

sHTS : A Streaming Architecture for Statistical Parametric Speech Synthesis

Maria Astrinaki¹, Onur Babacan¹, Nicolas d’Alessandro², Thierry Dutoit¹

¹NUMEDIART Institute for New Media Art Technology - University of Mons (Belgium)

²Media and Graphics Interdisciplinary Centre - University of British Columbia (Canada)

{maria.astrinaki, onur.babacan, thierry.dutoit}@umons.ac.be, nda@magic.ubc.ca

Abstract

In this paper, we present a prototype for real-time speech synthesis. Statistical parametric speech synthesis is a relatively new approach to speech synthesis. Hidden Markov model based speech synthesis, one of the techniques in this approach, has been demonstrated to be very effective in synthesizing high quality, natural and expressive speech. In contrast to unit selection techniques, HMM-based speech synthesis provides high flexibility as a speech production model, with a small database footprint. In this work we modified the publicly available HTS engine to establish a streaming architecture, called streaming-HTS or sHTS, which provides us with a basis for further research for a future fully real-time speech synthesis system. Quantitative evaluations of the system showed that the degradation of speech quality in sHTS is small with reference to HTS. These results were supported by subjective evaluation, which confirmed that HTS and sHTS can hardly be distinguished.

Keywords: HMM, speech synthesis, statistical parametric speech synthesis, real-time, performative, streaming

1. Introduction

While the the mainstream application area of speech synthesis is Text-To-Speech (TTS) synthesis, real-time interaction with speech synthesis systems has received growing interest these last years.

At University of MONS, the research activity of the NUMEDIART Institute for digital art technology aims at developing audio, video, and sensor-based systems for enhancing the interaction between artists (or their installations) and their public. One research axis is devoted to digital luthery, i.e. on the design of new digital instruments, or of augmented instruments, i.e. instruments built by adding digital interfaces on existing instruments.

This work broadly falls into the recently-shaping “performative speech synthesis” field. Previous research work

in this area from our group include MaxMBROLA [1], a diphone-based synthesizer ported to Max/MSP and used in an avant-garde opera performed by robots (Armageddon, created by French musical creation center ArtZoyd [2]). We also developed RAMCESS [3], a real-time singing synthesizer using an analysis/resynthesis method based on mixed-phase speech modeling, and the HandSketch [4], a bi-manual controller designed from the ground up for digital instruments.

For the next steps in our research, we want to benefit from the recent outgrowth of statistical parametric synthesis. As a matter of fact, in contrast with concurrent concatenative synthesis approaches which require the recording of very large databases for covering the wide variations found in expressive speech, statistical parametric speech synthesis systems, train statistical models with various features using speech databases, and generate speech from the trained models. A prominent method in this approach employs hidden Markov models (HMMs). An implementation of an HMM-based speech synthesis, HTS [6] is readily available and widely used. HTS features small database size, easily changeable voice characteristics and models a large amount of contextual factors. It produces highly intelligible speech.

To the best of our knowledge, a performative HTS-based speech synthesis system has not been attempted yet. Our aim in this work is thus to restructure the HTS code into a real-time back-end system, and evaluate the new systems performance in terms of segmental quality.

In this paper we give an overview of HTS as a comprehensive implementation of HMM-based speech synthesis. We continue with the description of our streaming architecture for HTS. Finally results from objective and subjective evaluations are presented.

2. HMM-Speech Based Synthesis with HTS

As computer technology evolves and the power and resources of computer increases, the task of building more and more natural and intelligible synthetic voices progresses rapidly. The need for smaller synthesizer footprints and more flexible control of synthesis has brought researchers to envision other ways of using the knowledge contained in the database. The so-called statistical parametric speech synthesis does not use real speech samples at synthesis runtime.

Instead, the pre-recorded database is analyzed and various production features are extracted, e.g. spectral envelopes,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

p3s 2011, March 14-15, 2011, Vancouver, BC, CA.

Copyright remains with the author(s).

fundamental frequency, duration of the phonemes, as well as as first and second time derivatives of these features [7].

Then the extracted features are used to train a statistical model. In HTS, for instance, each phoneme is typically modeled by a 5-states HMM, and the multidimensional Gaussian pdf associated to each state is made by the concurrent training of a context-dependent decision tree. At synthesis time, the HMM models of each phoneme in the target sentence are concatenated, and synthetic speech spectral, pitch, and duration trajectories are generated from HMMs themselves, based on a maximum likelihood criterion [8]. Finally the statistical model is used for generating trajectories, regarding a given target. These trajectories are used to control a voice production model in order to synthesize the speech [9].

As a result, the overall footprint of the system is fairly small (around 1MB per voice), as just the trained statistical model is need to run the system. This makes HMM speech synthesis a perfect candidate for portable applications such as those targeted in digital lutherie. On the other side, the use of a production model rather than real waveforms introduce a small loss in segmental quality, but definitely not as much as in the early rule-based synthesis systems.

Figure 1 is a block diagram of the HMM-based speech synthesis system. The basic architecture of an HTS system consists of two parts, the training and the synthesis, that will be discussed in the next subsections.

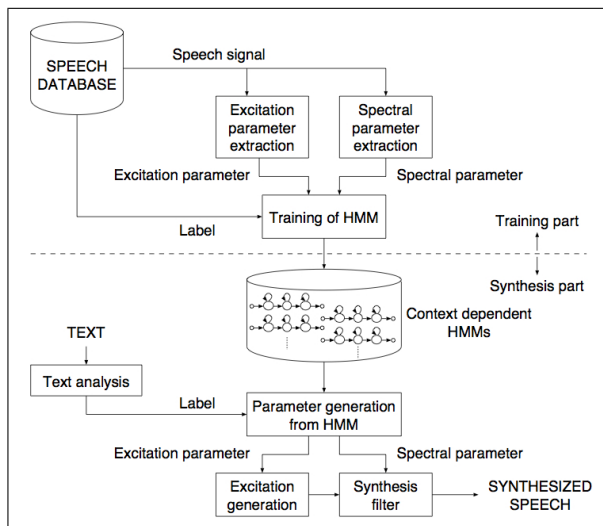


Figure 1. Block diagram of the HMM-based speech synthesis system: the training and synthesis parts [12].

2.1. Training Part

In the training part of HTS, both spectrum (mel-cepstral coefficients and their dynamic features) and excitation (log F0 and its dynamic features) parameters are extracted from the given database of natural speech and then are modeled by a Figure 2. Decision trees for context clustering [12] set of context-dependent HMMs. Notice that not only phonetic,

but also linguistic and prosodic context is taken into account. More specifically, the output vector of HMM consists of the mel-cepstral coefficient vector, including the zeroth coefficients, their delta and delta-delta coefficients and of the log fundamental frequency vector, its delta and delta-delta coefficients. In order to model speech in time, HMMs model the state duration densities by using a multivariate Gaussian distribution [10]. As it was described above, in order to handle the contextual factors, such as phone identity factors, stress or accent related factors that affect the targeted synthetic speech output, we use context-dependent HMMs. As the number of these factors increase, greater variety of prosodies, intonations, emotions and tones of voice become available, as well as different speaker individualities and speaking styles, leading to higher degrees of naturalness. On the other hand, increasing the number of factors increases the number of possible combinations exponentially. Thus, the model parameter estimation is not precise enough when the training data are sparse, i.e. not covering the entire contextual space. In HTS, this problem is solved by applying decision-tree based context clustering techniques [11]. As mentioned, magnitude spectrum, fundamental frequency and duration are modeled independently, therefore there is a different phonetic decision tree for each one, as illustrated in Figure 2.

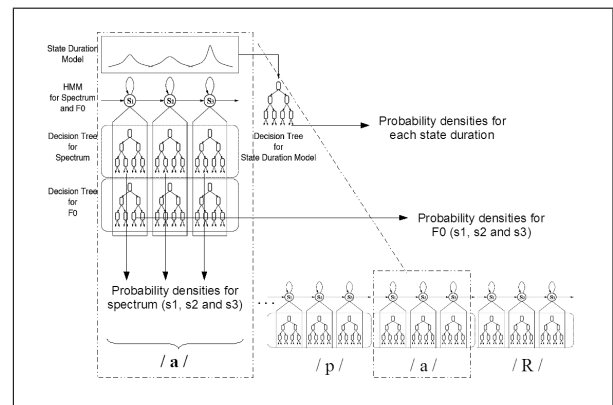


Figure 2. Decision trees for context clustering, [12].

2.2. Synthesis Part

In the synthesis part of HTS, the initial input is the target text to be transformed into synthetic speech. This text is parsed and mapped into a context-dependent phonetic label sequence, which is then used to construct an HMM sentence by concatenating the context-dependent HMMs according to this label sequence. When the sentence HMM is constructed, the sequences of spectrum and excitation parameters are generated, [13]. Finally, by using these generated parameters and a synthesis filter module, in this case an MLSA filter [14], a speech waveform is created. Based on the nature of statistical parametric speech synthesis, by modifying the HMM parameters we can obtain different voice

qualities of synthesized speech. It has been proved that by using speaker adaptation [16], speaker interpolation [17], or eigenvoice techniques [15] the voice characteristics can be modified.

2.3. Advantages of HTS

Compared to other systems, the main advantage of HTS is its flexibility. It is possible to change the voice characteristics, speaking styles, emotions and prosodic features simply by transforming the parameters of the model. Eigenvoice techniques [15] can be applied in order to create new voices. Interpolation techniques enable us to synthesize speech with various voice characteristics, speaking styles, and emotions that were not included in the natural speech database used for training our system and by applying multiple regression we can control these voice characteristics.

Additionally, HTS has a small number of tuning parameters; it is based on well defined statistical principles and it uses the source filter representation of speech, providing the flexibility to control and modify the magnitude spectrum, fundamental frequency and duration of speech are output separately. Furthermore, a small amount of training data is enough to create statistical parametric speech synthesis systems, which leads to a very small footprint. HTS has also a memory-efficient, low-delay speech parameter generation algorithm and a computationally-efficient speech synthesis filter.

All the characteristics mentioned above make HTS suitable not only for static embedded applications and mobile devices, such as Text-To-Speech applications (TTS) but also for real time performative speech synthesis, vocal expressivity and interaction.

3. Real-Time HTS : Streaming HTS

The concept of real-time can usually be understood in various ways. In computer science, it often means that the update of the output information of a given process occurs at the same rate as that of new input information. In other words, a real-time computer system may require some knowledge of the future of its input to produce its current output : the system is still causal, and its requirement for time look-ahead is implemented in a buffer, which implies that the process exhibits some delay. When we move to the field of performative realtime, real-time implies the possibility to use a process in a musical performance, in which delays must be strictly controlled. For the specific case of a speech synthesis system, for which coarticulation is known to have forward and backward effects (the latter being the most challenging for the delay issue), this implies that a given phoneme cannot be produced correctly if some future information is not known. The main question in this case comes down to: “how much of the future of a linguistic stream must be known to be able to produce natural

sounding speech?”. Statistical parametric synthesizers offer a wonderful testbed for answering such a question.

In order to make this possible, we needed to apply some modifications in the initial HTS offline run-time code. As a matter of fact, as described in the previous section, HTS synthesizes speech on the basis of complete sentences. Thus, in the offline architecture of HTS, all the contextual information from the full input is used. In other words, the intrinsic delay of this architecture if considered in a real-time perspective is one sentence.

3.1. Towards a streaming architecture of HTS

In direct contrast to the offline HTS approach and to the targeted real-time performative system, we achieved an intermediate step in this work. The streaming version of HTS we have tested here, also called sHTS, works on a phoneme-by-phoneme basis, i.e. it produces the samples for a given phoneme each time a new phoneme is sent to it as input. What is more, its delay (i.e. the number of future phonemes required for synthesizing the current phoneme) can be constrained (to one, two, or three phonemes, for our experiments). More specifically, we use an already trained HTS system, in which we change the way the HMM that models the sentence is constructed from the phonemes, and consequently the way the sequences of spectrum and excitation parameters are generated. We still use pre-computed context dependent phonetic labels that contain all the contextual information of a full text sentence, but all this information is not directly used for the feature synthesis step. Instead, for each given label, a sentence HMM is constructed, and for this single label the sequence of spectrum and excitation parameters are generated. The speech waveform synthesized from these parameters contains the synthetic speech that corresponds only to the given input phonetic label.

In this way, the sHTS parameter generation is achieved by computing many locally-optimal paths for the decision tree, which, when combined make a sub-optimal total path, instead of computing one total optimal path for the whole input sentence.

In the current implementation, even though the complete sentence context information is available, the system only has access to information corresponding to a single label at each iteration. In Figure 3 we present a simple block diagram of how labels are processed in the synthesis part of HTS and modifications that lead to sHTS.

3.2. sHTS Look-ahead

In a first experiment, we synthesized speech by processing the input phonetic labels one-by-one, using only the current label at each iteration (i.e., no information related to past labels, and no information about future labels either), and obtained the complete sentence waveform by simply concatenating the phoneme-level waveforms. As expected, this approach synthesizes phonemes as if they would be produced in isolation, i.e., surrounded by silence, and embark-

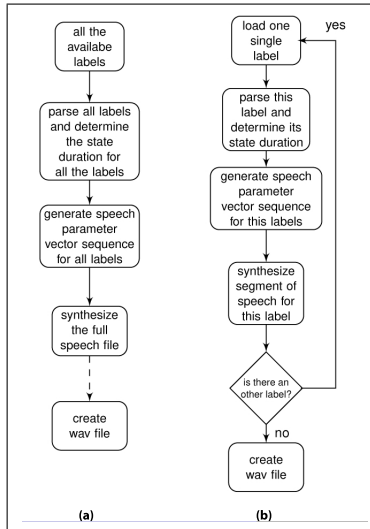


Figure 3. Block diagram of synthesis part of (a) offline HTS and (b) streaming HTS.

ing some kind of on- and offset. In order to overcome this problem we decided to introduce a small look-ahead buffer, whose length could be imposed at run-time. This greatly improved the output speech quality, even in the single label look-ahead case: the resulting synthesized speech sounds almost as natural and as intelligible as the synthesized speech from the offline version of HTS. Section 4 details the tests we made to assess this first impression.

4. Objective and Subjective Tests

For our tests we used the speaker-dependent training demo in English that is provided in [18]. Speech was sampled at 16 kHz and we used 5-state left-to-right HMMs. The following is a list of the contextual factors we took into account for training the English voice:

- phoneme
 - {preceding, current, succeeding} phonemes¹
 - position of current phoneme in current syllable
- syllable
 - number of phonemes at {preceding, current, succeeding} syllable
 - accent of {preceding, current, succeeding} syllable
 - stress of {preceding, current, succeeding} syllable
 - position of current syllable in current word
 - number of {preceding, succeeding} stressed syllables in current phrase

¹ The two preceding and the two succeeding phonemes of the current phoneme are taken into account

- number of {preceding, succeeding} accented syllables in current phrase
- number of syllables {from previous, to next} stressed syllable
- number of syllables {from previous, to next} accented syllable
- vowel within current syllable

- word
 - guess at part of speech of {preceding, current, succeeding} word
 - number of syllables in {preceding, current, succeeding} word
 - position of current word in current phrase
 - number of {preceding, succeeding} content words in current phrase
 - number of words {from previous, to next} content word
- phrase
 - number of syllables in preceding, current, succeeding phrase
 - position in major phrase
 - ToBI² endtone of current phrase
- utterance
 - number of syllables in current utterance

At run time, we use full label files, where each line is an alpha-numerical character sequence encoding all the information listed above for one phoneme, which is then input into the system. In HTS, all of the labels are used at once. In sHTS, one line (label) is input into the system at a time.

We synthesized speech produced by both HTS and sHTS, using all the pre-computed label files. We immediately observed that the state durations were different between the two approaches. This is because the durations are modeled more smoothly when all the information from the phonetic labels is provided to the system.

We forced both systems to use the same state durations (those provided by the HTS system), so as to be able to compare our results on a frame-by-frame basis. With this modification in place, we evaluated the quality of speech synthesized by sHTS by comparison to the original offline HTS synthetic speech, by using both objective and subjective measurements. More specifically, we evaluated the sHTS results using look-ahead buffers of one, two and three phonetic labels, which we will refer to as sHTS1, sHTS2, sHTS3, respectively.

² Tones and Break Indices

4.1. Objective Evaluation

In order to evaluate the distortion introduced by sHTS, we created a test database consisting of 40 sentences synthesized by HTS, sHTS1, sHTS2 and sHTS3.

We chose two metrics previously used in related research to evaluate test data.

The first metric we used is *mel-cepstral distortion* [19], a distance measure calculated between mel-cepstral coefficients. Mel-cepstral distortion is defined by

$$Mel - CD = \frac{10}{\ln(10)} \sqrt{2 \sum_{d=1}^D (mc_d^{(HTS)} - mc_d^{(sHTS)})^2} \quad (1)$$

where mc_d are mel-cepstral coefficients generated by the two different systems, and D is the mel-cepstral coefficient order. We applied this metric to the mel-cepstral coefficients generated by three test groups (sHTS1, sHTS2, sHTS3) and the results are presented in Table 1.

Table 1. Mean mel-cepstral distortion (Mel-CD) and 95% confidence intervals between HTS and sHTS with one, two and three future label buffers, in dB.

Mel-CD (dB)	Male Voice	Female Voice
sHTS1	2.76 ± 0.31	2.49 ± 0.33
sHTS2	2.68 ± 0.32	2.40 ± 0.34
sHTS3	2.63 ± 0.32	2.38 ± 0.34

The second metric we used is spectral distortion (SD), [20], which is defined by

$$SD_n^2 = \frac{20^2}{2\pi} \int_{-\pi}^{\pi} \left(\log_{10}|H_n(\omega)| - \log_{10}|\tilde{H}_n(\omega)| \right)^2 d\omega \quad (2)$$

where $H_n(\omega)$ are short-time Fourier transforms of corresponding (n -th) time-windowed frames from HTS and sHTS synthesis waveforms.

We applied discrete-time analogs of SD to our three test groups (sHTS1, sHTS2, sHTS3) and the results are presented in Table 2.

Table 2. Mean spectral distortion (SD) and 95% confidence intervals between HTS and sHTS with one, two and three future label buffers, in dB.

SD (dB)	Male Voice	Female Voice
sHTS1	0.88 ± 0.12	1.21 ± 0.19
sHTS2	0.75 ± 0.10	1.19 ± 0.21
sHTS3	0.73 ± 0.10	1.20 ± 0.22

Both methods show that as the buffer size is decreased, the degradation increases as expected, albeit not significantly.

This shows that using only one look-ahead phonetic label (similar to what is essentially done in diphone-based synthesis) results in segmental quality that is very close to (if not hardly distinguishable from; see below) using more look-ahead phonetic information.

4.2. Subjective Testing

For the subjective evaluation of the three sHTS approaches compared to the HTS we used the ABX method [21]. Our test database contains 66 different speech samples with durations of two to four seconds for each method. By using this database we created an ABX test with 30 questions.

The first 10 questions compare the synthesized speech output of HTS to the synthesized speech output of sHTS that uses one future label (sHTS1); in the same format, the next 10 questions compare HTS output to speech output of sHTS that uses two future labels (sHTS2) and the last 10 questions compare the output of HTS to the speech output of sHTS that uses three future labels (sHTS3). For each user a uniquely randomized test was generated, and for each question of the test, A and B options were randomly selected between the HTS sample and the version of sHTS that was being tested. In total we conducted 59 different tests that were completed by both speech and non-speech experts. Note that in the ABX method, 50% error rate means perfect confusability, i.e. that the two methods being tested are indistinguishable from each other. The results we obtained show relatively high confusability, enough to confirm that sHTS can be used in place of HTS.

Table 3. Error rate between HTS and sHTS1, sHTS2, sHTS3

sHTS1 vs. HTS	sHTS2 vs. HTS	sHTS3 vs. HTS
37.83%	37.66%	38.83%

5. Conclusions

Converting HTS from the original offline architecture to a streaming architecture takes us one step closer to a full real-time performative system. The results we have obtained so far are very encouraging, as they confirm that HTS can be used with only one look-ahead phoneme, as in diphone-based synthesis, with no significant distortion compared to full sentence look-ahead. Going further will require interfacing this streaming version of HTS with controllers for inputting phonetic labels on the fly, and for controlling pitch, duration, and voice quality in real-time. This of course requires further architectural modification in both the synthesis and the training part of an sHTS, and certainly the development of an appropriate interface that will combine gestures and voice synthesis. Finally, the main issue we will have to face (as anyone working on performative speech synthesis) is that of the ability of a human performer to control a large amount of control dimensions.

Acknowledgments

M. Astrinaki and O. Babacan's work is supported by a PhD grant funded by UMONS and ACAPELA-GROUP SA. The authors are grateful to Media and Graphics Interdisciplinary Centre - University of British Columbia (Canada) for organizing their internship, providing facilities to work and triggering new discussions on Human-Computer Interaction and performative speech synthesis.

References

- [1] D'Alessandro, N., Bozkurt, B., Dutoit, T., Sebbe, R., "MaxMBROLA: A Max/MSP MBROLA-Based Tool for Real-Time Voice Synthesis", in *Proceedings of the EU-SIPCO'05 Conference, September 4-8, 2005, Antalya (Turkey)*, 2005.
- [2] "Armageddon", [Web site], Available: http://www.artzoyd.net/_old/site2005/spectacles-armageddon.html
- [3] D'Alessandro, N., Babacan, O., Bozkurt, B., Dubuisson, T., Holzapfel, A., Kessous, L., Moinet, A., Vlieghe, M., "RAM-CESS 2.X framework - expressive voice analysis for realtime and accurate synthesis of singing", in *Journal on Multimodal User Interfaces (JMUI)*, Springer Berlin/Heidelberg, 2008, Vol. 2, Nr. 2, pp. 133-144.
- [4] D'Alessandro, N., Dutoit, T., "HandSketch Bi-Manual Controller: Investigation on Expressive Control Issues of an Augmented Tablet", in *Proceedings of the 7th International Conference on New Instruments for Musical Expression (NIME'07)*, 2007, pp. 78-81.
- [5] Hunt, A., Black, A., "Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database", in *Proc. IEEE International Conference of Acoustics, Speech, and Signal Processing*, 1996, pp. 373-376.
- [6] Zen, H., Tokuda, K., and Black, A., "Statistical Parametric Speech Synthesis", in *Speech Communication*, 2009, 51:11, pp. 1039-1064.
- [7] Zen, H., Tokuda, K., Kitamura, T., "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences", in *Comput. Speech Lang*, 2006c, 21 (1), 153-173.
- [8] Toda, T., Black, A., Tokuda, K., "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory", in *IEEE Trans. Audio Speech Lang. Process.*, 2007, 15 (8), 2222-2235.
- [9] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T., "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis", in *Proc. Eurospeech*, 1999, pp. 2347-2350.
- [10] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T., "Duration Modeling in HMM-based Speech Synthesis System", in *Proc. of ICSLP*, 1998, vol.2, pp. 29-32.
- [11] Zen, H., Tokuda, K., Kitamura, T., "Decision tree based simultaneous clustering of phonetic contexts, dimensions, and state positions for acoustic modeling", in *Proc. Eurospeech*, 2003b, pp. 3189-3192.
- [12] Tokuda, K., Zen, H., Black, A.W., "An HMM-based speech synthesis system applied to English", in *IEEE Speech Synthesis Workshop*, 2002.
- [13] Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., Kitamura, T., "Speech parameter generation algorithms for HMM-based speech synthesis", in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2000, Vol. 3, pp. 1315-1318.
- [14] Fukada, T., Tokuda, K., Kobayashi, T., Imai, S., "An adaptive algorithm for mel-cepstral analysis of speech", in *Proc. of ICASSP92*, 1992, vol.1, pp.137-140.
- [15] Shichiri, K., Sawabe, A., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T., "Eigenvoices for HMM-based speech synthesis", in *Proceedings of International Conference on Spoken Language Processing*, 2002, pp. 1269-1272.
- [16] Tamura, M., Masuko, T., Tokuda, K., Kobayashi, T., "Adaptation of pitch and spectrum for HMM-based speech synthesis using mlir", in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2001, Vol. 2, pp. 805808.
- [17] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T., "Speaker interpolation in HMM-based speech synthesis system", in *Proceedings of European Conference on Speech Communication and Technology97*, 1997, Vol. 5, pp. 2523-2526.
- [18] "HMM-based Speech Synthesis System (HTS)," [Web site], Available: <http://hts.sp.nitech.ac.jp>
- [19] Picart, B., Drugman, T., Dutoit, T., "Analysis and Synthesis of Hypo and Hyperarticulated Speech", in *Proceedings of the Speech Synthesis Workshop 7 (SSW7)*, 22nd - 24th of September 2010, NICT/ATR, Kyoto, Japan, 2010, pp. 270-275.
- [20] Norden, F., Eriksson, T., "A Speech Spectrum Distortion Measure with Interframe Memory", in *Proc. IEEE International Conference on Audio, Speech and Signal Processing*, 2001, May.
- [21] Clark, D. "High-resolution subjective testing using a double-blind comparator", in *J. Audio Eng. Soc.*, 1982, 30, 330-338.