GLOTTAL ESTIMATION USING A NONLINEAR ARMA FILTER

Yang Zhang, Mark Hasegawa-Johnson

University of Illinois, Urbana-Champaign
Department of Electrical and Computer Engineering

ABSTRACT

EGG, which depicts the degree of contact between vocal folds, is a measurable proxy for glottal pressure wave, and thus contains information of pitch and GCI. In this project, I proposed a TDRNN, or a nonlinear ARMA filter, that turns input speech waveforms to approximation of EGG, which can be further applied to pitch tracking or GCI location. Two baseline systems, a linear MA filter and a nonlinear MA filter, are also experimented and evaluated. Preliminary experiments have confirm the ability of the proposed system to estimate EGG, and further analysis discloses more interesting findings.

Index Terms— TDRNN, EGG, nonlinear ARMA filter

1. INTRODUCTION

According to the popular source-filter model of speech, the derivative of glottal pressure wave is generally regarded as the excitation of the source-filter model [1]. Therefore, it bears information of group delay and excitation period, and is essential to speech processing tasks such as pitch tracking and pitch-synchronous analysis.

However, so far there hasn't been a robust way of measuring glottal pressure wave. Although there have been many algorithms estimating it [2], these algorithms can only be partly verified by either synthetic data or pitch tracking.

A measurable proxy for glottal pressure wave is the electroglottograph [3], or EGG, which measures the degree of contact between two vibrating vocal folds. Experiments show that there exist a reliable nonlinear relationship between EGG and glottal pressure wave [3]. Although EGG contains less information than glottal pressure wave (information is lost when the vocal folds are completely apart), it still preserves information of GCI, or glottal closure instance, where short-time energy is highest within a period, and thereby group delay and pitch. In fact, there are many datasets whose pitch labels are obtained from EGG.

Therefore, if there is a way to learn the relationship between speech waveform and EGG, it will have great potentials in GCI locating and pitch tracking. Neural network is a popular and effective approach to learn nonlinear relationship between two signals, given that its architecture and nonlinearity are carefully determined, and so is suitable for our task.

This paper proposes to estimate EGG from speech waveform using neural network. The proposed system produces EGG estimates in real-time, and thus can be regarded as an non-linear filter. As will be analyzed in the next section, feedback loop is needed for more accurate estimation, making it an ARMA filter. The resulting architecture is thus time delay recurrent neural network (TDRNN).

The rest of the report is organized as follows. Section 2 discusses the architecture of the neural network; section 3 derives necessary training algorithms; section 4 gives some results and some interesting analyses of some preliminary experiments; and section 5 concludes the report and points out further research directions.

2. ARCHITECTURE

The general architecture of the proposed TDRNN is given by figure 1, which combines DFI representation, where Z^{-1} represents unit step time delay. From this figure, it is straightforward that the architecture can indeed be regarded as a nonlinear ARMA filter. There is a single output node, which approximates the real-time EGG waveform. The following subsections will discuss detailed settings, such as the number of nodes and layers etc.

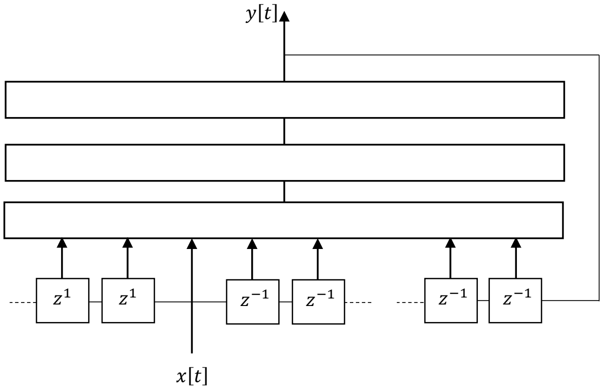


Fig. 1. General architecture of the proposed TDRNN.

2.1. Source of Nonlinearity

To avoid determining the network architecture too heuristically, it is necessary to discuss about the sources of nonlinearity. As mentioned in section 1, the relationship between speech waveform and EGG is connected by glottal wave, and so the nonlinearity must come from the relationship between either glottal wave and EGG, or glottal wave and speech waveform.

The first source of nonlinearity is the nonlinear relationship between glottal wave and EGG, which is obvious. The second nonlinearity comes from the coupling between vocal tract and glottal vibration. It has been well studied [4] that the wave reflected by the vocal tract and propagating back to the glottal area will interact with the glottal vibration, which induces nonlinearity, generally in the form of wider first formant.

The third source, which I believe is the most important one, lies in the changing vocal tract. Although, according to the source-filter model, vocal tract can be regarded as a LTI system within a very short time period, which means the relationship between glottal wave and speech waveform is linear in short time scope, the system keeps changing slowly over time, contributing to nonlinearity. An intuitive way of understanding this is that in order to determine which linear filter to apply, a phone recognition procedure has to be performed first, and this recognition procedure is generally nonlinear.

2.2. MA and AR orders

MA order is how many points of speech waveform should be included as input. The choice of MA order should be based on two considerations. On one hand, it should be no less than the number of poles of the vocal tract filter, because the neural net performs the inverse filtering to the vocal tract filter. It is generally believed that 10 poles are enough for approximating the vocal tract, and so MA order should be at least 10.

On the other hand, as discussed in the previous subsection, the speech input should bear enough information for phone recognition. Common features for speech recognition, such as MFCC and LPC, are generally extracted from frames that are around 30ms long. In our experiment, the sampling rate of the input speech is 10kHz. Based on these considerations, I set the MA order to 301, 150 lag and 150 ahead.

AR order is how many previous output should feed back to the network. By similar consideration, AR order should at least match the zeros of the vocal tract. Vocal tract zeros are induced by the parallel connection of vocal tract, nasal tract and glottal tract, each of which can be regarded as an all-pole system. With simple derivations, it can be proved that the number of zeros shall not exceed the number of poles, which is 10 by previous discussion. For research purpose, I set AR order to 10.

2.3. Layers and Hidden Nodes

To improve convergence and for simplicity, I apply only 1 hidden layer. According to section 2.1, the number of nodes in this hidden layer should suffice to approximate nonlinear mapping between glottal wave and EGG, and perform phone recognition. While, heuristically, a few hidden nodes are enough for the former purpose, the latter purpose requires the number of hidden nodes at the order of the total number of phones in the language. In English, there are around 50 phones, and so shall be the number of hidden nodes. However, within limited time, it is too hasty to train such a network on a training set large enough to cover all the phones. Instead, I set the number of hidden nodes to 3 and select a small test set. The impact on generalization capability will be discussed in section 4.

2.4. Nonlinearities and Stability

The choice of nonlinearities should take into account stability issue. In filter design, we are often concerned about BIBO stability, which means if the input is bounded, then the output is bounded. In linear systems, a sufficient condition for BIBO stability is that the system is causal and all the poles, if any, lie within the unit circle. In nonlinear systems, however, I don't know of any simple theorems, even if they exist. Instead, stability can be achieved by nonlinearities with bounded output, such as the hyperbolic tangent.

In the proposed system, hyperbolic tangent is applied to all the hidden nodes, but no nonlinearity, or linear nonlinearity, is applied to the output node. This is because we don't know beforehand what the output EGG is bounded by, and it is inefficient to predefine the range. It can be easily proved, by Cauchy-Schwarz, that the boundedness of the hidden nodes is enough to ensure boundedness of the output:

$$\begin{aligned}\|y\|_2 &= \left\|c + \boldsymbol{w}^T\boldsymbol{z}\right\|_2 \\ &\le \sqrt{c^2 + \|\boldsymbol{w}\|_2^2}\sqrt{1 + \|\boldsymbol{z}\|_2^2} \\ &\le \sqrt{c^2 + \|\boldsymbol{w}\|_2^2}\sqrt{1+K}\end{aligned} \tag{1}$$

where y is the output node; $\boldsymbol{z}$ is the hidden nodes; $\boldsymbol{w}$ and c are weights connected the hidden layer and the output node; K is the total number of hidden nodes. The notation will be discussed in detail in the next section. From (1), it can be seen that the network learns the output range by learning the weights.

2.5. Baseline Systems

To compare system performance, two alternative systems are also experimented. The first system is a simple linear MA filter, or equivalently two-layer neural net without nonlinearities, with the same MA orders. The second system is a nonlinear MA filter, which is the same as the proposed system, except that there are no feedback loops.

Theoretically, the proposed system should perform best, followed by the nonlinear MA filter, and then by linear MA filter, but the convergence rate is reversed. Linear MA filter has analytical solutions, and the nonlinear MA filter should converge faster than the proposed system. An efficient training algorithm, therefore, will be to initialize the weights based on these alternative systems, which will be discussed in the next section.

3. TRAINING

3.1. Notations

Before any training algorithm is derived, it is necessary to define notations, which is similar to those used in class. $s[t]$ denotes speech waveform, $l[t]$ denotes EGG waveform, $y[t]$ is the output of nonlinear filter. $y[t]$ should be as close to $l[t]$ as possible. The squared error is adopted as the error metric.

$$\varepsilon = \sum_t \varepsilon_t = \sum_t (l[t] - y[t])^2 \tag{2}$$

The input vector at time t is defined by

$$\boldsymbol{x}_t = [s[t-P], \cdots, s[t+Q], y[t-M], \cdots, y[t-1]]^T \tag{3}$$

where $P = Q = 150$, and $M = 10$ in my settings. Let $\boldsymbol{z}_t$ denote the hidden nodes at time t and $\boldsymbol{a}_t$ denote intermediate output that satisfies

$$\boldsymbol{z}_t = \tanh(\boldsymbol{a}_t) = \tanh(\boldsymbol{b} + \boldsymbol{W}\boldsymbol{x}_t) \tag{4}$$

where $\boldsymbol{b}$ and $\boldsymbol{W}$ are weights connecting input nodes and hidden nodes. $\boldsymbol{z}_t$, $\boldsymbol{a}_t$ and $\boldsymbol{b}$ are all K-by-1 vectors where $K = 3$ in this experiment. $\boldsymbol{W}$ is a K-by-$(P+Q+M+1)$ matrix. The relation between the hidden nodes and output is given by

$$y[t] = c + \boldsymbol{w}\boldsymbol{z}_t \tag{5}$$

where c and $\boldsymbol{w}$ are weights connecting hidden nodes and the output. $\boldsymbol{w}$ is a 1-by-K vector.

For the nonlinear MA filter, notation is largely the same, except that the input vector becomes

$$\boldsymbol{x}_t = [s[t-P], \cdots, s[t+Q]]^T \tag{6}$$

The notations for linear MA filter differ further by the input-output relationship:

$$y[t] = \boldsymbol{b} + \boldsymbol{W}\boldsymbol{x}_t \tag{7}$$

3.2. Training Algorithms for Baseline Systems

Training algorithms for baseline systems have been well covered in class. Here I list all the relevant algorithms only to demonstrate that I understand them.

3.2.1. Pseudo-Inverse for the Linear MA Filter

The optimal weights for the linear MA filter can be solved analytically using pseudo-inverse:

$$[\boldsymbol{b}_{opt}, \boldsymbol{W}_{opt}] = \left(\boldsymbol{X}\boldsymbol{X}^T\right)^{-1} \boldsymbol{X}\boldsymbol{y} \tag{8}$$

where

$$\boldsymbol{X} = [\boldsymbol{x}_{1+P}, \cdots, \boldsymbol{x}_{N-Q}] \tag{9}$$

and

$$\boldsymbol{y} = [y[1+P], ..., y[N-Q]]^T \tag{10}$$

3.2.2. Error Back Propagation for the Nonlinear MA Filter

The optimal weights for the nonlinear MA filter can be solved using gradient descent, and the gradient of weights can be obtained using error back propagation. Define

$$\begin{aligned} \delta_{y,t} &= \frac{\partial \varepsilon}{\partial y[t]} = \frac{\partial \varepsilon_t}{\partial y[t]} = 2\left(y[t] - l[t]\right) \\ \boldsymbol{\delta}_{z,t} &= \frac{\partial \varepsilon}{\partial \boldsymbol{a}_t} = \frac{\partial \varepsilon_t}{\partial \boldsymbol{a}_t} \end{aligned} \tag{11}$$

We have the following relation

$$\boldsymbol{\delta}_{z,t} = \boldsymbol{w}^T \delta_{y,t} \times \boldsymbol{z}_t \times (1 - \boldsymbol{z}_t) \tag{12}$$

where × denotes element-wise multiplication. Therefore, the gradients of weights can be calculated by

$$\begin{aligned} \frac{\partial \varepsilon}{\partial \boldsymbol{w}} &= \sum_t \delta_{y,t} \boldsymbol{z}_t \\ \frac{\partial \varepsilon}{\partial c} &= \sum_t \delta_{y,t} \\ \frac{\partial \varepsilon}{\partial \boldsymbol{W}} &= \sum_t \boldsymbol{\delta}_{z,t} \boldsymbol{x}_t^T \\ \frac{\partial \varepsilon}{\partial \boldsymbol{b}} &= \sum_t \boldsymbol{\delta}_{z,t} \end{aligned} \tag{13}$$

3.3. Error Back Propagation Through Time

For Nonlinear ARMA filter, there are dependencies among nodes at different times, and therefore the gradient calculation is different from that for nonlinear MA filter. However, the TDRNN is still a feed forward network, and BP algorithm is still applicable.

Since the output is used as future input, the gradient with respect to the output at time t should include errors propagating back from future times. Specifically

$$\begin{aligned} \delta_{y,t} &= \frac{\partial \varepsilon}{\partial y[t]} \\ &= 2\left(y[t] - l[t]\right) + \sum_{\tau=1}^{M} \boldsymbol{m}_{P+Q+1+\tau} \boldsymbol{\delta}_{z,t+\tau} \end{aligned} \tag{14}$$

where $\boldsymbol{m}_i$ is the transpose of the i-th column of the weight matrix $\boldsymbol{W}$. $P+Q+1+\tau$ is where $y[t]$ is located in the future input vector $\boldsymbol{x}_{t+\tau}$ when $1 \le \tau \le M$.

The definition of $\delta_{y,t}$ is the same as in (11), but notice that it is no longer equal to $\partial\varepsilon_t / \partial \boldsymbol{a}_t$. Since the architecture within time t is the same as in the case of nonlinear MA filter, (12) and (13) still hold. The time complexity of this algorithm is still $O\left(WN\right)$, where W is total number of weights.

3.4. Initialization

To avoid being tracked in local optimum and to accelerate convergence, initial weights should be set carefully. It is generally believed that deep architectures converge more slowly than shallow structures due to increased nonlinearity. Although there is only one hidden layer for each time, experiments show that the feedback loops make convergence rate low, possibly due to the high equivalent depth. Comparatively, the nonlinear MA filter converge much faster, and therefore it is useful to initialize using the converged weights of the nonlinear MA filter.

Another advantage for this method is that the convergence of the nonlinear MA filter can be accelerated by Lavenberg-Marquadt algorithm, but the nonlinear ARMA filter cannot. This is because LM approximation requires calculating $\partial\varepsilon_t / \partial w$. Since ε_t will propagate back to all previous times, calculating $\partial\varepsilon_t / \partial w$ requires $O\left(t\right)$ operation. The total complexity, therefore, is $O\left(W^2 N^2\right)$ instead of $O\left(W^2 N\right)$.

The initialization in my implementation is as follows. After the optimum of $\boldsymbol{W}_{opt}$ and $\boldsymbol{w}_{opt}$ are obtained for the nonlinear MA filter, set

$$\begin{aligned} \boldsymbol{W}_0 &= [\boldsymbol{W}_{opt}, \boldsymbol{0}] \\ \boldsymbol{w}_0 &= \boldsymbol{w}_{opt} \end{aligned} \tag{15}$$

as the initial weights for the nonlinear ARMA filter, where **0** is a K-by-M zero matrix. The idea is to set the weights for AR component to 0 and keep the rest the same.

4. EXPERIMENTS

4.1. Configuration

The experiments are performed on the Edinburgh dataset[5], which contains 50 utterances of a male speaker and 50 of a female speaker, as well as their simultaneous EGG's. Both the speech waveforms and EGG's are down sampled to 10kHz. As is mentioned in section 2.3, the training set should be set small to accelerate convergence. Therefore, the training set contains only 1 utterance "Where can I park my car" by the male speaker, and the test set contains the rest

49 utterances by the same speaker. Female speech is not used for now. All 3 architectures are trained and tested on the same partition of the dataset.

4.2. Statistics and Overview

The average squared error (ASE) is defined as

$$\text{ASE} = \frac{1}{\sum_i N_i} \sum_i \sum_t (y[t] - l[t])^2 \tag{16}$$

where i goes through all test/training utterances and N_i is the total number of sample points in the i-th utterance.

Table 1 shows the ASE on the training and the test set of the three different systems. From the result on training set, we can see that by adding a 3-node hidden layer, the error drops by more than half, which confirms my previous analysis that the relation between speech waveform and EGG is indeed nonlinear. The error drops further by around 5% when feedback loop is introduced. The improvement is not huge, which may indicate that the zeros of the vocal tract exist, but are not significant.

System	Training	Test
Linear MA	0.0051	0.0039
Nonlinear MA	0.0018	0.0036
Nonlinear ARMA	0.0016	0.0035

Table 1. Average Squared Error

The results on the test set reflect the problem of over-fitting, which results from insufficient model complexity as well as small test dataset. First, although the advantage of the nonlinear ARMA architecture over the other two is generalizable the test set, the difference is too small to be significance. Second, the ASE on the training set is abnormally greater that those on the test set, especially for the linear MA filter, which suggests that the test utterance may be an outlier. This problem can be solved by increasing complexity and test set, which will be a future research direction.

Figure 2 shows the nonlinear ARMA output as well as its EGG of two consecutive voiced segments of the training utterance. Generally speaking, the fine structure of the speech waveform is largely filtered out, though there is some remaining. The differences in output waveform between the two voiced segments, where the original speech waveform differs a lot, are greatly eliminated. However, a serious problem is that the system is unable to perform reliable UV decision, which can be seen from the burst of energy, which is supposed to be removed, at the beginning of the voice segment. This, again, results from insufficient model complexity.

4.3. Signal Processing Analyses on the Linear MA Filter

For the linear MA filter, signal processing analyses can bring some interesting perspectives.

The impulse response is simply the weights connecting the input and output layer, constant excluded. Figure 3(a) shows the impulse response of the filter. It is reasonable that the impulse response concentrate around 0, but it is quite surprising that there is larger anti-causal component, the impulse response before 0, than causal component. A tentative explanation for this is that the nonlinearity may be better approximated by introducing group delay, which changes the causality of the system. This hypothesis can be partly verified from the observation that the nonlinear filters are less anti-causal.

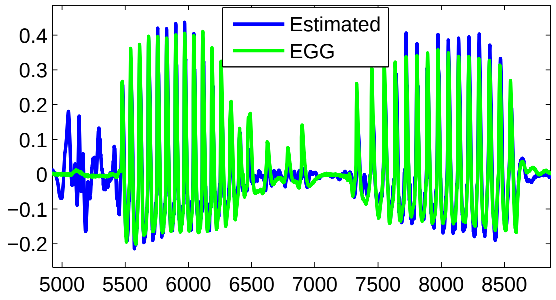
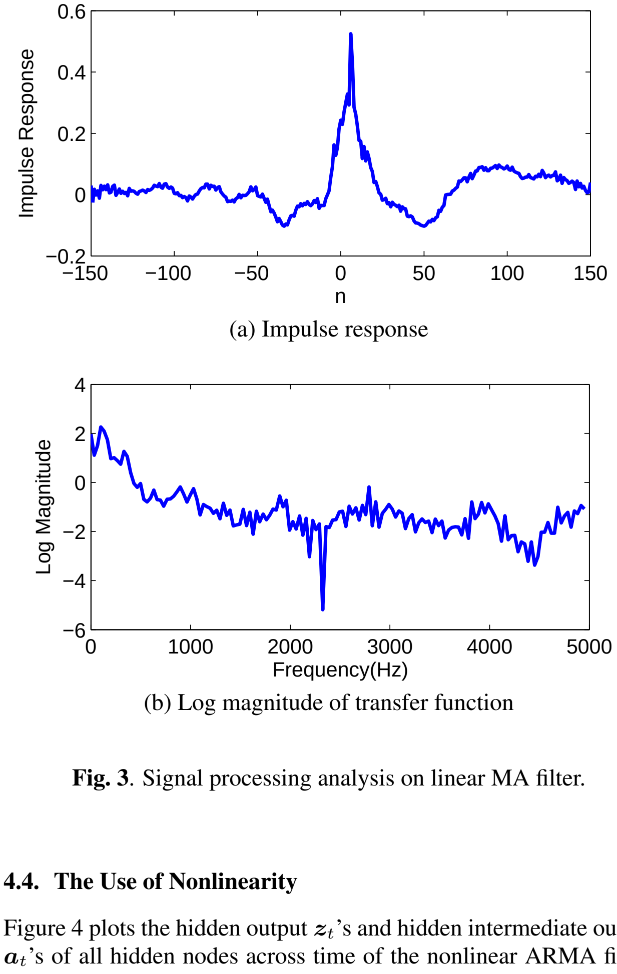


Fig. 2. Output of nonlinear filter versus EGG in two consecutive voiced segments.

The magnitude of the transfer function is plotted in figure 3(b). One immediate observation is that the filter is essentially a low-pass filter, which is reasonable because EGG signal is essentially a low-passy signal. It is also interesting to observe that there is some vague formant structures in the transfer function, which is probably the result of averaging of the formant structures of the few phones in the training utterance.



(a) Impulse response

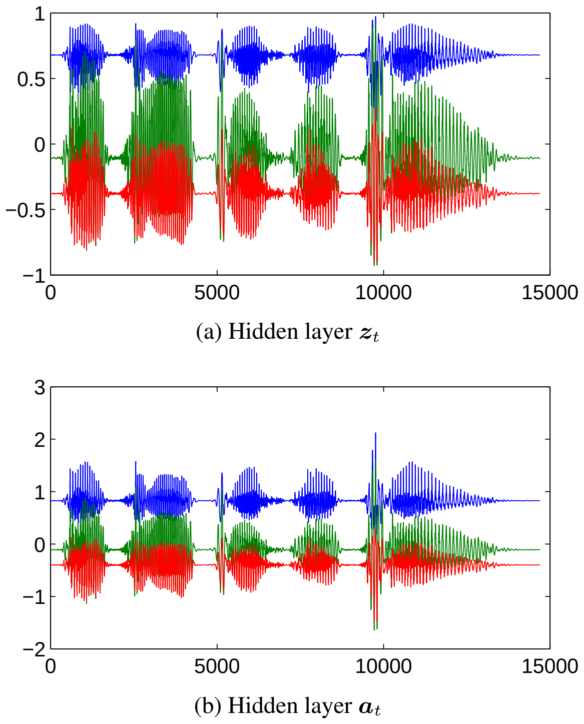
(b) Log magnitude of transfer function

Fig. 3. Signal processing analysis on linear MA filter.

4.4. The Use of Nonlinearity

Figure 4 plots the hidden output $\boldsymbol{z}_t$'s and hidden intermediate output $\boldsymbol{a}_t$'s of all hidden nodes across time of the nonlinear ARMA filter, in which we can have a very straightforward idea of how linearities of the hidden nodes are applied. Notice that the three hidden outputs are of similar shape, but there is a large distinction in DC offsets. The intermediate output of the upper signal exceedes 1 and that of the lower one exceeds -1. Both signals are curbed by the hyperbolic tangent function. The effect of nonlinearity is less significant for the signal in the middle.

Further, notice that the weights connecting the upper, middle and lower hidden outputs to the output layer are -2.0153, 2.1788 and -3.1902 respectively. Therefore, we can view the middle signal as the "main signal", and the other two as the "modification signals" to be subtracted. The modification signals are offset to where convexity/concavity of the sigmoid function is obvious, which in turn introduces nonlinear adjustment to the main signal. This is an interesting way of utilizing the nonlinear sigmoid functions.



(a) Hidden layer $\boldsymbol{z}_t$

(b) Hidden layer $\boldsymbol{a}_t$

Fig. 4. Demonstration of how the neural network utilizes nonlinearity. Each sequence is the time domain signal of a single node.

4.5. The Poles of the Nonlinear ARMA Filter

Finally, let's take a look at the poles of the system. Although the interpretation of poles is less straightforward for nonlinear filters, it does have interesting meaning in our case, because, according to section 4.4, the main signal is less distorted by the hyperbolic tangent, which is a good approximation to a linear filter.

Figure 5 shows the poles of the feedback loop connecting to the three hidden nodes. The distribution of the poles displays amazing patterns. First, all poles strictly lie within the unit circle, which further ensures stability. Second, the poles corresponding to the two modification hidden nodes have similar angles, and their norms have strict orders. As analyzed previously, these poles may correspond to the zeros of the vocal tract system, and the highly uniform angles of these pole may indicate the general distribution of spectral zeros.

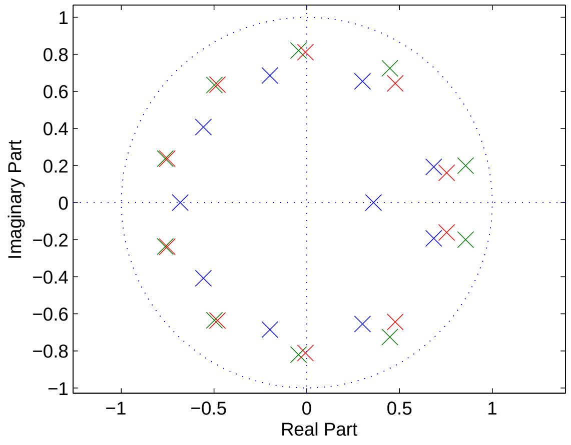


Fig. 5. Poles of the nonlinear ARMA filter. Each color denotes a set of poles corresponding to a single hidden node.

5. CONCLUSIONS AND FUTURE WORKS

In this project, I proposed a TDRNN, or a nonlinear ARMA filter, that turns input speech waveforms to approximation of EGG, which can be further applied to pitch tracking and GCI location. Two baseline systems, a linear MA filter and a nonlinear MA filter, are also experimented and evaluated. Preliminary results show that the proposed neural network is able to learn this nonlinear mapping better than the baseline systems, although the generalized advantage is not significant and UV decision is poor. Further analyses show that the proposed system has an interesting way of utilizing the nonlinearities of the hidden layer, and AR component learns an amazing pattern that might correspond to the zeros of the vocal tract system.

One of the biggest drawback regarding the current architecture is that the number of layers/hidden nodes is too small, and thereby the training set is too small, to learn the general nonlinear mapping, and to perform UV decision. Therefore, it will be useful to implement more efficient learning algorithms to train a more complex architecture efficiently. The second future direction is to use the estimated EGG to perform pitch tracking and GCI location and compare it to the current state-of-the-art. The potential advantage of this algorithm is that the training set contains richer information than simply pitch and GCI labels, which is often used in other supervised algorithms. It is interesting to investigate how this additional information can help.

6. REFERENCES

[1] Thomas F Quatieri, *Discrete-time speech signal processing*, Pearson Education India, 2002.

[2] Thomas Drugman, Baris Bozkurt, and Thierry Dutoit, "Causal–anticausal decomposition of speech using complex cepstrum for glottal source estimation," *Speech Communication*, vol. 53, no. 6, pp. 855–866, 2011.

[3] FLE Lecluse, MP Brocaar, and J Verschuure, "The electroglottography and its relation to glottal activity," *Folia Phoniatrica et Logopaedica*, vol. 27, no. 3, pp. 215–224, 1975.

[4] Ingo R Titze, "Nonlinear source–filter coupling in phonation:

Theory," *The Journal of the Acoustical Society of America*, vol. 123, pp. 2733, 2008.

[5] Paul C Bagshaw, Steven M Hiller, and Mervyn A Jack, "Enhanced pitch tracking and the processing of f0 contours for computer aided intonation teaching.," in *Proc. Eurospeech*. International Speech Communication Association, 1993.