IDIAP RESEARCH REPORT

A NEURAL MODEL TO PREDICT PARAMETERS FOR A GENERALIZED COMMAND RESPONSE MODEL OF INTONATION

Bastian Schnell Philip N. Garner

Idiap-RR-10-2018

JULY 2018

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

A Neural Model to Predict Parameters for a Generalized Command Response Model of Intonation

Bastian Schnell, Philip N. Garner

July 9, 2018

Abstract

The Generalised Command Response (GCR) model is a time-local model of intonation that has been shown to lend itself to (cross-language) transfer of emphasis. In order to generalise the model to longer prosodic sequences, we show that it can be driven by a recurrent neural network emulating a spiking neural network. We show that a loss function for error backpropagation can be formulated analogously to that of the Spike Pattern Association Neuron (SPAN) method for spiking networks. The resulting system is able to generate prosody comparable to a state-of-the-art deep neural network implementation, but potentially retaining the transfer capabilities of the GCR model.

Index Terms: Speech synthesis, prosody modelling, recurrent neural network, Fujisaki model

1 Introduction

We are interested in general in speech to speech translation, and specifically in transfer of paralinguistics from one language to another. For instance, if a speaker expresses emotion or emphasis in an input language, we would like those features to be present in the synthetic speech resulting from machine translation of speech recognition output. In previous work with colleagues [1], we studied a model of prosody (actually intonation, F_0) based on the Command-Response (CR) model of Fujisaki [2]. By contrast to the CR model, this *Generalised* CR (GCR) model can be extracted easily from an intonation contour using a matching pursuit algorithm [3]. The time-local nature of its constituent *atoms* was shown (by design) to lend itself to transfer of emphasis. In particular, sections of intonation contours can be replaced with others that carry different meaning, all whilst retaining naturalness.

In this work, we report on an investigation into how to use GCR to generate longer intonation contours for more general contour models. Of course, such contours can be generated by any modern Text-to-Speech (TTS) system (we use Merlin [4] to train a baseline system). However, we hope to retain the transfer capability of the GCR. GCR also enables analysis of the underlying physiological process.

Given that GCR atoms approximate (groups of) muscle responses to neural spikes, it would make sense to use a Spiking Neural Network (SNN) to generate these atoms. The generated spikes would be filtered by muscle responses to generate the pitch contour. However, the choice of a spiking network paradigm is not obvious. Rather, given the authors' familiarity with conventional back-propagation based deep learning algorithms and toolkits, we emulate a SNN. In this work we use a bidirectional Recurrent Neural Network (RNN) which is capable of generating spikes, hence atoms, for a given text. This in turn allows us to introduce a loss function for the training of spiking outputs which is inspired by losses in SNNs. Furthermore, we explain how to weight the loss of different frames with respect to spike positions and Voiced/Unvoiced (V/UV) decision to achieve good generalisation.We test the hypothesis that prosody generated by our neural model sounds natural, even though it might vary from the ground truth and that generated by a baseline model.

In the following sections, we give a brief overview of the GCR model in the context of other work. We go on to show that a bidirectional GRU based RNN can simulate the spikes that might be expected to come from a biological spiking network. We generalise the Spike Pattern Association Neuron (SPAN) algorithm [5] from the literature to construct a loss function from GCR atoms, and show that it can be backpropagated to allow the system to generate natural prosody. We compare our model in terms of objective and subjective measures with a strong baseline system (bidirectional RNN + post-processing) which is trained to predict LF_0 per frame.

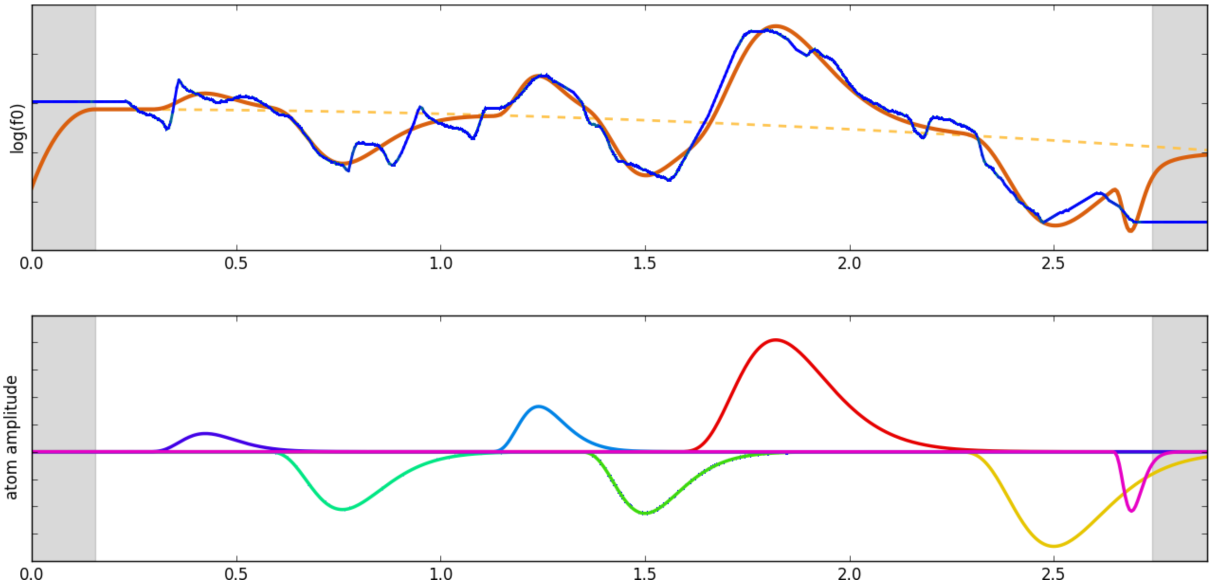

Figure 1 – Atom decomposition of Log-F_0 (LF_0) contour. Upper plot: Original LF_0 contour (blue, solid), reconstruction from atoms (orange, solid), phrase component (yellow, dashed). Lower plot: Atom impulse responses with one colour per atom.

2 Relation to prior work

Numerous approaches to modelling prosody by the superposition of multiple F_0 contours exist. The Tilt model [6] describes the pitch contour as a sequence of events with specific shapes that can be automatically extracted. The INSINT (INternational Transcription System for INTonation) model [7] allows automatic parameter extraction of the Tone and Break Indices (ToBI) model [8] that divides prosody into multiple tiers of linguistic focus. The General Superposition Model of Intonation [9] models the pitch contour through a decomposition of microprosodic segmental perturbations, an accent and a phrase curve. The Superposition of Functional Contours (SFC) model [10] is a data driven approach that models the pitch contour by a superposition of intonation prototypes. A common drawback of all models above is that none of them is based on observations of the physiological production aspect. The proposed GCR model is a physiologically based intonation model which has the same representative power as the CR model of Fujisaki [2]. It generates the LF_0 contour by a superposition of impulse responses to critically damped second order systems modelling muscle responses (Figure 1). The impulse response of a critically damped second order system is a gamma kernel

$$G_{k,\theta}(t) = \frac{1}{\theta^k \Gamma(k)} t^{k-1} e^{-t/\theta} \quad \text{for} \quad t \geqslant 0 \tag{1}$$

with k being the system order, Γ being the gamma function, and θ determining the length of the kernel. For a critically damped second-order system as well as the CR model $k = 2$, however previous research [11] has found that $k = 6$ gives better approximations of the original LF_0 contour. A phrase additionally consists of a phrase atom (phrase command in CR model) which models the general shape of the contour and is correlated mainly to the physics of the speakers' lung volume (dotted line in upper plot of Figure 1). Experiments have shown that the proposed model is capable of producing good representations and can transplant emphasis from one language to another [12].

The work closest to our approach is that of Hojo et. al. [13] where the CR model is represented by a constrained HMM, and a Neural Network (NN) predicts the posteriori probability of its states. A Viterbi-like algorithm extracts the most probable sequence based on the posteriors. The LF_0 generation based on the sequence is then straight forward and has been done before [14].

3 Atom Prediction

3.1 Spiking Neural Network

Rather than use an explicit spiking paradigm such as leaky integrate and fire (LIF), we instead emulate such a network using a conventional backpropagation network. This is achieved using a bidirectional RNN as described in [15]. Rather than use LSTMs with peepholes as in that paper, we use the GRU of [16] where peepholes are moot. The error is defined using the learning rule described in the following section.

3.2 Atom Loss

For a regression task which targets spiking output of varying amplitudes the commonly used Mean-Squared-Error (MSE) is not an appropriate loss function as it does not consider any temporal information of spikes. The problem breaks down to measuring the distance between two spike trains. Various methods exist to compute such a distance such as the Victor-Purpura metric [17], the Van Rossum Similarity Measure [18], the Schreiber *et al.* Similarity Measure [19], the Hunter-Milton Similarity Measure [20], Event Synchronization [21], Stochastic Event Synchrony (SES) [22], and the modulus- and max-metrics [23]. In general we are interested in a learning rule that uses such temporally-aware measurements to compute losses during training. We have not found a suitable learning rule in the literature for feed-forward NN or RNNs but instead in the field of SNNs. The closest precedent to the learning rule we propose is the SPAN method [5]. In SPAN, each spike is convolved with an "alpha" kernel which adds temporal information of the spike to all surrounding/succeeding frames. On the resulting continuous output MSE can be used as the learning rule. The authors of the SPAN method state that other kernel functions are possible as Gaussian, linear and exponential kernels [24] . The choice of kernel in the literature is driven by the supposed shape of the post-synaptic potential of neurons in the human brain. However, the spikes we are interested in represent muscle impulses with responses modelled by a gamma kernel as described in 2. We therefore use the gamma kernel as the kernel function. The length θ of the kernel is by no means obvious. While the desired length of correctly placed spikes is known, no ground truth is available for incorrectly placed spikes. We found that a single short kernel with $\theta = 0.01$ for all convolutions adds the required temporal information to each spike.

Let us define the matrix G which has the coefficients of the gamma kernel on its leading and above leading diagonals with size $(T \times T)$ where T is the number of frames in a training sample. Further define y_o as the output of the NN and y_d as the desired output each of size $(T \times 1)$. All spikes can be convolved independently from each other with the kernel function by $\mathrm{diag}(y) \cdot G = Y$ (compare Figure 2). We denote $\tilde{y}_d$ as the desired enveloped output given by the sum of all rows of Y_d which corresponds to a superposition of envelopes. The error at each time step t is computed by

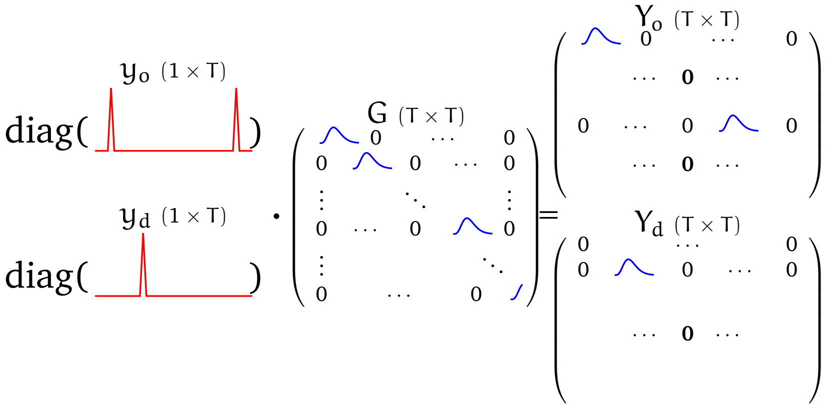

Figure 2 – Frame-wise convolution of NN output y_o and desired output y_d.

$$\mathrm{err}_t = \sum_{i=t}^{t+\Delta t} (Y_{o,t,i} - \tilde{y}_{d,i})^2 \tag{2}$$

with $Y_{o,t,i}$ being the t-th row and i-th column of Y_o, and $\tilde{y}_{d,i}$ being the i-th entry in $\tilde{y}_d$. Δt is given by the length of the gamma kernel used to convolve each spike and represents the number of frames where a spike takes effect. To limit the interval of the sum to $[t, t + \Delta t]$ is critical so that the error is not affected by succeeding parts of the sequence where the spike cannot take effect. To compute the sum efficiently we define the matrix S of size $(T \times T)$ which is the same matrix as G but with ones at non-zero entries of G and zero otherwise. By utilizing the Hadamard product $E = S \otimes Y_E$, with $Y_E = \text{square}[Y_o - \mathbf{1}\tilde{y}_d]$ and square[ ] being the element-wise square operation, entries outside the $[t, t + \Delta t]$ interval are zeroed. The error at time step t is given by the squared norm of the t-th row of E.

$$\text{err}_t = \|E_t\|_2^2 \tag{3}$$

Note that the error is computed frame-wise without the superposition of the enveloped NN output, which means that neighbouring spikes cannot interfere. When allowing the interference of spikes two problems arise:

- The NN learns to represent a single target spike by multiple smaller spikes.
- The NN predicts many spikes with opposite amplitude which cancel out.

The former problem is an acceptable variation to our model when assuming that a muscle response is not triggered by a single nerve impulse but multiple ones. However, the latter problem gives clearly unintended and physiological implausible behaviour. Therefore we use the frame-wise atom loss hereafter. A NN trained with the above learning rule gives an activation around each spike position and thus requires a post-processing step to identify the peaks.[1]

3.3 Amplitude Prediction

For any atom the prediction of position, amplitude and length θ is required. A single position flag trained with atom loss introduced above in section 3.2 gives good estimates of the position of spikes ($y_{d,t} \in \{-1, 0, 1\}$) but cannot predict amplitude and length at the same time. Unfortunately, we were not able to train a NN to predict a θ value directly. Instead, besides the position flag, the NN is trained with MSE to predict one amplitude per θ for a fixed set of θs. The set of θs needs enough values to allow an approximation of the target LF_0 contour with low error, but is limited which corresponds to the limited number of articulators in the human larynx. When training on amplitude spikes the problem of a highly unbalanced training set arises (>99.8% of all frames are zero). A network trained with MSE will therefore uniformly predict zeros and achieve a >99.8% accuracy. The problem can be solved by small adaptations to data and loss. First each amplitude spike is convolved by a normal distribution in time with a window of 51 frames. Secondly the loss of frames which are non-zero in the desired output are increased while all others are decreased resulting in a Weighted Mean-Squared-Error (WMSE).

3.4 Voiced/Unvoiced Prediction

The network also predicts a flag for V/UV LF_0 where values >0.5 are mapped to voiced frames. The target V/UV flag is used to decrease the weight of both losses (atoms and amplitudes) by 0.5 on unvoiced frames. The value of 0.5 was confirmed by a heuristic search. By this the network spends less effort on improving parts which are silent after synthesis.

4 Experiments

In running experiments, we mean to test the hypothesis that the basic procedure described above is a plausible approach to generate natural sounding intonation. The system is preliminary. A-priori we do not expect it to generate state-of-the-art intonation contours; rather, we simply aim to validate that the approach merits further research.

[1] We use the `scipy.signal.find_peaks_cwt` function.

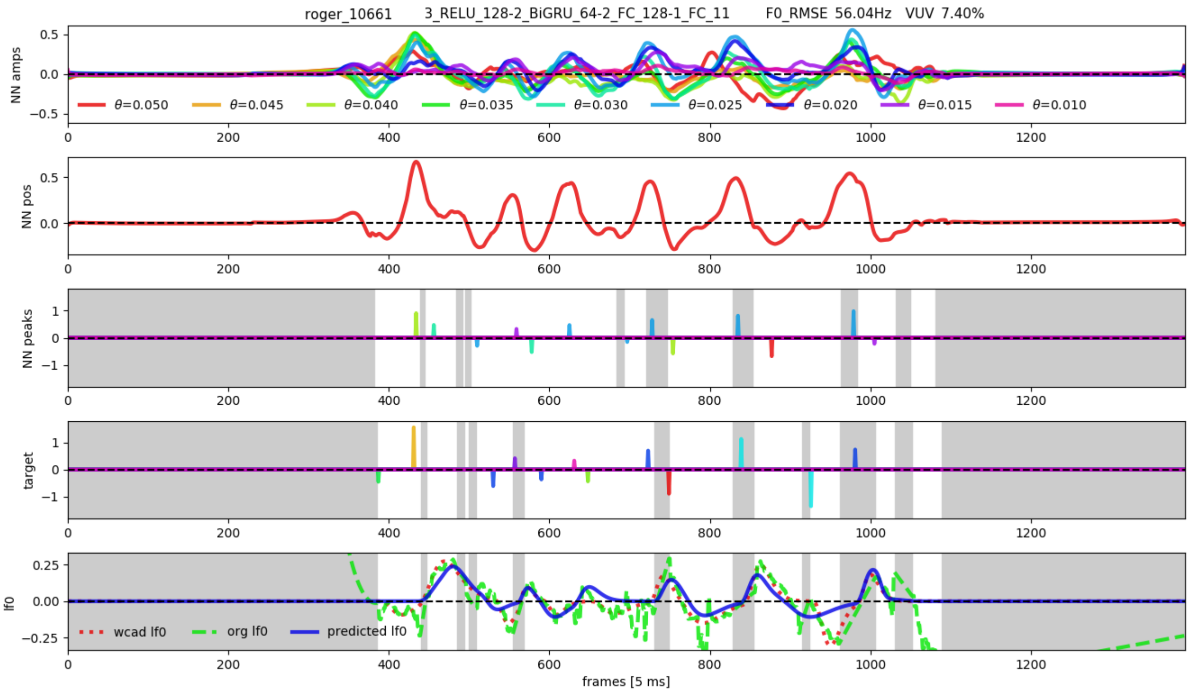

Figure 3 – Synthetic features on a temporal scale of 5 ms per frame. Plot descriptions from top to bottom: **1:** Nine amplitude outputs (one per θ). **2:** Spike position flag before post-processing. **3:** Atom spikes generated from spike position and amplitude max/min, V/UV flag (unvoiced frames grey). **4:** Target atom spikes and target V/UV flag (unvoiced frames grey). **5:** LF$_0$ (without phrase component) NN reconstruction (blue, solid), target reconstruction (red, dotted), original (green, dashed) and target V/UV flag (unvoiced frames grey).

4.1 Experimental Setup

We test our proposed model on the speech database released for the 2008 Blizzard Challenge [25] on a subset (carroll, arctic, theherald 1,2,3) of the native English Voice A (Roger) of about 6.5 hours on a 16 kHz sampling rate. We only use those samples which can be represented by a single phrase atom. 5% of all samples are set aside for testing which corresponds to approximately 20 minutes.

Festival [26] is used to obtain phone sequences from text which are force-aligned by context-independent HMMs with the help of Merlin [4]. Merlin is used again to characterise phones with 416 text-derived binary and numerical features such as quin-phone identities, part-of-speech, positional information relating to syllables, words, and phrases, which are normalised to [0.01, 0.99]. These questions are used as input for all systems.

The WORLD vocoder [27] (D4C edition [28]) is used to extract 60-dimensional Mel Frequency Cepstral Coefficients (MFCC), one band aperiodicity (BAP), and fundamental frequency (F$_0$) on log scale at 5-ms frame step. Dynamic features are also computed but are only used in the baseline system. LF$_0$ is interpolated before training and a binary V/UV flag is used to capture voicing information. The acoustic features are mean-variance normalised.

From the extracted LF$_0$ atoms are computed by matching pursuit as proposed in [1] including a single phrase atom. Atom amplitudes are mean-variance normalised. The length of an atom is limited to nine discrete values $\theta \in \{0.01, 0.015, 0.02, \ldots, 0.05\}$ which were found to be able to model the LF$_0$ contour with low error in previous research [11].

4.2 Network Topologies

The baseline system is similar to the one used in [29] following the usual approach by predicting acoustic features plus their dynamic components. It consists of two feed-forward RELU layers of 1024 nodes, three bi-directional GRUs with 512 nodes each, and a final linear output layer with 187 nodes. The model is trained with Adam [30] on 35 epochs (learning rate 0.002).

The model we propose consists of three feed-forward RELU layers with 128 nodes, two bi-directional

GRUs with 64 nodes each, two feed-forward RELU layers with 128 nodes, and a final linear output layer with 11 nodes. It predicts one V/UV flag, nine amplitudes (one per θ), and a spike position flag. The model is trained with Adam on 55 epochs (learning rate 0.0002). In both cases we use $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$ for Adam.

4.3 Synthesis

For all tests the original durations, MFCCs, and BAPs are used as we are only interested in the impact of different LF_0s on naturalness. For the baseline system LF_0 is improved by maximum likelihood parameter generation (MLPG) [31] using variances computed from the training data. The waveform is synthesized by the WORLD vocoder.

In our proposed model, the spike position flag is post-processed to identify its peaks which results in a value of $\{-1, 0, 1\}$ per frame. Atoms are constructed by taking the maximum of the nine predicted amplitudes for positive spikes and the minimum for negative spikes respectively. The θ value is implicitly given by the index of the selected amplitude within the nine outputs. LF_0 is reconstructed by superposition of all predicted atoms and the original phrase atom (Figure 3). We plan to predict the phrase atom as well in the future.

4.4 Objective Results

To objectively compare the models we compute the Root-Mean-Squared-Error (RMSE) of F_0 on all frames which are voiced either in the target data or in the network prediction, and the V/UV error rate. Our model preforms slightly worse than the baseline system (compare Table 1) but certainly close enough to validate our hypothesis that the approach is plausible.

Model	F0 RMSE	V/UV
baseline	44.46 Hz	5.43 %
atom	49.89 Hz	5.94 %

Table 1 – Objective results.

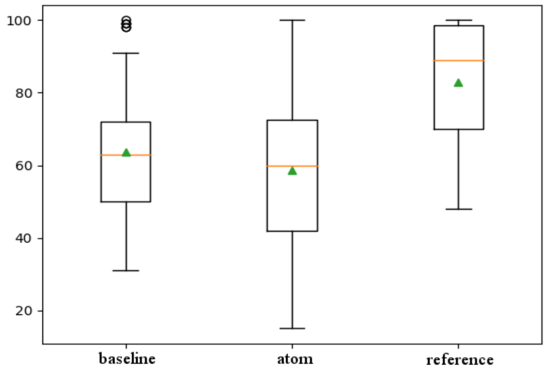

Figure 4 – Subjective score of MUSHRA intonation test. Medians as orange lines. Sample averages as green triangles. Outliers as circles.

4.5 Subjective Results

We measured the naturalness of the synthesised speech by a MUSHRA test[2] (Figure 4) where we compared our model (*atom*) and the baseline (*baseline*) with the speech produced by the vocoder with the original acoustic features (*reference*). We randomly selected a subset of 20 samples from the test set excluding those where the speaker takes a breath half way through as those samples require further phrase atoms. 17 non-native but fluent English speakers participated in the test. Each of them was asked to listen to 5 randomly selected samples from that subset and rate them on a scale from 0 to 100. They were told to focus on prosody only and ignore minor fuzzy/buzzy artefacts. As the most natural prosody is found in the reference sample, we excluded 18 results where the listener rated the baseline or the atom system more than 10 points higher than that reference. A two-tailed paired t-test on the individual ratings for the baseline and atom system gives a p-value of $p = 0.12 > 0.05$ supporting our assumption that the two system are not significantly different on a difference level of 0.05. The two-tailed paired t-tests show that both systems are significantly different to the reference. The p-value for baseline – reference is $p = 1.39e{-}9$, and for atom – reference: $p = 2.54e{-}12$.

[2] Listening test are designed with the BeaqleJS toolkit [32].

5 Conclusions

We have shown that the combination of an emulated spiking network, a dictionary of atoms representing muscle responses, and a SPAN-inspired training algorithm can generate reasonable intonation contours. Although "reasonable" is open to interpretation, the algorithm produces subjective results that are not significantly different from an accepted baseline. The proposed training algorithm for spiking targets enables the use of DNNs in other research fields currently dominated by SNNs. Future work includes the prediction of phrase atoms, exploiting the capabilities of the GCR model to produce / transfer affect, and reduce the number of heuristics identifying hyper-parameters.

6 Acknowledgements

This research was supported by SNSF Project number 165545, MASS: Multilingual Affective Speech Synthesis, http://p3.snf.ch/Project-165545.

References

[1] P.-E. Honnet, B. Gerazov, and P. N. Garner, "Atom decomposition-based intonation modelling," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4744–4748.

[2] H. Fujisaki, S. Ohno, and C. Wang, "A command-response model for F0 contour generation in multilingual speech synthesis," in *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*, 1998. [Online]. Available: http://www.isca-speech.org/archive_open/archive_papers/ssw3/ssw3_299.pdf

[3] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[4] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system," *Proc. SSW, Sunnyvale, USA*, 2016.

[5] A. Mohemmed, S. Schliebs, S. Matsuda, and N. Kasabov, "Training spiking neural networks to associate spatio-temporal input–output spike patterns," *Neurocomputing*, vol. 107, pp. 3–10, 2013.

[6] P. Taylor, "Analysis and synthesis of intonation using the tilt model," *The Journal of the acoustical society of America*, vol. 107, no. 3, pp. 1697–1714, 2000.

[7] D. Hirst, A. Di Cristo, and R. Espesser, "Levels of representation and levels of analysis for the description of intonation systems," in *Prosody: Theory and experiment*. Springer, 2000, pp. 51–87.

[8] 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*, 1992. [Online]. Available: http://www.isca-speech.org/archive/icslp_1992/i92_0867.html

[9] J. P. Van Santen and B. Möbius, "A quantitative model of F0 generation and alignment," in *Intonation*. Springer, 2000, pp. 269–288.

[10] G. Bailly and B. Holm, "SFC: a trainable prosodic model," *Speech communication*, vol. 46, no. 3-4, pp. 348–364, 2005.

[11] B. Gerazov, P.-E. Honnet, A. Gjoreski, and P. N. Garner, "Weighted correlation based atom decomposition intonation modelling," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015. [Online]. Available: http://publications.idiap.ch/index.php/publications/show/3145

[12] P.-E. Honnet and P. N. Garner, "Emphasis recreation for TTS using intonation atoms," in *9th ISCA Speech Synthesis Workshop*, no. EPFL-CONF-220902, 2016.

[13] N. Hojo, Y. Ohsugi, Y. Ijima, and H. Kameoka, “DNN-SPACE: DNN-HMM-based generative model of voice f0 contours for statistical phrase/accent command estimation,” *Proc. Interspeech 2017*, pp. 1074–1078, 2017.

[14] H. Kameoka, K. Yoshizato, T. Ishihara, K. Kadowaki, Y. Ohishi, and K. Kashino, “Generative modeling of voice fundamental frequency contours,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 6, pp. 1042–1053, 2015.

[15] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, “Learning precise timing with lstm recurrent networks,” *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[16] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.

[17] J. D. Victor and K. P. Purpura, “Metric-space analysis of spike trains: theory, algorithms and application,” *Network: computation in neural systems*, vol. 8, no. 2, pp. 127–164, 1997.

[18] M. v. Rossum, “A novel spike distance,” *Neural computation*, vol. 13, no. 4, pp. 751–763, 2001.

[19] S. Schreiber, J.-M. Fellous, D. Whitmer, P. Tiesinga, and T. J. Sejnowski, “A new correlation-based measure of spike timing reliability,” *Neurocomputing*, vol. 52, pp. 925–931, 2003.

[20] J. D. Hunter and J. G. Milton, “Amplitude and frequency dependence of spike timing: implications for dynamic regulation,” *Journal of neurophysiology*, vol. 90, no. 1, pp. 387–394, 2003.

[21] R. Q. Quiroga, T. Kreuz, and P. Grassberger, “Event synchronization: a simple and fast method to measure synchronicity and time delay patterns,” *Physical review E*, vol. 66, no. 4, p. 041904, 2002.

[22] J. Dauwels, F. Vialatte, T. Rutkowski, and A. S. Cichocki, “Measuring neural synchrony by message passing,” in *Advances in neural information processing systems*, 2008, pp. 361–368. [Online]. Available: http://papers.nips.cc/paper/3322-measuring-neural-synchrony-by-message-passing.pdf

[23] C. V. Rusu and R. V. Florian, “A new class of metrics for spike trains,” *Neural Computation*, vol. 26, no. 2, pp. 306–348, 2014.

[24] F. Ponulak and A. Kasiński, “Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting,” *Neural computation*, vol. 22, no. 2, pp. 467–510, 2010.

[25] V. Karaiskos, S. King, R. A. Clark, and C. Mayo, “The blizzard challenge 2008,” in *Proc. Blizzard Challenge Workshop, Brisbane, Australia*, 2008. [Online]. Available: http://www.festvox.org/blizzard/bc2008/summary_Blizzard2008.pdf

[26] A. Black, P. Taylor, R. Caley, and R. Clark, “The festival speech synthesis system,” 1998. [Online]. Available: http://www.cstr.ed.ac.uk/projects/festival/

[27] M. Morise, F. Yokomori, and K. Ozawa, “WORLD: a vocoder-based high-quality speech synthesis system for real-time applications,” *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[28] M. Morise, “D4C, a band-aperiodicity estimator for high-quality speech synthesis,” *Speech Communication*, vol. 84, pp. 57–65, 2016.

[29] S. Ronanki, O. Watts, and S. King, “A hierarchical encoder-decoder model for statistical parametric speech synthesis,” *Proc. Interspeech 2017*, pp. 1133–1137, 2017.

[30] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[31] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, “Speech parameter generation algorithms for hmm-based speech synthesis,” in *Acoustics, Speech, and Signal Processing, 2000. ICASSP’00. Proceedings. 2000 IEEE International Conference on*, vol. 3. IEEE, 2000, pp. 1315–1318.

[32] S. Kraft and U. Zölzer, “BeaqleJS: HTML5 and javascript based framework for the subjective evaluation of audio quality,” in *Linux Audio Conference, Karlsruhe, DE*, 2014. [Online]. Available: https://github.com/HSU-ANT/beaqlejs