Gregor Rozinaj, Andrej Páleník

Corpus-based Variable Synthesis of Slovak Intonation

UDK xxx.xxx.xx
IFAC x.x;x.x.x (filled by editors)

Paper classification (filled by editors)

In this paper we present new and flexible corpus-based method for prediction of intonation from text which was implemented as part of Slovak TTS framework at UT FEI-STU in Bratislava. Proposed solution is based on classic technique – Viterbi algorithm. Intonation contours are generated by concatenation of selected pieces of real F0 contours. Selected sequence of units minimizes the overall cost that combines two types of elementary costs – target and concatenation cost. Whereas for calculation of concatenation cost some simplifying assumptions have been applied in order to lower computation costs, on the other hand calculation of target cost can achieve arbitrary complexity thanks to configurable set of contextual rules. Main advantage of our approach is the ability to produce alternative melodic contours given the same text input without the need to manipulate F0. We argue that the optimal place where variability should be introduced is not at the phonetic level (manipulation of F0) but at some more abstract level (e.g. adjustment of feature weights in particular contexts and thus affecting the selection process). In our solution the rules affecting weights used in computation of target cost are transformed from functional requirements. Hence the presented method can be considered as functionally-driven approach.

Key words: Intonation Synthesis, Viterbi Algorithm, Speech Synthesis

Croatian translation of the title. If one of the authors is a native Croatian speaker, then please type the abstract translation in Croatian here. Otherwise, please leave this paragraph as it is. The editors will take care of the translation.

Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka. Ovo je primjer sažetka.

Ključne riječi: Croatian, translation, keywords

1 INTRODUCTION

Synthesis of adequate prosody is a key element in pursuit of attaining natural sounding speech synthesis. Every TTS system must address the problem of predicting prosodic parameters from text (= from information that is available as a result of text analysis) in some way. From among different phonetic variables that materialize various prosodic features probably the most prominent variables are segment duration and fundamental frequency (F0). Segment (or phone) duration realizes timing aspects of prosody (e.g. rhythm, tempo, etc…) whereas F0 is the principal correlate of speech melody (intonation). Mutual interplay of these variables in speech enables realization of number of linguistic (stress, boundaries, prominence, focus) and non-linguistic (mood, attitude, etc…) functions. The process of synthesis of prosody can be understood as creation of mapping between linguistic and acoustic domain. The mapping may or may not be straightforward, single or multiple layers of representation may be chosen and different generative techniques may be employed. Therefore before proceeding with actual presentation of our own approach a brief overview of intonation modeling and synthesis will be given in chapter 1. Chapter 2 contains detailed description of proposed intonation synthesis method. At first the theoretical justification of basic principles is provided following with the description of the algorithm itself. Implementation details are briefly overviewed and finally the experiments performed with implemented solution are presented. In chapter 3 the solution is summarized and further research steps are outlined.

1.1 Intonation Modeling Overview

In each language the prosody has a different role and various functions are supported by prosody with varying weight. Synthesis of intonation must always take into account specifics of the target language.

Traditional intonation models are tightly coupled with phonological theory of the language. They are based on finite set of abstract elements/symbols that comprise the basic prosodic inventory of the language. This also stands for more current models such as the widespread ToBI scheme [12] which is inspired by Pierrehumbert's intonation phonology whose fundament is the use of two distinctive tones (H – high and L – low). The main challenge for phonological models is to find mechanism of converting abstract representation into F0 target.

The actual measured or estimated course of F0 is the starting point for phonetic modeling. The aim of modeling is to find optimal approximation of real F0 contours. Usually some fitting procedure is employed to estimate quantitative parameters of model components. For example Fujisaki model [6] is based on two types of components (phrase component and accent component) whose contributions are summed in a superpositional framework.

When corpus-based techniques rose into prominence there emerged a need for intonation representation that could be obtained by automated procedure due to large amount of speech data [13]. Output of such procedure could be used in unit selection synthesis frameworks. Some phonetic models, e.g. MOMEL [8], don't have the ambition to capture or explain the relation between acoustic and linguistic domain. Their aim is just to provide efficient and non-redundant coding (stylization) of real F0 contours. Usually they serve as the first step for further analytical modeling. One of the main benefits of having low-level F0 stylization is removal of microprosodic perturbations. Although microprosody is important for perception of naturalness it is irrelevant with regard to global melodic pattern.

Perception-based models are special subclass of F0-based modeling approaches [10]. In perceptual modeling only F0 movements which produce audible change are relevant for the model. If neglecting of certain F0 shape does not change its perception it can be omitted from the model.

Although the majority of intonation models is concerned with grasping of the intonation forms there are a few different approaches where the prosodic function is the starting point. An example of such approach is SFC [2]. In this approach there are no prior requirements on the shape of F0. Resulting F0 contour is a mere gross output of a number of functional generators. Generators themselves are trained on speech corpus.

1.2 Intonation Synthesis Overview

Crucial task for both types of models – phonetic and phonological – is to find the link between linguistic description and fundamental frequency. In reality the mapping is complex and of many-to-many character. Single sentence can be uttered with different acceptable melodic patterns. On the other hand analysis of different F0 contours can result in the same sequence of phonological symbols. In phonological approaches the prediction of symbols from text is usually straightforward because the rules that govern the placement and allowable combinations of symbols are part of the phonological model itself. This is however not the case for all symbolic models, for example INTSINT [14] offers just a symbolic inventory and the phonological model must be developed on top of it. More complicated part is the conversion from symbols to F0 targets. This can be achieved, for example, by means of linear regression [3],[14].

Data-driven synthesis approaches are usually used in conjunction with phonetic models. During preparation stage some automatic analysis procedure is performed. Later at synthesis time standard techniques for corpus-based speech synthesis are used (e.g. Viterbi search, CART-based clustering [1][4], neural networks [2]. Unlike standard speech synthesis the solution for synthesis of intonation must cope with the fact that there is no basic unit of prosody. Most systems use syllable-sized units as a minimal building blocks. Larger units (intonation groups / stress groups / accent groups) are usually constructed as sequences of syllables with usually single stressed syllable [7][9].

Some methods do not use any sort of intonation modeling at all. Sampled real F0 contours are used for training the generators and at synthesis time the F0 targets are directly predicted by the synthesis engine based on linguistic input. While some methods use the speech corpus to derive "average" F0 shapes [15], other methods attempt to exploit the variability stored in corpus to produce variable synthesis either by directly modifying F0 within the scope given by calculated probability distribution [5] or by intervening into the unit selection process [11].

2 Theoretical Justification of Proposed Model

At present there exists no model of Slovak intonation suitable for usage within corpus-based synthesis framework. Existing theory of Slovak intonation is mainly prescriptive and descriptive. Therefore we were not limited nor biased with the choice of model to be used. Suitable phonological model was not at hand and we opted not to use any phonetic model apart from basic F0 stylization. Design of our synthesis method was inspired by three basic principles:

1. Perform manipulation of F0 as little as possible.
2. Functional requirements should drive the synthesis process.

3. Recorded speech corpus contains rich variability of prosodic realizations that should be utilized.

In order to avoid modification of F0 we should try as much as possible to utilize the benefit of having large number of natural F0 contours in the speech corpus. In data-driven approaches the upper bound of quality of synthesis is only determined by the quality of the training corpus. If realizations of some prosodic feature are completely missing in the corpus we would not be able to produce it by using any method. On the other hand if there are multiple instances of particular prosodic feature in the corpus we are given the luxury of choice. Selection of the optimal sub-contour must, however, always take into account all other requirements for given sentence's prosodic features. Thus the resultant synthesized F0 contour must be globally optimal – with respect to all raised requirements. At the same time negative effects caused by concatenation of sub-contours must be minimized as well.

These considerations lead to further questions: How to define optimal selection? What requirements should be raised? We were inspired by approach of SFC model [2] which comes out of a small set of prosodic functions. Each of these functions has a defined scope – number of syllables on which it has impact. For example let's assume that indication of phrase boundary influences N syllables before and M syllables after the actual boundary. In SFC framework each prosodic function is produced by trained functional generator (neural network). The most important input for the functional generator is the actual scope and syllable's position within actual scope. Overlapping contributions of various functional generators are summed up within superpositional framework. We chose different generation mechanism but the starting point is very similar. In our approach we define functional requirements and transform them into rules affecting unit selection process. For example let's assume that original functional requirement states: "Let the sentence have typical melodic pattern for yes/no question!". Such requirement can be broken down as follows: What is typical for melodic patterns of yes/no question? It is the typical fall-rise-(fall) pattern placed on the subject of the question. For simplicity let's assume that position of the subject in standard yes/no question is at the end of the sentence. Hence the transformed requirements can be formulated in following points:

a) For target syllables from the final prosodic word of the sentence always take into account only candidate syllables from sentences of the same type.
b) For target syllables from the final prosodic word of the sentence prefer candidate syllables from final prosodic words of the sentence.

Such simple requirements although very simple and probably incomplete can be directly taken into consideration during unit selection process.

For synthesis of intonation the optimality criterion is rather tricky problem because, as was already stated, single sentence can be uttered with many acceptable intonation patterns. In many TTS systems the construction of intonation prediction module is motivated by the effort to obtain some kind of „average" = most probable contours. We take different approach – we try to utilize the richness of prosodic variations stored in the speech corpus not for the purpose of obtaining statistical model but to permit alternative intonation contours be synthesized. So, in our approach the answer to an interesting question: „How to achieve intonation variability?" would be the most trivial one: We select alternative intonation pattern from the corpus. We think that variability achieved by alternative selection has lower risk of producing „weird" melodic patterns as opposed to approaches where variable intonation is obtained by modification of F0 values on the basis of statistic model.

3 SYNTHESIS ALGORITHM

As was already mentioned several times we adopt standard unit selection technique based on search for optimal sequence of units. Output F0 contour is constructed by concatenation of partial F0 sub-contours. Basic unit of synthesis is F0 segment whose boundaries are aligned with boundaries of the voiced part of syllable.

3.1 Descriptive Features

Each unit in the corpus as well as each unit of the synthesis target is described by a vector of binary features. We chose the binary form of features for a number of reasons. Firstly, all features can be equally handled. Furthermore, conditions based on binary values can be formulated more easily and are usually more comprehensible. Finally, operations on binary sequences can be implemented efficiently. All features needed to be transformed into binary form. For example the original feature – number of phonemes in a syllable – was broken down into 5 binary features in the form: "Does the syllable consists of at least X phonemes?" where $X = 2 \ldots 6$. Original features (before the transformation was applied) describe the following attributes:

- Number of phonemes in syllable
- Number of syllables in prosodic word
- Number of prosodic words in phrase
- Position of syllable within prosodic word
- Position of prosodic word within phrase

- Part of speech of corresponding word
- Characteristics of the syllable (syllable structure, properties and length of onset and coda, distance between neighboring nuclei, presence of stress)

In total 43 binary features were used $\mathbf{p} = (p_1, p_2, \ldots, p_{43})$. Only such features which can be estimated or derived from input text were considered.

3.2 Viterbi Algorithm

For the search of optimal sequence of candidates Viterbi algorithm was implemented. Viterbi search is performed in a sequence of steps. Each step corresponds to single target syllable. Set of candidate syllables is examined in each step and number of optimal sub-sequences – from the start leading to each candidate syllable are kept. After completing the t-th step when N_t candidate syllables were examined exactly N_t optimal sub-sequences are kept for further processing. Every sub-sequence involving j-th candidate unit in t-th step is characterized by certain value of total cost obtained by following formula:

$$\delta_t(u_{t,j}) = \min_{1<i<N_{t-1}} \left[\delta_{t-1}(u_{t-1,i}) + a_t(u_{t-1,j}, u_{t,j})\right] \quad (1)$$

The increment of total cost is given by term a_t which consists of two types of cost:

$$a_t(u_{t-1,i}, u_{t,j}) = T(s_t, u_{t,j}) + J(u_{t-1,i}, u_{t,j}) \quad (2)$$

Target cost represented by term $T(s_t, u_{t,j})$ expresses how well does the candidate syllable $u_{t,j}$ fit to target syllable s_t. On the other hand concatenation cost $J(u_{t-1,i}, u_{t,j})$ penalizes possible concatenation of candidate syllables $u_{t-1,i}$ and $u_{t,j}$. The overall optimal sequence has the lowest total cost out of all sub-sequences considered in the final step. In order for the algorithm to work efficiently and as desired the definition of target and concatenation cost is crucial. Input to our Viterbi algorithm is a sequence of M syllable-sized units from the input sentence $\langle syl_1, syl_2, \ldots, syl_M \rangle$ where each syllable is described by feature vector $S = \langle s^1, s^2, \ldots, s^M \rangle$ consisting of N features each – $(s_1^m, s_2^m, \ldots, s_N^m)$. The target cost computed between two feature vectors in the basic form is given by the following formula:

$$T(\mathbf{s}, \mathbf{u}) = \sum_{j=1}^{N} w_j T_j(s_j, u_j) \quad (3)$$

The target cost weighted by w_j is computed as follows:

$$T(s,u) = s \oplus u \quad (4)$$

By default (without integration of the contextual rules introduced in subsection **Error! Reference source not found.**) all weights are equal to 1 so basically the binary fields representing two units being compared are XOR-ed and the count of 1s is obtained. Hence whenever the two units do not match in n-th feature the candidate is penalized by incrementing the target cost by 1.

Concatenation cost should be defined with caution due to large number of its computations in the course of algorithm flow. For the sake of computational simplicity we defined the concatenation cost only on the basis of whether the two units are natural neighbors in their original contexts. If yes, then concatenation cost is zero otherwise the concatenation is always penalized by addition of fixed experimentally derived constant.

3.3 Contextual rules

Synthesis algorithm formulated in its basic form was further enhanced by introducing the contextual rules. This is the key concept which enables flexibility in formulating functional requirements and variability in the unit selection process. All rules are defined on a set of descriptive binary features $(p_1, p_2, \ldots, p_N)$. Some rules may be formulated conditionally. In these cases the conditional expressions can also be only based on the same set of features:

$$\langle condition \rangle : p_i = value \quad (5)$$

There are three types of rules. Strict requirement of exact match on j-th feature may be enforced by formulating so called mandatory rule where the condition is optional:

$$MANDATORY\_RULE : M(p_j, \langle condition \rangle).$$

If the condition is satisfied then only candidate syllables that have exact match on value of feature p_j can be considered for unit selection. An example of loosely formulated mandatory rule: „If the current target syllable being synthesized occurs in the first prosodic word of the sentence then consider for the selection only those candidate syllables that are stressed (if the target syllable is stressed) or unstressed (if the target syllable is unstressed).“. Applying the mandatory rule narrows the number of candidate units. Exact opposite to mandatory rule is an ignore rule with optional condition:

$$IGNORE\_RULE : I(p_j, \langle condition \rangle).$$

By introducing ignore rules the j-th feature p_j may be

completely omitted from target cost computation if the condition is satisfied. This means that candidate units would not be penalized for mismach based on value of p_j.

The most versatile is the third type of rule called the variant rule. By using variant rules we are able to increase penalization for feature value mismatch in certain contexts determined by mandatory condition:

$$VARIANT\_RULE : V\left(p_j, b, e, \langle condition \rangle\right)$$

b (baseline) and *e (extra)* are scalar values which should be taken as a contribution into overall target cost in case of mismatch based on feature p_j. If the condition is satisfied the mismatch based on value of p_j is penalized by value of *extra* otherwise the mismatch is penalized by value of *baseline*. As an example let us take the functionally derived requirement b) from chapter 2. If we transform it into variant rule it could have looked like the following: "If the current target syllable being synthesized occurs in the last prosodic word of the sentence then penalize all candidate syllables that do not occur in last prosodic word of the sentence in their original contexts with penalty value = 10, otherwise penalty value = 1.". Variant rules offer great flexibility in expressing requirements for the unit selection process.

Enhanced unit selection algorithm with integration of the contextual features works almost the same as in its basic form. Slight differences are introduced in two steps – construction of the candidate unit set and computation of target cost. The candidate set may be narrowed as a result of applying all mandatory rules. Theoretically the candidate set may be left completely empty. Computation of target cost gets complicated because the weights w_j need to be computed first. For each feature p_j relevant rules must applied and the resultant weight w_j is obtained by formula:

$$w_i = \prod_j RULE_j\left(p_j, b_j, e_j, condition\right) \qquad (6)$$

where $p_j = p_i$.

If there is any valid ignore rule the weight would be equal to zero. Otherwise the weight will be calculated as a product of penalty values from all relevant variant rules.

3.4 Implementation of the Synthesis Engine

For the purpose of proper evaluation of the proposed synthesis approach new experimental speech corpus was recorded. The corpus consists of total number of 143 sentences, 754 words, 1679 syllables and 3982 phonemes. The average sentence length is 5 words, the minimal sentence length is 1 syllable and the maximum sentence length is 14 words. It was carefully manually constructed in order to contain limited number of different prosodic features. All sentences are simple yes/no questions and are composed of single syntactic phrase. No syntactic markers, enumerations or parentheses were allowed. While we severely restricted the number of supported prosodic functions on the other hand the corpus contains large number of variations of the supported prosodic features. Yes/no questions were chosen because of their characteristic intonation pattern. Working with distinctive pattern enables to better judge how well the prosodic function was realized.

The recorded speech corpus was automatically analyzed using MOMEL algorithm [8]. Obtained quadratic F0 stylizations were used as the inventory of F0 contours. Each syllable-sized intonation unit was described by three F0 values aligned at 10, 50 and 90% of the voiced portion of the underlying syllable. The actual implementation of the algorithm was implemented in Python and the manual steps of speech annotation were supported by excellent tool – Praat[1]. The rules that control the unit selection are stored in an XML format and can be manually edited. For low-level speech synthesis and evaluation of synthesized contours well-known MBROLA diphone synthesizer[2] was used.

3.5 Experiments

Several experiments were performed in order to test the performance of the synthesis engine. First of all the optimal value for penalization of concatenation cost was searched for heuristically. Reasonable balanced value of the penalty can always be found but it appears that with the increasing number of rules the value of concatenation penalty should also be increased.

Next experiment was performed to test the flexibility of the unit selection algorithm. Intonation of the same sentence was repeatedly synthesized while the number of rules grew. In order for the rules to be more comprehensive for the reader semantic description of used features are given in Table 1. Concatenation penalty value for the whole experiment was set at c = 10. Results of this experiment are demonstrated in Fig. 1-6. At the beginning of the experiment only single rule was employed:

R1 : M(phr_has_plus5_pw)

The sentence being synthesized was a simple yes/no question: „Máme dobrú náladu?" (English: "Do we have good mood?"). After applying the initial set of rules

[1] http://www.fon.hum.uva.nl/praat/
[2] http://tcts.fpms.ac.be/synthesis/

intonation contour displayed in Fig 1a was produced and selection of units illustrated in Fig 4 was performed. Audibly the result was rather acceptable but it was not very good overall. Rule R1 limited the set of possible candidates to syllables from sentences of similar length. From Fig. 4 it is clear that the high value of concatenation penalty has caused selection of units from single sentence. However the last prosodic word from which the syllables were taken is two syllables longer and therefore the typical melodic pattern on the last prosodic word of yes/no question was cut. In order to correct this effect the second rule was introduced. It effectively penalizes selection of syllables that are not the final syllables in their original sentences in the place of the last target syllable in the sentence.

R2: V(*syl_final_in_pw*, 1.0, 6.0, (*pw_final_in_phr*, 1))

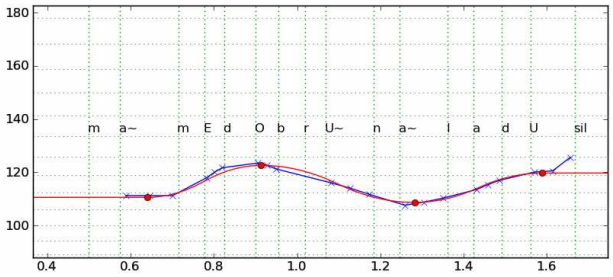

Figure 1. F0 contour generated using 1 rule.

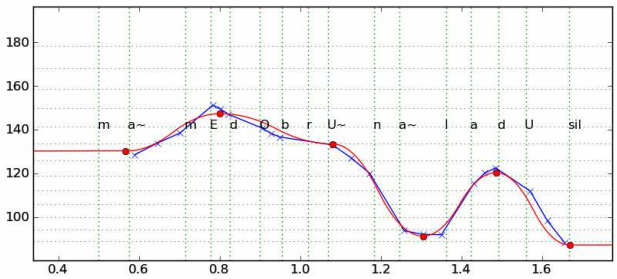

Figure 2. F0 contour generated using 2 rules.

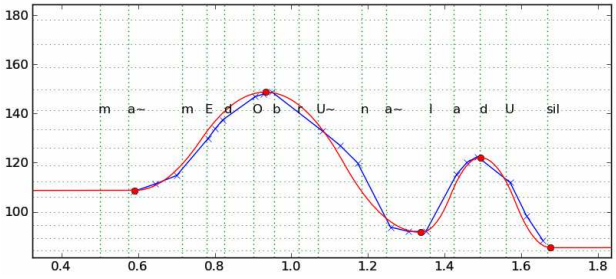

Figure 3. F0 contour generated using 3 rules.

Rule R2 achieved alignment of the final part of sentence with candidate units having the same properties. Again selection of all units was drawn entirely from single sentence in the corpus. The resultant F0 contour and the unit selection are depicted in Fig 2 and Fig 5 respectively. This time the beginning of the sentence was cut-out which brought in unnatural melody at the onset of the target sentence. Addition of the third rule removed also this undesireable effect:

R3: V(*syl_1st_in_pw*, 1.0, 6.0, (*pw_1st_in_phr*, 1))

Now both the beginning and the end of sentence are composed of intonation sub-contours that are aligned, typical melodic pattern for yes/no question is preserved and the sentence has natural intonation onset. This was all achieved using only 3 control rules. All of the rules that were added in the course of the experiment were derived from logical functional requirements. This demonstrates the flexibility of the framework and its ability to easily introduce variability to intonation synthesis.

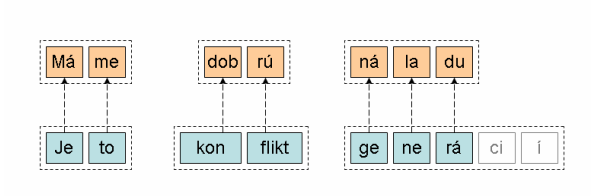

Figure 4. Illustration of unit selection with 1 rule.

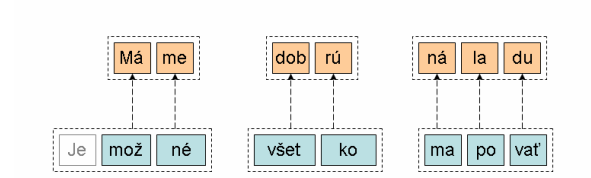

Figure 5. Illustration of unit selection with 2 rules.

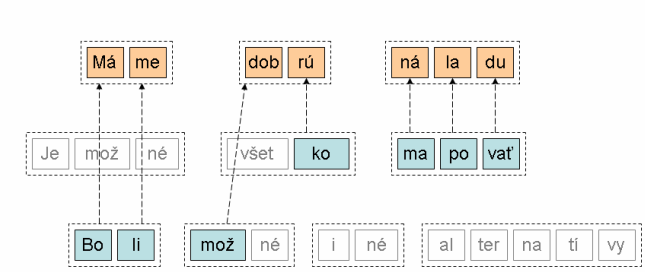

Figure 6. Illustration of unit selection with 3 rules.

4 CONCLUSION

4.1 Advantages and Disadvantages

Proposed system for synthesis of intonation has some really nice features:

1) It allows flexible adjustment of weights in a comprehensible manner.
2) Output intonation contours are (at least piece-wise) always formed from real F0 sub-contours.
3) Acoustic parameters are not taken into account

for unit selection which enables computationally less expensive ways of calculating costs.

Although the listed pros are quite comprehensible there are also some serious drawbacks that should be mentioned. First of all, there is a problem how to efficiently implement this algorithm. Each new introduced rule causes the number of computations to linearly grow since the rule must be applied to each candidate syllable. For standard-sized corpus containing approximately 50000 syllables each rule requires cca 50000 tests. Idea for improvement worth testing is to define strict set of mandatory rules in order to consider only reasonably low number of candidates. To compensate for continuity which would be affected by such massive reduction of candidates the candidate set should always contain the natural neighbor of the optimal candidate for previous syllable. This would have had also a positive effect that the candidate set would then almost never be empty (with some rare exceptions).

Second source of troubles may be caused by the fact that growing number of rules may lead to uncontrollable growth of target cost values. Finally, there were some really strong simplifications made regarding concatenation cost. Neglecting of F0 in computation of concatenation cost is rather strong simplification. For smaller corpora with limited prosodic scope (like the one used in this study) negative effects may not be encountered often.

However for larger corpora with broad prosodic coverage there is higher probability of spurious jumps in F0 at concatenation points.

Table 1. Description of features used in experiment.

Feature Name	Feature Description
phr_has_plus5_pw	*Does the phrase have more than 5 prosodic words?*
syl_final_in_pw	*Is the current syllable final in the prosodic word?*
pw_final_in_phr	*Is the current prosodic word final in the phrase?*
syl_1st_in_pw	*Is the current syllable the first in the prosodic word?*
pw_1st_in_phr	*Is the current prosodic word the first in the phrase?*

4.2 Summary and Future Work

Current study presents only the 1st attempt in the field of intonation synthesis performed on limited speech corpus. From among many possible improvements we highlight only the most urgent topics apart from obvious ones (e.g. test the implementation on a larger speech corpus):

By ignoring F0 in calculation of concatenation costs we substantially reduced the overall computation cost but only at the price of probable spurious discontinuities at concatenation points. In our current setting no negative effects were observed but for larger corpora this problem will surely inflate. Therefore some mitigation procedure must be found – either modification of the unit selection process or modification of the output F0.

The second urgent issue concerns the way how the actual F0 is concatenated. Currently the F0 values taken from the winning candidate are stretched accordingly in order to fit on the target phonetic stream. However when the mismatch between candidate and target syllables in terms of syllable structure is too big some more elaborate fitting procedure would be suitable. Rather big mismatch may have influence on the global tune – speaker would prefer different melodic pattern. These effects should be analyzed in order to find the balance – when the mismatch in phonetic structure has such influence that it should be penalized.

Some gaps in our current solution which urgently need to be solved have been pointed out. However there are also some optimistic options for further research. Our synthesis algorithm proved to be flexible enough so that even ad hoc rule creation causes the synthesis output to improve in intended way. Formulation of the rules however need not to be purely human job. We believe that our current synthesis engine can be turned into intelligent machine-learning system after some further improvements. Human teacher would guide the automatic learning system by judging its output. This presents a real challenge for us now.

ACKNOWLEDGMENT

This work has been supported by the Grant Agency of the Slovak Republic, VEGA 1/0718/09 Algorithms and Methods of Multimedia Signal Processing for Human Machine Interface and FP7-ICT-2011-7- 287848 – FP7 HBB-Next - Next-Generation Hybrid Broadcast Broadband.

REFERENCES

[1] P.D. Agüero, et al., “Automatic Analysis and synthesis of Fujisaki’s intonation model for TTS,” in *Proceedings of Speech Prosody 2004*. (Nara, Japan), pp.427-430, March 2004.

[2] G. Bailly, B. Holm, “SFC: A trainable prosodic model,” *Speech Communication*, vol 46 (3-4). Special issue on Quantitative Prosody Modelling for Natural Speech Description and Generation, pp. 348-364, 2005.

[3] A. W. Black, A. J. Hunt, “Generating F0 contours from ToBI labels using linear regression,” in *Proceedings of. ICSLP'96*. vol. 3, (Philadelphia, USA), pp. 1385-1388, October 1996.

[4] K. Dusterhoff, A. W. Black, “Generating F0 contours for speech synthesis using the Tilt intonation theory,” in *Proceedings of ESCA Workshop on Intonation: Theory, Models and Applications*. (Athens, Greece), pp. 107-110, September 1997.

[5] D. Escudero, et al. “Corpus based extraction of quantitative prosodic parameters of stress groups in Spanish,” in *Proceedings of ICASSP 2002*, (Orlando, USA), pp. 481–484, May 2002.
[6] H. Fujisaki, S. Ohno, “Prosodic parameterization of spoken Japanese based on a model of the generation process of F0,” in *Proceedings of ICSLP'96*, vol. 4, (Philadelphia, USA), pp. 2439-2442, October 1996.
[7] J. M. Gutiérrez-Arriola, et al., “New rule-based and data-driven strategy to incorporate Fujisaki’s F0 model to a text-to-speech system in Castillian Spanish,” in *Proceedings of the International Conference on Acoustics and Signal Processing ICASSP’ 2001*, (Salt Lake City, USA), pp. 821-824, 2001.
[8] D. Hirst, R. Espesser, “Automatic modelling of Fundamental Frequency Using a Quadratic Spline Function,” in *Travaux de l'Institut de Phonétique d'Aix en Provence*, vol.15, pp. 75-85, 1993.
[9] F. Malfrére, et al., “Automatic Prosody Generation Using Suprasegmental Unit Selection,” in *Proceedings of the 3rd ESCA/COCOSDA Workshop on Speech Synthesis*, (Jenolan Caves, Australia), pp. 323-328, November 1998.
[10] P. Mertens, C. d’Alessandro, “Pitch contour stylization using a tonal perception model,” in *Proceedings of 13th International Congress of Phonetic Sciences*, vol. 4, (Stockholm, Sweden), pp. 228-231, August 1995.
[11] J. Romportl, “Structural Data-Driven Prosody Model for TTS Synthesis,” in *Proceedings of Speech Prosody 2006*, vol. II, (Dresden, Germany), pp. 549-552, May 2006.
[12] K. Silverman, et al., “TOBI: a Standard for Labelling English Prosody,” in *Proceedings of second International Conference on Spoken Language Processing ICSLP'92*, (Banff, Canada), pp. 867-870, October 1992.
[13] P. Taylor, “Analysis and Synthesis of Intonation using the Tilt model,” in *Journal of the Acoustical Society of America*, vol. 107, No. 3, pp. 1697-1714, 2000.
[14] J. Véronis, et al., “A stochastic model of intonation for text-to-speech synthesis,” in *Speech Communication*, vol. 26, no. 4, pp. 233-244, 1998.
[15] Y. Yamashita, T. Ishida, “Stochastic F0 contour model based on the clustering of F0 Shapes of a syntactic unit,” in *Proceedings of 7th European Conference on Speech Communication and Technology EUROSPEECH 2001*, (Aalborg, Denmark), pp. 533-536, September 2001.