Evaluating Prosodic Processing for Incremental Speech Synthesis

Timo Baumann

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

David Schlangen

Faculty for Linguistics and Literary Studies
Bielefeld University, Germany
david.schlangen@uni-bielefeld.de

Abstract

Incremental speech synthesis (iSS) accepts input and produces output in consecutive chunks that only together result in a full utterance. Systems that use iSS thus have the ability to adapt their utterances while they are ongoing. However, starting to process with less than the full utterance available prohibits global optimization, leading to potentially suboptimal solutions. In this paper, we present a method for incrementalizing the symbolic pre-processing component of speech synthesis and assess the influence of varying "lookahead", i. e. knowledge about the rest of the utterance, on prosodic quality. We found that high quality incremental output can be achieved even with a lookahead of less than one phrase, allowing for timely system reaction.

Index Terms: speech synthesis, spoken dialogue systems, incrementality, prosody

1. Introduction

Incremental processing is a mode of operation in which the first bits of output are produced before all input has been obtained. This makes it possible for spoken dialogue systems to generate output faster by folding much of the processing time into the (relatively slow) speech delivery resulting in faster turn-taking behaviour [1], and to react to aspects of the user's ongoing input (e. g. by giving feedback [2]). Incremental speech *output* processing in particular, i. e. the incremental generation and synthesis of the system's turn, allows for adaptations of the ongoing utterance; for example, the system may react by rephrasing or repeating parts of the utterance if its interlocutor makes a puzzled face, or if noise interfered with delivery [3].

So far, work on incremental processing has mostly focussed on incrementalizing input processing components such as speech recognition [4, 5], or NLU [6, 7]. Incremental input processing also needs to determine whether a preliminary hypothesis will turn out to be correct [8], and if this decision is not taken timely, performance degrades gracefully. In contrast, output generation operates under real-time constraints: if the system has begun to produce observable output, it needs to continue to do so, even when information necessary to produce optimal continuations is yet unavailable; of course, the system may hesitate but the hesitation should be modelled plausibly and not result from an interruption of the audio stream.

Most state-of-the-art speech synthesis technology relies on the full (textual) specification of the utterance that is to be produced [9] and works top-down from this specification to the audio; this inhibits its use in incremental systems. Furthermore, modern speech synthesis systems combine different and complex techniques on various levels of linguistic and acoustic processing and it is hence far from trivial to devise incremental versions of all of these techniques (if at all possible).

Some of these processing steps, such as HMM optimization and vocoding, are processing intensive but fairly independent of other steps (since they just require the phonetic target sequence that is to be synthesized); they can be incrementalized to fold their processing time into speech delivery time. Others, such as symbolic linguistic pre-processing and prosodic assignment, however, use various types of linguistic information about the full utterance, in order to produce optimal output. For example, prosodic assignment takes into account the full utterance when assigning pitch targets to speech segments. However, these symbolic processing steps are relatively fast, which makes it possible to integrate them into an incremental system by re-executing them when new information becomes available.

In this paper, we analyze the impact on prosody of using the non-incremental linguistic pre-processing steps of MaryTTS [10] in various reduced lookahead scenarios as part of an incremental speech output system. Our results indicate that incremental processing can lead to good intonation contours if at least some lookahead is available. In the next section, we review some related work before describing our component for incremental speech synthesis in Section 3, and the design space for incremental prosodic processing in Section 4. We then describe our experiment on lookahead conditions in Section 5 and close with conclusions in Section 6,

2. Related Work

In the psycholinguistics literature (e. g. [11]), it is rarely disputed that speech is produced incrementally. Specifically, Levelt claims that rhythm can be assigned with a one word lookahead, and intonation with even less [11]. Current main-stream speech synthesis systems, however, do not perform incremental prosody processing [9], and use e. g. CARTs with non-local features (such as '#words to utterance end') for prosody assignment [10].

The desirability of incremental speech synthesis has been commented on before; Edlund [12], e. g., lists as desiderata *interruptability* during the utterance, *giving feedback* on delivery to other system components, and, of course, *real-time performance*. However, his system only performs non-incremental diphone synthesis prior to utterance delivery. For HMM synthesis, an incremental version of the (previously global) HMM emission optimization has recently been proposed [13], with only a small degradation of synthesis quality.[1] This component, however, was not integrated into an incremental synthesis system, which we aim at here.

[1] We are not aware of any work on incremental unit selection synthesis, and we believe that the non-locality of the search problem in unit selection would make this much harder. Thus, we chose HMM synthesis in this paper. However, the synthesis method itself has no impact on our results, as we target the slightly higher level of prosodic processing.

In this paper we extend earlier work on our incremental speech synthesis (iSS) component [14] which builds on INPROTK [15], an incremental dialogue processin architecture [16]. We recently built a system [3] that combined iSS with a component for incremental natural language generation (iNLG), which produces output in chunks that roughly correspond to intonation units. In that system, the incremental approach lead to radically faster system response time (an average reduction around 1 second, or 66 %) and allowed for a strategy to cope with interfering noise that was rated as highly more natural than strategies building on non-incremental processing.

The work presented in this paper assesses the prosodic quality (pitch and duration deviation) of incrementally produced output under various lookahead conditions. This has not previously been analysed for iSS. Following previous work on incrementality evaluation [17] we compare against a non-incremental gold standard, focussing on the deviation of ideal delivery [18]. We analyse the lookahead vs. quality tradeoff similarly to [4].

3. Incremental Speech Synthesis

In our view, incremental speech synthesis (iSS) needs more than the requirements cited above (*interruptability* and *execution feedback*). We require that iSS also support the *extension* of an ongoing utterance and the *adaptation* of as-yet unspoken parts of it. These requirements are best met by a processing paradigm in which processing takes place *just-in-time*, i. e., where processing steps are taken as late as possible, avoiding re-computations.

As mentioned in the introduction, speech synthesis is conventionally performed top-down, requiring the higher level to finish before processing starts on the next lower level. However, doing it like this is not a logical necessity, as we found, and working out all the details at one level is not a precondition to starting work at the next level. To produce a preliminary sentence-level intonation, some knowledge about overall utterance structure is required, but individual words need not necessarily be known. Likewise, post-lexical phonological processes can be computed as long as a very limited local context (one word) is available; vocoding parameter computation (which must model co-articulation effects) requires even less context (about one phoneme); and vocoding does not need any lookahead at all (aside from audio buffering considerations).

Thus, our data structure, which is composed of incremental units [19], is built incrementally in a triangular top-down and left-to-right fashion with different amounts of pre-planning (see Figure 1) by processing modules that work concurrently. As can be seen in the figure, just enough audio is being vocoded to keep audio buffers full, and just enough vocoding parameter frames are computed by HMM parameter optimization [13] as are necessary to model co-articulation for the current segment.

Our component is fed with *chunks*, short sub-sequences of words from a higher-level component such as an incremental NLG. Chunks are handed to MaryTTS' linguistic processing and the results are translated into our incremental data structures based on INPROTK [15]. These are then integrated into the current utterance structure, becoming available to the lower-level incremental sub-modules for synthesis.

In Figure 1, the moment at which the higher-level component is queried for more words to be appended to the ongoing utterance is two phonemes before the system would otherwise run out of more synthesis material (this allows to to account for co-articulation). In the following section we discuss the influence of the choice of when new material is requested on the quality of the prosodic assignment that can be achieved.

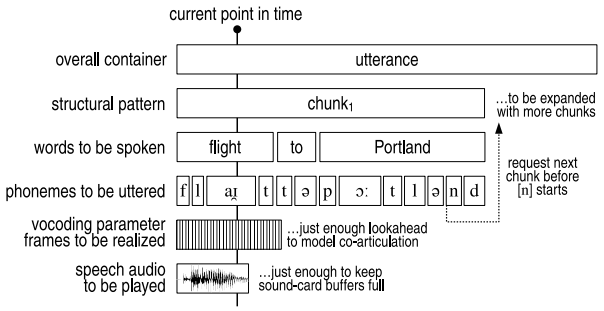

Figure 1: Hierarchic structure of incremental units describing an example utterance as it is being produced during delivery, showing the event-based just-in-time processing strategy.

4. Incrementalized Prosodic Processing

Prosody is influenced by *long-range dependencies* (e. g., phrase intonation depends on the finality of the phrase, and rhythm clashes potentially many words ahead may influence stress assignment [11]). Nonetheless, human performance indicates that most often prosody can be generated on-the-fly. Thus, our goal is to generate a prosodic assignment for the utterance incrementally (with only limited amounts of the utterance available and requesting further words just-in-time) that matches the prosody produced non-incrementally by the TTS as closely as possible.

Imagine the system produces the utterance in Example 1 (with 'I' denoting chunk boundaries from NLG) incrementally:

(1) your flight | on September 8th 2012 | to PDX via EWR | has been booked

After synthesis starts, the question arises of *when* more words need to be added, and of *how many* words should be added at a time. These are the questions of lookahead and granularity, respectively.[2] When new words are processed, there is also the question of how much of the already known words should be taken into account as *left context*. The full design space for incremental prosody production is illustrated in Figure 2.

In our system, we add words at a *granularity* of chunks (as they are generated by an incremental NLG component), which roughly correspond to prosodic phrases. Adding full phrases at a time has the advantage of feeding the TTS' symbolic processing component with 'sensible' input for prosodic assignment (i. e., querying for "to PDX via EWR" will likely better match the full utterance than querying for "8th 2012 to PDX").

Using *left context* may provide important clues to the symbolic processor, and our system uses all of this information (potentially at the cost of some processing time).

Regarding *lookahead*, we have to decide on when to demand a next chunk. Our just-in-time principle would dictate to do this as late as possible (as in Figure 1; in the example in Figure 2: at position w_n). However, combined with using left context, if we have available more lookahead, it may be possible to produce a smoother transition to the next chunk by re-computing even the current phrase in light of its now known continuation, and adapting the prosody of those parts of it that have not been spoken yet. In our example, when we append chunk_3 ("to PDX via EWR"), we can process this chunk together with chunks 1 and 2 as left context. It is very likely that now the part of the utterance corresponding to chunk_2 ("on September 8th 2012") will be assigned

[2] iSS in general is agnostic to both lookahead and granularity. However, both are crucial in devising a plausible prosody.

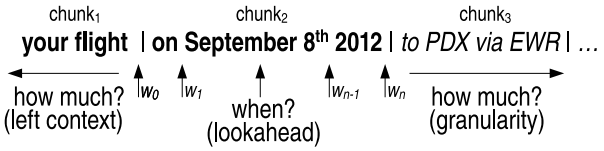

Figure 2: Design space for incremental prosody production. To append the chunk "*to PDX via EWR*", we need to decide (a) when to do this, (b) how much left context to use, and (c) how much to add at a time.

a much better intonation. If we use lookahead, we are still able to use this better intonation instead of the originally computed one in the yet unspoken parts of chunk$_2$.[3]

The more lookahead, the more the ongoing phrase can be reworked—but there is a trade-off, as this also makes the strategy less incremental, requiring more planning-ahead. The exact extent of this trade-off is what we explore systematically with the experiment below.

5. Experiment

5.1. Experimental Conditions

As described above, we explore the influence of lookahead on prosodic quality. The extreme settings of incrementality in the design space are as follows:

non-incremental we use this as the control condition;

trivially incremental synthesize every phrase in isolation when the current phrase ends (a strategy used e. g. by [20]);

only left context no lookahead (like **trivial**), but use left context, allowing better prosodic connection, esp. at the onset of phrases; (All other settings below also use left context.)

full phrase lookahead require the next phrase before starting production of the current phrase; this corresponds to the arrow labelled w_0 in Figure 2 and was done in our previous experiment [3].

We denote **intermediate settings** with w_i: w_1 is the setting that integrates the next chunk after the first word of the current phrase, w_{n-1} integrates one word before the end of the current phrase, and so on. (Hence 'full phrase lookahead' and 'only left context' settings can be described as w_0 and w_n, respectively.)

In all settings, we add a full phrase at a time, i. e., we do not vary the update granularity.

We expect that providing left context in the w_n setting will improve quality (as measured by similarity to non-incremental, full-utterance synthesis) of the beginning of each new chunk, and that increasing lookahead (i. e. moving "to the left" in Figure 2) will additionally improve quality of the endings of each chunk. We tested these assumptions by analysing output generated under these conditions.

5.2. Exemplary Analysis

Figure 3 shows parts of exemplary pitch tracks generated in different settings (omitting some intermediate settings for clarity).

As can be seen, the prosody of the output when following the "trivially incremental" strategy deviates rather strongly from the non-incremental condition, both at phrase beginnings and endings. Condition w_n successfully reduces the deviation at phrase beginnings as these are produced with left context (i. e. avoiding onset intonations within the utterance). However, w_n still shows strong deviations at phrase endings (strong shifts within the phrase-final word), when a preliminary utterance-final intonation is replaced with a continuation intonation. Condition w_1 follows the non-incremental pitch track relatively closely.

We will quantify these findings on more data and for all settings in Subsection 5.4 below.

[3] We should avoid to introduce discontinuities into the pitch track, as they would sound unnatural. This can easily be avoided e. g. by gradually adjusting and enforcing a maximum gradient. We have not implemented this for the quantitative evaluation reported in Section 5.4; it would somewhat increase RMSE.

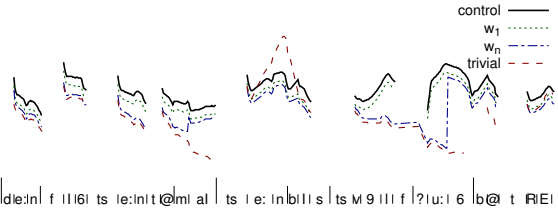

Figure 3: Exemplary pitch tracks of part of an utterance under some synthesis conditions; phrase boundaries are marked by higher vertical markings between segments.

5.3. Data

We generated six complex utterances from a calendar domain (informing about upcoming events, conflicts, or rescheduled appointments) in German. Each utterance contains 6–7 'chunks' (content units produced by incremental NLG), and on average 27 words (composed of on average 5.3 phonemes). We synthesized all utterances (even the non-incremental version) using our limited-lookahead HMM optimizer and vocoder, resulting in about 70 seconds of speech per condition.

We then compared the prosodic assignments (pitch and segment durations), based on the synthesis logfiles, against non-incremental synthesis results as a gold standard. An incremental system can impossibly reliably outperform its non-incremental sibling, which is thus an adequate upper bound.

5.4. Results

Table 1 summarizes how the prosody (duration and pitch assignments) under the various settings deviates from that when synthesizing non-incrementally, using root mean squared error (RMSE) as metric. We additionally show the 95 % quantile of the error magnitude for pitch.

As can be seen in the table, the RMSE for both duration and pitch in the w_0 (full phrase) condition is remarkably low at less than a millisecond for duration and about 7 Hz for pitch (95 % of pitch values are within 10 Hz of the gold standard and 98 % remain within 25 Hz). In our corpus, mean phoneme duration is 81 ms, thus a duration's RMSE of 0.81 ms is a deviation of 1 %, well below the just-noticeable difference (JND) for speech tempo of roughly 5 % [21]. Likewise, mean pitch was 172 Hz thus an RMSE of 7.08 Hz is a deviation of 4.1 %, close to the JND for pitch in speech [22].

Quite importantly, using left context drastically cuts down on pitch and timing deviation even if no lookahead at all is available. Also, we see that deviation increases the later phrases are appended, i. e., the less lookahead is available. The almost linear relationship between lookahead and prosodic quality is shown in Figure 4.

condition	timing deviation RMSE (in ms)	pitch deviation (in Hz) RMSE	95 % quantile
w_0 (full phrase)	0.81	7.08	10
w_1	1.16	8.32	19
w_2	3.37	11.27	27
w_3	5.01	15.10	37
w_{n-1}	5.01	17.40	46
w_n (w/ left context)	5.47	18.42	50
w_n (trivial)	14.70	28.42	67

Table 1: Deviation of look-ahead conditions from the non-incremental control condition.

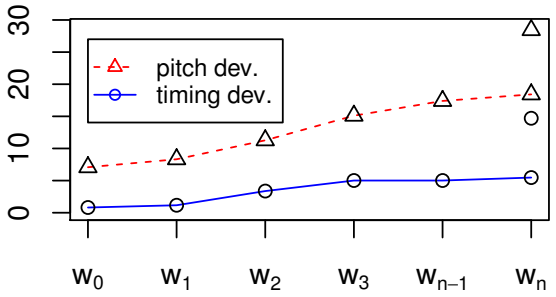

Figure 4: Deviation of pitch and timing plotted against lookahead. The more lookahead available, the better the results. (The single points on the right represent the trivial setting's performance.)

6. Conclusions

We conclude that good prosodic assignments can be achieved using incremental processing. However, some *lookahead* is desirable to allow the prosody processor to take some right context into account. The more lookahead is available, the closer the produced prosody is to non-incremental processing, reaching levels below or close to JND for one phrase of lookahead.

We expect that our results from the objective analysis carry over to subjective user ratings. Our informal trials and our previous experience with the component [3], where iSS output was actually preferred, indicate this, especially because iSS allows for otherwise more natural system behaviour.

However, lookahead also comes at a cost, namely the timeliness of picking up changes in the outside world. We believe that roughly one phrase of lookahead is a good rule-of-thumb for the tradeoff between timeliness and speech output quality. w_0 has the utterance-initial disadvantage of needing to wait for the first two chunks from iNLU before delivery may start. In contrast, the slightly inferior w_1 condition may fold the processing time for chunk$_2$ into the first word spoken. This is why we suggest the w_1 condition, at least if NLG is processing intense.

Our approach of wrapping an existing, non-incremental prosody processor works surprisingly well. However, we believe that fully integrated, incremental prosody processing might be able to cope with even less lookahead. It might especially better be able to allow for variable update timings instead of fixed lookaheads as used in this paper, or to sensibly integrate errors that are due to very 'late' changes of the utterance; something we plan to address in future work.

Acknowledgements The authors would like to thank Hendrik Buschmeier for providing iNLG output.

7. References

[1] G. Skantze and D. Schlangen, "Incremental dialogue processing in a micro-domain," in *Proceedings of EACL*, Athens, Greece, Apr. 2009, pp. 745–753.

[2] O. Buß, T. Baumann, and D. Schlangen, "Collaborating on utterances with a spoken dialogue system using an ISU-based approach to incremental dialogue management," in *Proceedings of SigDial*, Tokyo, Japan, Sep. 2010, pp. 233–236.

[3] H. Buschmeier, T. Baumann, B. Dorsch, S. Kopp, and D. Schlangen, "Combining incremental language generation and incremental speech synthesis for adaptive information presentation," in *Procs. of SigDial*, Seoul, Korea, 2012.

[4] T. Baumann, M. Atterer, and D. Schlangen, "Assessing and improving the performance of speech recognition for incremental systems," in *Proceedings of NAACL-HLT*, Boulder, USA, Jun. 2009, pp. 380–388.

[5] E. Selfridge, I. Arizmendi, P. Heeman, and J. Williams, "Stability and accuracy in incremental speech recognition," in *Proceedings of SigDial*, Portland, Oregon, Jun. 2011, pp. 110–119.

[6] K. Sagae, G. Christian, D. DeVault, and D. Traum, "Towards natural language understanding of partial speech recognition results in dialogue systems," in *Proceedings of NAACL-HLT*, Boulder, USA, Jun. 2009, pp. 53–56.

[7] A. Peldszus, O. Buß, T. Baumann, and D. Schlangen, "Joint satisfaction of syntactic and pragmatic constraints improves incremental spoken language understanding," in *Procs. of EACL*, 2012.

[8] D. DeVault, K. Sagae, and D. Traum, "Can I finish? Learning when to respond to incremental interpretationresults in interactive dialogue," in *Procs. of SigDial*, London, UK, Sep. 2009, pp. 11–20.

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

[10] 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, 2003.

[11] W. Levelt, *Speaking: From Intention to Articulation*. MITP, 1989.

[12] J. Edlund, "Incremental speech synthesis," in *Second Swedish Language Technology Conference*, Stockholm, Sweden, Nov. 2008, pp. 53–54, System Demonstration.

[13] T. Dutoit, M. Astrinaki, O. Babacan, N. d'Alessandro, and B. Picart, "pHTS for Max/MSP: A streaming architecture for statistical parametric speech synthesis," numediart Research Program on Digital Art Technologies, Tech. Rep. 1, Mar. 2011.

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

[15] ——, "The INPROTK 2012 release," in *Proceedings of SDCTD*, Montréal, Canada, 2012.

[16] D. Schlangen, T. Baumann, H. Buschmeier, O. Buß, S. Kopp, G. Skantze, and R. Yaghoubzadeh, "Middleware for incremental processing in conversational agents," in *Proceedings of SigDial*, Tokyo, Japan, Sep. 2010, pp. 51–54.

[17] T. Baumann, O. Buß, and D. Schlangen, "Evaluation and optimisation of incremental processors," *Dialogue & Discourse*, vol. 2, no. 1, pp. 113–141, 2011, Special Issue on Incremental Processing in Dialogue.

[18] T. Baumann and D. Schlangen, "Predicting the micro-timing of user input for an incremental spoken dialogue system that completes a user's ongoing turn," in *Proceedings of SigDial*, Portland, USA, Jun. 2011, pp. 120–129.

[19] D. Schlangen and G. Skantze, "A general, abstract model of incremental dialogue processing," in *Proceedings of EACL*, Athens, Greece, Mar. 2009, pp. 710–718.

[20] G. Skantze and A. Hjalmarsson, "Towards incremental speech generation in dialogue systems," in *Proceedings of SigDial*, Tokyo, Japan, Sep. 2010, pp. 1–8.

[21] H. Quené, "On the just noticeable difference for tempo in speech," *Journal of Phonetics*, vol. 35, no. 3, pp. 353–362, 2007.

[22] S. G. Nooteboom, "The prosody of speech: Melody and rhythm," in *The Handbook of Phonetic Sciences*, W. J. Hardcastle and J. Laver, Eds. Oxford: Blackwell, 1997, pp. 640–673.