Partial Representations Improve the Prosody of Incremental Speech Synthesis

Timo Baumann

Department of Informatics, Universität Hamburg, Germany
baumann@informatik.uni-hamburg.de

Abstract

When humans speak, they do not plan their full utterance in all detail before beginning to speak, nor do they speak piece-by-piece and ignoring their full message – instead humans use partial representations in which they fill in the missing parts as the utterance unfolds. Incremental speech synthesizers, in contrast, have not yet made use of partial representations and the information contained there-in.

We analyze the quality of prosodic parameter assignments (pitch and duration) generated from partial utterance specifications (substituting defaults for missing features) in order to determine the requirements that symbolic incremental prosody modelling should meet. We find that broader, higher-level information helps to improve prosody even if lower-level information about the near future is yet unavailable. Furthermore, we find that symbolic phrase-level or utterance-level information is most helpful towards the end of the phrase or utterance, respectively, that is, when this information is becoming available even in the incremental case. Thus, the negative impact of incremental processing can be minimized by using partial representations that are filled in incrementally.

Index Terms: incremental processing, prosody, speech synthesis, spoken dialogue systems, simultaneous interpreting

1. Introduction

Incremental speech synthesis (iSS) produces speech from a specification that is only finalized after utterance delivery has already started. This capability is required in novel, interactive applications such as simultaneous interpretation [1, 2], live commentary, for example in sports domains [3], or highly responsive dialogue applications [4]. All of these applications break the conventional metaphor in speech synthesis processing of "reading out aloud" a text that is fully known in advance.

A particular problem for iSS is the generation of plausible utterance prosody, because prosody contains long-range dependencies such as rhythm clashes several words ahead that influence the current realization. These long-range dependencies can be dealt with by (a) having synthesis lag behind such that all (or most) dependencies are satisfied, (b) ignoring dependencies and lagging behind less, or (c) using non-final data in case the correct data is not yet available. The trade-off between prosodic quality and low processing lag is a general property of incremental processing – while the trade-off cannot be solved, it can be *optimized* to lead to good results given the circumstances.

In speech synthesis, prosody generation is conventionally split into two parts: the assignment of symbolic intonation information (e. g. determining intonation phrases and stress marks on words using the ToBI system [5]), and the derivation of synthesis parameters (tempo and pitch, in the form of segment durations and f_0) from the symbolic representation.

This paper investigates possibilities to improve the parameter derivation from symbolic features. The method's flexibility allows it to make use of symbolic features when they are available and to use default values when features are unavailable. This results in an optimization of the incremental processing trade-off that leads to results almost on par with non-incremental processing. Specifically, we find in Section 6 that this is caused by the fact that features are likely to be available when they are most needed (towards the ending of the bearing unit).

2. Related work

Psycholinguistic research has found human speech to be produced incrementally [6], yet most speech synthesizers do not make use of this insight [7, 8]: instead, they process – in a top-down manner – each layer of abstraction after the other, assuming all higher-level information to be complete before processing on the next layer starts. Many of the processing layers have recently been shown to work reasonably well incrementally, with limited contexts.

Most straightforwardly, non-incremental speech synthesis technology can be used repeatedly with piecewise extended input and stitching together the produced outputs [9]. The prosodic quality obtained in this manner depends on the *lookahead*, that is, how soon further input must become available in order to give contextual hints to current processing. Previous work found that one intonation phrase of future context (beyond the phrase that is currently being delivered) is necessary to result in acceptable prosody under this approach [10], but without differentiating the influence of symbolic intonation assignments, HMM state selection, or further influences of incremental processing. The goal of the present paper is to state more precisely, as well as to reduce, this lookahead requirement.

The selection of HSMM states, durations, and pitch is based on symbolic intonation and segmental assignments by higher-level linguistic pre-processing. It is often performed using decision trees (such as CARTs [11]). In HMM synthesis, multiple trees (based on identical feature vectors) are used for duration, pitch, cepstral, and aperiodicity parameters, respectively. Constraining the features which are available to decision-making has been investigated by several authors, with the purpose of abandoning higher-level features [12], for speech coding [13], and with incrementality in mind [14], in the latter case limiting features to the current syllable, word, or current phrase. While cepstral and aperiodicity parameter assignments work well with strongly limited contexts (e. g. including up to the current word), duration and f_0 assignments appear to require larger contexts. For this reason, the present papers focuses on prosodic parameters.

All works on the relevance of feature usage for HSMM state selection find that quality deteriorates when features are left out. In all cases, features are grouped into feature classes; the

Table 1: Counts of decision features, categorized along the temporal axis and by levels of linguistic abstraction (indicating granularity), for German. Feature classes are encircled.

	past	current	future
phone	20	10	19
syllable	3	8	2
word	2	7	3
phrase/accentuation	11	10	10
full utterance	—	5	—

classes used by [14] and re-used in the present paper are shown in Table 1. To the best of the author's knowledge, previous work has not investigated the conditional usage of features (or feature classes) depending on the context availability during incremental processing. For example, phrase-level features are likely available during phrase-final words, but may be unavailable before. Investigating the use of features when they are likely available in practice is the novel contribution of the present work.

'Below' the level of this work, HSMM parameter optimization has previously been shown to work well within local contexts [15, 16], global variance optimization [17], which greatly improves HSMM synthesis quality, has recently been supplemented with a local alternative [18], for which Chunwijitra et al. [18] find superior results to GV with a lookahead of only 50 ms (10 frames into the past, 10 frames into the future). Finally, STRAIGHT vocoding [19] is inherently incremental.

3. System architecture

Making use of as much information as is available for decision-making is a straightforward idea which is, however, not well supported by conventional speech synthesis architectures. The processing layers and their (simplified) associated data is sketched in Figure 1. Conventional systems process information top-down and layer-by-layer. That is, processing on a lower layer only ever starts when processing on all higher layers has been completed. In contrast, incremental synthesis starts outputting speech as soon as the first phones can be realized and further processing on every layer is triggered either by the availability of what is to be spoken, or the need for some information by a lower processing layer.[1] The extent to which processing depends on 'future' data (such as the identity of the next but one phoneme) is described by the *lookahead requirement* of the processing mode.

For the task at hand, determining HSMM states with decision trees that use features from all layers of symbolic processing (cmp. Table 1), the lookahead requirement depends on what features are to be used. For example, following the approach in [14], using phrase-level features requires that all words and phones in the current phrase have been determined.

The simplistic approach from [14], which presupposes the (un)availability of data on a certain level of abstraction is inefficient in several circumstances: (a) for the last word in a phrase, it is (almost) certain that all phrase-level information is available – yet this may often not be the case for early words of the phrase; (b) for a given (multi-syllabic) word, features relating to the next syllable will be available – yet, relying on the next syllable being available implies to know the next phrase (at least its beginning) for phrase-final syllables. Thus, we propose to flexibly use all features that are available and to skip those features that are unavailable.

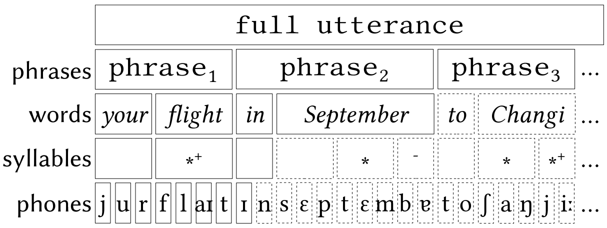

Figure 1: Simplified representation of symbolic data available for HMM state selection (dashed boxes: data yet undetermined in incremental synthesis).

Granularity is an important concept in incremental processing and describes the size of the units that form the input and output of a processor. Previous systems have assumed input in the form of multiple word chunks [9] which in [10] have been required to co-incide with intonation phrases. We here propose a more flexible 'mixed granularity' approach, in which different types of input (phrasing information, words, etc.) are specified independently. We show that this is more suitable, and (as will be shown below) better matches the problem of providing features to decision-making. While the architecture of our system provides a mixed-granularity approach, the programming interface to manage partial representations is yet to be finished.

4. Implementation

Our system is implemented in InproTK [23] and extends Inpro_iSS [9] which uses adapted code from MaryTTS [7] for linguistic analyses, HSMM optimization and vocoding.

Previous versions of Inpro_iSS have used the HMM states that were non-incrementally computed by MaryTTS and only performed parameter optimization and vocoding incrementally. In the present system, we additionally decide on HMM states using feature vectors that are incrementally determined from the current state of the IU network.[2]

In addition, our system implements incremental prosody assignment only, that is, the assignment of parametric values (durations, f_0, cepstral, and aperiodicity means and variances) based on non-incremental symbolic intonation from MaryTTS. In our system, we assume that a phrase's symbolic intonation can be determined as soon as its final word is known. Section 6 adds a smaller experiment to determine in how far this assumption holds true. Also, even though symbolic intonation can be hard to determine in text-to-speech tasks (as used in the experiments below) because all structural information needs to be reconstructed from the textual form, dialogue systems or simultaneous interpreters should have much of this structural information readily available already during utterance delivery.

[1] Of course, input must be provided sufficiently quickly as to not interrupt speech delivery – however, as speech is a relatively slow communication channel, significant adequacy gains are possible using iSS in practical applications [20, 21, 22].

[2] So far, no real feature extraction from the IU network has been implemented, but instead the non-incremental feature vectors are adapted to contain only those features that are available (and valid) given the context. Features that are not available in the incremental setting are replaced by default values (similarly to and using the same defaults as [14], but on a per-feature basis). Thus, while the current system is not fully lazy/just-in-time incremental, it provides identical results as if feature extraction were completely re-implemented (which we plan to do in the near future).

Table 2: Prosody (f_0 and duration) deviation relative to non-incremental (full-context) processing (BITS-1 voice, IPdS corpus) of settings with and without context-sensitive feature usage.

	f_0 in Cent		duration in ms	
setting	RMSE	MAE	RMSE	MAE
context–1phone	192	0.0	28.9	3.32
current word*	219	14.3	26.4	1.46
context sensitive	170	0.0	25.5	0.63
context+utterance	32.5	0.0	2.0	0.0
current phrase*	167	0.0	25.4	0.36

*experimental settings as previously reported in [14].

Of course, in a deployed incremental system, information should always be considered as soon as possible (especially if it can revise later on). However, for experimentation outside of specific application areas, some assumptions on the availability of information need to be made. Specifically, we decided that features relating to the *next syllable* (and the *second to next phone*) are only available if these are part of the current word (or current syllable, respectively). We also decided that *phrase-level information* is only available (and should be trusted) during the final word of the current phrase. In addition, information pertaining to the *full utterance* is only available on the utterance-final word (or not at all, this is one of the conditions tested in the following section). We also analyze the influence of the *next phone* features being always available (vs. not, if the next phone already belongs to the next word) in one of the conditions below.

5. Experiments using full symbolic intonation

We use the same corpus of 598 German utterances from [24] as used in [14], which are synthesized using the BITS-1 voice [25] deployed with MaryTTS (version 4.3). Utterances are relatively short, totalling only 751 phrases (for an average of 1.26 phrases per utterance, 6.1 words per phrase).

We compare the following experiment conditions:

current word condition from [14] which includes all features pertaining to the current word and future syllables,

current phrase similar to *current word*, including features up to (including) the current phrase and future words; the *current word* and *current phrase* conditions form a lower and upper bound for the performance of the new method,

context sensitive the context sensitive method (using phrase-features when available) as described in Section 4,

context+utterance context sensitive method including full-utterance level information for the utterance-final word,

context–1phone context sensitive method but excluding the next phone if it does not pertain to the current word.

We report root-mean-squared error (RMSE), as well as the *median* absolute error (MAE) for per-state f_0 (measured in Cent [26], 1 semitone = 100 Cent) and phone durations (measured in milliseconds) in Table 2. Clark and Dusterhoff [27] found that RMSE between pitch contours best correlates with user ratings from perceptual evaluation among a number of measures. Settings are ordered in the table by the mimimal lookahead that they require.

As can be seen in the table, errors decrease with the minimal amount of lookahead used by each condition and the *context sensitive* method performs very well: it does not use any more lookahead than the *current word* method, but its performance is actually very close to that of the *current phrase* method which requires current-phrase features to be available in the whole phrase, not only during the phrase-final word.

Table 3: Performance impact (in terms of RMSE) of using features in the final word of the phrase (or utterance).

		non-final		final	
	setting	f_0	dur.	f_0	dur.
phrase-level	w/o phrase inf.	88.2	3.68	426.0	53.6
	w/ phrase inf.			340.6	53.3
utterance-level	w/o utt. inf.	21.6	2.00	370.5	56.2
	w/ utt. inf.			64.2	2.1

In addition, the context sensitive method with addition of full-utterance features (yet, only for the utterance-final word) gives a dramatic improvement and is almost as good as non-incremental processing, with an f_0 RMSE of less than $^{1}/_{3}$ of a semitone and a duration RMSE of 2 ms. Also, distinguishing the improvements between f_0 and duration, phrase-level features appear to be important for f_0 whereas utterance-level features are important for duration estimation.

Finally, the *context–1phone*, which is truly word-by-word incremental, still outperforms the f_0 assignments of the *current word* method. However, next (and, to a lesser degree, second-to-next) phone features are very important for cepstral and aperiodicity assignment [14]. They should hence not be skipped entirely. An (informal) listening experiment confirms this issue.

In order to validate the extremely good results of the context sensitive conditions, we performed another experiment in which we compare the performance of the *context sensitive* approach with and without context-sensitive phrase-level information on all phrase-final material and on all non-phrase-final material (where phrase-level information is unavailable in either case). Similarly, we compare the *context+utterance* approach non-finally and finally (with and without utterance-level features). The results are shown in Table 3.

As can be clearly seen, both phrase- and utterance-level features are most important towards the end of phrases (respectively utterances), with the final words causing the vast majority of the overall error (in the case of phrase-level features even when the features are available, potentially because important utterance-level features are still missing). It must, however, be noted that errors are still high in the final portions of the phrase or utterance, which indicates that prosody of phrase/utterance-endings is more complex than mid utterance.

We conclude that both phrase-final and utterance-final features are most important *when they are actually available* (i. e. during the phrase/utterance final word) even in incremental processing using a word-by-word granularity. The negative performance impact of incremental processing on HMM state selection can be reduced immensely by using partial representations.

6. Experiments using limited symbolic intonation

We finally estimate the performance of our new method under limited, incrementally produced symbolic intonation instead of full, non-incrementally produced symbolic intonation. We use the previous restart/rewrite method from [9, 10], as our system still lacks a truely incremental intonation processor. The method works best with full phrase increments and hence we use the

Table 4: Performance impact of using incremental intonation and/or prosodic parameter assignments in terms of RMSE as compared to non-incremental intonation and non-incremental prosodic parameter determination.

symbolic intonation	parametric prosody	f_0	dur.
non-incremental	context sensitive	143	21.8
	context+utterance	47.2	6.93
w_{n-1} incremental	non-incremental	0.0	1.81
	context sensitive	162	21.8
	context+utterance	51.9	6.81
w_n incremental	non-incremental	227	31.5
	context sensitive	201	19.4
	context+utterance	223	30.5

Calendar domain data [20] as in [10].[3] This corpus contains 9 utterances with 6-7 phrases each (totalling 59 phrases and 243 words).

We generated both phrase-incremental as well as non-incremental symbolic intonation. Intonation information that derives from processing the next phrase was either used only after the current phrase (this corresponds to the w_n condition in [10]) or integrated back into the last word of the current phrase (corresponding to the w_{n-1} condition). We combined these symbolic intonation sources with non-incremental and *context sensitive* (normal and including utterance-level features) prosodic parameters.

The results for f_0 and duration RMSE relative to completely non-incremental processing are shown in Table 4. It turns out that integrating the next phrase and recomputing symbolic intonation before speaking the last word of the current phrase (the w_{n-1} condition) only incurs a very small performance penalty as compared to non-incremental symbolic intonation assignments. In both cases, *context+utt* outperforms plain *context sensitive*, but still leaves considerable error (RMSE of roughly $1/2$ a semitone) as compared to completely non-incremental sub-symbolic prosody assignments. The zero f_0 error of w_{n-1} combined with non-incremental prosody assignment indicates that f_0 assignments only differ in the final word of the phrase, at least in these nine utterances; the small duration error must stem from the timing differences in penultimate words of phrases. Finally, incorporating symbolic intonation only after the phrase (w_n condition) is too late to lead to plausible mid-utterance prosody and the fact that using less features for parameter estimation performs better indicates that the substituted defaults are in fact better than the sentence-end intonations falsely produced by incremental intonation processing without lookahead.

7. Conclusion

This paper has analyzed the advantage of flexibly using all available symbolic intonation features when assigning sub-symbolic prosody parameters (f_0 and duration). Compared to statically limiting the feature context to some class (which then results in a certain lookahead requirement in incremental processing), the *context sensitive* method performs much better: with a lookahead of one word, it radically outperforms the static *current word* condition and approaches the performance of the static *current phrase* method (which, however, requires a lookahead of a whole phrase). The performance gain is maximized when utterance-level information is taken into account for utterance-final words.

We find the majority of errors to occur phrase- and utterance-finally, that is, when utterance/phrase-final information can be reasonably considered to be available. This corresponds well with the fact that speech itself is an incremental phenomenon and, as human speakers often change or extend their utterances while producing them, only relies on information that is known at the time of realization.

We test our method with simplistic incremental intonation assignments and find that integrating a next phrase before the last word of the ongoing phrase already leads to results that are similar to the non-incremental intonation condition and only differ by ½ semitone and 7 ms RMSE from complete non-incremental intonation and prosody assignments. These results radically outperform [10] which require almost a full phrase of lookahead (w_1 condition in [10]) for similar results.

Our current implementation does not yet determine all features from the incrementally available data, but uses non-incrementally produced feature sets that are trimmed off the features that are not available or reliable in the incremental setting. This implementation is inefficient and we work on changing it to truly incremental feature extraction from the IU network. Another area of future work is to build decision trees that are specifically designed for incremental use-cases, for example with an implementation that explicitly supports missing features instead of using global default values, with a possible intermediate solutions of context-dependent defaults.

We have, so far, only analyzed the word-by-word extension of ongoing synthesis. It will be interesting to investigate how *changes* (of the words to be spoken, or phrase-level intonation to be realized) need to be handled, for example, how quickly a change of phrase-level intonation should be realized by prosodic parameters. Finally, we plan to validate the reported numerical evaluation results in a listening experiment.

8. Resources

The methods presented in this paper have been added to the iSS component of InproTK. InproTK is free and open-source software and is available at `http://inprotk.sf.net`. MaryTTS, which forms the basis of the present work, is available at `http://mary.dfki.de`. Example audio files for the various lookahead conditions are included with the proceedings.

9. Acknowledgements

This work is supported through a PostDoc grant by Daimler and Benz Foundation. The author would like to thank Arne Köhn, Wolfgang Menzel, and Nigel Ward for valuable remarks.

10. References

[1] W. Wahlster, *Verbmobil: Foundations of Speech-to-Speech Translation.* Springer Verlag, 2000, ch. Mobil Speech-to-Speech Translation of Spantaneous Dialogs: An Overview of the Final Verbmobil System, pp. 3–21.

[2] S. Bangalore, V. K. Rangarajan Sridhar, P. Kolan, L. Golipour, and A. Jimenez, "Real-time incremental speech-to-speech translation of dialogs," in *Proceedings of NAACL-HTL 2012*, Montréal, Canada, Jun. 2012, pp. 437–445.

[3] Note however that for several reasons the numbers in [10] cannot directly be compared to the numbers presented in this section.

[3] D. L. Chen and R. J. Mooney, “Learning to sportscast: A test of grounded language acquisition,” in *Proceedings of 25th International Conference on Machine Learning (ICML-2008)*, Helsinki, Finland, Jul. 2008.

[4] T. Baumann, “Incremental spoken dialogue processing: Architecture and lower-level components,” Ph.D. dissertation, Universität Bielefeld, Germany, 5 2013.

[5] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, “ToBI: A standard for labeling english prosody,” in *Second International Conference on Spoken Language Processing*, Alberta, Canada, Oct. 1992.

[6] W. J. Levelt, *Speaking: From Intention to Articulation*. Mit Pr, 1989.

[7] M. Schröder and J. Trouvain, “The German text-to-speech synthesis system MARY: A tool for research, development and teaching,” *International Journal of Speech Technology*, vol. 6, no. 3, pp. 365–377, Oct. 2003.

[8] P. Taylor, *Text-to-Speech Synthesis*. Cambridge Univ Press, 2009.

[9] T. Baumann and D. Schlangen, “INPRO_iSS: A component for just-in-time incremental speech synthesis,” in *Procs. of ACL System Demonstrations*, Jeju, Korea, July 2012.

[10] ——, “Evaluating prosodic processing for incremental speech synthesis,” in *Proceedings of Interspeech*. Portland, USA: ISCA, Sep. 2012.

[11] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Wadsworth, Monterey, USA, 1984.

[12] O. Watts, J. Yamagishi, and S. King, “The role of higher-level linguistic features in HMM-based speech synthesis,” in *Proc. Interspeech*, Makuhari, Japan, Sep. 2010, pp. 841–844.

[13] M. Cernak, P. Motlicek, and P. N. Garner, “On the (un)importance of the contextual factors in HMM-based speech synthesis and coding,” in *Proceedings of ICASSP*, 2013.

[14] T. Baumann, “Decision tree usage for incremental parametric speech synthesis,” in *Proceedings of the International Conference on Audio, Speech, and Signal Processing (ICASSP 2014)*, 5 2014.

[15] T. Dutoit, M. Astrinaki, O. Babacan, N. d’Alessandro, and B. Picart, “pHTS for Max/MSP: A streaming architecture for statistical parametric speech synthesis,” Université de Mons, Tech. Rep. 1, 3 2011. [Online]. Available: http://www.numediart.org/docs/numediart_2011_s13_p2_report.pdf

[16] M. Astrinaki, N. d’Allessandro, B. Picart, T. Drugman, and T. Dutoit, “Reactive and continuous control of hmm-based speech synthesis,” in *IEEE Workshop on Spoken Language Technology (SLT 2012)*, Miami, USA, 2012.

[17] T. Toda and K. Tokuda, “A speech parameter generation algorithm considering global variance for hmm-based speech synthesis,” *IEICE transactions on information and systems*, vol. 90, no. 5, pp. 816–824, 2007.

[18] V. Chunwijitra, T. Nose, and T. Kobayashi, “A speech parameter generation algorithm using local variance for hmm-based speech synthesis,” in *Proceedings of Interspeech*. Portland, USA: ISCA, Sep. 2012.

[19] H. Kawahara, “Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited,” in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., IEEE International Conference on*, vol. 2. IEEE, 1997, pp. 1303–1306.

[20] H. Buschmeier, T. Baumann, B. Dorsch, S. Kopp, and D. Schlangen, “Combining incremental language generation and incremental speech synthesis for adaptive information presentation,” in *Proceedings of SigDial*, Seoul, Korea, 2012, pp. 295–303.

[21] T. Baumann and D. Schlangen, “Interactional adequacy as a factor in the perception of synthesized speech,” in *Proceedings of Speech Synthesis Workshop (SSW8)*, 2013.

[22] S. Kousidis, C. Kennington, T. Baumann, H. Buschmeier, S. Kopp, and D. Schlangen, “Situationally aware in-car information presentation using incremental speech generation: Safer, and more effective,” in *Dialogue in Motion Workshop*, 4 2014.

[23] T. Baumann and D. Schlangen, “The INPROTK 2012 release,” in *Proceedings of SDCTD*, Montréal, Canada, 2012.

[24] K. Kohler, “Erstellung eines Textkorpus für eine phonetische Datenbank des Deutschen,” in *Arbeitsberichte des Instituts für Phonetik der Universität Kiel (AIPUK)*, K. Kohler, Ed., 1992, vol. 26, pp. 11–39.

[25] T. Ellbogen, F. Schiel, and A. Steffen, “The BITS speech synthesis corpus for German,” in *Proceedings of LREC*, 2004.

[26] DIN 13320:1979-06, “Acoustics; spectra and frequency curves, concepts, representation,” German Institute for Standardization (DIN), Jun. 1979.

[27] R. A. Clark and K. E. Dusterhoff, “Objective methods for evaluating synthetic intonation,” in *Proceedings of Interspeech*, 1999.