JNDSLAM: A SLAM extension for Speech Synthesis

Rasmus Dall[1]*, Xavi Gonzalvo*[2]

[1]The Centre for Speech Technology Research, University of Edinburgh, UK
[2]Google Inc., UK

r.dall@sms.ed.ac.uk, xavigonzalvo@google.com

Abstract

Pitch movement is a large component of speech prosody, and despite being directly modelled in statistical parametric speech synthesis systems very flat intonation contours are still produced. We present an open-source fully data-driven approach to pitch contour stylisation suitable for speech synthesis based on the SLAM approach. Modifications are proposed based on the Just Noticeable Difference in pitch and tailored to the need of speech synthesis for describing the movement of the pitch. In an anchored Mean Opinion Score (MOS) test using oracle labels the proposed method shows an improvement over standard synthesis. Long Short-Term Memory Neural Networks were then used to predict the contour labels, but initial experiments achieved low prediction rates. We conclude that using current linguistic features for pitch stylisation label mapping is not feasible unless additional features are added. Furthermore an open-source implementation is released.

Index Terms: HMM, TTS, LSTM, prosody, pitch contour, speech synthesis

1. Introduction

Statistical parametric speech synthesis (SPSS) has long been deemed to overtake unit selection as the synthesis method of choice. SPSS is more flexible [1, 2] and has a smaller footprint [3], however it is still not deemed as high quality as the best unit selection systems [3]. Although recent technological advances [4, 5] are closing the gap, there is one area in which SPSS is still greatly lacking. Namely prosody. SPSS is known for having a flat intonation with little prosodic variance, and this is noticeable even in synthesis of neutral read speech. While unit selection will also falter in more advanced prosodic elements, it does not have a comparable problem for neutral read speech as it will always have "natural" pitch movements from the selected units. SPSS, on the other hand, will deliver an averaged pitch contour which will tend to be flat. Furthermore studies have shown that current higher level linguistic features contribute little to the overall perception of the system [6, 7], it is thus desirable to investigate potential new features.

In this paper we present an extension of the Stylisation and LAbelling of speech Melody (SLAM) method of Obin et al. [8] more suitable for speech synthesis and initial attempts at predicting these from text. SLAM is a pitch contour stylisation method and while the pitch contour is not the only relevant element of prosodic intonation contours (other phenomena include durations, pausing and energy), it is considered one of the main essential elements and is, besides durations, the only prosodic element directly modelled in SPSS systems [9].

Stylisation of pitch has been done before (e.g. [10, 11, 12, 13]), but in TTS most famously through the Tone and Break Indices (ToBI) labelling scheme [14]. However, the main advantage of SLAM over other methods is that it is a fully data-driven method which requires no human labelling effort and works on any segment length desired.

ToBI can be semi-automatically [15] or fully automatically labelled [16, 17]. However, for automatic labelling, classifiers are used and while reasonably accurate these do not match human annotator agreement [16]. Furthermore, the inventory set is fixed and if it is to be used for a new language then the inventory must be modified [18]. Stem-ML [12] can automatically label data without the use of classifiers, however needs some manual guidance on the inventory decision and boundary settings [19]. Tilt [10, 20] can label data and infer the inventory, however needs a detection algorithm to determine where an intonational event happens. INTSINT [13] can label data and infer an inventory without a detection algorithm, however the unit size of each label is not clearly defined, except that some number of labels exist in each intonational phrase. For TTS, in order to label our data we use phonemes as the unit size and it must be clear to which label each phoneme belongs, and it is not clear how INTSINT, currently, would guarantee this.

SLAM has none of the above limitations as the inventory is derived from and defined by the data on any given unit size. As such it lends itself easily to application in TTS as a label can be related to each phoneme, although that label may not be determined at the phoneme level. That SLAM can derive an inventory set on any segment length desired is attractive, and it can do this while still having a sufficiently small inventory describing most of the data [8]. SLAM was, however, not designed specifically with TTS in mind. As a consequence the inventory cut-off points are not based on perceptually noticeable differences and may fail to take into account the amount of movement of the pitch. We here present an extension of the method more suitable for TTS by taking into account the Just Noticeable Difference (JND) of pitch and a more explicit modelling of the movement of the pitch. This method is presented in two versions, a standard and simplified.

This paper is therefore organised as follows. SLAM is summarised in Section 2, our extension, JNDSLAM, presented in Section 3. Section 4 shows evidence that JNDSLAM can improve SPSS and Section 5 present an initial attempt at predicting JNDSLAM labels before concluding in Section 6.

2. SLAM

SLAM provides a stylisation of the pitch of any given speech segment. In this paper we focus solely on the syllable, however the method is the same for any length of speech. From the raw pitch values the speakers mean pitch is calculated over the entire corpora. Then for each syllable the pitch values are smoothed and all values are converted into semitone deviance from the mean pitch. Note, in the implementation from [8] pitch is ex-

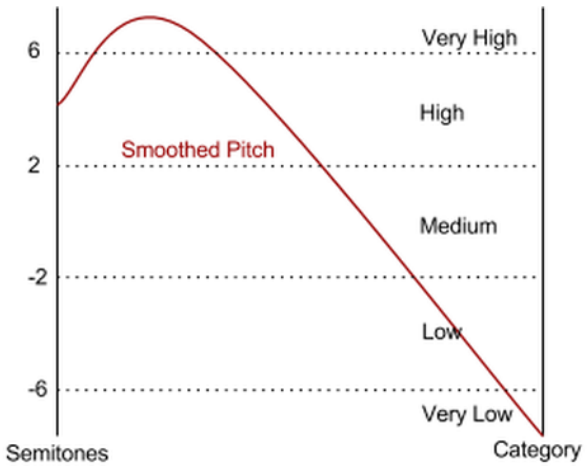

Figure 1: A sample pitch contour. The starting value is High. End is Very Low and a Very High extreme position exists in the beginning of the segment.

tracted using SWIPE [21] and smoothed using LOWESS [22], though in principle any methods can be used. In our case we used the pitch tracker REAPER [23] instead of SWIPE.

Three values define the resulting stylisation:

- The starting level of the pitch relative to the mean
- The ending level of the pitch relative to the mean
- Any extreme position, its position in the segment and it's level.

Five levels, of 4 semitones size, for each position is defined relative to the mean semitone.

- Very High: 6 semitones or more above the mean.
- High: 2-6 semitones above the mean.
- Medium: -2 to 2 semitones around the mean.
- Low: -2 to -6 semitones below the mean.
- Very Low: -6 semitones or more below the mean.

Of these, extreme positions are the most involved. Start and End is simply the level relative to the mean semitone value of the pitch, but extremes are more relational. To determine the presence of an extreme, the difference from the starting and ending position to the most extreme (positive or negative) pitch value in the segment is calculated. If the *smallest* of these differences is more than 2 semitones, an extreme is present and its level is determined. The position of the extreme is recorded as in which third of the segment it appears in. Figure 1 illustrates a sample smoothed pitch contour with an extreme position. With a label for an unvoiced segment this results in a label set of 401 possible labels. Of the 401 possible labels, 311 were present in our data (see below for corpora details). However, 14 of these represent more than 95% of the data. This is higher than the 8 contours in the original study [8]. This is likely due to the fact that our corpora is in English and the original authors use French - the two languages likely differ in inventory naturally. It does however demonstrate that the method is not language specific as the relevant inventory is still small and within the 10-20 suggested as reasonable by SLAM's original authors [8].

2.1. Corpora

Throughout this paper the corpora used was from a single US English Female speaker. The corpora was recorded for the purpose of speech synthesis and contains 641k syllables. This is

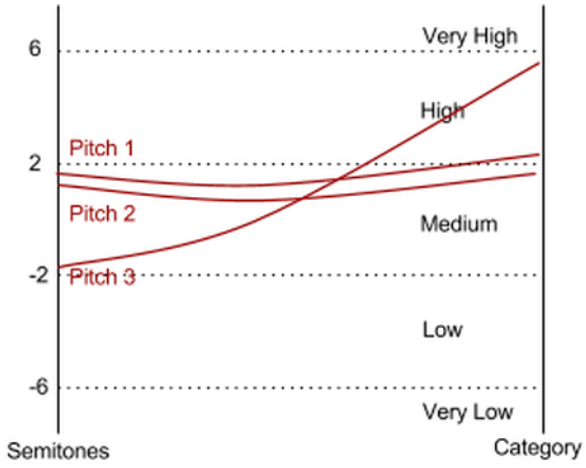

Figure 2: SLAM would categorise Pitch 1 and 3 as the same. Evidently Pitch 1 and 2 are more similar and JNDSLAM catches this distinction by defining the end point in terms of direction and strength of movement from the start.

over an order of magnitude larger than the test set used in [8] which contained 43k syllables.

3. Just Noticeable Difference SLAM

For the use of SLAM in SPSS there are two issues. Firstly the level categories do not necessarily represent a perceptually noticeable difference in pitch. This is an issue as we are interested in reproducing perceptually meaningful movements in SPSS. We thus suggest to use the Just Noticeable Difference (JND) in pitch as a perceptually meaningful categorisation, this is 1.5 semitone [24, 25]. By using categories of two JND in size we ensure that from the mean value there is always at least 1 JND of movement to the next level, this is a slightly smaller range than the original SLAM's size of 4 semitones. Secondly SLAM may categorise pitch contours that are very similar as being quite different due to the static nature of the End element. Figure 2 illustrates the issue showing three pitch contours. Contour 1 and 2 are very similar, whereas contour 3 is different. However because contour 1 starts at 1.8 semitones above the mean and ends at 1.9, and contour 2 goes from 1.95 to 2.05 they are classified as being different (Medium to medium and medium to high). Thus contour 2 and 3 are being classified as the same and 1 as different. This is not ideal as in TTS we wish to capture the movement of the pitch in order to recreate it. The proposal is therefore to define the End position as the movement relative to the Start position, with movement levels following the categorisation. A similar issue exists with extreme positions around the category borders, and we propose to simply define any extremes as being present if the smallest difference to the Start or End is above the JND, whether it is positive or negative and its position in the segment. Just Noticeable Difference SLAM (JNDSLAM) thus stylise a pitch with the following three values:

- The starting position of the pitch relative to the mean
- The direction and strength of movement of the pitch relative to the starting position
- Any extreme value as determined by the JND and its position in the segment

With the following five positional and movement levels (except for extreme which is either positive or negative):

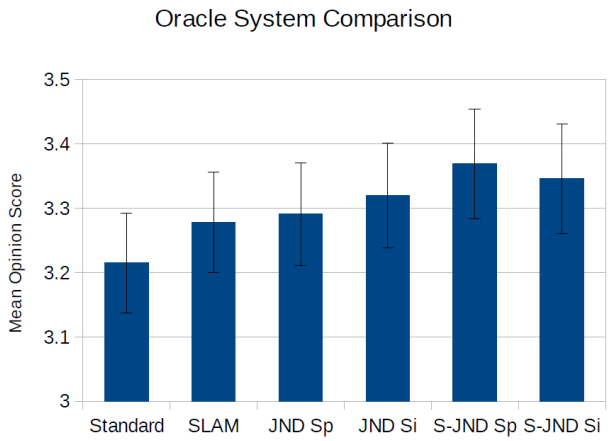

Figure 3: Mean Opinion Score results for Oracle synthesis of voices using SLAM. Standard = Standard HMM system, SLAM = Original SLAM, JND = JNDSLAM, S-JND = Simplified JNDSLAM, Sp = Label split as three elements, Si = Label as single element.

- Very High/Up: 3 JND's or more above the mean/of movement.
- High/Up: 1 to 3 JND above the mean/of movement.
- Medium/Straight: -1 to 1 JND around the mean/of movement.
- Low/Down: -1 to -3 JND below the mean/of movement.
- Very Low/Down: -3 or more JND below the mean/of movement.

Another benefit of removing the levels for the extremes is a large reduction of the label set from 401 to 176 labels, of these 169 are present in the data and 23 represent 95% of the data. While this is larger than the 10-20 suggested in [8] we consider this a good thing as it shows we've managed to unify many labels which have not been used much together and also get a label set with more meaningful labels.

3.1. Simplified JNDSLAM

While JNDSLAM provides labels better suited for TTS, initial experimentation with prediction (see Section 5 for more detail) revealed two potential issues. Firstly, no extremes were predicted. These are also rare in the data with only one extreme containing label in the 20 most frequent labels of JNDSLAM. Secondly few Very High/Low/Up/Down labels are predicted since those are fairly rare in the data. This suggests the task may be too complex and to alleviate this JNDSLAM was simplified. The simplification consists of not determining the position of any extreme and to only have positions and movements above and below 1 JND. Resulting in the following stylisation:

- The starting position of the pitch relative to the mean (High/Medium/Low).
- The direction of movement of the pitch relative to the start (Up/Straight/Down).
- Any extreme position (Positive/None/Negative).

While this simplification does remove granularity necessary for good analysis of pitch contours for phoneticians (if that is your interest we recommend the full JNDSLAM), it is likely to be beneficial to TTS as it simplifies the prediction task and introduces less variables for the classifier to predict. This is evident as it reduces the label set to just 28 labels all of which are present in the data. Furthermore, 19 labels represent 95% of the data, and 10 labels with extremes are in the top 20 as opposed to 1 for JNDSLAM.

4. Oracle Synthesis Evaluation

To evaluate the effects of using SLAM labels for voice training, and to see if any improvements are obtained from JNDSLAM and its Simplified version, HMM voices were trained. The labels were used as part of the linguistic context information, added at the syllable level and for each phone the label of its syllable and the preceding and following syllable was used. Thus their effect is on the decision tree context clustering stage of voice training. For the original SLAM algorithm the label was treated as one entity, however for both Simplified and standard JNDSLAM a second HMM voice was also trained. It is likely that some elements of the SLAM labels are more important for different clusterings and so splitting them may be beneficial, furthermore as sentences tend to have an overall falling pitch the starting position is likely to follow this pattern whereas the other elements may be more independent. It also has an effect on prediction of the features as they may be predicted separately which should be a simpler task (see Section 5). To do this the label was decomposed into its three elements (starting position, direction of movement and extreme value) and each of these were used as separate context features. In total six voices were trained, all on the same Female US English corpora described earlier. A standard unmodified HMM, one using the original SLAM, two (as one label or split) using JNDSLAM and two using Simplified JNDSLAM. 20 utterances were held out from voice training and used for the test. These were synthesised using each voice, and using each of the SLAM methods the sentences were labelled and these Gold standard labels were used in synthesis. This constitutes what we call Oracle synthesis and represent the upper bound for how well an automatic system can do using these labels. Using crowdsourcing each sentence from each system was evaluated 20 times in an anchored mean opinion score (MOS) test, for a total of 400 evaluations of each system.

4.1. Anchored MOS

An anchored MOS test is slightly different from a standard MOS test. In a standard test participants hear only samples from the systems used and the MOS scores become relative within those systems. To mitigate this, natural speech is often added as a topline system and to act as an anchor-point on the scale. In an anchored MOS test subjects are given samples of speech corresponding to each of the 5 levels of the scale. This make the results relative to the anchor samples and less relative between the systems. The benefit is that if we use the same anchor samples we can compare results from different tests, however, we may see less differences between systems unless these are also important relative to the anchors.

4.2. Results and Discussion

The results are summarised in Table 3. All SLAM based systems score higher than the Unmodified HMM system, but only the Simplified algorithm is significantly better. Both versions of JNDSLAM receives higher scores than the original SLAM and while these differences are not significant the tendency is clear and consistent. There is no difference between the split and single label HMM's suggesting that splitting the labels had no positive nor negative effect. It is notable that simplifying the label set improved MOS. The reason for this is quite likely

that with a reduced set each label provides more information to base clustering around. To see how far up the tree, and thus how important the labels are, we looked at where in the decision tree the contour related questions enter for the Simplified JND with each label separate. For the lf0 tree contour questions were present at the very top of each of the HMM states tree, accounting for 42% of the questions in the first 5 levels of the trees. Questions also appear in the 5 first levels of the duration, all of the mixed excitation and most of the mcep trees. Clearly the labels are useful in the build process. This also suggests that pitch is not entirely independent from the other streams, something which has also been noted in [26, 27], and it is thus important to model pitch well not only to get the correct pitch but also to improve other elements of the system.

5. Predicting JNDSLAM

That the labels yield an improvement over the unmodified HMM is encouraging. However, in the standard TTS situation we do not have access to the acoustics to derive the labels, these must be predicted. In an initial investigation into predicting the labels Long Short Term Memory Neural Networks (LSTM) [28] were used. These were chosen as it is a modern machine learning method, which also has the added benefit of being useful as an SPSS system [4] making potential later integration easier.

To predict JNDSLAM a standard linguistic feature set was used similar to that of HTS [9]. The hope being that, while these features are not sufficient for direct prediction of pitch movement as evidenced by the flat SPSS intonation, by using an intermediate representation which yields a good acoustic model and a simpler task (i.e. Simplified JNDSLAM), these features would be sufficient for acceptable prediction. In earlier studies using automatic ToBI prediction this approach has had a mixed success [6, 18]. These studies use much smaller amounts of labelled data and our increased dataset could mitigate this. Labels were predicted either in full form or as its three separate parts. The bidirectional LSTM consisted of four layers. A sparse input layer, a forwards LSTM layer, a backwards LSTM layer and an output layer. If the labels were predicted in full directly, the output layer was a sparse output layer, however if predicted in parts, a multidistribution output layer was used. This layer consisted of three sublayers, one for each element of the label. Four LSTM's were trained on 577k syllables with 32k for testing and 32k for tuning (5% each of the full corpora). We included one simple baseline of picking the most likely label for each method. Table 1 summarise the results. The overall prediction rates are above the baseline in all cases, showing that we do indeed capture some of the prosodic variation in the speech. The best overall performance is Simplified JNDSLAM predicted as one label with 25.2% correct.

5.1. Prediction Discussion

While the prediction accuracy is low, it is still possible that prediction yields reasonable contours despite this, as in [18]. However, when informally listening to output samples two tendencies arise. Directly predicting the labels yields more movement but also ask for a generally low starting point whereas predicting the labels individually yields a better starting point but less movement. Predicting more movement but in the wrong place is likely to annoy listeners and predicting less movement, i.e. flatter prosody, is already an issue in TTS. It is thus clear that current linguistic features are not useful for predicting pitch contour labels, and we therefore leave a synthesis evaluation of predicted labels as future work to be done after better prediction is obtained. The reason for why these features are not so useful is probably down to the fact that very few word level and above features are used, and those which are have little effect [6].

	Start	Direction	Extreme	Overall
JNDSLAM				
Baseline	32.7%	31.5%	69.8%	10.7%
Single	40.3%	38.8%	76.8%	18.1%
Split	39.9%	38.0%	75.4%	15.9%
Simplified				
Baseline	32.7%	37.7%	69.8%	13.3%
Single	46.7%	42.0%	76.0%	25.2%
Split	46.3%	46.5%	75.4%	22.2%

Table 1: Prediction accuracy of the LSTM's for each element of the label.

There are two strategies that we would like to investigate in the future. On the one hand, as JNDSLAM is an automatic method we are not limited to the use of data from one speaker, it is thus possible to obtain information from multiple speakers of the same accent to generate accent based pitch contour predictors. On the other hand, we believe a fruitful future approach would be to ensure prosody exists over simply no prosody as noted in [18]. For this purpose a lexicon-based approach could be useful. Just as words contain a set of phonemes with accent and contextual variation, they also have a standard prosody. That could be encoded and supplied to the HMM's models during synthesis time.

6. Discussion and Conclusion

We have presented a fully automatic pitch contour stylisation method suitable for speech synthesis. It builds upon the SLAM system of [8] by taking into account the JND of pitch difference, recasting end position as direction of movement and extreme positions as relative to its strength. We've called this JNDSLAM and have also presented a simplified version[1]. MOS results show that all versions of SLAM provide an improvement over the standard HMM system, particularly the Simplified version, however splitting the label had no effect. Furthermore, initial exploration of JNDSLAM prediction shows fairly low overall accuracy. That we cannot predict prosodic labels from text using the current linguistic features highlights the need for research into new potential features that are capable of this. Other work has noted the unimportance of current higher level features [6, 7, 29], and this work adds to a growing body of evidence that current linguistic features are insufficient. We have shown that JNDSLAM has the potential to impact synthesis quality if used correctly, and have proposed several possible solutions to this which involve both improvements to the automatic prediction methods, but also a potential offline encoding of "neutral" contours in the dictionary.

JNDSLAM has been presented as a means to improve standard TTS systems, however it also has other potential applications. In applications such ASR and Voice Cloning where the correct pitch values are available, these stylisations could provide evidence for a variety of prosodic events. Furthermore current TTS systems are unresponsive to requests for specific pitch contoursand JNDSLAM provides a means to request specific contours. This has potential uses in dialogue systems where we know what we wish to emphasise, whether we are asking a questions etc., phenomena which has specific prosodic realisations which can be supported by the TTS system.

[1] A liberally licensed C++ implementation is freely available at https://github.com/RasmusD/JNDSLAM

7. References

[1] M. Tachibana, J. Yamagishi, and T. Masuko, "Speech Synthesis with Various Emotional Expressions and Speaking Styles by Style Interpolation and Morphing," *IEICE Tansactions on Information and Systems*, vol. E88-D, no. 11, pp. 2484–2491, 2005.

[2] J. Yamagishi, K. Onishi, and T. Masuko, "Acoustic Modeling of Speaking Styles and Emotional Expressions in HMM-Based Speech Synthesis," *IEICE Transactions on Information and Systems*, vol. E88-D, no. 3, pp. 502–509, 2005.

[3] A. Gutkin, X. Gonzalvo, S. Breuer, and P. Taylor, "Quantized HMMs for Low Footprint Text-To-Speech Synthesis," in *Proc. Interspeech*, Makuhari, Chiba, Japan, 2010.

[4] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Proc. ICASSP*, Brisbane, Australia, 2015.

[5] Y. Agiomyrgiannakis, "Vocaine the Vocoder and Applications in Speech Synthesis," in *Proc. ICASSP*, Brisbane, Australia, 2015.

[6] O. Watts, J. Yamagishi, and S. King, "The role of higher-level linguistic features in HMM-based speech synthesis," in *Proc. Interspeech*, no. September, 2010, pp. 841–844.

[7] M. Cernak, P. Motlicek, and P. N. Garner, "On the (Un)Importance of the Contextual Factors in HMM-Based Speech Synthesis and Coding," in *Proc. ICASSP*, Vancouver, Canada, 2013, pp. 8140–8143.

[8] N. Obin, J. Beliao, C. Veaux, and A. Lacheret, "SLAM: Automatic Stylization and Labelling of Speech Melody," in *Proc. Speech Prosody*, Dublin, Ireland, 2014.

[9] H. Zen, K. Tokuda, T. Masuko, T. Kobayasih, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 5, pp. 825–834, 2007.

[10] P. Taylor, "Analysis and synthesis of intonation using the Tilt model." *The Journal of the Acoustical Society of America*, vol. 107, no. June 1999, pp. 1697–1714, 2000.

[11] C. D'Alessandro and P. Mertens, "Automatic pitch contour stylization using a model of tonal perception," *Computer Speech & Language*, vol. 9, pp. 257–288, 1995.

[12] G. P. Kochanski and C. Shih, "STEM-ML: Language independent prosody description," in *Proc. ICSLP*, Beijing, China, 2000.

[13] D. Hirst, A. D. Cristo, and R. Espesser, "Levels of representation and levels of analysis for the description of intonation systems." in *Prosody: Theory and Experiment*, M. Horne, Ed. Kluwer Academic Press, 2000.

[14] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, "TOBI: A Standard for Labelling English Prosody," in *Proc. ICSLP*, Banff, Canada, 1992, pp. 12–16.

[15] A. K. Syrdal, J. Hirschberg, J. McGory, and M. Beckman, "Automatic ToBI prediction and alignment to speed manual labeling of prosody," *Speech Communication*, vol. 33, pp. 135–151, 2001.

[16] A. Rosenberg, "Autobi a tool for automatic tobi annotation," in *Proc. Interspeech*, 2010.

[17] ——, "Automatic detection and classification of prosodic events," PhD Thesis, Columbia University, 2009.

[18] T. Boro, A. Stan, O. Watts, and S. D. Dumitrescu, "RSS-TOBI - A Prosodically Enhanced Romanian Speech Corpus," in *Proc. LREC*, Reykjavik, Iceland, 2014, pp. 316–320.

[19] C. Shih and G. Kochanski, "Chinese tone modeling with stem-ml," in *Proc. ICSLP*, 2000, pp. 67–70.

[20] P. Taylor, "The tilt intonation model," in *Proc. ICSLP 98*, 1998, pp. 1383–1386.

[21] A. Camacho, "SWIPE: A Sawtooth Waveform Inspired Pitch Estimator," Ph.D. dissertation, University of Florida, 2007.

[22] W. S. Cleveland, "LOWESS: A Program for Smoothing Scatterplots by Robust Locally Weighted Regression," *The American Statistician*, vol. 35, no. 1, p. 54, 1981.

[23] "REAPER: Google's open source pitch tracker," http://https://github.com/google/REAPER.

[24] A. C. M. Rietveld and C. Gussenhoven, "On the relation between pitch excursion size and prommence," *Journal of Phonetics*, vol. 13, pp. 299–308, 1985.

[25] M. Mehrabani, T. Mishra, and A. Conkie, "Unsupervised Prominence Prediction for Speech Synthesis," in *Proc. Interspeech*, Lyon, France, 2013, pp. 1559–1563.

[26] T. Merritt, T. Raitio, and S. King, "Investigating source and filter contributions, and their interaction, to statistical parametric speech synthesis," in *Proc. Interspeech*, Lyon, France, 2014.

[27] G. E. Henter, T. Merritt, M. Shannon, C. Mayo, and S. King, "Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech," in *Proc. Interspeech*, Lyon, France, 2014.

[28] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, Dublin, Ireland, 2014.

[29] M. P. Aylett, R. Dall, A. Ghoshal, G. E. Henter, and T. Merritt, "A Flexible Front-End for HTS," in *Proc. Interspeech*, Singapore, Singapore, 2014.