INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Unsupervised Training of a DNN-based Formant Tracker

Jason Lilley, H. Timothy Bunnell

Nemours Biomedical Research, Wilmington, DE, USA
jason.lilley@nemours.org, tim.bunnell@nemours.org

Abstract

Phonetic analysis often requires reliable estimation of formants, but estimates provided by popular programs can be unreliable. Recently, Dissen et al. [1] described DNN-based formant trackers that produced more accurate frequency estimates than several others, but require manually-corrected formant data for training. Here we describe a novel unsupervised training method for corpus-based DNN formant parameter estimation and tracking with accuracy similar to [1]. Frame-wise spectral envelopes serve as the input. The output is estimates of the frequencies and bandwidths plus amplitude adjustments for a prespecified number of poles and zeros, hereafter referred to as "formant parameters." A custom loss measure based on the difference between the input envelope and one generated from the estimated formant parameters is calculated and back-propagated through the network to establish the gradients with respect to the formant parameters. The approach is similar to that of autoencoders, in that the model is trained to reproduce its input in order to discover latent features, in this case, the formant parameters. Our results demonstrate that a reliable formant tracker can be constructed for a speech corpus without the need for hand-corrected training data.

Index Terms: speech analysis, formant estimation, formant tracking, deep learning, acoustic models of speech

1. Introduction

Formant estimation is a crucial component in acoustic phonetic studies of human speech, but rarely used in other areas of speech science such as speech recognition or modern speech synthesis systems. In large measure this may be due to the difficulty of reliably estimating formant frequencies within a single windowed frame of natural speech. Moreover, tracking continuous trajectories of formants as they vary over time in natural utterances is notably error-prone, typically requiring manual review and correction when high accuracy is required. There are, consequently, few large datasets of precisely tracked formants (and those that exist typically include only the first 2-4 formants). One such dataset is VTR-TIMIT [2], a 516-utterance subset of TIMIT [3] whose first three formant frequencies were first derived automatically and then hand-corrected. Although other datasets exist, e.g. [4, 5], most are smaller and contain only measurements of isolated words or single vowels.

For these reasons, the development of automatic formant measurement algorithms is still an ongoing topic of research [1, 6-12]. [13] compared the automatically estimated formant tracks from three commonly used formant trackers [14-16] against the hand-corrected VTR-TIMIT trajectories for F1 – F3, and found they can be off by hundreds of Hertz on average if default parameters are used. Parameter tuning is possible for many of these and other trackers, but speaker-specific tuning may be impractical for large datasets.

Most formant tracking algorithms begin with independently obtained formant frequency estimates for every analysis frame within an utterance. The individual frame-by-frame estimates are then collected into consistent formant trajectories over time using techniques such as dynamic programming or statistical clustering methods. Most work focuses on improving the formant estimation step [7-9, 11], the trajectory formation step [6, 17], or both [10, 18]. For example, the MSR algorithm of [18] (used for the initial estimation of VTR-TIMIT) involves iterative Kalman filtering of LPC cepstra. Likewise, the "KARMA" algorithm [17] uses Kalman inference on autoregressive moving average (ARMA) cepstral coefficients. Recently [7, 9] compared several LPC-based algorithms and demonstrated a new approach, called quasi-closed phase forward-backward covariance-based (QCP-FBCOV) linear prediction analysis, for which they showed lower error rates on VTR-TIMIT and synthetic data.

More recently, Dissen et al. [1] approached the task using supervised machine learning. They trained two neural network models—one an LSTM-based RNN model they call "DeepFormants" (henceforth called DF-RNN) and one a combined RNN-CNN model (henceforth DF-RCNN)—on the 384-utterance training subset of VTR-TIMIT to predict the first 3 formant frequencies, and tested on the 192-utterance test subset. The input features of DF-RNN were 350 cepstral coefficients of various types, while those of DF-RCNN were 55*50 raw spectrograms. Their results showed lower error rates on the VTR-TIMIT test set than other trackers [14, 17, 19], and at least on par with MSR [18]. Likewise, [13] included DF-RNN in their study, and found it generally performed better than the three LPC-based automatic trackers [14-16]. However, the models trained on VTR-TIMIT had higher error rates on two other datasets [4, 5]. [1] present a domain adaptation method that successfully reduces the error rates down to the VTR-TIMIT level, but the adaptation method requires formant measures from the new datasets. Hence, using their approach on other datasets, either from scratch or via adaptation, requires some amount of hand-labeled data.

In the following, we present a corpus-based method of training a neural network to predict formant frequencies—as well as bandwidths and amplitudes—that requires no prior formant measurements as training data. Instead, spectral envelope estimates for each analysis frame are fed to a network that predicts vocal tract resonance features (pole and zero parameters) as output. Since the correct output features are not known in advance, the output feature predictions are used to generate a new estimate of the input spectral envelope from which a loss can be calculated for back propagation. This approach is similar in concept to an autoencoder [20, 21], except that in this case the "latent variables" are interpretable as vocal tract resonance features, and the "decoder" is a pre-

http://dx.doi.org/10.21437/Interspeech.2021-1690

determined algorithmic reconstruction of the spectral envelope from the latent feature values. Without the need for prior measurements, large training datasets can be used, and the trained network can be used to generate formant trajectories on unseen speech data. We demonstrate this method, which we call **FormantNet**, on VTR-TIMIT, showing error rates on par with [1] and lower than other published methods on this dataset.

2. Approach

The FormantNet approach owes much of its theoretical motivation to early work on an analysis-by-synthesis approach described in [22]. That work used an iterative Newton-Raphson technique to solve simultaneous equations relating formant frequencies and bandwidths to spectral shape by minimizing the mean-squared error between a cepstral-smoothed input speech spectrum and the spectrum predicted by the formant parameters. For tractable computation on computers of that era, only three formants were predicted, with fixed bandwidths, and using 64-sample spectra in the range 0 – 5 kHz. Given advances in computer technology and machine learning over the past 50 years, we can implement a related approach to identifying and tracking formant parameters using deep learning.

As mentioned, although not an autoencoder, our approach resembles autoencoding strategies in which a bottleneck layer is used to force discovery of latent features that greatly reduce the dimensionality of the input feature set. However, rather than allow the network to discover the latent features, we constrain the latent features to be interpreted as formant parameters via a special loss function. The mapping from formant parameters to spectral features used in our model, due to [23], is given in (1), which describes the spectrum level h at frequency f for the impulse response of a single formant with resonant frequency F, bandwidth B, and amplitude weighting factor A.

$$h(f) = \frac{A \times (F^2 + {B^2}/{4})}{\sqrt{((f-F)^2 + {B^2}/{4}) \times ((f+F)^2 + {B^2}/{4})}} \quad (1)$$

To create a comparison spectrum, (1) is evaluated for each formant and for each f in the discrete frequency spectrum. In the parallel formant model that we are using [24], the amplitude-weighted contributions of individual linear formant spectra are added to form the composite spectrum. When zeros are also included in the model to be trained, the final spectrum is the summed formant spectra divided by the summed zero spectra. The loss function used in training our models is the mean squared difference between this estimated spectrum and the input spectrum (both converted to decibels).

One restriction of the approach we have used is that the number of formants and zeros to be learned for an entire corpus is fixed by the network design. This should be based on the average number of resonances expected for the corpus population in the input bandwidth. Our experiments found that networks with six poles and one zero tend to converge best over the speakers and sampling rate of VTR-TIMIT. With fewer formants, the full spectrum was not adequately modeled; with more, the network often tried to squeeze the extra formants where they were unneeded, such as between F2 and F3.

3. Methods

We outline our methods and procedures here; upon publication, full details and code will be made publicly available at https://github.com/NemoursResearch/FormantNet.

3.1. Materials

We developed our models with the training portion of TIMIT. Note that unlike [1], we used the entire TIMIT training set, not just the 324 files of the VTR-TIMIT training set. In fact, we made no use of the formant measurements of the VTR-TIMIT corpus, except in evaluation. To test model convergence, we selected 48 speakers (3 men and 3 women from each of the 8 dialect regions; 480 utterances total) to be used as validation data; the remaining 4140 utterances were used for training.

3.2. Input features

The RMS amplitudes of the input waveforms were normalized to an overall 68 dB for the utterance before feature extraction. Then we used Iterative Adaptive Inverse Filtering (IAIF) [25] to remove the estimated contribution of the source signal from the speech signal. The remaining signal was then converted to Hann-windowed frames of 32 msec in length every 5 msec. The Discrete Fourier Transform was taken to convert the signals to 257-point spectra. From these we approximate spectral envelopes by iteratively replacing narrow scale minima and applying a 3-point smoothing function to the raw harmonic spectra. This approach tends to trace a smooth curve over harmonic peaks while also retaining spectral valleys possibly associated with zeros. After a floor of 0.001 was added, the linear envelopes were converted to the decibel scale. These envelopes serve as both input and target for the model, except that the input spectra are normalized by subtracting the mean and dividing by standard deviation of the training set; the target spectra used in computing the loss are not.

To give temporal context to the CNN models described below, the input to the model at each time step consisted of a window of 21 frames: the target frame whose formants are to be estimated, as well as the 10 preceding and 10 following frames (the initial or final frame was duplicated as needed to fill out this structure). For the LSTM models, which model temporal sequences directly, only the target frame was provided per time step, but we also experimented with bidirectional LSTM (BLSTM) models, to see if performances would improve with access to following frames as well as preceding ones. For both, the training sequence length was set to 64 time steps.

3.3. DNN models

We experimented with several different model architectures, including convolutional neural network (CNN) and recurrent neural network (RNN) architectures. All models were implemented in TensorFlow 2.3 [26].

The CNN models each had 3 convolutional layers of 16, 32, and 64 units. Each convolutional layer was followed by a max-pooling layer. Both convolutional and pooling layers were two-dimensional, operating over both the time and frequency axes of the input windows. The final pooling layer was followed by two hidden dense layers of 1024 and 256 units, respectively, followed finally by the output layer. The activation of all intermediate layers was ReLU. The RNN models consisted of either one or three LSTM layers of 512 units apiece, followed immediately by the output layer. For BLSTMs, the outputs of the forward and backward passes were concatenated before being passed to the output layer.

The output layer of all models produced 3 values (frequency, bandwidth, and amplitude) per pole and 2 values (frequency and bandwidth) per zero (whose amplitude factors

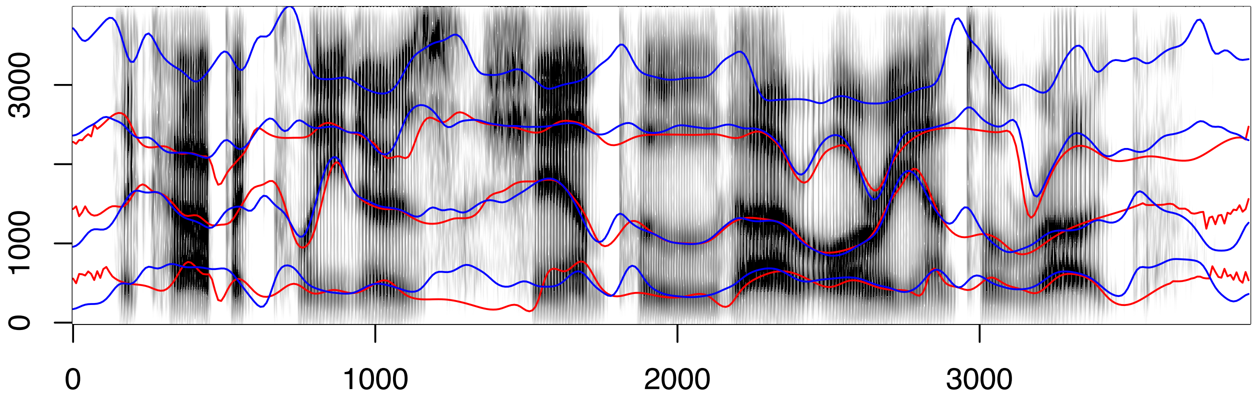

Figure 1: *Spectrogram (0-4 KHz) of the utterance "The carpet cleaners shampooed our oriental rug." Hand-corrected formant tracks F1-F3 in red; model-predicted tracks F1-F4 in blue.*

are fixed at 1.0), for a total of 20 output values in our 6-pole, 1-zero models. For all models, the output activation layer was sigmoid, producing values between 0 and 1. The frequencies were rescaled to values between 0 and 8000 Hz, the bandwidths to 20-5020 Hz, and the amplitude adjustments to between -100 and +100 dB.

Models were trained using the Adam optimizer, with a learning rate of 0.0001, and the loss function described in Section 2. The batch size was 32. During training, models were tested on the validation set after every epoch; training was halted once the validation loss did not improve after 20 epochs, or for a maximum of 200 epochs. The model iteration with the lowest validation loss was used for evaluation.

3.4. Evaluation method

For evaluation, input spectra were calculated and normalized per Section 3.2, and model output was rescaled as per Section 3.3. The formant outputs were sorted by mean frequency to determine which represents F1, which F2, and so on, and 10 rounds of 3-point binomial smoothing were applied to reduce frame-to-frame jitter in formant frequency values.

4. Results

Table 1 presents some of the models we constructed and evaluated, as well as their performance on the test set of VTR-TIMIT. The "Arch" column indicates the basic architecture of the model; e.g., "CNN3" indicates a model with 3 convolutional layers.

Our results indicated that despite their larger size and the surrounding context frames provided as input, the CNN models did not do as well as the LSTM models in our tests. A simple one-layer LSTM was sufficient to produce lower error rates, and in fact, adding two more layers did not improve performance. Note also that the bidirectional LSTM model was no better than the simple LSTM, indicating that the following context was not necessary for formant tracking. Figure 1 illustrates the output of the LSTM1 model on a test VTR-TIMIT utterance, alongside the hand-measured formants for comparison.

Table 1: *Model architectures, and their mean absolute error on the VTR-TIMIT test set.*

	All segments				**Vowels**		
Arch.	**All**	**F1**	**F2**	**F3**	**F1**	**F2**	**F3**
CNN3	129	105	117	165	65	81	111
LSTM1	**114**	**100**	**115**	**126**	**64**	**75**	**90**
BLSTM1	**114**	102	**115**	**126**	65	77	**90**
LSTM3	131	102	146	146	65	81	96

Below we focus on the LSTM1 model, redubbed FormantNet below, and compare its performance with that the DeepFormants (DF) models of [1], as well as other results published in the literature on the VTR-TIMIT database. Table 2 reproduces results published in Table VI of [1], showing mean absolute error by formant and phonetic class. The two models of [1] are presented, as well as the popular tool WaveSurfer [19], and MSR [18], which was used for the initial frequency estimations before hand-correction. (For space we omit Praat, which had the highest error rates.) Note that FormantNet produced lower error rates than the other three methods for F3 of both vowels and semivowels, as well as F2 of semivowels. Some other F2 and F3 consonant measures (underlined) were also lower than the two DF models, though not lower than the MSR method.

Table 3 reproduces Table VII of [1], in which the DF models are compared to previous published results for KARMA

Table 2: *Mean absolute error over all speech in test set, divided by phonetic class.*

	WaveSurfer			**MSR**			**DeepFormants**			**DF-RCNN**			**FormantNet**		
Class	**F1**	**F2**	**F3**	**F1**	**F2**	**F3**	**F1**	**F2**	**F3**	**F1**	**F2**	**F3**	**F1**	**F2**	**F3**
vowel	70	94	154	64	105	125	54	81	112	**53**	**72**	108	64	75	**90**
semivowel	89	126	222	83	122	154	**67**	114	168	68	111	160	79	**93**	**124**
nasal	96	229	239	67	**120**	**112**	**66**	175	151	69	191	158	98	213	143
fricative	209	263	439	**129**	**108**	**131**	131	135	159	139	142	167	160	135	161
affricate	292	407	390	**141**	**129**	**149**	164	162	189	174	173	195	186	186	186
stop	168	210	286	130	**113**	**119**	131	135	168	**123**	135	170	135	158	166

[17] in terms of root mean square error (RMSE); the RMSE for FormantNet has been appended. FormantNet has a lower error for F3, and is in the middle of the pack on F2 and overall, but shows a somewhat higher error rate for F1.

Table 3: *Root mean square error over all segments, test set.*

Method	All	F1	F2	F3
KARMA	220	**114**	226	320
DF-RNN	**163**	118	**169**	204
DF-RCNN	173	127	180	213
FormantNet	173	143	177	**195**

[13] compared DF-RNN and three LPC-based trackers (Praat [14], SNACK [15], and *assp* [16]). Using the default parameters for each tracker, [13] measured RMSE for voiced frames only, and split the results by gender, noting the default parameters for each tracker tend to favor one gender. Since DF-RNN was trained on the training set, it was only evaluated on the test set, while the other methods were evaluated on the entire VTR-TIMIT dataset. Their results are reproduced in Table 4, along with comparable results for FormantNet, which has the lowest error rate for almost all measures.

Table 4: *Root mean square error over all segments, split by gender, voiced frames only.*

	F1		F2		F3	
Tracker	**f**	**m**	**f**	**m**	**f**	**m**
All utterances:						
SNACK	126	100	291	227	313	375
ASSP	113	**96**	479	211	512	225
PRAAT	116	234	217	338	249	404
FormantNet	**108**	99	**180**	**129**	**178**	**149**
Test utterances:						
DF-RNN	120	97	195	167	252	169
FormantNet	**104**	**93**	**182**	**132**	**192**	**157**

Finally, we compare our results with the quasi-closed phase (QCP) LPC method of [9], who examined only vowels and semivowels. Their formant detection rate (FDR) measure is defined as the percentage of frames in which the difference between estimated and reference formant frequencies is both lower than a specified absolute value, and within a specified percentage deviation from the reference value. Table 5 reproduces their results at three FDR thresholds, along with the comparable values for FormantNet. We see that the FDRs for FormantNet are better for F2 and F3 but a bit lower for F1.

Table 5: *Formant detection rates (FDR) at different thresholds, for vowels and semivowels in the test set.*

Ratio	F1	F2	F3
FDR within 20% and 200 Hz dev			
QCP-FBCOV	**84.9**	85.0	83.9
FormantNet	79.6	**93.7**	**90.4**
FDR within 25% and 250 Hz dev			
QCP-FBCOV	**90.4**	90.5	89.1
FormantNet	85.9	**96.5**	**93.8**
FDR within 30% and 300 Hz dev			
QCP-FBCOV	**93.4**	93.9	92.1
FormantNet	90.1	**97.9**	**95.5**

5. Discussion and Conclusions

In summary, we found that our unsupervised FormantNet model produced error rates on VTR-TIMIT similar to those of [1], and generally better than other methods. The model did particularly well with F3, but often had somewhat higher error rates than other methods on F1. This may be because there is still interference from F0; the IAIF method may not be sufficiently separating the source signal from the vocal tract signal. In the future, we plan to explore other methods of disentangling the source and vocal tract signals. We also found that a model with just one LSTM layer outperformed larger (B)LSTM and CNN models. It may be that a larger model needs more data to produce better results; on the other hand, perhaps the problem is not that difficult and thus cannot benefit much from a larger model.

The acoustic model of speech described here (Eqn. 1) is most appropriate for voiced speech, particularly vowels and other sonorants, and Tables 2 and 5 show that our model was most competitive with vowels and semivowels. On consonants, FormantNet's performance was comparable to the DF models, but both the DF models and ours had higher error rates than MSR. We point out that the MSR method was the one used as the starting point for the hand-corrected measurements, and so it is to be expected that there may be a bias in the dataset toward the MSR measures. Similarly, the DF methods were also trained on the same measurements, whereas FormantNet was trained completely independently of those measurements. Moreover, the contrast in results between Tables 2 and 4 suggests that DF-RNN's lower error rates on consonants are based largely if not entirely on voiceless frames.

In general, there are advantages and disadvantages to the available methods. Unlike the deep-learning methods of [1] and this paper, existing LPC-based techniques require no training data and are domain-independent, but generally have higher error rates, and may be difficult to tune, particularly on multi-speaker data or real-world data with variable audio quality. Between the deep-learning methods, it is not surprising that DNN models trained on hand-labeled data would perform better than those that are not, and models trained on hand-labeled data can be trained on a much smaller amount (only 324 utterances for [1], versus 4620 for FormantNet). Yet hand-labeled data is scarce, and unsupervised methods allow all available data to be used for training, labeled or not. The DF models require adaptation to new corpora; likewise, our models trained on other corpora fare less well on the VTR-TIMIT dataset than when trained on TIMIT data. But again, it may be more feasible to train a new model with our method, without the need for labeled data. Thus, the FormantNet method may be most appropriate for large multi-speaker corpora, for which no hand-labeled data exists and for which speaker-specific tuning of automatic trackers may be infeasible.

Note also that the FormantNet method produces not only frequency estimates of the lower formants, but also bandwidth and amplitude correction estimates, as well as estimates of the higher poles and zeros, all together comprising a model of the entire vocal-tract signal within the input frequency range, which has potential applications in speech synthesis, speaker identification, and the analysis of disordered speech.

6. Acknowledgements

Funding for this research was provided by NSF grant IIS-1816726 and Nemours Biomedical Research.

7. References

[1] Y. Dissen, J. Goldberger, and J. Keshet, “Formant estimation and tracking: A deep learning approach,” *Journal of the Acoustical Society of America*, vol. 145, no. 2, pp. 642-653, Feb. 2019.

[2] L. Deng, X. Cui, R. Pruvenok, Y. Chen, S. Momen, and A. Alwan, “A database of vocal tract resonance trajectories for research in speech processing,” in *Proceedings of ICASSP 2006—IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. I-I, 2006.

[3] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, *The DARPA TIMIT acoustic-phonetic continuous speech corpus*. Linguistic Data Consortium, 1993.

[4] J. Hillenbrand, L. A. Getty, M. J. Clark, and K. Wheeler, “Acoustic characteristics of American English vowels,” *Journal of the Acoustical Society of America*, vol. 97, no. 5, pp. 3099-3111, 1995.

[5] C. G. Clopper and T. N. Tamati, “Effects of local lexical competition and regional dialect on vowel production,” *Journal of the Acoustical Society of America*, vol. 136, no. 1, pp. 1-4, 2014.

[6] B. H. Story and K. Bunton, “Formant measurement in children’s speech based on spectral filtering,” *Speech Communication*, vol. 76, pp. 93-111, 2016.

[7] D. Gowda and P. Alku, “Time-varying quasi-closed-phase weighted linear prediction analysis of speech for accurate formant detection and tracking,” in *Proceedings INTERSPEECH 2016 – 17th Annual Conference of the International Speech Communication Association,* San Francisco, USA, Sep. 8-12, 2016, pp. 1760–1764.

[8] D. Gowda, M. Airaksinen, and P. Alku, “Quasi closed phase analysis of speech signals using time varying weighted linear prediction for accurate formant tracking,” in *Proceedings of ICASSP 2016—IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 4980-4984, 2016.

[9] D. Gowda, M. Airaksinen, and P. Alku, “Quasi-closed phase forward-backward linear prediction analysis of speech for accurate formant detection and estimation,” *Journal of the Acoustical Society of America*, vol. 142, no. 3, pp. 1542-1553, 2017. Doi: 10.1121/1.5001512

[10] R. Sharma, L. V. Vignolo, G. Schlotthauer, M. A. Colominas, H. L. Rufiner, and S. R. M. Prasanna, “Empirical mode decomposition for adaptive AM-FM analysis of speech: A review,” *Speech Communication*, vol. 88, pp. 39-64, 2017. Doi: 10.1016/j.specom.2016.12.004

[11] S. Fulop and C. Shadle, “Automated formant tracking using reassigned spectrograms,” *Journal of the Acoustical Society of America*, vol. 143, no. 3, p. 1870, 2018. Doi:10.1121/1.5036138

[12] M. A. Ramírez, “Hybrid autoregressive resonance estimation and density mixture formant tracking model,” *IEEE Access*, vol. 6, pp. 30217-30224, 2018.

[13] F. Schiel and T. Zitzelberger, “Evaluation of automatic formant trackers,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018, pp. 2843-2848.

[14] P. Boersma and D. Weenink, “Praat, a system for doing phonetics by computer,” *Glot International*, vol. 5, no. 9/10, pp. 341-345, 2002.

[15] K. Sjölander, *Snack-Sound-Toolkit.* http://www.speech.kth.se/snack

[16] M. Scheffer, *Advanced Speech Signal Processor (libassp).* http://www.sourceforge.net/projects/libassp

[17] D. D. Mehta, D. Rudoy, and P. J. Wolfe, “Kalman-based autoregressive moving average modeling and inference for formant and antiformant tracking,” *Journal of the Acoustical Society of America*, vol. 132, no. 3, pp. 1732-1746, 2012.

[18] L. Deng, L. J. Lee, H. Attias, and A. Acero, “A structured speech model with continuous hidden dynamics and prediction-residual training for tracking vocal tract resonances,” in *Proceedings of ICASSP 2004—IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. I-557, 2004.

[19] K. Sjölander and J. Beskow, “WaveSurfer – an open source speech tool,” in *Proceedings INTERSPEECH 2000 – 1st Annual Conference of the International Speech Communication Association,* Beijing, China, Oct. 16-20, 2000, pp. 464–467.

[20] M. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE Journal*, vol. 37, no. 2, pp, 233-243, 1991. Doi: 10.1002/aic.690370209

[21] G. E. Hinton and R. S. Zemel, “Autoencoders, minimum description length and Helmholtz free energy,” *Advances in Neural Information Processing Systems 6*, pp. 3-10, 1994.

[22] J. P. Olive, “Automatic Formant Tracking by a Newton-Raphson Technique,” *Journal of the Acoustical Society of America*, vol. 50, pp. 661-670, 1971. Doi: 10.1121/1.1912681

[23] G. Fant, *Acoustic Theory of Speech Production*. The Hague: Mouton, 1970.

[24] J. N. Holmes, "Research report - formant synthesizers - cascade or parallel," *Speech Communication*, vol. 2, pp. 251-273, 1983.

[25] P. Alku, “Glottal wave analysis with pitch synchronous iterative adaptive filtering,” *Speech Communication*, vol. 19, pp. 459–476, 1992.

[26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015. Software available from tensorflow.org. Version 2.3.