

# Novel Text-To-Speech Reading Modes for Educational Applications

Lukas Latacz, Yuk On Kong, Wesley Mattheyses and Werner Verhelst

Laboratory for Speech and Audio Processing

Department of Electronics and Informatics

Vrije Universiteit Brussel

Pleinlaan 2, B-1050 Brussels, Belgium

Phone: +32 (0)2 629 3955 Fax: +32 (0)2 629 2883

E-Mail: {llatacz|ykong|wmatthey|wverhels}@etro.vub.ac.be

*Abstract*— In this paper we describe the development of two new text-to-speech reading modes for use in a computerised reading tutor for dyslexic children: a phoneme spelling mode which spells words phoneme by phoneme and a syllable mode, which allows speech to be synthesised as either isolated or lengthened syllables. These modes are used for modelling (i.e. to give a good example) and for giving feedback to the child. They are non-existent in most of the current reading tutors but will improve the flexibility of the tutor and allow better therapy.

*Keywords*— speech modification, speech synthesis, educational application, syllabification

## I. INTRODUCTION

Speech and language technologies are being applied in educational applications, such as tools for CALL (Computer-Assisted Language Learning) and reading tutors. This paper focuses on extending the functionality of a computerised reading tutor for dyslexic children. Speech output is an important aspect of such reading tutor. Although the quality of synthesized speech has improved tremendously over the last decade, natural speech is still used in most recent reading tutors (e.g., [7] and [10]).

Existing speech synthesizers usually provide a speaking style that corresponds to fluently read text. In contrast, speech therapists use different speaking styles when interacting with their patients. Additional speaking styles or reading modes increase the effectiveness of the reading tutor, as demonstrated in [5]. Phoneme-by-phoneme spelling and syllabified speech are needed for modelling and for giving feedback. Two types of syllabified speech are considered in this paper, using either isolated or lengthened connected syllables. Lengthened connected syllabification is needed to assist more advanced readers to build up their fluency and reading speed. We will describe these reading modes in detail.

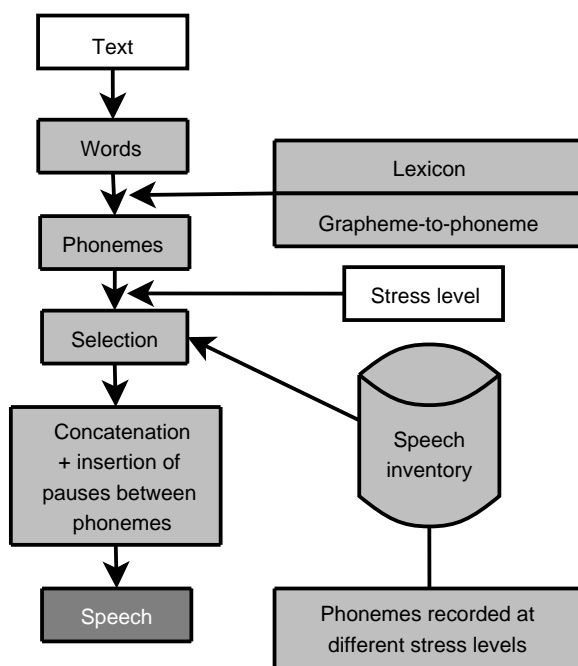


Fig. 1. Overview of phoneme-by-phoneme spelling mode synthesis.

## II. SPELLING MODE

Orthographic spelling mode, or letter-by-letter spelling, is implemented in most current speech synthesizers. A phoneme-by-phoneme spelling mode which allows over-stressing is, however, more useful for training people with reading disabilities.

The spelling mode can be considered a case of limited domain synthesis [1]. The domain, in this case, consists of single phonemes, spoken at different stress levels. Spelled speech is achieved by concatenating recordings of these phonemes with a silence of fixed duration in between. Phonemes of the same stress level should be recorded with similar pitch, speech rate and loudness to obtain natural-sounding synthesized speech.

An overview of this system is presented in figure 1.

The input text is processed into a list of words. Words are converted into a stream of phonemes using a lexicon and a grapheme-to-phoneme conversion. The appropriate phonemes with their associated stress-levels are then selected from a speech inventory and concatenated with a silence in between. The domain could be extended in order to synthesize feedback like "not *h a t* but *h a d*", which is straightforward to implement.

Utterances were spelled phoneme-by-phoneme using our system. The quality of the synthesized speech is found to be very high. This agrees with the results of other limited-domain speech synthesizers.

### III. SYLLABLE MODE

The purpose of this reading mode is to synthesize speech as either isolated or lengthened syllables. Most concatenative speech synthesizers synthesize speech by concatenating small units such as diphones or demiphones. The best result would be obtained by recording and playing back longer units, but in practice this is not feasible due to the sheer amount of recordings needed. A different approach is proposed, which is based on modifying natural fluent speech to synthesize syllabified speech.

#### A. Isolated syllables

Natural speech is transformed into speech which consists of isolated syllables by inserting a silence using phoneme-silence and silence-phoneme diphones at each syllable boundary. The prosody is modified afterwards to obtain natural-sounding speech with isolated syllables. The input speech has to be segmented and labeled into *phones*, as we will explain later. The amount of diphones needed is small and is maximally twice the amount of phonemes. For example, there are 52 phonemes in Dutch, including false diphthongs. Fewer, in fact, should be needed since not every phoneme could occur at both the beginning and the end of a syllable. For example, the voiced glottal fricative [h] in Dutch cannot be the last phoneme of a syllable. Note that the extra diphones have to be recorded in exactly the same conditions as the speech input.

##### A.1 Inserting diphones

Concatenating two speech signals is usually not a straightforward task. Differences in spectrum, pitch and energy of those signals at join positions often introduce artefacts which reduce the quality. An example is shown in figure 2. The choice of concatenation method is even more important in our case since only one instance of each diphone is used. Concatenating often involves spectral

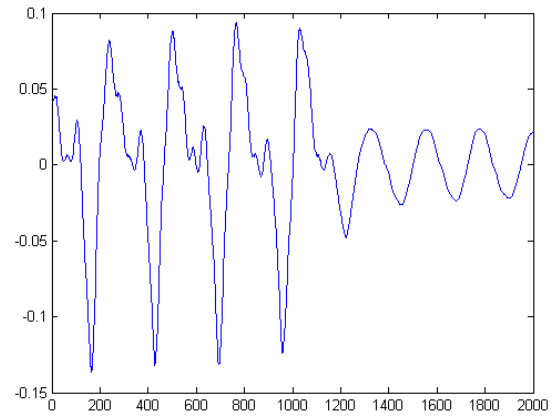


Fig. 2. An illustration of the problems that could occur when using an unoptimized method to concatenate two voiced speech signals. The concatenated signals represent the same phoneme /n/. Differences in spectrum, pitch and energy of the two signals cause noticeable artefacts in the resulting speech, although no abnormal pitch periods appear.

smoothing to minimize the differences between the concatenated signals. As pointed out in [2], spectral smoothing can at times noticeably improve speech quality, yet it may degrade it further instead. Sometimes it is better to perform no processing at all. A novel adaptive approach is therefore proposed. The amount of smoothing will depend on the spectral properties of the signals to be joined. Our algorithm contains five steps:

1. Analysis of the speech signals.
2. Calculation of the optimal join position.
3. Selection of frames from *both* speech signals.
4. Modifying the energy of the selected frames.
5. Overlap-add synthesis of selected frames to form the output speech.

In the rest of this section, it is assumed that a *phoneme-silence* diphone is added. The time scales should be reversed to concatenate a *silence-phoneme* diphone instead.

The time intervals containing the phone at syllable boundary are selected from the input speech and the diphone used. These are divided into overlapping analysis frames (e.g., 60 ms and 87.5 % overlap between overlapping frames). Note that the overlap has to be large enough, as will be explained later. Feature vectors containing MFCC's are calculated for each analysis frame. Each analysis frame of the diphone has a counterpart belonging to the input speech. These corresponding frames are selected by minimizing mismatch as indicated by a Euclidean spectral distance measure based on MFCC's. For instance, to find a frame  $A_y$  of the input speech which cor-

responds to the frame  $B_x$  of the diphone:

$$y = \operatorname{argmin}_k \sqrt{\sum_i b_{x,i}^2 - a_{k,i}^2} \quad (1)$$

where  $\vec{b}_x$  and  $\vec{a}_k$  are the feature vectors of analysis frames  $B_x$  and  $A_k$  respectively.

The optimal join position selection is based on optimal coupling [3]. The cutting positions of the diphones and syllables are not fixed in advance, but are chosen automatically in order to minimize mismatch. The overall minimum mismatch between frames defines the cutting positions.

The next step is the selection of synthesis frames from either the speech input or the diphone. These are used to construct the output speech. This is illustrated in figure 3 and explained below. Note that synthesis frames should be shorter than analysis frames in duration. Each synthesis frame comes from an analysis frame of either the diphone or the input speech. Experiments show that the proportion of the length of an analysis frame to that of a synthesis frame has to be moderate (e.g., at least 2 times) to obtain good results. The length of the synthesis frames is set to be twice the overlap between neighbouring analysis frames (e.g., 15 ms) so that successive synthesis frames overlap each other by 50%. A large overlap between neighbouring analysis frames is therefore required.

Assume that analysis frames  $A'_i$  and  $B_i$  are selected from the input speech and the diphone respectively as the frames which contain the optimal cutting positions. The input speech is then cut at the middle of frame  $A'_i$ . Let  $s_1$  be the last synthesis frame of the input speech before cutting.  $s'_1$  is the frame following  $s_1$ . Because overlapping and adding  $s_1$  and  $s'_1$  would recreate a portion of the original input speech, we need to find the frame  $s_2$  which resembles  $s'_1$  as much as possible.

This frame is selected either from analysis frame  $B_{i+1}$  or analysis frame  $A'_{i+1}$ .  $A'_{i+1}$  is the analysis frame which corresponds to  $B_{i+1}$ . The frames  $s_{diph,2}$  and  $s_{is,2}$  are the most similar frames to  $s'_1$  from  $B_{i+1}$  and  $A'_{i+1}$  respectively. The positions of these frames are found by maximizing a similarity measure (e.g cross average magnitude difference function (cross-AMDF), cross-correlation, ...). This approach is similar to WSOLA [9]. The cross-AMDF measure was used in our experiments. The algorithm continues using the most similar frame.  $s_{is,2}$  is the most similar in figure 3.

Frame  $s'_1$  is a so-called *template* frame. The next template frame is  $s'_2$  which is the frame following  $s_2$ . Frame  $s_3$  is selected from either analysis frame  $B_{i+2}$  or its corresponding frame. This procedure continues until no more

analysis frames  $B_i$  are available.

The energy of the selected frames is modified by multiplying the frames  $s_i$  with an energy modifier  $m_i$ . Assume that the total amount of selected synthesis frames equals  $N$  and the energy of a frame  $x$  is  $E_x$ . If  $s_i$  originates from the diphone, the energy modifier is calculated as

$$m_{diphone,i} = \sqrt{\frac{E_{A_m}}{E_{B_n}}} \cdot \frac{N-i}{N} + \frac{i}{N} \quad (2)$$

Otherwise the energy modifier equals

$$m_{inputspeech,i} = m_{diphone,i} \cdot \sqrt{\frac{E_{B_i}}{E_{A'_i}}} \quad (3)$$

Frames  $A'_i$  and  $B_i$  contain the optimal cutting positions, as mentioned earlier. Finally, each synthesis frame  $s_i$  will be overlapped and added to form the output. An example is shown in figure 4. The same signals are joined as in figure 2.

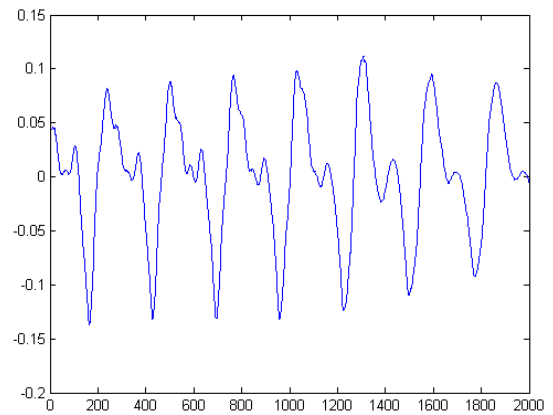


Fig. 4. Result of joining two voiced speech signals using our concatenation method. The concatenated signals represent the same phoneme /n/.

A synthesis frame belonging to the input speech is selected if that frame is more similar to the template frame than the best synthesis frame from the diphone. Using that frame minimizes mismatch. The join is therefore smoothed by selecting frames from the input speech. The amount of frames used to smooth the join is not fixed. For example, if the input speech and diphone are similar at cutting position, it is more likely that frames from the diphone are selected. Once a synthesis frame from the diphone is selected, no frames from the input speech will be selected. The opposite is also possible. No frames from the diphone will be selected if the signals differ too much. Note that sometimes a voiceless phoneme at a syllable boundary could become voiced instead because of

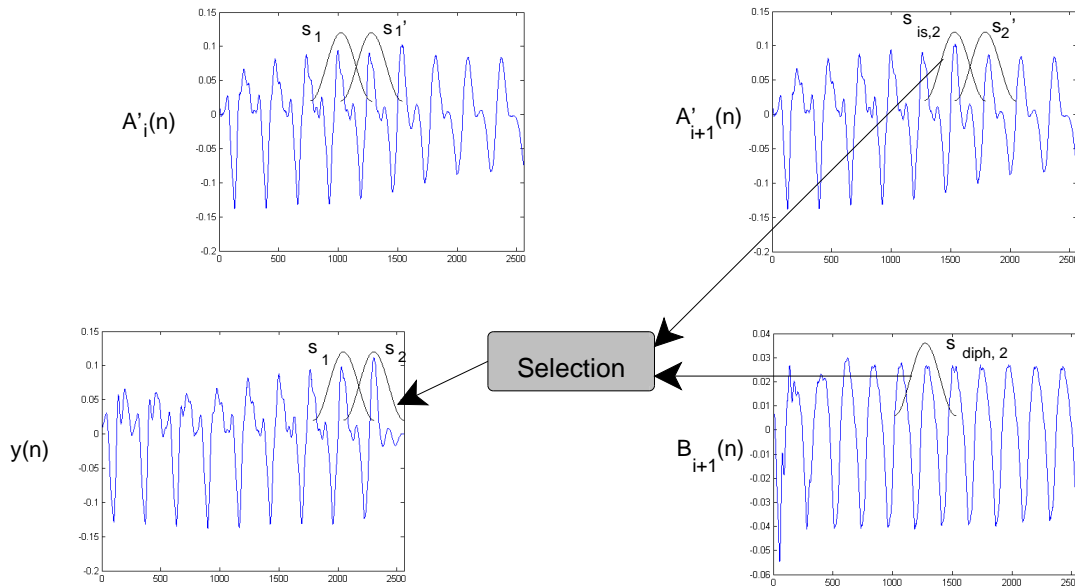


Fig. 3. Illustration of our concatenation method.  $y(n)$  is the output speech.

coarticulation. Potential problems can be avoided by selecting frames from the diphone only, if the voicing of the phone is not what is expected.

Informal listening experiments show a high output quality, which is comparable to that from other concatenation algorithms.

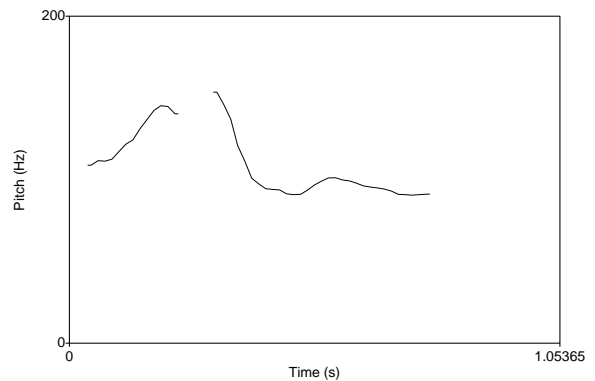
#### A.2 Prosody of speech in isolated syllables

Prosody is usually expressed in acoustic terms as pitch, loudness and duration. Once the target prosody is known, a prosody modification algorithm such as PSOLA [8] can be used to obtain the required prosody. Two prosodic dimensions will be considered: pitch and duration.

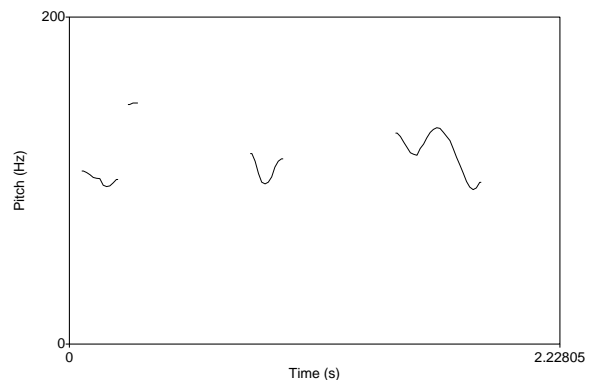
To our knowledge, the prosody of syllabified speech has not yet been investigated. In figure 5 a recording of syllabified speech is compared to a recording of fluent speech. By performing such empirical comparison, we were able to adapt the front-end of NeXTeNS [6] at the symbolic level so as to synthesize speech with isolated syllables instead of fluent speech. The processes involved in generating prosody from the text input are shown in figure 6. For more information, the reader is referred to [4].

Two approaches will be considered. The first approach is a simple approach. Pauses are inserted between syllables and the total duration of each syllable is adjusted by a fixed percentage. The pitch contour will be stretched but its shape retained. Only time-scaling modifications will be needed if fluent speech is modified.

The second approach inserts phrase breaks at syllable boundaries unless such boundaries occur at the end of a phrase or a sentence. A common categorisation of phrase



(a)Fluent natural speech



(b)Natural speech in isolated syllables

Fig. 5. The prosody of fluently read speech and that of speech in isolated syllables are not the same, as can be seen here: the pitch contours of the Dutch word 'lettergreep' (English: *syllable*) in both cases.

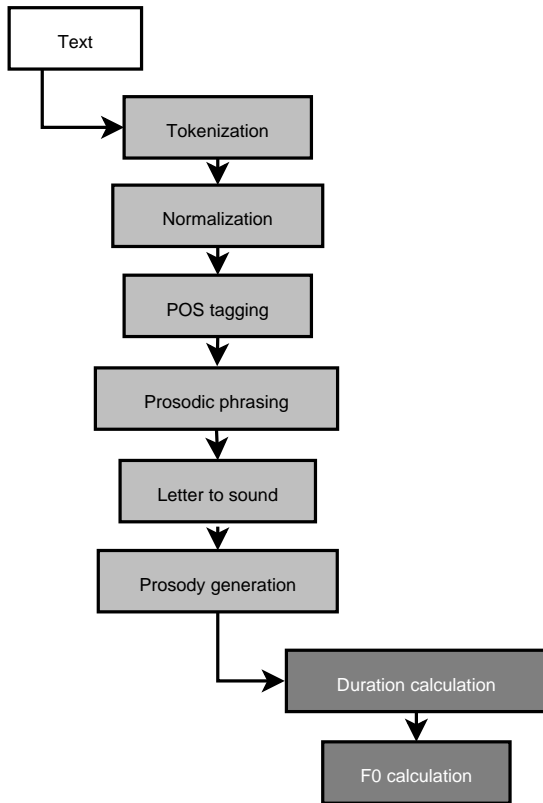


Fig. 6. In general, generating prosodic parameters involves these processes. *Tokenization* is to convert a string of characters into a list of tokens. Each token is then converted into zero or more words (*normalization*). Abbreviations, numbers, ... are expanded. *Part of speech* (POS) describes how each word functions syntactically. Next, prosodic phrase breaks are predicted and the pronunciation of each word is found. Pauses are inserted based on the predicted phrase breaks. Intonational accents are assigned by the *prosody model*. Finally, durations are assigned to each phone and the pitch contour is calculated.

breaks is to classify them as either light, medium or heavy, as in NeXTeNS. Pauses are inserted when these breaks occur. Medium breaks are inserted between words, and light breaks are inserted between syllables of the same word. The existing prosody model for fluent speech uses these breaks, whether existing or extra ones at syllable boundaries, in the calculation of the prosody. This results in the prosodic parameters for syllabified speech.

### A.3 Modifying prosody

In text-to-speech, prosody is usually generated explicitly by a model or rules. Prosodic changes on a small time-scale or *microprosody* are often not captured by the prosodic generation model. They are, however, important for obtaining natural-sounding speech. We will modify the macroprosody of the input speech while keeping the microprosodic variations, as explained later.

Assume that  $g_i$  and  $g'_i$  are the durations generated by

the prosody models of fluent and syllabified speech respectively. The duration  $d'_i$  of phoneme  $x_i$  is calculated as

$$d'_i = \frac{g'_i}{g_i} \cdot d_i \quad (4)$$

The duration  $d_i$  is the original phoneme duration.

The pitch contour  $p_i(t)$  expresses the rise and fall of the pitch of the  $i^{\text{th}}$  syllable of input at time  $t$ . These contours are time-scaled using the calculated durations, which results in contours  $p_{ts,i}(t)$ . The pitch contours  $q_i(t)$  and  $q'_i(t)$  are the contours of the  $i^{\text{th}}$  syllable generated by the prosody models of fluent and syllabified speech respectively. Each pitch contour  $q_i(t)$  is time-scaled using the calculated durations, to obtain  $q_{ts,i}(t)$ . The new pitch contours  $p'_i(t)$  can now be calculated as

$$p'_i(t) = \frac{q'_i(t)}{q_{ts,i}(t)} \cdot p_{ts,i}(t) \quad (5)$$

PSOLA is used to apply these changes. Informal listening experiments show that the simple approach is very suitable for short utterances, like single words or short sentences. The phrase break model seems more appropriate for longer utterances.

### B. Lengthened and connected syllables

Timing is the most important difference between fluent speech with normal speech rate and speech with lengthened and connected syllables. Note that the latter is not the same as slow, fluent speech. The goal is to change the timing of natural fluent speech so that it could fit some of the clinical needs for therapy.

In natural speech with lengthened and connected syllables, we observed that some parts of syllables are lengthened more than others. For example, the Dutch word *nemen*<sup>1</sup> has two syllables, *ne* and *men*. The first and last phonemes of the word are the same. If spoken in lengthened connected syllables, they will be lengthened to different extents. The vowels are lengthened differently, although they have the same location in the syllable. A non-uniform time scaling approach is therefore required. Our algorithm is based on WSOLA [9], which can be used to time-scale with time-varying time-scaling factors. In our case, these factors are updated each 7.5 ms.

A piecewise linear model of the time-scaling factors is proposed. Each phone  $x_i$  is lengthened using its own time-varying time-scaling factor  $\beta_i$ . These factors are interpolated across phone boundaries to handle phoneme transitions. The modified signal is slowed down if  $0 < \beta(t) < 1$  and speeded up if  $0 > \beta(t)$ . We observed that the kind

<sup>1</sup>to take, transcription in SAMPA: /nem@n/

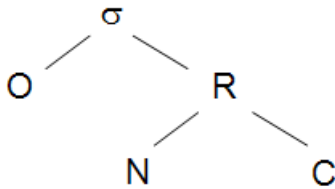


Fig. 7. Tree representation of a syllable. The nucleus  $N$  and the coda  $C$  form together the rime  $R$ .  $O$  is the syllable onset. Only the nucleus is obligatory for all languages. This is always a vowel in Dutch.

Phoneme classes	Example phonemes	$\gamma$
Plosives	p, b, t, k	0
Short vowels	A, E, I, O	2
Long vowels	a, e, i, o	3
Nasals	m, n	3
Voiceless fricatives	f, x	1
Voiced fricatives	v, z	0.5
Other	l, r	0.5

TABLE I

VALUES OF THE TIME-SCALE PARAMETER  $\gamma$  USED IN OUR EXPERIMENTS FOR LENGTHENING SYLLABLES. PHONEMES ARE EXPRESSED IN SAMPA NOTATION.

of phone and the position of the phone in the syllable are the most important aspects for calculating the time-scaling factors:

$$\beta_i = \frac{1}{1 + \gamma_i \cdot \phi_j} \quad (6)$$

$\gamma_i$  is the so-called *time-scale factor modifier* for phone  $x_i$ . A large  $\gamma_i$  indicate that it is very likely that the phone will be lengthened. Vowels and nasals are examples of these phonemes. Plosives, on the other hand, usually have a very small time-scale factor modifier. Each syllable has, at most, three parts: onset, nucleus and coda. These could be represented as a tree as shown in figure 7.  $\phi_i$  is the time-scale factor modifier for a syllable part. The onset is usually not lengthened much.

Currently, the parameters of our model are set empirically. An overview of the parameter values can be found

Part of syllable	$\phi$
Onset	0.5
Nucleus	2
Coda	2

TABLE II

VALUES OF THE TIME-SCALE PARAMETERS  $\phi$  USED IN OUR EXPERIMENTS FOR LENGTHENING SYLLABLES.

in tables I and II. Future work includes the use of parameters derived from real-life data, if possible, taken from clinical sessions.

#### IV. CONCLUSION

We successfully developed two new text-to-speech reading modes for use in a Dutch reading tutor for dyslexic children. Initial experiments using recordings of a female voice indicate high naturalness and quality of the output. Current investigations target the use of these modes to improve the performance of slow readers.

#### V. ACKNOWLEDGEMENTS

Part of the research reported in this paper was supported by IWT project SPACE (SBO/040102). The authors would like to thank Pol Ghesquière and Leen Cleuren from the Centre for Disability, Special Needs Education and Child Care of the Department of Educational Sciences, Katholieke Universiteit Leuven, Belgium for their assistance and comments.

#### REFERENCES

- [1] A. Black and K. Lenzo. Limited domain synthesis. In *Proc. IC-SLP*, volume 2, pages 411–414, Beijing, China, Oct. 2000.
- [2] D. Chappell and J. Hansen. Spectral smoothing for speech segment concatenation. *Speech Communication*, 36:343–373, 2002.
- [3] A. Conkie and S. D. Isard. Optimal coupling of diphones. In J. P. H. Santen, R. W. Sproat, J. P. Olive, and Hirschberg, editors, *Progress in Speech Synthesis*. Springer, 1996.
- [4] T. Dutoit. *An introduction to text-to-speech synthesis*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.
- [5] C. Heiner, J. E. Beck, and J. Mostow. Improving the help selection policy in a reading tutor that listens. In *Proceedings of the InSTIL/ICALL Symposium on NLP and Speech Technologies in Advanced Language Learning Systems*, pages 195 – 198, June 2004.
- [6] J. Kerkhoff and E. Marsi. Nextens: a new open source text-to-speech system for dutch. In *Proc. 13th meeting of Computational Linguistics in the Netherlands*, 2002.
- [7] J. Mostow and G. Aist. Evaluating tutors that listen: An overview of project listen. In K. Forbus and P. Feltovich, editors, *Smart Machines in Education: The coming revolution in educational technology.*, pages 169 – 234. MIT/AAAI Press, 2001.
- [8] E. Moulines and F. Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech using diphones. *Speech Communication*, 9:453–470, 1990.
- [9] W. Verhelst and M. Roelands. An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech. In *In Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 554–557, Minneapolis, USA, 1993.
- [10] B. Wise, R. C. S. van Vuuren, S. Schwartz, L. Snyder, N. Ngamapatpong, and J. Tuantranont. Learning to read with a virtual tutor: Foundations to literacy. In C. K. Kinzer and L. Verhoeven, editors, *Interactive Literacy Education: Facilitating Literacy Environments Through Technology*. Erlbaum Publishers, Mahway, NJ, 2005.