signal is not too noisy).

4 Results

The first stage of the work was to evaluate the error surface for the optimization problem. The motivation was to get insight about the difficulty an algorithm would face. This section also presents results for synthesizing vowels with GA using the intraframe architecture. These results are followed by figures that illustrate how speech analysis can be used in the knowledge-based architecture. The fitness function used in these experiments was the spectral distortion, which outperformed the MSE.

4.1 Studying the error surface

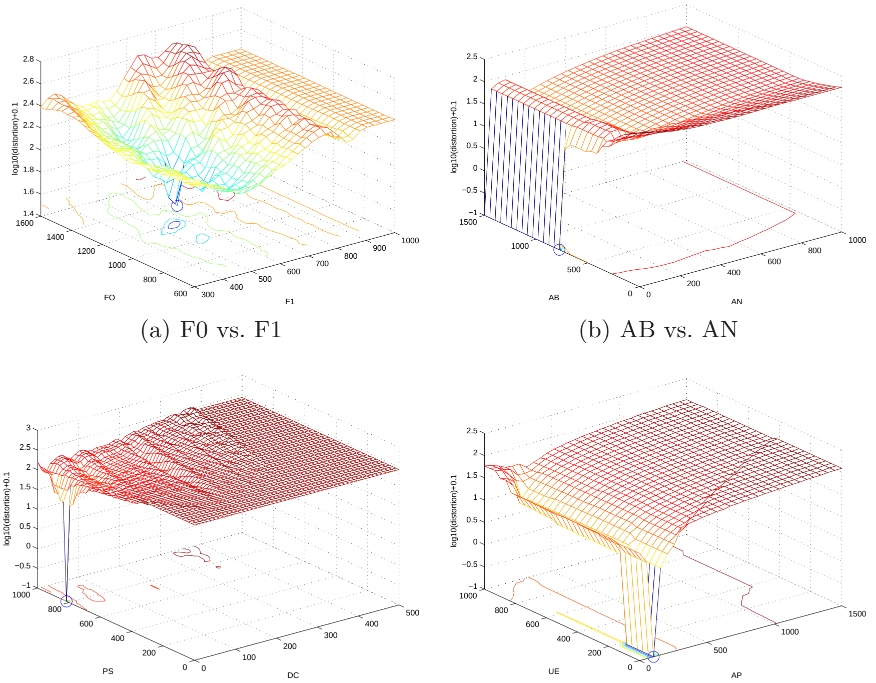
The experiment used artificial vowels that were synthesed by HLsyn itself. Because the "right" parameters were known, the experiment can be easily controlled. First, a set of 13 HLsyn parameters were used to produce few frames of a stationary sound. Later, new versions of this sound were obtained by varying a couple of parameters. The task of the GA algorithm was to find the correct pair of values corresponding to the modified parameters. This allows to easily visualize the error surface as shown in Figure 1, which describes results of four different pairs of parameters.

Figure 1(a) shows the error surface when the correct values of F0 and F1 are 1200 and 500, respectively. These values correspond to a fundamental frequency F0 of 120 Hz and a first formant F1 of 500 Hz. One can see that the curve has many local minima. The other three pairs also show a similar situation. From these four graphs, one can note that the sensitivity to each parameter also varies considerably.

4.2 Synthetic Vowels

This subsection shows some results when using GA to obtain the parameters of vowels generated by the Klatt synthesizer. The motivation was to study the GA convergence and tune its parameters, such as the mutation probability. Figure 2 illustrates the convergence when the studied pair of parameters were the first two formants F1 and F2 with correct values of 500 and 1000 Hz, respectively. The GA population was 50 individuals. The figures indicate that, in this case, around 50 iterations were sufficient for having individuals close to the optimal values. After 100 iterations, all individuals were very close to the optimum point.

The next subsection discusses more ellaborated experiments with whole sentences.



(a) F0 vs. F1

(b) AB vs. AN

(c) PS vs. DC

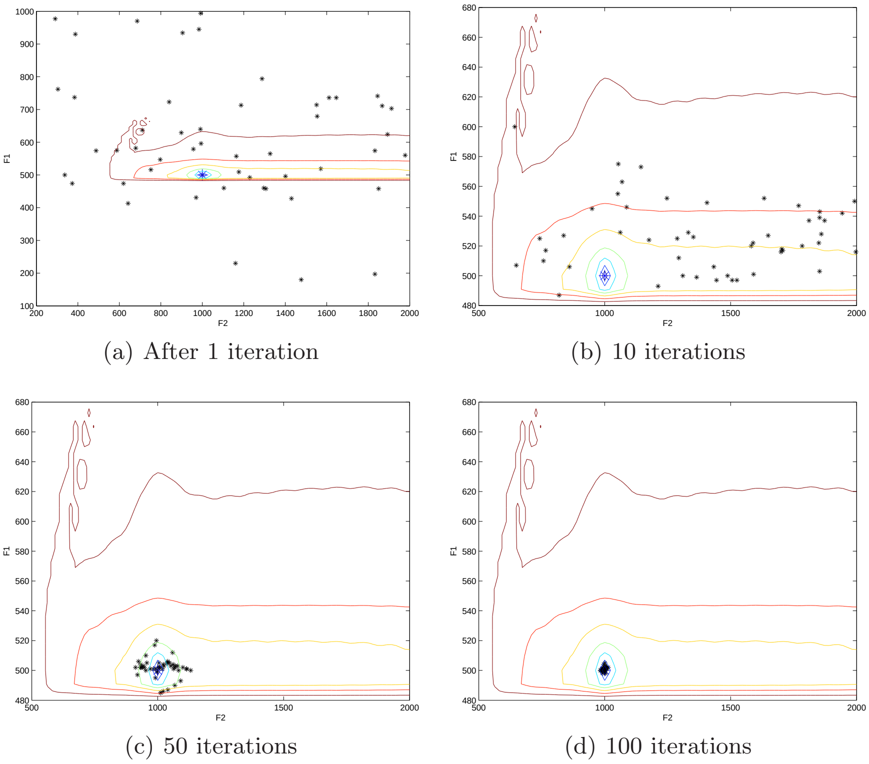
(d) UE vs. AP

Fig. 1. Error surfaces and contours of four pairs of parameters. The curves illustrate the local minima and varied sensitivity of parameters.

4.3 Synthetic Sentences

Before studying the synthesis of natural (human-generated) sounds, the problem to be circumvented is the computational cost of the optimization procedure. It takes too long for the synthesis of a whole sentence. Each frame is a GA problem to be solved, and even using the interframe architecture, the time is still too long.

The approach adopted was to invest on the knowledge-based architecture, using speech analysis algorithms to estimate parameters. For extracting these speech parameters two widely known speech analysis toolkits were used: Praat (www.praat.org) and Snack (www.speech.kth.se/snack). Both provide support to scripts. We used these tools to estimate formants (central frequencies and bandwidths) and F0 (fundamental frequency). Given the estimated parameters for a specific frame, these parameters become the means of Gaussians that are used to randomly initialize part of the GA population (the other part comes from a percentage of the population from the previous GA problem, as in the interframe architecture). The variances of these Gaussians were empirically obtained.



(a) After 1 iteration

(b) 10 iterations

(c) 50 iterations

(d) 100 iterations

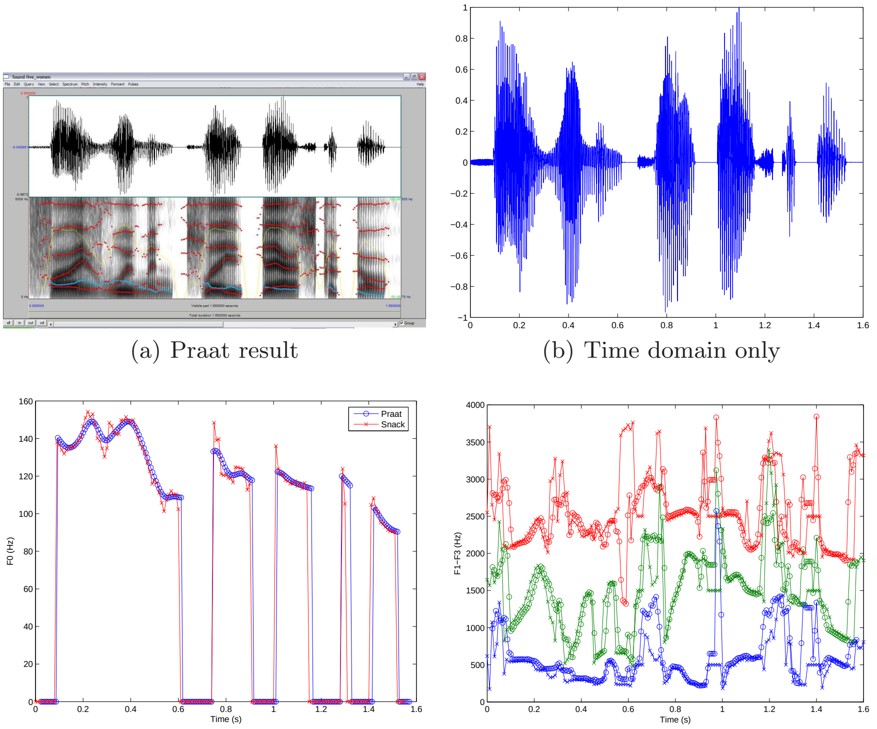
Fig. 2. Error contour and location of the GA individuals when finding the optimum F1 (y-axis) and F2 (x-axis) values for a synthetic vowel. Panel (a) has different axes range. The optimal values F1=500 and F2=1000 Hz are indicated by a blue $*$).

The use of speech analysis was quite effective to speed up convergence when dealing with F0 and the formants. Figure 3 shows that Praat and Snack achieve a reasonable result and basically agree in their estimations. The utterance was "five women played basketball", generated by HLsyn for a female voice (available at *www.sens.com/hlsyn_overview.htm*).

The next stage, after tuning the system with synthetic sentences, is to conduct a formal evaluation using the TIMIT corpus, which is the most popular among the corpora distributed by the LDC (*www.ldc.upenn.edu*). Currently, the main difficulty is that there are no ready-to-use routines for estimating the other parameters of the synthesizers and the synthesis of a long utterance still takes considerable time.

5 CONCLUSIONS

A new framework for automatically extracting the input parameters of the Klatt and HLsyn synthesizers was presented. A formal evaluation is still required, but



(a) Praat result

(b) Time domain only

(c) F0 tracks by Praat and Snack

(d) Formant tracks by Praat and Snack

Fig. 3. Whole sentence analyzed by Praat and Snack.

some preliminary results with synthetic sounds illustrated the problem characteristics (e.g., error surfaces) and possibile solutions. Currently, new algorithms for estimating parameters through speech analysis are under development. The idea is to split the task into two: speech analysis to obtain key parameters, followed by tuning of these and the rest of the parameters through evolutionary computing (GA, particle swarm, etc.). The final product of this research will be a tool for speech analysis, helpful to speech therapists, phoneticians and other professionals in related areas.

References

1. Klatt, D.: Software for a cascade / parallel formant synthesizer. Journal of the Acoustical Society of America **67** (1980) 971–95
2. Klatt, D., Klatt, L.: Analysis, synthesis, and perception of voice quality variations among female and male speakers. Journal of the Acoustical Society of America **87** (1990) 820–57
3. Bickley, C.A., Stevens, K.N., Williams, D.R.: Control of Klatt speech synthesizer with high-level parameters. The Journal of the Acoustical Society of America **91** (1992) 2442

4. Hansona, H.M., Stevens, K.N.: A quasiarticulatory approach to controlling acoustic source parameters in a Klatt-type formant synthesizer using HLsyn. J. Acoust. Soc. Am. **112** (2002) 1158–82
5. Langdon, W.B., Poli, R.: Foundations of Genetic Programming. Springer-Verlag (2002)
6. Ding, W., Campbell, N., Higuchi, N., Kasuya, H.: Fast and robust joint estimation of vocal tract and voice source parameters. In: IEEE ICASSP. (1997) 1291–4
7. Breidegard, B., Balkenius, C.: Speech development by imitation. In: Proceedings Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems. (2003) 57–64
8. Heid, S., Hawkins, S.: Procsy: A hybrid approach to high-quality formant synthesis using HLsyn. In: Proceedings of the 3rd ESCA/COCOSDA. (1998)
9. PROCSY: http://kiri.ling.cam.ac.uk/procsy/ (Visited in January, 2006)
10. Allen, J., Hunnicutt, M.S., Klatt, D.H., Armstrong, R.C., Pisoni, D.B.: From text to speech: The MITalk system. Cambridge University Press (1987)
11. van Santen, J., Hirschberg, J., Olive, J., Sproat, R., eds.: Progress in Speech Synthesis. Springer-Verlag, New York (1996)
12. Holmes, J.N., Holmes, W.J.: Speech Synthesis and Recognition. T & F STM (2001)
13. Dutoit, T.: An Introduction to Text-To-Speech Synthesis. Kluwer (2001)
14. Jitca, D., Apopei, V.: Conclusions on analysis and synthesis of large semivocalic formantic transitions. In: International Symposium on Signals, Circuits and Systems. (2003) 177 – 180
15. De C.T. Gomes, L., Nagle, E., Chiquito, J.: Text-to-speech conversion system for Brazilian Portuguese using a formant-based synthesis technique. In: SBT/IEEE International Telecommunications Symposium. (1998) 219–224
16. Rutledge, J., Cummings, K., Lambert, D., Clements, M.: Synthesizing styled speech using the Klatt synthesizer. In: International Conference on Acoustics, Speech, and Signal Processing (ICASSP). (1995) 648 – 651
17. Saikachi, Y., Hillman, R., Stevens, K.: Analysis by synthesis of electrolarynx speech. J. Acoust. Soc. Am. **118** (2005) 1965
18. Klautau, A.: Classification of Peterson and Barney's vowels using Weka. Technical report, UFPA, *http://www.laps.ufpa.br/aldebaro/papers* (2002)
19. Pinto, N., Childers, D., Lalwani, A.: Formant speech synthesis: Improving production quality. IEEE Transactions on Acoustics, Speech and Signal Processing **37** (1989) 1870–1887
20. Gilbert, J., Fosler, E.: http://www.icsi.berkeley.edu/eecs225d/klatt.html (Visited in December, 2005)
21. Valbret, H., Moulines, E., Tubach, J.P.: Voice transformation using PSOLA technique. speech **11**(2-3) (1992) 189–194
22. Dutoit, T., Pagel, V., Pierret, N., Bataille, F., der Vrecken, O.V.: The MBROLA project: Towards a set of high quality speech synthesizers free of use for non commercial purposes. In: Proc. ICSLP '96. Volume 3., Philadelphia, PA (1996) 1393–1396
23. Paliwal, K., Atal, B.: Efficient vector quantization of LPC parameters at 24 bits/frame. In: ICASSP. (1991) 661–4
24. Hosom, J.P.: Automatic Time Alignment of Phonemes Using Acoustic-Phonetic Information. PhD thesis, Oregon Graduate Institute of Science and Technology (2000)