Prosody Modelling in Czech Text-to-Speech Synthesis

Jan Romportl, Jiří Kala

Department of Cybernetics, Faculty of Applied Sciences
University of West Bohemia, Pilsen, Czech Republic
rompi@kky.zcu.cz, jkala@kky.zcu.cz

Abstract

This paper describes data-driven modelling of all three basic prosodic features – fundamental frequency, intensity and segmental duration – in the Czech text-to-speech system ARTIC. The fundamental frequency is generated by a model based on concatenation of automatically acquired intonational patterns. Intensity of synthesised speech is modelled by experimentally created rules which are in conformity with phonetics studies. Phoneme duration modelling has not been previously solved in ARTIC and this paper presents the first solution to this problem using a CART-based approach.

1. Introduction

Concatenative text-to-speech (TTS) synthesis of the Czech language has been researched, elaborated and implemented already for a significant period of time. During this period various prosody models have been proposed, yet at least to our knowledge there has not been implemented and practically applied any complex *data-driven* (in the sense of automatic training using very large real speech databases) prosody model of all three basic prosodic characteristics (i.e. fundamental frequency (F0), intensity and segmental duration altogether).

This paper tries to present such a prosody model implemented in the TTS system ARTIC, developed at the Department of Cybernetics, University of West Bohemia [1]. The model is formally based on a linguistically motivated structural prosody description framework, which explicitly separates prosodic function from its form. The fundamental frequency generation part of the model is based on our data-driven intonation model previously introduced for example in [4], whereas intensity modelling is rule based. The most recent advance presented in this paper consists in incorporating a CART-based duration model trained on a large speech corpus.

2. Prosody description framework

The prosody model used in TTS system ARTIC is based on explicit distinction between prosodic form and function. The importance of such a form of linguistic stratification has already been frequently discussed (let us at random mention for instance [2]).

2.1. Prosodic form and function

In our conception each input sentence is represented in form of a prosodic structure. The prosodic structure is a result of parsing a sentence using a specific set of linguistically motivated transformation rules collectively called *prosodic grammar*. The prosodic structure of a sentence formally corresponds to a prosodic function while a prosodic form (i.e. how prosody is eventually realized by acoustic means – "surface" prosody) is then derived from it (i.e. the allowed prosodic forms depend purely on the prosodic function together with phonotactics restrictions, not on the text or sentence itself).

In other words – the prosodic structure determines a parameterisation of input text and this parameterisation is then used in a system for prosodic form assignment (i.e. a classifier, knowledge base, unit selection algorithm, etc.). It is not a goal of this paper to fully describe the prosodic structures and grammar – the discussion on this topic can be rather found in [3]. The following paragraphs just briefly summarise some information necessary as a background for our TTS prosody model.

2.2. Prosodic grammar

The prosodic grammar tries to capture structuring of a sentence relevant for prosody functioning. Using generative-based rules it decomposes a sentence into its immediate constituents (terminals and non-terminals) and mutual relations between these constituents formalise the prosodic function. The grammar (or rather its equivalent Chomsky's normal form) is designed to be implemented in a stochastic grammar parser, which is now being developed and tested. We distinguish the following language units serving as the grammar terminal and non-terminal constituents (parenthesised symbols are used in the respective grammar rules):

Prosodic sentence (PS)

Prosodic sentence is a prosodic manifestation of a sentence as a syntactically consistent unit, yet it can also be unfinished or grammatically incorrect.

Prosodic clause (PC)

Prosodic clause is such a linear unit of a prosodic sentence which is delimited by pauses. A prosodic sentence generally consists of more prosodic clauses.

Prosodic phrase (PP)

Prosodic phrase is such a segment of speech where a certain intonation scheme is realized continuously. A prosodic clause generally consists of more prosodic phrases.

Prosodeme (P0), (Px)

Prosodeme is an abstract unit established in a certain communication function within the language system. We have postulated that any single prosodic phrase consists of two prosodemes: so called "null prosodeme" and "functionally involved prosodeme" (where (Px) stands for a type of the prosodeme chosen from the list shown below), depending on the communication function the speaker intends the sentence to have. In the present research we distinguish the following prosodemes (for the Czech language; other languages may need some modifications):

- P0 – null prosodeme
- P1 – prosodeme terminating satisfactorily (a reply is not expected)
  - P1-1 unmarked
  - P1-2 marked directive
  - P1-3 marked expressive
  - P1-4 specific
- P2 – prosodeme terminating unsatisfactorily (a reply is expected)
  - P2-1 unmarked (supplementary, "wh-questions")
  - P2-2 marked declaratory ("yes/no questions")
  - P2-3 marked disjunctive (questions with disjunctive "or")
  - P2-4 specific
- P3 – prosodeme nonterminating
  - P3-1 unmarked
  - P3-2 marked bound (involved in a function primarily held by P1 or P2)
  - P3-3 specific

Prosodic word (PW)

Prosodic word (sometimes also called phonemic word) is a group of words subordinated to one word accent (stress). Languages with a non-fixed stress position would need a stress position indicator too.

Semantic accent (SA)

By this term we call such a prosodic word attribute, which indicates the word is emphasised (using acoustic means) by a speaker.

There are two more terminal symbols used ("$" and "#") standing for pauses differing in their placement (inter- and intra-sentential). The terminal symbol *(w_i)* stands for a concrete prosodic word from a lexicon and ∅ means an empty terminal symbol. Note that *Px* is only an "abbreviation" for each prosodeme (i.e. P1-1, etc.). The rules should be understood this way: "(PC) → (PP) {1+} # {1}" means that the symbol *(PC)* (prosodic clause) generates one or more *(PP)* symbols (prosodic phrases) followed by one # symbol (pause).

(PS) → (PC) {1+} $ {1}

(PC) → (PP) {1+} # {1}

(PP) → (P0) {1} (Px) {1}

(P0) → ∅

(P0) → (PW) {1+}

(Px) → (PW) {1}

(Px) → (SA) (PW) {1+}

(PW) → w_i {1}

Figures 1 and 2 show two possible prosodic structures of the Czech sentence: "It is not a singular transformation of a long vowel into a diphthong." However, the second variant bears a semantic accent on the word "singular" so as to bring forward the contrastive focus as the opposite of e.g. "frequent".

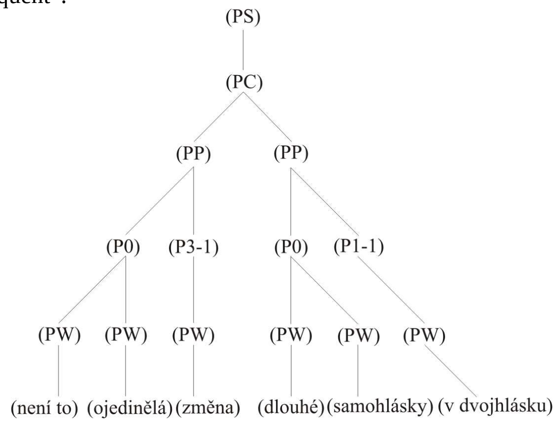

Figure 1: Czech sentence prosodic structure in a neutral form.

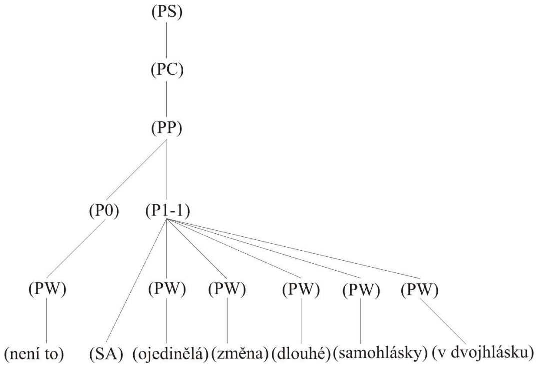

Figure 2: Czech sentence prosodic structure with a semantic accent.

It is not a simple task to infer the full prosodic structure from the surface form of a sentence. This can be done using a probabilistic grammar parser similar to a parser used for syntax analysis – on one hand the prosodic parser is simpler due to far less complex grammar, but on the other hand the relations among prosodic constituents are not as clear and straightforward as among syntactic constituents (in case of prosody many phenomena are facultative, singular or even random). Hence the goal of the prosodic parser is not to create couple of "definitely correct" prosodic structures of a given sentence; rather it should delimit a class of prosodic structures acceptable in a given context.

Because of such peculiarities we have not yet implemented fully working automatically trained parser into ARTIC and the task of prosodic structure parsing is carried out by a set of heuristic rules. These rules are obviously far from performing optimally (for example they are very inaccurate in prosodic phrase detection and semantic accents have to be omitted at all) but they are treated as a temporary solution.

3. F0 modelling

It is beyond the scope of this paper to fully describe the data-driven model of F0 implemented in ARTIC – more information on this (including the model evaluation) can be found in [4]. However, the basic idea is in conformity with the aforementioned considerations about duality of prosodic form and function.

From the formal point of view all information about prosodic function of each word is encoded in the prosodic structure itself and hence the position of the word within the structure. Therefore the prosodic form realised by means of F0 behaviour depends purely on positions of the prosodic words within the prosodic structure of a given sentence.

The position of a prosodic word ("position" not in the exact meaning – rather we would use it in the sense of mutual configuration between prosodic words and their parent prosodic constituents) is described by a set of features (we refer to it as description array – DA) which include for instance: index of the prosodic word within its neighbours with the same parent node, type of its parent node and its index (and this recursively up to the root node), and also various quantitative features concerning syllabic, stress and phoneme structure of the word. More details on DA can be found in [4].

The relation between prosodic function (formulated through DA) and its form is represented by a function in the mathematical sense, which we refer to as *realization function* (because it realizes the function through the form). The realization function is created from a suitable speech corpus (ideally the same one used for a particular speech segment database creation) with transcribed utterances, prosodic structure tags (i.e. the transcribed sentences are prosodically parsed) and F0 contours (e.g. acquired by electroglottograph measuring). Speech must be segmented at least on the level of prosodic words (i.e. time intervals of prosodic words must be known).

The F0 contours are segmented according to the prosodic words – this way the F0 contour of each prosodic word token is acquired (let us call such a segment a *sub-contour*). The corpus used in ARTIC consists of 5,000 sentences involving 55,655 sub-contours which are then clustered into so called *cadences* (abstract intonational patterns – as will be described further in the text).

3.1. Realization function

The realization function is defined as

$$R\colon DA \rightarrow I \times pot(C)$$

where $I = \{i_1, \ldots, i_l\}$ is a set of initial conditions, $C = \{c_1, \ldots, c_m\}$ is a set of cadences and $pot(C)$ is a power set of C. A cadence is an intonational pattern which fits into an interval of a single prosodic word. The set C can also be called a cadence inventory. Initial conditions say where on the frequency scale a cadence chosen for a prosodic word starts.

Fujisaki shows [5] that F0 can be modelled in a logarithmic space as a sum of outputs of two linear systems. In the linear space this summation corresponds to a multiplication of values, therefore each sub-contour (as a segment of a whole F0 trajectory) acquired from the corpus can be decomposed into two components: (a) the initial F0 value of the sub-contour; (b) the rest of the sub-contour relatively to the initial value (in its multiples).

The realization function also consists of two components. The first one is constructed from the corpus by linking each DA occurring in the corpus with the initial F0 value of the respective sub-contour occurring with this DA in the corpus. Since a particular DA is often assigned to several prosodic word tokens in the corpus, there are usually more possible initial value links. In such cases the first sub-contour with a given DA occurring in the corpus (supposing indeed arbitrary, yet constant sentence numbering) is considered – this ensures the synthesised prosodemes to be intonationally "consistent" as for the prosodic word initial conditions because the initial F0 values of the prosodic words within a particular synthesised prosodeme are all selected from the same sentence (otherwise it could happen that each initial condition in the synthesised prosodeme is selected from a different sentence, although with the same DA).

The set $C = \{c_1, \ldots, c_m\}$ (the cadence inventory) is created by a clustering algorithm based on repeated bisections and cosine similarity function, applied on all F0 sub-contours from the corpus. Prior to this, the sub-contours are represented by vectors with the dimension x (i.e. by approximating each sub-contour with x equidistant points relatively to its initial value – this ensures sub-contour normalisation over time intervals and F0 values). The elements of C (i.e. cadences) are constructed as either centroids of the clusters, or there is one (or more) vector chosen from each cluster as its representative (using various methods, such as elimination of outliers according to Mahalanobis distance).

We have experimented with various values of m (the number of cadences) ranging from 3 to 200. Good results are achieved for example with the number of clusters m=30. In this case the smallest cluster consists of 911 vectors (sub-contours) and the largest of 3571. The cadence inventory is created from the cluster centroids.

We say a cadence *belongs* to a particular DA provided that the sub-contour occurring in the corpus with this DA is an element of the cluster represented by the given cadence. The second component of the realization function is constructed from the corpus by linking each DA occurring in the corpus with the set of all cadences belonging to this DA. Thus if we have a prosodic word w_j, then

$$R(DA(w_j)) = <i_j, C_j>$$

where $i_j \in I$ is the assigned initial condition and $C_j \subseteq C$, $C_j = \{c_{j,1}, c_{j,2}, \ldots, c_{j,lj}\}$ is a set of the assigned cadences. Now let the synthesised sentence S be given as:

$$S\colon\ w_1\, w_2 \ldots w_p$$

The resulting generated F0 contour of the sentence S is then constructed from the initial conditions and cadences given by the realization function for each prosodic word w_1, … w_p – the initial conditions are F0 values at the beginnings of the prosodic words and the cadences actually fill the gaps between neighbouring initial conditions by F0 values calculated as multiples of the initial conditions. As it can be seen from the definition of the realization function, the set of several suitable cadences is given for each prosodic word – only one of them must be chosen at a time. This is done by a criterion function, minimised over all combinations of proposed cadencies. One of the choices for the criterion function is for example a sum of differences of F0 values on the boundaries of the prosodic words – to avoid or at least minimise F0 discontinuities in junctures where one cadence ends and the next one (based on a different initial condition) starts. This process of cadence concatenation is described together with the criterion function in more detail in [4].

3.2. Prosodic homonymy

One can easily see no corpus can offer all possible DAs and therefore it is impossible to construct the realization function ideally. Hence the crucial importance for the realization function has the *relation of indistinguishableness* [4]. Two description arrays are in the relation of indistinguishableness provided that their different deep prosodic-semantic functions can be realized by the same functor (i.e. same surface prosodic means) – two different DAs are homonymous in terms of their surface realization and thus mutually interchangeable. Informally: the realization function is defined also for those possible DAs not occurring in the corpus; namely if a set of appropriate cadences is to be determined for a DA not occurring in the corpus, another DA which occurs in the corpus and is homonymous according to the aforementioned relation, is taken instead and the set of cadences and initial conditions is determined for the new DA.

A question is how to determine the relation of indistinguishableness. The best method is probably an automatic analysis of heldout corpus data – this presupposes that the heldout data include DAs not occurring in the training data (i.e. factually unobserved) and the relation of indistinguishableness can be determined by a feasible generalisation of the mutual relation between the training and heldout data. This generalisation can be formalised for instance by a specific DA space metrics which allows to find a homonymous DA in terms of the minimum vector distance.

However, research in this field has not been successfully finished yet and thus our TTS system ARTIC must now settle for a workaround in the form of performing a number of limited perturbations of the least significant (heuristically and experimentally determined) components of an unobserved DA (e.g. exact length of a prosodic word in phonemes, exact number of prosodic clauses in a sentence, etc.) which eventually transform the unobserved DA into such a DA that occurs in the corpus and is very likely to be still homonymous.

4. Intensity modelling

It has been often discussed in Czech phonetics literature that intensity (or loudness – as a psychological correlate of intensity) is of far less importance than fundamental frequency with respect to suprasegmental features of speech, therefore our prosody model pays significantly less attention to it.

Moreover, we have undertaken theoretical considerations of modelling intensity analogically to fundamental frequency, i.e. by “intensity cadencies”. However, since intensity is much more interconnected with segmental qualities of speech, the application of such a model is not as straightforward as in the case of fundamental frequency (intensity can be treated as sort of a distinguishing feature of a phoneme, unlike F0 which is basically present at voiced phonemes and not present at unvoiced phonemes).

Considering the aforementioned, our prosody model currently incorporates only a simple rule for intensity modelling. Czech phonetics studies usually mention some increase of intensity (or perceived loudness) on stressed syllables. We have experimentally revealed that linear increase of speech signal amplitude by 1.3 on stressed syllables is well assessed by listeners evaluating the resulting synthesised speech. This is in conformity with [6] stating that stressed syllables usually feature increase of intensity level by 1 – 3 dB.

5. Segmental duration modelling

All previous versions of our prosody model did not comprise any explicit duration modelling techniques and have been using only average lengths of phonemes from segmented speech corpus. However, in our recent research we have incorporated and implemented a Classification and Regression Tree (CART) approach for segmental duration modelling, mainly because of possibility of its straightforward application and rich experience of other research teams. Our experiments are similar to [7], [8] but there is one important difference – we do not use only one regression tree for all phonemes, rather we have trained an independent tree for each phoneme (experiments with a single universal tree have reached worse score for us).

5.1. Training data

Training data for tree construction consists of 5,000 indicative sentences recorded by a female voice talent (the same data have been used also for the acoustic unit inventory creation and for fundamental frequency modelling). These recordings have been automatically segmented by a statistical approach (HMM-based). Resulting inventory counts over 400,000 phonemes where each of them has been represented by 172 features (as it is described further).

5.2. Phoneme features

For the sake of the CART-based classification each phoneme token (i.e. occurrence of a phoneme) is represented (or described) by a set of 172 features which can be methodologically divided into five groups. Since an independent tree is built for each phoneme type (the word “type” is used here in the sense of commonly understood duality “token/type” – “type” is the phoneme itself and “token” its textual occurrence), the phoneme type itself is not included among the features.

5.2.1. Basic feature groups

These groups of features are derived from phoneme types of neighbouring phonemes and their categorisation into phoneme classes such as vowel, consonant, fricative, plosive, etc.

Features defined by neighbour type form the first group:

- **previous_type/next_type** – the type of the previous/next phoneme. If the phoneme stands as the first/last one in a sentence, the symbol "_" (underscore) is used as a value of this feature.
- **previous2_type/next2_type –** the type of a phoneme which stands over one phoneme before/after. Identically as in the previous case the underscore symbol is used in case the type of the phoneme cannot be obtained.

The second group is based on membership of a phoneme type into specified phoneme classes. The classes are distinguished by various articulatory and phonational criteria (e.g. vowel quantity, sonority, articulation place and manner, etc.). Values of the features are either true or false – depending on whether a phoneme type is or is not a member of the given class.

5.2.2. Feature groups based on prosodic grammar

The next feature groups describing phonemes are based on the prosodic grammar described in Section 2 of this paper (although not all grammar attributes are used). Every sentence is thus structured hierarchically into the constituents resulting from the prosodic grammar, i.e. prosodic sentence, prosodic clause, prosodic phrase, prosodeme, prosodic words – and in addition to them – syllables and phonemes.

The constituents are hierarchically sorted from the parent ones down to their children. Each of them contains one or more child elements. For example every phoneme stands somewhere in a syllable and each syllable contains one or more phonemes; a syllable stands in a prosodic word and each prosodic word contains one or more syllables.

Features in the third group have their values derived from the “length” of a prosodic sentence constituent in the phoneme token context. This length is determined for each constituent by the number of its child constituents (the number of phonemes in a syllable, syllables in a prosodic word, etc.).

The fourth group consists of features which indicate the position of a child constituent within its parent constituent in the phoneme token context – from the beginning and from the end of the parent constituent (the numeric representation is used). Again, not just the position of the constituent within its immediate parent is used, but the positions in the whole parent hierarchy are taken into account as well.

The last group of features is similar to the previous one with the difference that the values are not represented by numbers, but positions are categorised into these possibilities:

- FIRST/LAST – the child is positioned within its parent as the first/last one (from beginning)
- MIDDLE – in other cases

5.3. Training process

The duration model training has been carried out using the *wagon* CART building program, a part of the Edinburgh Speech Tools Library. Root mean squared error (RMSE) and correlation coefficient (CORRC) values, presented in the evaluation further in this paper, have been therefore computed by *wagon*.

Since our segmented speech data contain more than 400,000 phoneme tokens, there are enough occurrences of each phoneme type and thus we have decided to train individual regression tree for each phoneme type.

The first 80 percent of sentences from the whole corpus have formed a training set and the rest of the data then has been used for testing.

5.4. Experiments

Several training and evaluation experiments have been carried out. The very first training experiments used only some of the features from the groups described in Section 5.2. However, due to poor results the feature set has then been extended to the final number of 172 features.

As described in the text above, an independent tree for each phoneme type is used, therefore the phoneme duration estimator is built as a composition of all individual regression trees where the root (i.e. first) questions is about the phoneme type. After that the algorithm continues in a standard way.

In one of the training experiments the features based on phoneme classes were excluded. However, this way we have reached too high values of RMSE and CORRC (see Table 1) and thus the approach had to be improved. The next couple of experiments were characterised by leaving out the features based on the position and then also on the categorised position because of our hypothesis these features are strongly correlated. The results of these two experiments were very similar and – most importantly – worse than without excluding any features.

The next step consisted in adding the features based on neighbour phoneme type and because this way we have achieved better results, we have expanded the feature set to the full form described hereinbefore. The results achieved by such classifier and feature configurations eventually reached the applicable level and are comparable to results presented by other reports [9], [10], [11].

Since our speech corpus segmentation is based on a statistical approach (HMM) and not conducted by human experts, it sometimes can happen that segment boundaries are placed relatively far from the position where they should be. To prevent these errors from negatively influencing segmental duration estimation we have tried to eliminate them from the training data by excluding phoneme tokens with statistically improbable duration. We have experimentally set this statistical relevance so that only phoneme tokens with duration between 5 and 95 percent fractile (computed for each phoneme type independently) have been included into the training data (sort of a “fractile pruning”). This way we have achieved the best results in terms of the values of RMSE and CORRC.

We have also performed calculation of RMSE and CORRC for a “dummy” duration estimator previously used in our system which gives each phoneme token the length equal to the average length of the respective phoneme type computed from the training data (i.e. actually no estimator because each occurrence of a certain phoneme type has the same length). The results of this experiment are quite important and illustrative since they give an idea of the theoretically lowest acceptable classifier performance. They are presented in the Table 1 as well.

5.5. Evaluation

The first aspect of evaluation of the phoneme duration estimator is mathematical (or rather quantitative). RMSE and correlation coefficient values of the previously described approaches are presented in the following table.

Approach	RMSE	CORRC
"dummy" estimator	24,47	0,85
excl. neighbour token classes	28,39	0,77
all features	22,56	0,75
all features – fractile pruning	18,89	0,92

Table 1: Duration model performance assessment

In comparison with results reported by other studies based on CART (see the Table 2), our experiments have come out slightly better (as for RMSE and CORRC). One cannot judge (concerning current research and evaluation methodology and techniques) whether this is a language or even speaker dependent phenomenon, or our set of features performs really better (the influence of the language is indubitable – e.g. more conservative duration behaviour in the Czech language in comparison with English). However, our model is still not in its final version and we will continue to analyse the results in more detail.

Language [source]	RMSE	CORRC
German [9]	22,71	0,83
English [10] (voice *lja*)	21,00	0,78
English [10] (voice *rjs*)	20,00	0,80
English [10] (voice *erm*)	24,00	0,82
Korean [11]	26,48	0,73
Czech [7]	20,30	0,79
Czech – this paper	18,89	0,92

Table 2: Results comparison with other studies

The second, for our work actually more important aspect of the evaluation is overall quality of produced synthetic speech. We have not yet carried out formal inter-subjective listening tests which quantitatively represent perceptional difference between the baseline "dummy" estimator and the evaluated one. However, according to informal judgement based on listening to synthesised sentences our CART estimator with all features and fractile pruning performs same or better than the baseline technique.

6. Conclusion

The research concerning F0 modelling is currently focusing mainly on the issues connected with prosodic homonymy. We have been able to prove that the current version of synthesised intonation is very well assessed and we expect that further improvement of prosodic structure parsing brings in more naturalness, especially in the field of semantic coherence of the synthetic speech. The presented approach in duration estimation has also performed well in our case and future work in this area will involve mainly more precise perceptual evaluation and also accuracy improving.

7. Acknowledgements

This research is supported by the Ministry of Education of the Czech Republic, project LC 536, Grant Agency of the Czech Republic, project 102/05/0278, and 6th FP IST-034434 COMPANIONS.

8. References

[1] Matoušek, J., Tihelka, D. and Romportl, J., "Current state of Czech text-to-speech system ARTIC", *LNAI Vol. 4188*, Springer, Berlin, 2006, p. 439 – 446.

[2] Hirst, D. J., "Form and function in the representation of speech prosody", *Speech Communication Vol. 46*, 2005, p. 334 – 347.

[3] Romportl, J., Matoušek, J., "Formal prosodic structures and their application in NLP", *LNAI Vol. 3658*, Springer, Berlin, 2005, p. 371 – 378.

[4] Romportl, J., "Structural data-driven prosody model for TTS synthesis", *Proceedings of Speech Prosody 2006, Studientexte zur Sprachkommunikation, Vol. 40*, Dresden, 2006, p. 549 – 552.

[5] Fujisaki, H., "A note on the physiological and physical basis for the phrase and accent components in the voice fundamental frequency contour", New York, 1988.

[6] Ptáček, M., "Akustika řeči", Praha, 1996 ("Speech acoustics", in Czech).

[7] Batůšek, R., "A duration model for Czech text-to-speech synthesis", *Proceedings of Speech Prosody 2002*, Aix-en-Provence, p.167 – 170.

[8] Öztürk, Ö., Çiloğlu, T., "Segmental duration modeling in Turkish", *LNAI Vol. 4188*, Springer, Berlin, 2006, p. 669 – 676.

[9] Reidi, M. P., "Controlling Segmental Duration in Speech Synthesis Systems", dissertation thesis, Zurich, 1998.

[10] Goubanova, O., King, S., "Predicting consonant duration with Bayesian belief networks", *Proceedings of InterSpeech 2005*, 2005, p.1941 – 1944.

[11] Chung, H., Huckvale, A. M., "Linguistic factors affecting timing in Korean with application to speech synthesis", *Proceedings of Eurospeech 2001*, 2001.