18

Adapting Prosody in a Text-to-Speech System

Janez Stergar[1] and Çağlayan Erdem[2]
[1]University of Maribor, Faculty of Electrical Engineering and Computer Science
[2]Siemens Corporate Technology, Dept. CTIC 5
[1]Slovenia
[2]Germany

1. Introduction

The requirements of the evolving information communication technologies (ICT) place new demands on text-to-speech (TTS) systems. The modern high quality TTS system has to be capable of fast and high-quality adaptation to a new language, voice or even expressive speech. Thus adaptation to new voices with different prosodic characteristics is desired.

In this chapter a survey of recent and past approaches of prosodic processing in text-to-speech synthesis will be discussed.

Regardless of the different approaches which have been proposed ranging from generating prosody by rule to huge databases covering almost all prosodic patterns of a specific speaker there is clearly still much work to be done (van Santen et al., 2008).

Automatic learning techniques seem to offer the fastest solution in adapting a TTS system to a new language, voice or a new application. They allow automatic extraction of specific features (e.g. non-uniform unit selection, prosodic regularities extraction) from an appropriate database of natural speech. Such techniques depend on the construction of a large pre-processed corpora (properly segmented, labelled with appropriate prosody labels, etc.). Despite the overall impression that TTS is an inferior task compared to speech recognition, TTS research and development community was not able to produce massive series of consumer products since the early 80es (Dutoit, 2008). Since then a broad spectrum of systems has been developed and successfully implemented – prosody was one of the major tasks to tackle in such systems.

The term "prosody" covers a wide range of features characterizing "the musical qualities" of speech, including phrasing, pitch, loudness, tempo and rhythm. A number of studies suggest that prosody has a great impact on the intelligibility and naturalness of speech perception. Despite the fact that synthesized speech is nowadays mostly intelligible and in some cases sounds undistinguishable from human speech, it still lacks the flexibility and appropriate rendering of expressivity in the synthesized voice.

Text-to-prosody systems based on the use of prosodic databases extracted from natural speech are a key point for development of new TTS systems. One of the major problems in TTS synthesis consists in the automatic generation of natural and intelligible prosody. Therefore the preparation of suitable speech-corpora for automatic prosodic feature extraction is essential.

The pre-processing and labelling in the TTS front end can be performed either automatically or by hand. While automatic labelling can be less accurate than hand labelling, the latter is very time consuming (sometimes also inconsistent). However in some processes, such as segmentation to non-uniform units, which are essential for concatenative TTS synthesizers and verification of automatically labelled data, expert guided procedures can't be avoided. On the other hand many adaptation tasks can be realized automatically or semi-automatically.

In the following sections we will discuss the main three approaches in tackling prosody in a TTS system:

a. the rule based approach,
b. the statistical approach and
c. the minimalistic approach, using as-is prosody in unit selection process minimizing manipulation efforts of units.

We will emphasize the data-driven (corpus-based) approach of extracting prosodic features and focus on the design of a database. We will discuss the basic procedures in the design of the final corpus and steps taken for a suitable corpora preparation and adaptation of learning modules included in the TTS back end. Also all major processes in the TTS front end will be emphasized – the process of labelling the corpus with appropriate tags (e.g. symbolic prosody labels) and the construction of suitable modules for prediction (modelling of different labelling categories). Nevertheless the most important part the adaptation of the TTS back end will also be discussed.

A selective method for classification of different symbolic tags will be introduced and a NN structure based on auto-associative classificators for modelling prosody presented. The symbolic information in this stage is ported to the final and most important module – the module for acoustic modelling. In this section state of the art NN structures suitable for adaptation to a new language without language expertise were implemented. Our discussion will be concluded with the evaluation tests performed on the adapted multilingual TTS system for Slovenian language.

2. Prosody and TTS

The conversion of text to speech can be described as essentially a two-stage process (Fig. 1) – these are an analysis stage, which derives a linguistic structure from the input text and generation stage, where linguistic structure is used for speech synthesis including the generation of intonation, rhythm and so on.

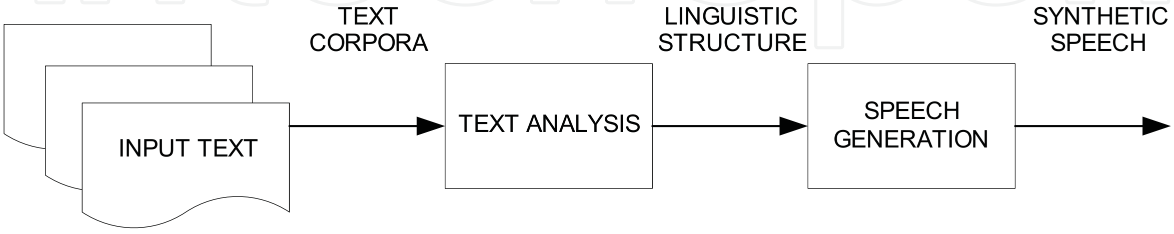

Fig. 1. Text-to-speech process.

The result of text and linguistic analysis in the text front end stage is detailed linguistic information about the structure of the input text, in particular, syntactic, lexical, and phonological with some semantic information. Only the proper choice of prosodic

parameters given by sound duration and intonation contours enables a TTS backend to produce natural-sounding, high quality, synthetic speech (Edgington et al., 1996).
One of the major problems in text-to speech synthesis system consists in the automatic generation of a natural and intelligible prosody. There are two main approaches to the prediction of prosodic structures - rule-based and stochastic.

2.1 Rule based prediction

Many of the rule-based approaches stem from early work on performance structures based on experimental data, such as pausing and parsing values. This work sought to account for the disparity between linguistic phrase-structure theories and actual performance structures produced by humans, and focused on recreating the pause data of several analyzed sentences from syntax (although they claimed that their method could easily account for other prosodic features). The central tenet of the work was that prosodic phrasing is a compromise between the need to respect both the linguistic structure and performance aspects of the sentence.
More recent efforts have extended the work on performance structure prediction to the prediction of prosodic phrasing. In this work, the basic rule-based approach is preserved, but other factors are introduced which are considered important for predictive purposes. For example, some of the researchers believe that syntax plays a lesser role in determining phrasing, and those certain prosodic performance constraints, such as length, override syntactic structure. They allow prosodic boundaries to cross-syntactic boundaries under certain conditions, whereas early work was essentially interclausal. Other modifications include counting phonological words rather than actual words when determining node strengths. A phonological word effectively functions as one spoken item, as the internal word-word boundaries are resistant to pausing. Typical examples are determiner-noun word groups, such as the `the + man'. Further extensions incorporate punctuation into the predictive models, or assign more importance to specific features (Edgington et al., 1996).

2.2 Data-driven or stochastic methods

With the availability of large corpora, annotated with prosodic information such as location and salience of pauses, temporal information on duration, etc., the stochastic-based approach will come more to the fore. Recently methods for automatically predicting prosodic information using decision tree models have been described. Generally, decision trees are derived by associating a probability with each potential boundary site in the text, and relating various features with each boundary site (e.g. utterance and phrase duration, length of utterance - in syllables/words - positions relative to the start or end of the nearest boundary location, etc). The resulting decision tree provides, in effect, an algorithm for predicting prosodic boundaries and their salience (i.e. relative importance) for new input texts.
It is interesting to note that evaluations of both rule-based and data-driven methods recently showed that similar results are achieved (Edgington et al., 1996).

3. The database construction

The first step towards a data driven learning process is the design of speech corpora suitable for (in our case concatenative) TTS synthesis. Also many other aspects in speech synthesis have to be considered e.g. modelling of speech specific attributes (e.g. prosody, emotions, etc.). One has to consider that literate native speakers have to deal with many important production tasks in the interpretation procedure of read speech (controlled laboratory speech). Pronouncing words correctly is only part of the problem faced by human readers. In order to sound natural and to sound as if they understand what they are reading, one must also appropriately assign prominence to some words, and de-emphasize others. It is unavoidable to 'chunk' the sentence into meaningful (intentional) phrases. Appropriate fundamental frequency (f0) contours have to be chosen and control of certain aspects of voice quality performed. One should also pay attention that a word should be pronounced longer if it appears in some positions in the sentence, that if it appears in others, since segmental durations are affected by various factors, including phrasal position (Sproat & Olive, 1995).

Despite the trend in high quality concatenative TTS systems, to gather as much as possible material (Campbell & Mokhtari, 2003), our goal was to make a compromise between quality and financial means available. We strove for a solution of all in one corpus with the goal to use only one large enough for all major data-driven tasks needed in adaptation of the multilingual TTS system.

3.1 The corpus

The corpus consists of app. 1200 sentences in the Slovenian language (orthography), which equals approximately three hours of speech. The selection of the corpus text was designed to ensure good coverage of the phones in the Slovenian language; therefore clauses were gathered and included from different text styles (e.g., literature and newspaper texts).

The main concern in corpus design was towards optimized suitability for concatenative speech synthesis (the best coverage of elementary segments). No intentional balancing of clause types was performed (declarative – interrogative – exclamations), dialogue context and syntax were not considered, and no semantic analysis was performed since only isolated sentences were included. Prosody was not the first concern for text selection.

The whole corpus was determined using a selection of clauses from a 31 million word corpus in the Slovenian language from e-newspapers, e-literature, the WWW or CD's. The major parts of the clauses covered daily-published news and Slovenian literature; the minority consisted of clauses taken from Slovenian poetry.

In the first step sentences not shorter then 15 and not longer than 25 words were pre-selected from the major corpus. Then, four different text corpora were generated and analyzed statistically (approximately 5000 sentences per corpus). The selection of sentences for the final corpus was based on a two-stage process. In the first stage an analysis based on statistical criteria was performed. In the second stage the final text was chosen based on the results of the first stage. In the following the two stages are described briefly.

After the grapheme-to-phoneme conversion the statistic analysis of corresponding units (mono-phones, diphones, …) generated in a non-uniform unit generator was performed at the sentence level for each of the four corpora in a separate module. The analysis module scans all non-uniform units and determines how frequently each unit appeared. The obtained statistics mirror the non-uniform unit richness and unit structure of all clauses for the corresponding text corpora (Rojc & Kačič, 2000).

After the described statistical analysis of the four different text corpora the final corpus was generated for each of the four. The criterion for the final text filtering was based on monophone, diphone, triphone and fivephone (non-uniform units) richness. Considering comprehension and frequency of units, a careful elimination of sentences was performed. Clauses with poor unit comprehension and unit duplicates were eliminated. In the final corpus 1200 sentences remained (Rojc & Kačič, 2000).

The statistically analyzed corpora had similar unit statistics, although the distribution of units was not the same. Three of the four corpora included many foreign names (clauses gathered predominantly from newspapers) that we replaced with Slovenian ones, essentially not influencing the statistics of non-uniform units. The corpus with the minimum changes of the non-uniform units after foreign name replacements was chosen as our final corpus.

3.2 Audio recordings

The audio database recordings were created in a studio environment with a male speaker reading aloud isolated sentences in the Slovenian language (fs = 44.1 kHz, 16 bit).

Because the speaker was a professional radio news speaker, the speech contained no disfluencies (i.e., filled pauses, repetitions and deletions) although there was some evidence of hesitations in the form of pauses and lengthening. Compared to the German used in Müller et al., 2000_b, the percentage of hesitations differed significantly (<0.5% German, >15% Slovenian). The stated comparison was estimated after statistical analysis of B9 tags inserted by the labellers in the procedure of phrase breaks labelling.

3.3 Phonetic transcription

The phonetic transcription was managed using a two-step conversion module. The first step is realized with a rule-based algorithm. The second step was designed with a data-driven approach (NN were used). The module was designed for the support of two approaches in grapheme–to–phoneme conversion. The first part was intended for those cases in which no morphological lexica were available. The first rule based stress assignment was applied, followed by a grapheme-to-phoneme conversion procedure. The step of stress marking before grapheme-to-phoneme conversion is very important for the Slovenian language, since it very much depends on the type and place of stress. If the phonetic lexicon is available, a data-driven NN approach, represented by the second part in the module, can be used. In the proposed data-driven approach, a phonetic lexicon was used as data source for training the NN (Rojc & Kačič, 2000).

The data preparation, generation of the training patterns and the training of NNs were done completely automatically. The transcription was performed in two steps. In the first step the graphemes were converted into phonemes, and syllable breaks were inserted in the phoneme string. In the second step the stress marks were inserted. The problem of how to perform mapping between graphemes and phonemes by generating training patterns for NNs (NN) was solved as proposed in Hain, 1999. For both NN tasks we applied a multilayer perceptron (MLP) feed-forward network with one hidden layer. As a learning algorithm, the back-propagation algorithm was chosen.

Pronunciation was derived from the IPA Alphabet. In order to represent the IPA symbols in ASCII characters the SAMPA format is widely used. In our grapheme-to-phoneme conversion module the SAMPA phonetic transcription symbols for the Slovenian language were used (Kačič & Zemljak, 1999).

3.4 Phonetic segmentation

The spoken corpus was phonetically transcribed using HTK. Along with standard nomenclature, two special markers were used for pauses between phonemes. “sil” denotes the silence before and after a sentence. “sp” denotes the silence between words in a sentence. Both were determined with a one-state HMM and all phonemes with three-state HMM in the HTK environment. The ‘sil’ and ‘sp’ tags were used in the semi-automatic process of phrase breaks labelling.

3.5 Part of speech tags

The text corpus was hand-labelled using 13 different classes of part-of-speech tags (POS) among which two were used for punctuation marking (end/intermediate). All tags were combined in an environment where tracking and correcting tags was simplified for the labellers. First an existent environment for reviewing in Microsoft Word was used, but due to stability problems it was replaced with a public domain editor using macros. The first environment was promising with the ability of different reviewing marks with accompanying time stamps and user (different labellers) specific tags (colour-marked).

Compared to the POS tag set in the German corpus used in Müller et al., 2000_b, the POS tag set for the Slovenian corpus is smaller. The difference in size occurs because the Slovenian corpus is hand-tagged, and no reliable tagger currently exists for a large POS tag set and the possibility of reliable automatic POS-tagging.

4. Symbolic prosody labelling

As automatic approaches usually depend on some manual examination and eventual corrections (verification) it seems to be appropriate approaching the problem of labelling with a semi-automatic method.

In our approach of corpora preparation we designed a graphical environment, which we applied for semi-automatic phrase break labelling. The tool was planed to simplify the labeller decisions and support the classification of different classes of breaks.

4.1 Phrase breaks labelling

Since no inventory for symbolic prosody breaks labels is defined for the Slovenian language, it was decided to use labels similar to those used in Kompe, 1997 and Mihelič et al., 2000. Thus the prosody break labels are determined through acoustic perceptual sessions, and the text was labelled speaker dependent (the decisions on labelling were made exclusively on perceptual criteria). The following inventory of prosody break labels was used for labelling the corpus:

- B3 – full intonational boundary with strong intonational marking, often with lengthening or change in speech tempo (we’ll refer to that label as a major break);
- B2 – intermediate phrase boundary with weak marking (we’ll refer to that label as a minor break);
- B9 – irregular prosodic boundary, usually disfluencies at hesitations, repairs etc. ; and
- B0 – normal word boundary.

The acoustic prosodic boundaries were determined by boundary indication, listening to audio files and visual output (pitch and energy) from specially designed graphical tool.

4.2 Labeling of prominent words

It was decided to distinguish between word accent, phrase accent and sentence (utterance) accent. Word accent is carried by a word emphasized through perceptual prosodic accent or pitch accent, where phrase accent by our definition is carried by a word most prominent within a phrase comprised of one or more accentuated words. The third accent defined in our inventory is the so-called sentence accent, which is (eventually) carried by a word most prominent in the considered sentence (it is not necessary that a distinction of the so-called sentence accent can be made between words being prominent). The classification of specified accents is a complex matter; therefore, an inventory adequate to distinguish among the three accents was chosen. However, the performed experiments concentrated only on the reduced categories defined in our accent-labelling inventory using two labels (AC = accented, NA = not accented). In the used inventory a phrase is a sequence of words within B2/B3 boundaries.

Distinction between accented and non-accented words was done within a phrase comparing syllable pitch envelope and normalized syllable mean average pitch changes (normalized on syllable mean average pitch changes for the concerned sentence). Energy and mean energy for syllables in each word were also considered. Through acoustic-visual sessions with our graphic tool also a classification in special cases was made where, depending on the accent type, the accented syllables had low average pitch compared to the sentence average. A German-like TOBI intonational description scheme was used for intonational marking (Benzmüller & Grice, 1999). Word prominence is classified according to four classes:

- EA = Emphatic accent,
- PA = Primary accent,
- SA = Secondary accent, and
- NA = No accent.

We consider primary accent to be assigned to normally accented words – words perceptibly most prominent within a phrase (lexical stress). Usually one or more words within a phrase carry a primary accent. We consider the secondary accent to be conveyed by an accentuated word within a phrase, not carrying a primary accent. Finally, the emphatic accent is reserved for accented and (lexical) non-accented words that are perceived as extremely stressed relative to other words or are carrying an emphatic function.

4.3 Selection of phrase breaks

A tool intended to help the labeller (novice or expert) to make decisions about prosody breaks within each sentence was designed (Stergar et al., 2003). The tool indicates possible prosody boundaries, which depend on the segmented pauses in spoken corpora. Prominent words are also indicated.

Experiments on multilingual databases (3 languages) have shown that the strategy of segmenting the speech signal with pauses yields a significant improvement in annotation accuracy (Vereecken et al., 1997). Therefore syllable and word boundaries were marked with vertical lines adding overview clearness, and *B* marks for symbolic prosody boundaries were inserted in the sentence concerned.

The designed tool indicates markers for prosody boundaries taking phonetic segmentation of pauses into account. Yet considering only the duration of silence between phrases, it indicates the position of prosody boundaries. The decision for break indication is made by comparison with a specific threshold. This threshold can be changed manually and tuned according to a specific speaker (Stergar & Hozjan, 2000). However, boundaries indicated by intonational marking or lengthening (without a pause) must still be hand-labelled.

4.4 Selection of prominent words

Two classes of prominence on word level were defined (Stergar & Horvat, 2003):

- perceptual prosodic accents (words being emphasized by stress) and
- pitch accents (words being emphasized by pitch movements).

Our aim was the selective detection of both classes automatically. The hand labelling of prominent words of our database is in progress but is due to a very time consuming process proceeding very slowly.

The first acoustic parameter involved in our experiments was band-pass filtered energy. We used a classical FIR with frequency bounds between 500 – 2000Hz. Experiments in Tamburini, 2002 for Italian and Sluijter & van Heuven, 1996 for American English and Dutch (both for male speakers), showed that this band of high frequencies is the most suitable. For every utterance we computed RMS of the band-pass filtered energy. Energy variations across different utterances were reduced with normalizing every syllable with mean syllable energy over the concerning utterance.

The second acoustic parameter was fundamental frequency – f0. As the extraction of the pitch contour is a delicate task we used a successful scheme for f0 estimation. Therefore we used a robust algorithm for periodicity detection in the autocorrelation domain, suggested in Boersma, 1993.

Additionally we processed every utterance and computed a measure for pitch changes – pitch dynamics (f_D) – for every syllable (Hozjan & Stergar, 2002):

$$f_{D_j} = \sum_{i=1}^{N} \left| x_{i+1} - x_i \right| \tag{1}$$

where j is indexing the current syllable and i the concerned sample.

The prominent syllables (words) were automatically classified according to the proposed statistical threshold selection criteria for perceptual and pitch prosodic accents (Stergar et al., 2003).

5. Acoustic prosody modelling

Data-driven prosody generation modelling by NNs was established first in Traber, 1992 and became state-of-the-art technique. The commonly used NN architecture is the multi-layer-perception with a direct recurrence (Traber, 1992) or without (Haury & Holzapfel, 1998; Erdem et al., 2000). The advantage of modelling with NN is fast and easy adaptation to new languages, speakers and speaking styles. The used TTS system has a language and speaker independent core with external knowledge sources like lexica and NN modules for special tasks. The f0-contour generation module is such an external adaptable source.

5.1 The concept of NN adaptation to a new language

Building up a NN for such a task for the first time or during the adaptation to a new speaker or even to a new language is a delicate task. There is a dilemma how many inputs available to use trying to avoid high input dimensions of a NN. An over-fitting problem of NN's with a high input dimension (which brings lower generalization abilities) is known (Prechelt, 1998). This problem was overcome using expert knowledge e.g. knowledge about the importance of the input parameters or time consuming heuristic approach e.g. testing different input parameter constellations to get the best performance (Sonntag et al., 1997).

We propose a parametric weight decay method, which enables to overcome the difficulties of data-driven techniques with fast adaptation without language expertise. This parametric weight decay systematically analyses the input vector of a NN. In an additional pre-processing unit of the original NN a diagonal matrix to a pre-processing cluster propagates the input parameters. The weight decay concept is applied only to these diagonal elements. These elements represent a weighting of the according input, which then allows an evaluation of input parameters.

The weight decay concept helps training NN models with reduced degree of freedom by adding a penalty term to the error function (Eq. 2). The first term $E_0(w)$ is the original error function and the second W_P (λ, w_i) is the standard weight decay penalty function given by:

$$W_P\left(\lambda, w_i\right) = \frac{\lambda}{2} \sum_{w_i} w_i^2 \tag{2}$$

where

λ denotes a penalty scaling factor,

w_i denotes the weights of the NN the penalty term is applied on.

During minimization small values for the weights w_i of the weight vector W were preferred as high valued weights lead to big penalties (Eq. 3). The penalty term used encourages smoother NN weight mappings (Bishop, 1995).

$$E\left(w^j\right) = E_0\left(w^j\right) + \frac{\lambda}{2} \sum_{w_i^j} \left(w_i^j\right)^2 \to \min_{w_i} \tag{3}$$

where

j denotes the number of the training epochs of the NN penalty scaling factor.

The weight adaptation is performed using the adaptation parameter η, which controls the step size during NN training (Eq. 4).

$$w^{j+1} = w^j - \eta \frac{\partial E\left(w^j\right)}{\partial w^j} \tag{4}$$

Substituting the second term in Eq. 4, with its partial derivative leads to:

$$w^{j+1} = w^j - \eta \lambda w^j - \eta \frac{\partial E_0\left(w^j\right)}{\partial w^j} \tag{5}$$

The weight vector w_{j+1} for the next epoch in Eq. 5 is computed as the difference of the prior weight vector w_j and the partial derivative of the error function (back propagation) as applied in Müller et al., 2000_a and Hain & Zimmermann, 2001. The final weight adaptation is performed with the modified weight vector w_j, where the weight decay term is extended with a parameter p in Eq. 6:

$$E\left(w^j\right) = E_0\left(w^j\right) + \frac{\lambda}{p} \sum_{w_i^j} \left|w_i^j\right|^p \to \min_{w_i}, \quad 0 < p \le 1 \tag{6}$$

and with a modified penalty term with its weight adaptation in Eq. 7:

$$w^{j+1} = w^{j} - \lambda\eta sign\left(w^{j}\right)\left|w^{j}\right|^{p-1} - \eta\frac{\partial E_0\left(w^{j}\right)}{\partial w^{j}} \tag{7}$$

In the following this modified weight decay function will be referred as *p-WD* (according to the term of the introduced parameter p). There are different ways to implement the weight decay within NN training, as one might apply the penalty term to all weights within the NN or to special connection areas. The input vector *x* is propagated by the *diagonal matrix* $W_{diag} \in R^{l\times l}$ to the pre-processing layer, which has a *tanh* activation function (Fig. 2). This diagonal matrix is the only connection that utilizes the penalty term in Eq. (6). Due to this fact the order of vector element x_i' is defined by the product of the *diagonal matrix* W_{diag} and the input vector x_i. Therefore w^{i}_{diag} gives a weighing of the input x_i. w_i is bound to the interval [0,1]. After pre-processing of input data the weighed inputs are propagated to the original NN with *n* hidden neurons and *m* outputs. The original NN will be explained in detail in the after going paragraphs. It is important to initialize W_{diag} with equal values, which are incremented or decremented according to the weight adaptation in Eq. (7) during training. This type of realization of weight decay aims at:

- Outlier cancellation: the asymptotic behaviour of *tanh* is used in the pre-processing layer as a delimiter. An element of W_{diag} growing too high is limited to the interval [-1;1] at the output side of the pre-processing cluster. With this precaution measure inputs are avoided to dominate the mapping. All inputs are limited to the interval [-1;1].
- Soft input pruning: elements of W_{diag} being pushed to zero are considered to have no significant influence on the NN model. This means that the corresponding input contains no important information for the training task with the used database. Inputs being close to zero might be removed afterwards by introducing a threshold and omitting inputs with values in W_{diag} below that threshold. We obtain soft input pruning, as there is no ultimate decision made during training. Unimportant inputs are faded out.
- Separation into subtasks: The original task and the input feature analysis are solved in a parallel manner. The feature analysis does not need a further training phase.

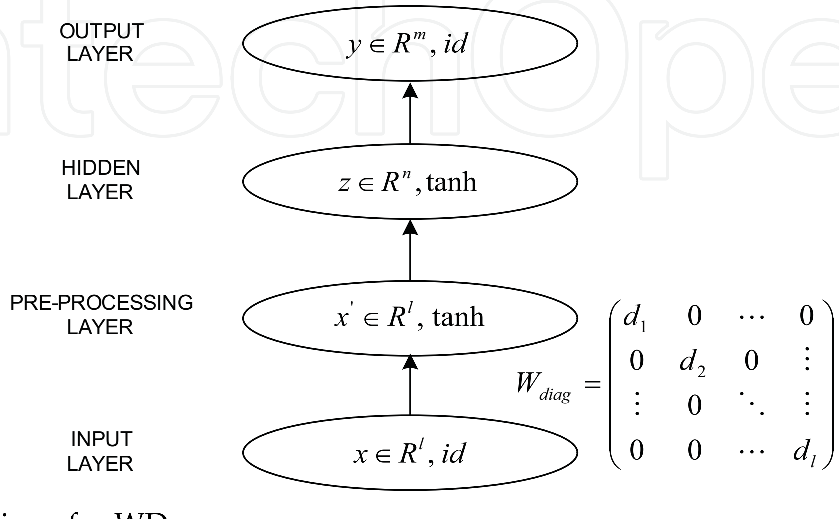
OUTPUT LAYER
$y \in R^m$, id
HIDDEN LAYER
$z \in R^n$, tanh
PRE-PROCESSING LAYER
$x' \in R^l$, tanh
INPUT LAYER
$x \in R^l$, id
$W_{diag} = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \vdots \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & d_l \end{pmatrix}$

Fig. 2. Realization of p-WD.

Dealing with highly correlated input features the application of *p-WD* should be preferred, as it helps to select one of the highly correlated features in contrast to standard weight decay. Weight decay does not select one of the highly correlated inputs. This different selection property of *p-WD* is exemplified using a NN with two highly correlated inputs. This can be formulated in Eq. (8).

$$x^{'} = \tanh\left(w_1 x_1 + w_2 x_2\right) \tag{8}$$

If we presume that inputs x₁ and x_2 are highly correlated we can rewrite Eq. (8) into:

$$x^{'} = \tanh\left(\left(w_1 + w_2\right) x_1 + 0 \cdot x_2\right) \tag{9}$$

The penalty terms for the standard weight decay are given by Eq. (10). The left side of the inequation according to Eq. (8) and the right side according to Eq. (9):

$$\frac{\lambda}{2}\sum_{w} w^2 \quad : \quad w_1^2 + w_2^2 \geq \left(w_1 + w_2\right)^2 + 0^2 \tag{10}$$

Using the Lagrange multiplier method and the penalty terms for Eq. (8) and Eq. (9) we can state the final inequation for the modified p-WD with the introduced parameter p as follows:

$$\frac{\lambda}{p}\sum_{w} |w|^p \quad : \quad \left|w_1\right|^p + \left|w_2\right|^p \geq \left|w_1 + w_2\right|^p + 0^p \tag{11}$$

Through the p-WD penalty function we achieve a minimized input feature set as the right side of Ineq. (11) is smaller than the left side. In experiments described in Erdem & Zimmermann, 2002_c, the stated behaviour of p-WD has been observed and the optimal parameter p determined.

5.2 f0 generation

The p-WD method presented in the foregoing section is implemented in the f0-contour generation module. The utilized NN has to map input parameters to an appropriate f0-contour. Regarding the syllable the mapping is performed to four f0-contour parameters (Fig 3). The solid line depicts a f0-contour on the syllable level. These contours are parameterized (dashed line) by four values: f0-maximum (p1= f0_max), f0-maximum position (p2= f0_maxPos), f0 at syllable start (p3= f0_Start), and f0 at syllable end (p4= f0_Stop). For the contour parameterization a maximum based description is used (Heuft et al., 1995), which mainly defines that f0-contours on syllable level for non-tonal languages can be described by a rising on the first part and a falling on the second part of the syllable.

The mentioned parameters p1, p2, p3, and p4 are the outputs y = {p1, p2, p3, p4} of the NN respectively (Fig. 4). Hence the dimension of the output cluster m = 4.

f0-contours are known to be influenced by long-term features (the sentence type), breath and local stress intention. The input parameters must contain information concerning local and global characteristics (symbolic prosody tags). For a good mapping it is also important to provide contextual information of the syllable. Due to computation reasons the context window length was chosen to be seven to the left (past) and seven to the right (future) of the syllable with the exception of the linguistic categories. The following input features are presented to the NN to solve this problem on the syllable level for each context unit:

- Phonetic information: The phonetic structure of a syllable to be processed is coded here. The vowel is presented as a one out-of-n coded input using the Slovenian SAMPA phoneme set. Neighbouring phonemes of the vowel are given in four classes (plosive, fricatives, nasal and liquids) and also as a one-out-of-n coded input in a symmetric context window of four phonemes.
- Positional information: Continuous positional information gives time distances of the syllable and its vowel. Discrete information denotes whether this syllable is an initial, medial or final one within the sentence, the phrase, and the word.
- Stress information: Flags denote the stress type of a syllable within the word and the phrase.
- Linguistic categories: The used linguistic category set consists of 14 tags, which are one-out-of-n coded and presented in a context of 3 to both sides.

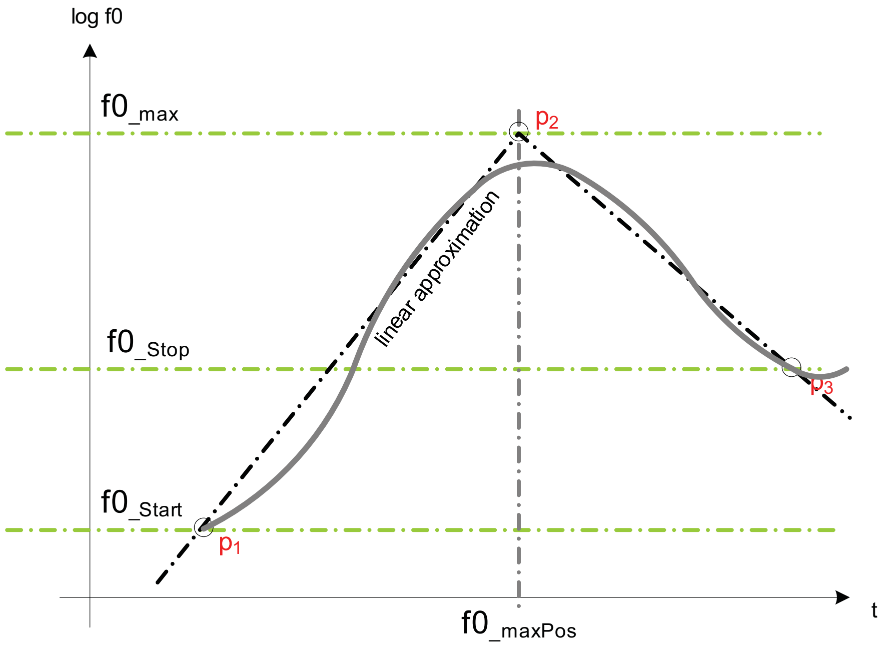

Fig. 3. Maximum based parameterization of f0-contours.

Hence by this input constellation the input dimension of x according to Fig. 2 and Fig. 4 is in the range between 500 and 600 Hz. The p-WD technique was applied to recordings of three hours of a Slovenian news speaker reading gathered isolated sentences from a large corpus as described in the foregoing section.

The patterns for training (80%) and testing (20%) were separated. A validation set of (20%) was selected randomly from the training set. The introduced parameter p in the weight decay penalty term of *p-WD* was optimized by experiments with varying parameters p (p=[0.1 - 2.0; step 0.1]). The optimum parameter was found to be p = 0.6 (Erdem & Zimmermann, 2002_a). This tuned NN module was then used to analyze the inputs and optimize the input feature selection.

6. Symbolic prosody modeling

Which parameters are the most relevant for symbolic prosody label prediction remains an open research question. A carefully chosen feature set can help to improve prediction accuracy; however, finding such a feature set is work-intensive. In addition, linguistic expert knowledge can be necessary and the feature set found can be language and task dependent. A feature set that is commonly used and seems to be relatively independent of language and task is part-of-speech (POS) sequences (Müller et al., 2000_b).

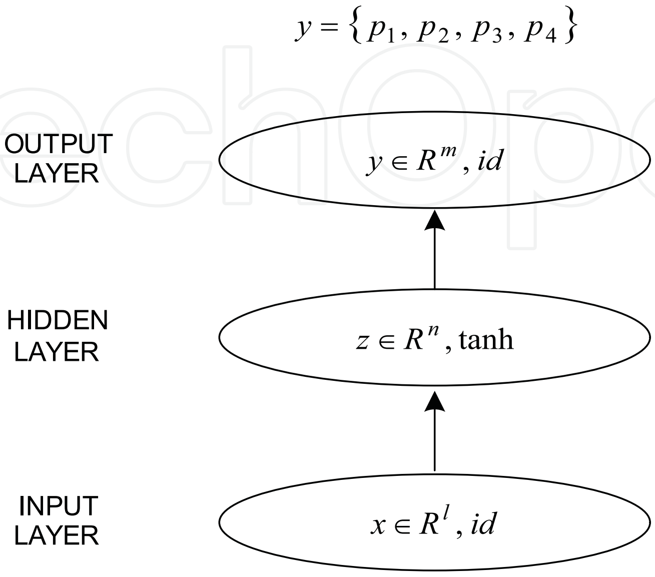

Fig. 4. NN structure for f0-contour modelling.

POS sequences of length four to the left and right of the position in question were used. For input to our prediction model the POS sequences were coded with a ternary logic (–1 for a non-active node, +1 for an active node, 0 for not valid).

Thus, for each POS tag a vector was obtained with a dimension of the size of the tag set. The size of our tag set was 13. Using a POS sequence length of four to the left and right for the Slovenian language, we achieved m = (4+1+4) * 13 = 117 dimensions.

The dimension of the applied input vector as well as tag set is similar to the German language prediction tests as reported in Müller et al., 2000_b where a tag set of length 14 was used.

6.1 Autoassociative NN classifier

We used a new approach of symbolic prosody tags prediction from POS with a NN structure based on autoassociators introduced in Müller et al., 2000_b. With the used architecture we minimized the problem of unbalanced information flow between the forward and backward path where many inputs are compressed into a single number for classification error. The architecture consists of two stages; STAGE 1 and STAGE 2 (Fig. 5).

The first stage consists of k different autoassociator models for k different classes (e.g. k=4 for B1, B2, B3, B9). Each model is trained only with data from the class it represents. The m-dimensional input vector x is mapped onto n-dimensional vector z, with n<<m. The NN are trained with the goal that the output vector x′ recovers as accurate as possible the original input x.

Thus an intermediate representation z of the data in a lower dimensional space is achieved with the compression of x via the matrix w1 and hence decompression of z via matrix w2. After training for each autoassociator a reconstruction error eREC is computed. The distance

The distance measure eREC = (x-x′)^2 is achieved through a squaring activation function of the upper cluster (Fig. 6, right) considering the difference between input x and x′ achieved using a negative identity matrix -id. The result is high dimensional error information as input into the classifier.

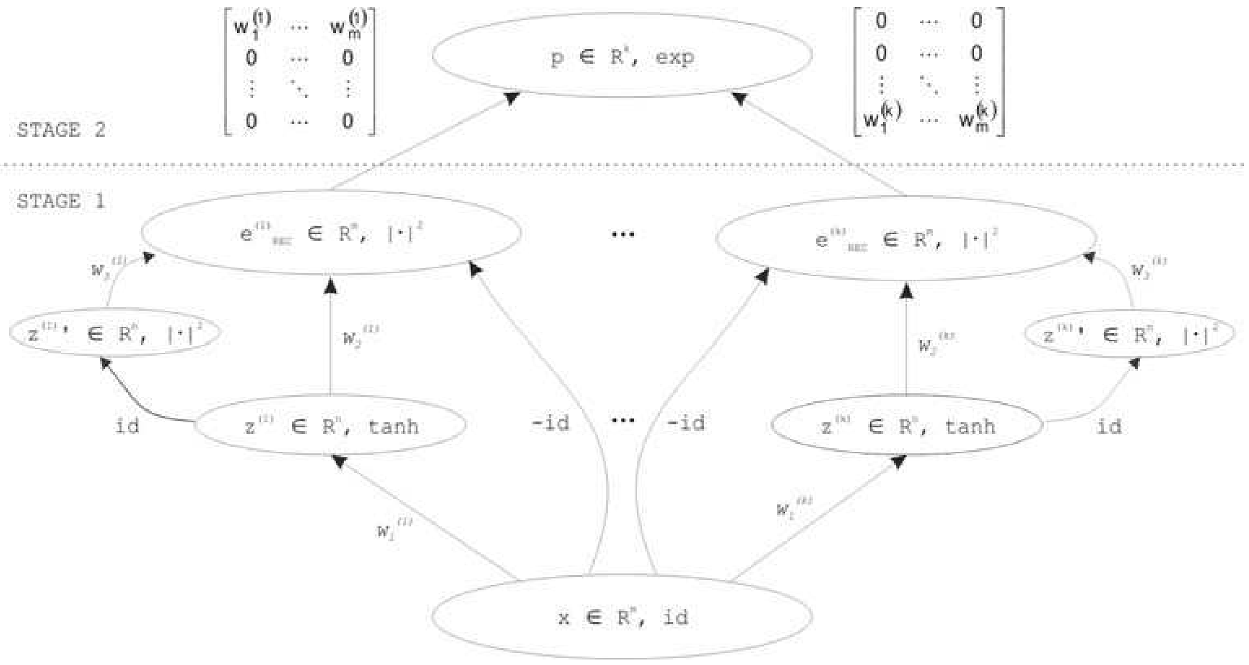
$\begin{bmatrix} w_1^{(1)} & \cdots & w_m^{(1)} \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}$
p ∈ R^k, exp
$\begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ w_1^{(k)} & \cdots & w_m^{(k)} \end{bmatrix}$
STAGE 2
STAGE 1
e$^{(1)}_{REC}$ ∈ R^n, |·|^2
...
e$^{(k)}_{REC}$ ∈ R^n, |·|^2
w$_3^{(1)}$
w$_3^{(k)}$
z$^{(1)}$' ∈ R^h, |·|^2
w$_2^{(1)}$
w$_2^{(k)}$
z$^{(k)}$' ∈ R^h, |·|^2
id
z$^{(1)}$ ∈ R^h, tanh
-id
...
-id
z$^{(k)}$ ∈ R^h, tanh
id
w$_1^{(1)}$
w$_1^{(k)}$
x ∈ R^n, id

Fig. 5. Autoassociative NN classifier.

The performance of the autoassociator can be improved with augmentation of the coordinate transformation $x' = w_h \cdot tanh(w_i \cdot x)$ additionally taking the squared representation of z into account (Fig. 6, right).

$$p_i = \frac{e^{\left(x - w_2^{(i)} \tanh\left(w_1^{(i)} \cdot x\right)\right)^T diag\left(w_1^{(i)}, w_2^{(i)}, ..., w_m^{(i)}\right)\left(x - w_2^{(i)} \tanh\left(w_1^{(i)} \cdot x\right)\right)}}{\sum\limits_{j=1}^{k} e^{\left(x - w_2^{(i)} \tanh\left(w_1^{(i)} \cdot x\right)\right)^T diag\left(w_1^{(i)}, w_2^{(i)}, ..., w_m^{(i)}\right)\left(x - w_2^{(i)} \tanh\left(w_1^{(i)} \cdot x\right)\right)}} \tag{12}$$

In STAGE 2 these detailed error information is used to determine which class (model) a given pattern on input x probably belongs to. The classifier is a NN that calculates the class conditional probabilities $p_i = p(x \in class_i)$ from the reconstruction error vectors of the different autoassociator models in Eq. (12). The experimental results confirm our assumption that the characteristics of the different classes can be captured by such autoassociators (Müller et al., 2002).

Our tests of phrase break prediction were performed with limited labelling material available (Stergar et al., 2003), app. ½ compared to Müller et al., 2000_b. The results are comparable to those for German and English (Müller et al., 2000_b; Black & Taylor, 1997). For the prediction of breaks (B correct), the results are equivalent to the achieved accuracy prediction of B correct (77.67 %) for German and nearly equivalent to the achieved accuracy prediction of B correct (79.27 %) for English despite the reduced inventory of clauses used for training.

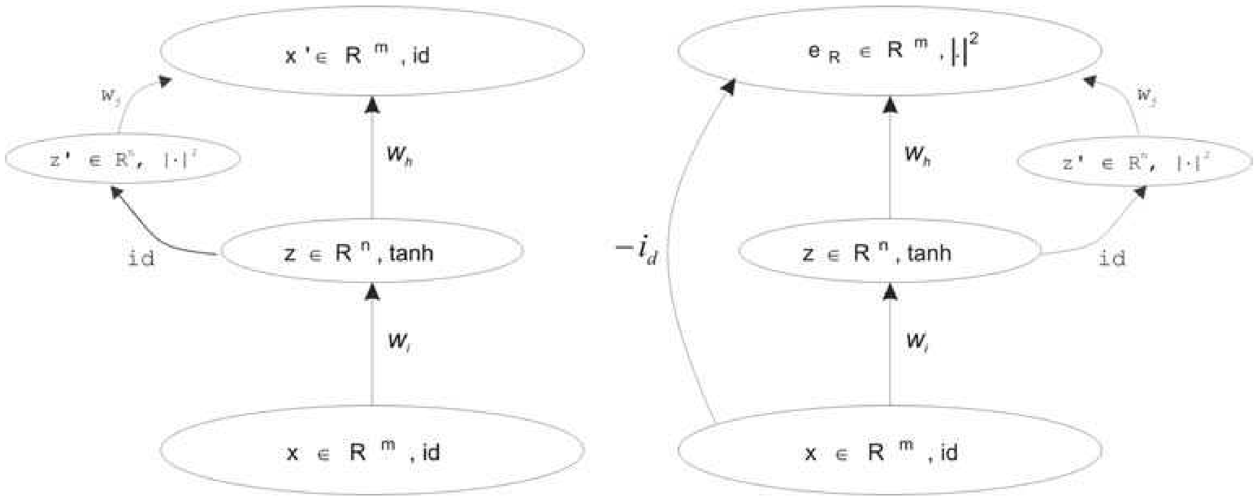

Fig. 6. Left: an autoassociator NN trained for a single class. Right: an autoassociator used for computation of reconstruction error.

7. Duration control

Different methods for duration control have been proposed. In Klatt, 1987, a rule based duration control method was presented, which depending on rules, modifies the duration of a segment by a multiplicative or additive scaling factor. In van Santen & Olive, 1990 the authors differentiate between different duration models, a data-driven method with good generalization abilities is presented in Campbell, 1992 using a NN for the syllable duration control.

Our acoustic prosody module consists of a duration control and f0-contour unit (Fig. 7). In the first approach the introduced duration control module used a classification and regression tree (CART) like method. The parameters for the nodes were derived from triphone clusters obtained by a tree-based clustering algorithm provided by standard clustering within HTK (Holzapfel, 1999). This approach will not be explained in detail. Additionally this statistic method was enhanced by a more sophisticated statistical approach considering larger contextual information on syllable and word level. Nevertheless there is a dilemma between robustness and significance of the statistics as the multilingual aimed TTS system utilizes a restricted speech database (app. 1000-1200 sentences). Therefore a NN approach is employed to solve this dilemma due to its generalization property.

As depicted in Fig. 7 both units' duration control and f0 modelling are modelled by NN. The segmental duration control is handled first – those segmental durations generated by the duration control unit are afterwards used as inputs to the f0-generation unit.

Continuous positional information of syllables is derived from these durations, which are important for the f0-generation task. A segmental duration module has to control the rhythm of a synthetic voice and the known effect of final lengthening.

Similar to the f0-contour prediction task the duration control unit uses left (past) and right (future) contextual information to establish the prediction. The input features of both modules are very similar. They are organized on syllable level for the f0-contour prediction and on phoneme (triphone) level for the duration control task. The state-of-the-art causal retro-causal modelling was presented in (Zimmermann et al., 2000) for the f0-prediction task. The shortcomings of that architecture are a fix-point recurrence, which causes stability problems during training, and a non-observance of the mentioned structural switching. The causal retro-causal error-correction (CRCEC) NN architecture is used as a basis for the modelling of the duration control task. Different architectures will be presented to overcome these two problems. First basics (finite unfolding, error correction) and new partial retro-causal expansion for the integration of the structural switching are discussed. Also the asymmetric architectures are mentioned which overcome the structural switching of the information flow. We will conclude with the implementation of the asymmetric modelling in the NN applied.

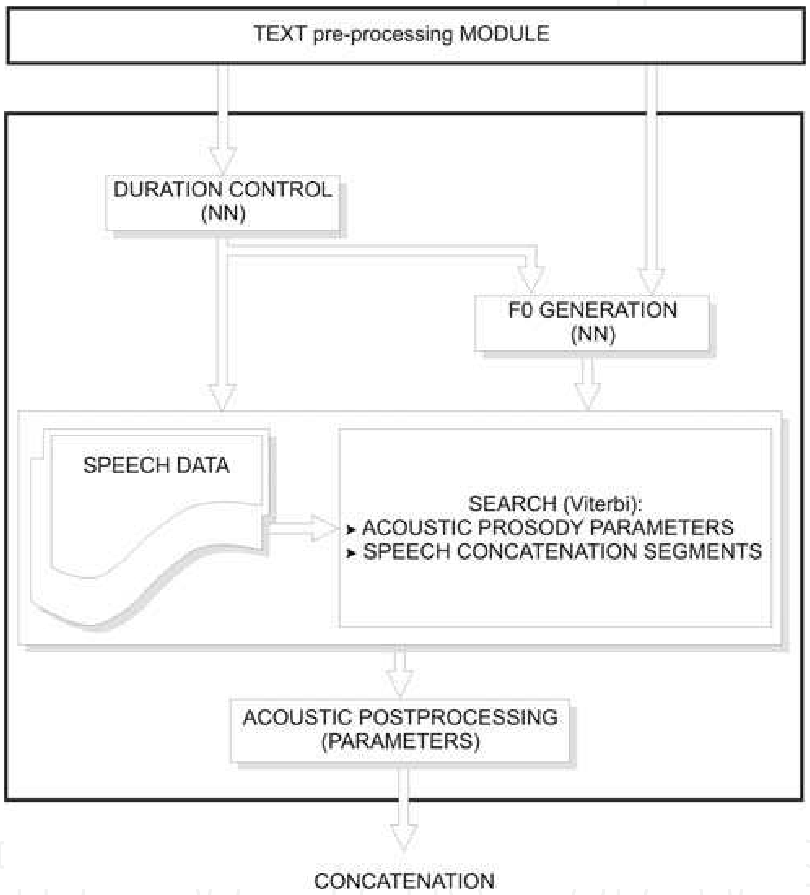
TEXT pre-processing MODULE
DURATION CONTROL (NN)
F0 GENERATION (NN)
SPEECH DATA
SEARCH (Viterbi):
▸ ACOUSTIC PROSODY PARAMETERS
▸ SPEECH CONCATENATION SEGMENTS
ACOUSTIC POSTPROCESSING (PARAMETERS)
CONCATENATION

Fig. 7. The acoustic part architecture of the used TTS system.

7.1 The causal retro-causal error-correction architecture

The causal retro-causal error-correction NN architecture (CRCECNN) for one time step (solid lines) is depicted in Fig. 8. This architecture uses shared weights and has a symmetrical extension to the neighbouring time steps (dotted lines). As can be seen there are two different information flows. In the upper part, there is a causal information flow denoted by the *matrix A* carrying state information (s_{t+i}) of the dynamics between neighbouring state clusters. This path allows the mapping of long-term forecasts. The *matrix F* in the lower part gives a retro-causal information flow (r_{t+i}). Within each time step *i* there are two error-correction parts incorporated. Both are coupled by the usage of one output cluster z_{t+i}. The error-correction will be explained using the causal information flow path.

While *matrix B* introduces external information u_t to the system, the *matrix C* transforms the state s_t to its expectation y_t. *Matrix D* propagates the model error (the expectation y_t being compensated by the observation y_t^d) to cluster s_{t+1}. The path:

$$s_t \to C \to z_t \to D \to s_{t+1} \tag{13}$$

allows to map local structures as shocks or short term effects (Eq. 13). The z-clusters represent the output clusters of the NN architecture. In cluster z_t the difference $z_t = C \cdot s_t - y_t^d$ (forecast error) between the expectation of the NN and the observation y_t^d is computed. Note that y_t^d is propagated by identity *Id* to z_t.

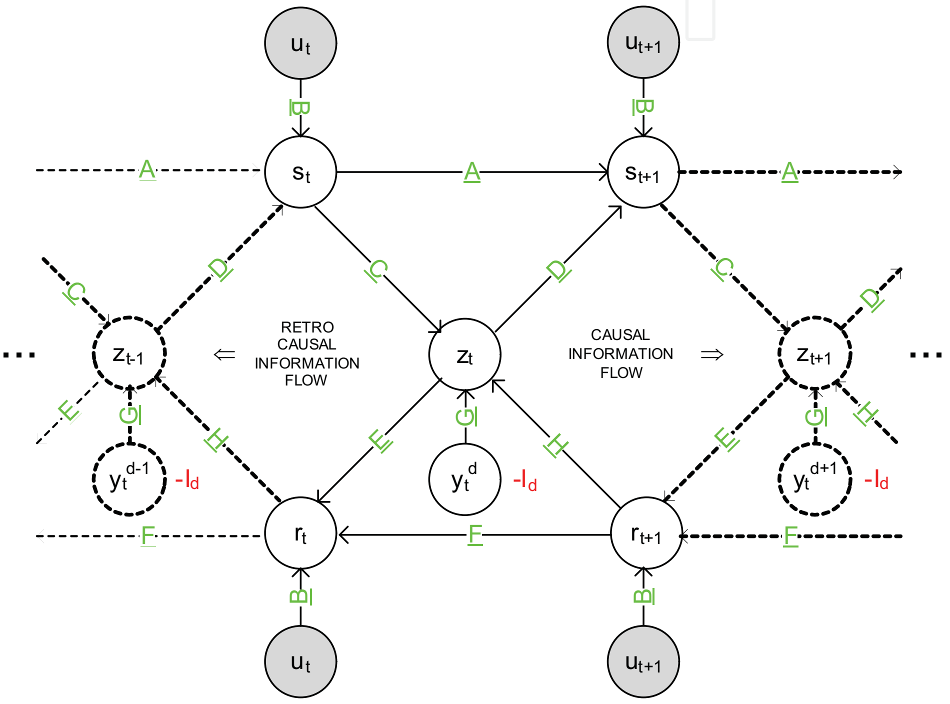

Fig. 8. Causal Retro-Causal structure of modelling.

This difference has its optimum in zero, since this denotes no forecast error having no forecast error results in a perfect description of the dynamics. Therefore the target vector z_t is set to zero during training.

If there is no mismatch between expectation and observation, then no further information is propagated by *matrix D* to state s_t and we almost obtain a simple finite unfolding NN. Existing mismatch delivers further input information to state s_{t+1}. This information is used during training for the adaptation of parameters. By this error correction principle we obtain in z_t an internal vector driving the transition of the system state together with external input u_t and previous states. These internal vectors generate the error flow, computed at each output cluster time step of the unfolding. If the internal autoregressive part coded in *matrix A* and all external driving forces of a dynamics are known, it would be possible to give a perfect description of the dynamical system. But if it is not possible to identify the dynamics due to missing or unknown externals or noise, the last model error is an indicator of the model misspecification. Since the model error is used as a measure of unexpected shocks, the learning of false dependencies is lowered and models generalization ability is improved (Rumelhart et al., 1986; Zimmermann et al., 2002).

Incorporating information flow from the right to the left captures retro-causal dependencies. If this is handled symmetrically over all time steps, this modelling result in fix point recurrences as depicted in Zimmermann et al., 2000. One closed loop is given by:

$$s_t \to C \to z_t \to E \to r_t \to H \to z_{t-1} \to D \to s_{t+1} \tag{14}$$

These fix point recurrences complicate computation of the resulting CRCECNN. Therefore a partial symmetric expansion in the following subsection which results in a *partial CRCECNN* (P-CRCECNN) is an approach to solve the stated problem (Erdem et al., 2002_b).

7.2 The partial causal retro-causal error-correction architecture

The NN depicted in Fig. 9 utilizes shared weights and finite unfolding. The coupling of both information flows is realized by only one output cluster z_t instead of the coupling at each time step within CRCECNN. By coupling that information flows within the present time step this new architecture does not contain fix-point recurrent loops, which might cause instabilities during training. In the following this architecture will be used for further adaptations applying structural switching.

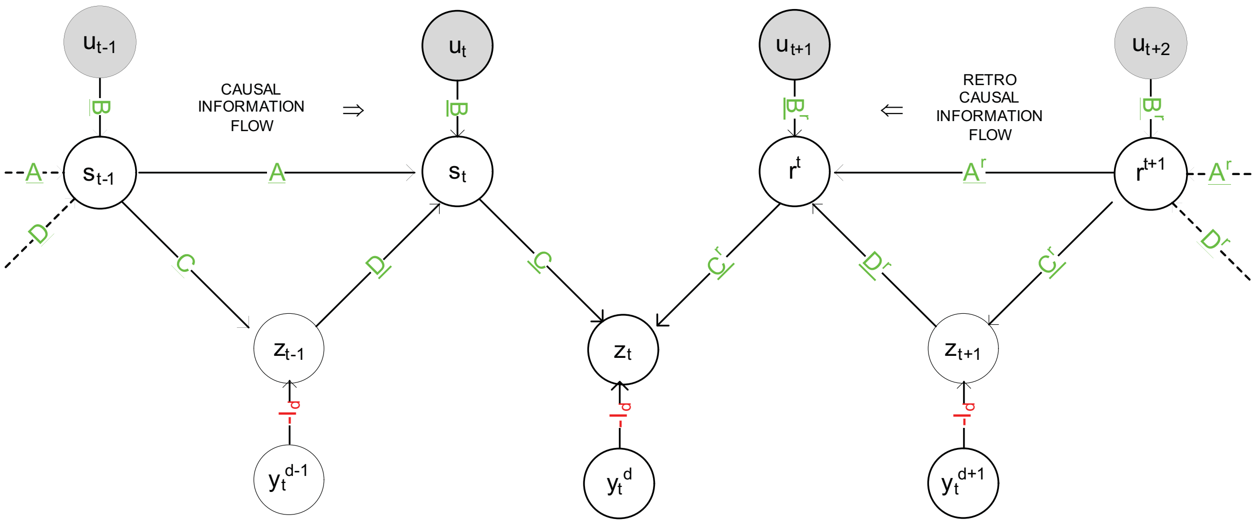

Fig. 9. The structure of P-CRCECNN.

During training all segmental durations modelled as observations are known. But within the application there are no observations available for i ≥ 0, because they are not predicted yet. For i < 0 predictions of the NN are re-utilized as observations. Because of this mismatch between training and application the retro-causal information flow has to be treated in a specific way. In the following to different ways of asymmetric P-CRCECNN are explained which overcome this mismatch. In Fig. 10 the idea of removing connections after training is depicted. The dotted connections *matrix* C^r and *matrix* D^r are trained. So the architecture is the same as depicted in Fig. 9 during training. But within the application connections $\boldsymbol{C}^r$ and D^r are removed. The resulting architecture is then a finite unfolding in time without the error correction principle for the retro-causal information flow during application. The next architecture is established by using finite unfolding in time for the retro-causal path during training and application (Fig. 10, the removed nodes and connections are shown as transparent).

7.3 The implementation in duration control NN module

In the following the application of asymmetric P-CRCRECNN's within the segmental duration control unit of our acoustic prosody NN module is presented. These data-driven methods are applied to recordings of approx. three hours of a Slovenian database as already described (The corpus). The patterns for training (80%) and testing (20%) are separated. A validation set of (20%) is selected randomly from the training set. The used database is the same as applied within the f0-generation task described in the foregoing section (f0 generation). The f0-generation task utilized patterns organized on syllable level – within this task, patterns are organized on triphone level.

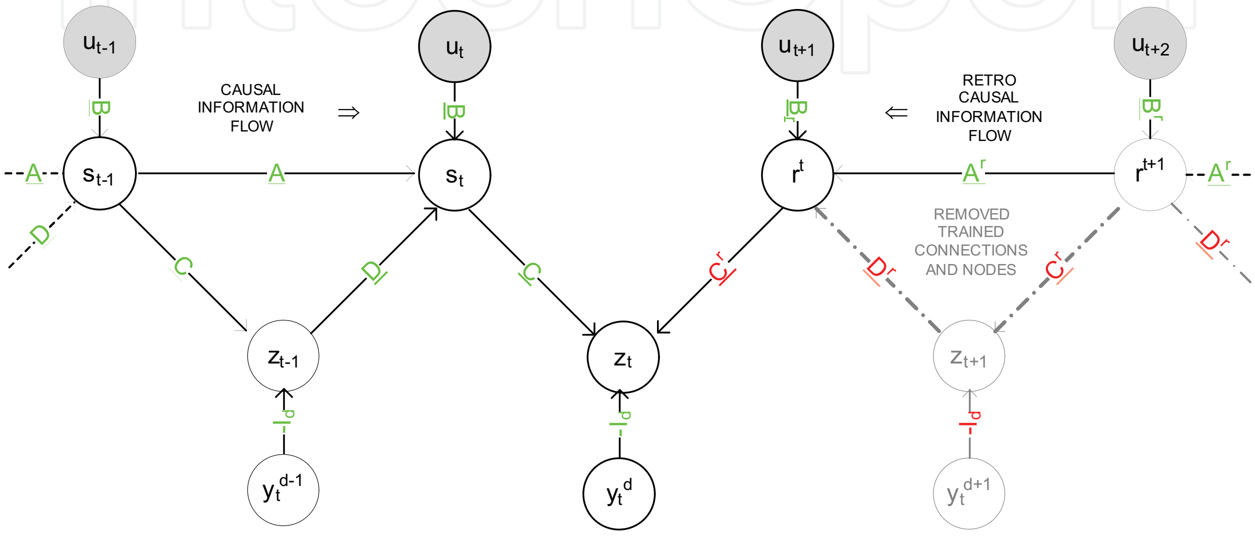

Fig. 10. Topology of modified P-CRCECNN removing trained connections with finite unfolding.

The following information (extracted from the database) is presented to the NN input in a context of seven phonemes to the left and right:

- Phonetic information: with one-out-of-n coding the phoneme index is presented here. A phoneme-set of 45 phonemes is used. Additionally the four phoneme classes (vowel, fricative, nasal, liquid, and plosive) are presented here.
- Positional information: discrete information denotes whether the according syllable is an initial, medial or final one within the phrase and the word. Continuous information is given by the relative syllable position within a sentence and phrase.
- Stress information: flags denoting the stress type of the according syllable are coded here. Four flags present Word level stress (prominence as described in section symbolic prosody). Sentence level stress consists of two stress marks.
- Linguistic categories: a one-out-of-n (set of 13 categories) coded linguistic category (POS) denotes the category type of the according word.

All listed input categories are presented at each time step of the unfolding clusters denoted by u_{t+i}. The according output vectors are modelled as observations and are presented at each time step in the clusters denoted by $u^d{}_{t+i}$. Target values for the NN are normalized to ensure an optimized signal-flow during training of the NN due to *tanh* activation function within the causal and retro-causal state clusters. A first normalization of segmental duration is obtained by the mean and standard deviation value from the used triphone classes. A second normalization was necessary to ensure an optimized signal flow during training of the NN. The mean and standard deviation were derived from the first normalized segmental durations.

For evaluation the trained NN were used to predict segmental durations of sentences that were in the test set. Three audio-files are generated with those predictions that are then used for evaluation.

The experiments showed that the best results are obtained with the P-CRCECNN modified with removing connections after training (Fig. 10). The obtained test showed that 85,6% of phrase breaks were realized with a clear final lengthening (Erdem et al., 2002_b).

As perception is a highly complex process not necessarily modelled appropriately by isolated physically distances, informal listening test were performed. Files generated by re-synthesis utilizing the different presented NN-architectures and the original one were presented to non-expert listeners. They had to judge which of the presented files were most/least pleasant. The target group also had to give a ranking. This ranking was then scaled on a value set from 1 to 5, with 5 denoting the acoustically most pleasant sentence and 1 reserved for unacceptable ones. The asymmetric P-CRCECNN with connections removed after training was evaluated to be most pleasant (average rating of 3,27). This architecture uses the error correction principle within the retro-causal path for modelling local prosodic structures. Hence it seems to help the long term forecast path improving its generalization ability, as this NN performs better than the PCRCECNN with final unfold, which does not utilize error correction. However the long term forecast path within P-CRCECNN structures with final unfold also has the ability to capture short time events.

7.4 Unit selection and Multi-level Viterbi-search algorithm

The unit selection module within the introduced multilingual TTS system uses a robust unit selection method based on syllable prosody parameters optimization (Erdem et al., 2002_c).

First isolated NN predictions of f0-contours (Erdem et al., 2002_a) and segmental durations (Erdem et al., 2002_b) are performed and then these parameters are re-utilized for a search in speech data (corpus) for best fitting of speech segments and acoustic prosody parameters. This search is realized by using a modified multi-level Viterbi-search algorithm that operates on syllable level but explicitly allows higher and lower levels of speech segments in the path search procedure. The selected units (words, syllables and triphones) are chosen from different utterances. Dealing with a limited database it is likely not all specified targets are fulfilled by the units found in the database. Thus a post-processing on the prosody parameters at the selected unit boundaries is necessary.

Regarding the phonetic and perceptive criteria it is crucial to find optimal speech segments. The optimization of physical distances does not necessarily result in a naturally sounding synthesized voice. Perception is a highly complex process, which usually can’t be modeled appropriately only by tackling single physical distances between segments. Evaluating the target distance for a candidate unit, or distance between a pair of units to be concatenated, returns only a physical measure of distance, and is not necessarily a reliable indicator of the perceived distortion that may occur (Holzapfel & Campbell, 1998). Therefore nonlinear weighting of the partial suitabilities by multiplying them in order to obtain a global suitability function is applied:

$$\mathrm{S}_{\mathrm{global}} = \prod_{n} CSC \cdot \prod_{i} LSC \qquad (15)$$

where
CSC = Continuity Suitability Cost (based on phonetic context, prosodic context and acoustic concatenation cost),
LSC = Local Suitability Cost (based on phonetic context, duration, log power and mean f0).

The computation of LSC is based on syllable level prosodic targets, as a syllable level maximum based description (Heuft et al., 1995) of f0-contours for non-tonal languages was used to train the f0-NN. The following targets are utilized for LSC computation (syllable level):

- f0-contour target parameters: p1 (initial f0-value), p2 (maximum f0-value), p3 (final f0-value), p4 (maximum position).
- segmental durations of the triphones,
- power of triphones.

Each LSC has to be in an acceptable range for an acceptable unit candidate. In contrast to a linear weighting by adding each partial suitability (Hunt, 1996) one single partial high mismatch already leads to a low overall S_{global}.

To calculate the different suitabilities a fuzzy logic motivated nonlinear suitability function is used that is composed of two half Gaussians and one constant region. The constant region is dissected to two non-equal regions representing a threshold. Within that threshold, distances to target values have no perceptual influence. Two parameters S_L and S_R control the shape of the Gaussian function.

$$S_{S_X}(t) = e^{\left(\frac{\left(t - t_T + Y\right)^2}{2 \cdot S_X^2}\right)} \tag{16}$$

where
S = nonlinear suitability function,
S_X = Left/Right distance in the Gaussian part of the local suitability cost function LSC; X= {L, R},
Y = threshold region.

S_L and S_R regulate the influence of a special target parameter. A small value of both parameters indicates a sharp criterion for selection, as for the same distance to the target a lower suitability is returned (Eq. 16). All the above mentioned targets (p1-p4) are calculated in this way.

For the CSC calculation a simple but efficient solution was experimentally selected instead of the original time consuming signal processing based on spectral analysis. Experiments showed only minor degradation in speech quality (Erdem et al., 2002_c).

7.5 Post-processing

Dealing with limited speech data (segments) for synthesis makes signal processing on speech elements at concatenation points unavoidable. Therefore we used simple but efficient post-processing on the selected segments prosody parameters. This new method was already applied and tested within the TTS system PAPAGENO for a German male news speaker (Erdem et al., 2002_c). It could be shown that it improves the quality of the used prosody generation module and of the selection process.

It was observed that the used NNs are giving good prosody modelling results within macro prosody. Therefore the general idea of this post-processing is a realignment of the obtained f0-contours according to the run of the f0-maxima of the triangles (Fig. 11). A shift is applied that operates on three levels (word-, syllable, and triphone-level). Values Δ1 and Δ2 give the difference to the according f0-maximum of the syllable or word predicted by the NN. After this shift a jump discontinuity may occur therefore a smoothing is applied using a declining linear function on each side of the jump discontinuities (Erdem et al., 2002_b).

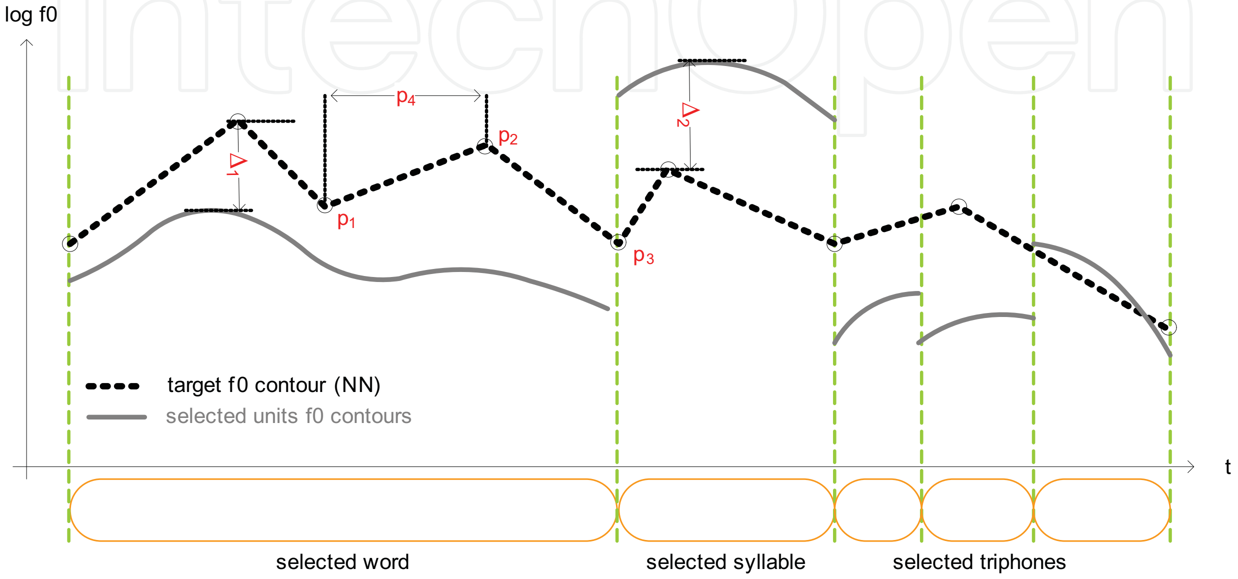

Fig. 11. Contours in selected units with jump discontinuities at unit boundaries.

After the post-processing, the f0-contours are then realized by modifying the speech-elements using a PSOLA like algorithm for speech synthesis.

8. Conclusion

In the foregoing sections we introduced the adaptation of a multilingual TTS system to Slovenian language. In the starting sections we explained our approach in the design of a suitable database used for adaptation of all modules in the used TTS system. We presented the procedure applied in creation of the final corpus, and steps taken for necessary basic pre-processing procedures (grapheme to phoneme conversion, insertion of syllable breaks, syllable stress marks placement, transcription and phonetic segmentation). All steps were performed completely automatically. We introduced a semi-automatic approach in symbolic tags marking for hierarchical prosody modelling used in the acoustical part of TTS system. The presented procedure for phrase breaks labelling is based on HTK tags for silence and is performed semi-automatically. The automatically selected tags were manually verified also other important marks had to be inserted after the process of verification manually. With the introduced approach we accelerated hand labelling and contributed to consistency in the labelling procedure. In comparison to other results our approach shows almost the same or slightly improved prediction performance with 50% less data used for training. However we have to mention the differences (selective labelling) in the labelling procedure used in our approach. In more detail we explained the used adaptable acoustical architecture combined of four modules. The first module introduced was the duration control NN module. We emphasized its basic structure with the new p-WD method applied. The p-WD method helps to select one of the highly correlated features in contrast to standard weight-decay. Hence through its penalty function we achieved a minimized input feature set. The NN duration control module introduced uses the modified causal retro-causal error correction architecture (CRCECNN). With the introduced architecture the module error is used as a measure of unexpected shocks, the learning of false dependencies is lowered and module generalization ability is improved. The fix point recurrences computation difficulties were solved with the proposed partial CRCECNN architecture. The performed experiments confirmed the suitability of the P-CRCECNN architecture. The problem of finding optimal speech segments was also mentioned. We used an approach of segment selection using a global parameterized non-linear suitability function in combination with a modified multi-level Viterbi search algorithm. Nevertheless due to the fact of a limited database a post processing approach had to be implemented.

The acoustical results of our adapted multilingual TTS system were presented to a group of 20 non-expert listeners. We generated an inventory of 216 test sentences not used for the training or validation process. The test performed during a 3 hour session (approx.) showed that our approach of adapting a multilingual TTS acoustic architecture based on NN architectures is suitable and promising. The average rating (1-5) was good-very good (3,28). We also conclude that the implementation of symbolic prosody tags into the architecture of acoustic modelling essentially contributed to naturalness of the synthesized speech without influencing the intelligibility of synthesized sentences.

9. References

Bishop C. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press, ISBN 978-0-19-853864-6, Oxford

Black A. W., Taylor P. (1997). Assigning Phrase Breaks from Part-of-speech Sequences, *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech)*, pp. 995-998, Rhodes, Greece

Boersma P., (1993). Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound, *Proceedings 17*, pp. 97-110, Institute of Phonetic Sciences, University of Amsterdam

Campbell N., (1992). Syllable-based segmental duration, In: *Talking Machines: Theories, Models and Designs*, Bailly G., Benoit C. and Sawallis T. R., (Ed.), pp. 211-224, Elsevier Science Ltd., ISBN: 978-0444891150, North-Holland

Campbell N., Mokhtari P., (2003). Voice Quality: The 4th Prosodic Dimension, *Proceedings of the 15th International Congress of Phonetic Sciences,* pp. 2417-2420, Barcelona, Spain

Dutoit, T. (2008). Corpus-Based Speech Synthesis, In: *Handbook of Speech Processing*, Benesty J., Sondhi M., Huang Y., (Ed.), Springer Handbook of Speech Processing, pp. 437-455, ISBN: 978-3-540-49125-5, Springer Berlin Heidelberg

Edgington M., Lowry A., Jackson P., Breen A. P., Minnis S. (1996), Overview of current text-to-speech techniques II – Prosody and speech generation, *BT technology journal*, Vol. 14, No. 1, pp. 84-99, Springer Dordrecht, ISSN 1358-3948, Holland.

Erdem C., Beck F., Hirschfeld D., Hoege H., Hoffman R. (2002_c). Robust unit selection based on syllable prosody parameters, *Proceedings of 2002 IEEE Workshop on Speech*

Synthesis, pp. 159 – 162, ISBN: 0-7803-7395-2, Santa Monica, California, USA, Sept. 11-13, 2002

Erdem C., Holzapfel M., Hoffmann R. (2000). Natural F0-contours with a new Neural-Network-hybrid approach, *Proceedings, 6th International Conference on Spoken Language Processing* (*ICSLP 2000*), pp. 227-230, Beijing, China, Oct. 16-20, 2000

Erdem C., Zimmermann H. G. (2002_a). A data-driven method for input feature selection within neural prosody generation, *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing ICASSP 2002*, pp. 477-480, Orlando, Florida, May 13-17, 2002

Erdem C., Zimmermann H. G. (2002_b). Segmental duration control by time delay neural networks with asymmetric causal and retro-causal information flows, *Proceedings of the 10th European Symposium on Artificial Neural Networks (ESANN 2002)*, pp. 269-274, ISBN 2-930307-02-1, Bruges, Belgium, Apr. 24-26, 2002

Hain H. U., Zimmermann H. G. (2001). A Multilingual System for the Determination of Phonetic Word Stress Using Soft Feature Selection by Neural Networks, *4th ISCA Tutorial and Research Workshop (ITRW) on speech synthesis (SSW4)*, Blair Atholl Palace Hotel, Perthshire, Scotland, Aug. 29 – Sept. 1, 2001

Hain H. U. (1999). Automation of the training procedure for neural networks performing multilingual grapheme to phoneme conversion, *Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech'99)*, vol. 5, pp. 2087-2090, Budapest, Hungary, Sept. 5-9, 1999

Haury R., Holzapfel M. (1998). Optimisation of a Neural Network or Pitch Contour Generation, *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '98)*, Vol. 1, pp.297-300, Washington State Convention Center, Seattle, Washington, USA, May 12-15, 1998

Heuft B., Portele T., Höfer F., Krämer J., Meyer H., Rauth M., Sonntag G. (1995). Parametric Description of F0-Contours in a Prosodic Database, *Proceedings of the 13th International Congress of Phonetic Sciences (ICPHS 95)*, Vol. 2, pp. 378-381, Stockholm, Sweden, Avg. 13-19, 1995

Holzapfel M. (1999). HMM based database segmentation and unit selection for concatenative Speech Synthesis, The Journal of the Acoustical Society of America, Volume 105, Issue 2, p.1031, Feb., 1999

Holzapfel M., Campbell N., (1998). A Nonlinear Unit Selection Strategy for Concatenative Speech Synthesis Based on Syllable Level Features, *Proceedings of the 5th International Conference on Spoken Language Processsing (ICSLP 1998)*, Vol. 6, pp. 2755-2758, Sydney Convention and Exhibition Centre, Darling Harbour, Sydney, Australia, Nov. 30 – Dec. 4, 1998

Hozjan V., Stergar J. (2002). Determination of prominence accent of prosodic segments in emotional speech, *Advances in Speech technology: 8th International Workshop*, pp. 229-235, Faculty of Electrical Engineering and Computer Sciences, Maribor, Slovenia, 2002

Kačič Z., Zemljak M. (1999). SAMPA - computer readable phonetic alphabet. The WEB portal of Department of Phonetics and Linguistics, University College London. http://www.phon.ucl.ac.uk/home/sampa/slovenian.htm

Klatt D., (1987). Review of text-to-speech conversion for English. Journal of the Acoustical Society of America, 82 (3), pp. 737-793, Sept., 1987

Kompe R., (1997). Prosody in Speech Understanding Systems, *Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence*, 1st Ed., ISBN: 978-3540635802, Springer Verlag, Berlin Heidelberg,

Mihelič F., Gros J., Nöth E., Dobrišek S., Žibert J. (2000). Recognition of Selected Prosodic Events in Slovenian Speech, *Proceedings of the 2nd Conference on Language Technologies*, pp. 45-48, ISBN 961 6303-25-2, Cankarjev dom, Ljubljana, Slovenia, Oct. 17-18, 2000, Institut Jožef Stefan, Ljubljana

Müller A. F., Stergar J., Horvat B., (2002). Designing Prosodic Databases for Automatic Modeling of Slovenian Language in a Multilingual TTS System*, Proceedings of the 3rd international conference on Language resources and Evaluation, LREC 2002*, Las Palmas, Canary Island, Spain, May 18-26, 2002

Müller A. F., Tao J., Hoffmann R. (2000_a). Data-driven importance analysis of linguistic and phonetic information, *Proceedings, 6th International Conference on Spoken Language Processing* (*ICSLP 2000)*, pp. 227-230, Beijing, China, Oct. 16-20, 2000

Müller A. F., Zimmermann H.G., Neuneier R. (2000_b). Robust Generation of Symbolic Prosody by a Neural Classifier Based on Autoassociators, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing* (ICASSP 2000), vol. 3., pp.1285-1288, Istanbul, Turkey, June 5-9, 2000

Prechelt L. (1998). Early Stopping - But When?, In: *Neural Networks: Tricks of the Trade*, Orr G. B. and Müller K. R., (Ed.), pp. 55-69, ISBN 3-540-65311-2, Springer Verlag, Berlin, 1998.

Rojc M., Kačič Z. (2000). Design of Optimal Slovenian Speech Corpus for use in the concatenative Speech Synthesis System, *Proceedings of the 2rd international conference on Language resources and Evaluation (LREC 2000)*, pp. 321-325, Athens, Greece, May 31 – June 2, 2000

Rumelhart D. E., Hinton G. E., Williams R. J., (1986). Learning internal representations by error propagation, In: *Parallel Distributed Processing: Explorations in the Microstucture of Cognition, Vol. 1, Foundations*, Rumelhart D. E. and McClelland J. L., (Ed.), pp. 318-362, The MIT Press/Bradford Books, ISBN 978-0262181204, Cambridge, MA, USA July, 1996

Sluijter A., van Heuven V., 1996. Acoustic correlates of linguistic stress and accent in Dutch and American English. *Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP96)*, pp. 630-633, Wynham Franklin Plaza Hotel, Philadelphia, PA, USA, Oct. 3-6, 1996

Sonntag G. P., Portele T., Heuft B. (1997). Prosody generation with a neural network: Weighing the importance of input parameters. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1997)*, Vol. 2, pp. 931-934, Apr. 21-24, 1997

Sproat R., Olive J. (1995). An Approach to Text-to-Speech Synthesis, In: *Speech Coding and Synthesis*, Kleijn W. B., Paliwal K. K., (Ed.), Elsevier Science Inc., ISBN 978-0444821690, New York, NY, USA, December, 1995

Stergar J., Horvat B. (2003). An Environment for Word Prominence Classification in Slovenian Language, *Proceedings of the 15th International Congress of Phonetic Sciences (ICPhS 2003)*, pp. 2087-2090, Universitat Autónoma de Barcelona, Barcelona, Spain, Avg. 3-9

Stergar J., Hozjan V. (2000). Steps towards preparation of text corpora for data driven symbolic prosody labeling, *Proceedings of the 2nd Conference on Language Technologies*, pp. 82-85, ISBN 961 6303-25-2, Cankarjev dom, Ljubljana, Slovenia, Oct. 17-18, 2000, Institut Jožef Stefan, Ljubljana

Stergar J., Hozjan V., Horvat B. (2003). Labeling of Symbolic Prosody Breaks for the Slovenian Language, *International Journal of Speech Technology*, Vol. 6, No. 3, pp. 289-299, July, 2003

Tamburini F., (2002). Automatic detection of prosodic prominence in continuous speech, *Proceedings of the 3rd international conference on Language resources and Evaluation, LREC 2002*, pp. 301-305, Las Palmas, Canary Islands, Spain, May 18-26, 2002

Traber C., (1992). F0 generation with a database of natural F0 patterns and with a neural network, In: *Talking Machines: Theories, Models and Designs*, Bailly G., Benoit C. and Sawallis T. R., (Ed.), pp. 287-304, Elsevier Science Ltd., ISBN: 978-0444891150, North-Holland

van Santen J., Olive J.(1990). The Analysis of Contextual Effects on Segmental Duration, *Computer Speech & Lanuage*, Vol. 4, Issue 4, pp. 359-390, Oct., 1990

van Santen J., Mishra T., Klabbers E., (2008). Prosodic Processing, In: *Handbook of Speech Processing*, Benesty J., Sondhi M., Huang Y. (Ed.), pp. 471-487, Springer, ISBN: 978-3-540-49125-5, Springer Berlin Heidelberg

Vereecken H., Martens J. P., Grover C., Fackrell J., Van Coile B. (1998). Automatic prosodic labeling of 6 languages, *Proceedings of the 5th International Conference on Spoken Language Processsing (ICSLP 1998)*, Vol. 4, pp. 1399-1402, Sydney Convention and Exhibition Centre, Darling Harbour, Sydney, Australia, Nov. 30 – Dec. 4, 1998

Vereecken H., Vorstermans A., Martens J. –P. and Van Coile B. (1997). Improving the Phonetic Annotation by means of Prosodic Phrasing, *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech 1997)*, Vol. 1, pp. 179-182, Rhodes, Greece, Sep. 22-25, 1997

Zimmermann H. G., Neuneier R., Grothmann R., (2002). Modeling of Dynamical Systems by Error Correction Neural Networks, In: *Modeling and Forecasting Financial Data, Techniques of Nonlinear Dynamics*, Soofi A. and Cao L., [Ed.], Vol. 2, pp. 237-262, Kluwer Academic Publishers, ISBN 0792376803, Boston/Dodrecht/London

Zimmermann H. G., Müller A. F., Erdem C., Hoffmann R., (2000). Prosody Generation by Causal Retro-Causal Error Correction Neural Networks, *Workshop on Multi-Lingual Speech Communication*, pp. 116-121, Kyoto, Japan, Oct. 11-13, 2000