Optimizations and Fitting Procedures for the Liljencrants-Fant model for Statistical Parametric Speech Synthesis

Prasanna Kumar Muthukumar1, Alan W Black1 H. Timothy Bunnell2

1Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA
2Nemours Biomedical Research, Wilmington, USA

Abstract

Every parametric speech synthesizer requires a good excitation model to produce speech that sounds natural. In this paper, we describe efforts toward building one such model using the Liljencrants-Fant (LF) model. We used the Iterative Adaptive Inverse Filtering technique to derive an initial estimate of the glottal flow derivative (GFD). Candidate pitch periods in the estimated GFD were then located and LF model parameters estimated using a gradient descent optimization algorithm. Residual energy in the GFD, after subtracting the fitted LF signal, was then modeled by a 4-term LPC model plus energy term to extend the excitation model and account for source information not captured by the LF model. The ClusterGen speech synthesizer was then trained to predict this excitation parameters from text so that the excitation model could be used for speech synthesis. ClusterGen excitation predictions were further used to reinitialize the excitation fitting process and iteratively improve the fit by including modeled voiced and segmental influences on the LF parameters. The results of all of these methods have been confirmed both using listening tests and objective metrics.

Index Terms: Speech synthesis, Liljencrants-Fant model, statistical parametric synthesis

1. Introduction

Excitation modeling is widely acknowledged to be important; yet, it has not received the amount of attention that is devoted to modelling the vocal tract. In an earlier paper [1], we were able to show that information contained in the excitation could be used to classify the emotions contained in speech. A good excitation model will therefore help us move closer towards the ultimate goal of making synthetic speech more expressive. In this paper, we attempt to construct an excitation model by using the classic Liljencrants-Fant model (LF model, for short) with a novel iterative way of estimating the parameters of this model for a given glottal flow. We begin by describing the LF model itself. We then describe a way to get a good separation between the source and the vocal tract using the Iterative Adaptive Inverse Filtering (IAIF) method [2]. The LF model is then fit to the residual obtained by the IAIF procedure. Any remnants of the residual which are not modeled by the LF model are modeled using a low order Linear Prediction fit. The vocal tract is modeled using standard LSPs [3]. The ClusterGen speech synthesizer is then used to learn a mapping between the text and the synthesis parameters. We then describe an iterative technique that we used to improve the quality of the LF model fit by initializing it with ClusterGen’s predictions of the fit. We also describe optimizations that we do with ClusterGen to improve predictability.

2. Related Work

A variety of methods have been proposed and tested for estimating excitation source parameters both as a means of accurately measuring voice quality [4, 5, 6], to provide an improved source parameterization for speech synthesis [7, 8, 9, 10, 11, 12], and as a means of improving the ability to convey emotion and expressiveness in synthesis [13, 14]. In most cases, the LF model has been chosen to represent the voiced source parameters, although several alternatives have also been explored (e.g., [15, 16, 17]). The process of estimating LF parameters typically begins by inverse filtering the acoustic speech signal either manually (e.g., [8]) or automatically (e.g., [11, 12, 9]) to remove the contribution of vocal tract resonances from the signal. Thereafter, estimates of the LF parameters can be derived either directly from measures of isolated pitch epochs [11, 12], or by using search strategies that attempt to locate the best parameters by minimizing an error measure either in the time domain [4] or a combination of the two [8].

Our current model is similar to the one proposed jointly in [17] and [18]. Both this method and the one we propose, combine voiced, mixed, and voiceless excitation by employing the LF model for a voicing source along with a separate model that is fit to information in the differentiated glottal flow that is not captured by the LF model (we use a low-order LPC while the previous model uses a Harmonic plus Noise Model [19]). This approach blends smoothly from voiced to voiceless regions by allowing the LPC fit to dominate the signal in regions where the LF model parameters are difficult or impossible to estimate.

While our model itself might be related to a previous model, the novelty of our paper lies in the fitting process itself and the unique way in which it makes use of a statistical parametric speech synthesizer.

3. Liljencrants-Fant model

The Liljencrants-Fant model [20] (LF model) was developed in the 1980’s as a mathematical way of describing the glottal flow derivative. The reason for using the derivative of the glottal flow, as opposed to the glottal flow itself, was that the derivative was easier to model.

The model itself consists of the following equation:

\[
\varepsilon(t) = \begin{cases} \frac{E_0 e^{\alpha t} \sin(\omega_g t)}{T_{c e}} [e^{-\varepsilon(t-T_{c e})} - e^{-e(t-T_{c e})}] & t < T_{ve} \\
T_{ve} < t < T_{v} 
\end{cases}
\]

Figure 1 depicts a plot of the LF model for typical values of the parameters. The parameters \( T_{pe}, T_{ve}, T_{ve}, T_{ve} \) are explained in the figure. The parameters \( E_0, \alpha \) and \( \varepsilon \) can be determined from the positive peak. The glottal frequency \( \omega_g \) can be determined from the fundamental period \( T_0 \).
4. Iterative Adaptive Inverse Filtering

The spectra of the vocal tract and the glottal source are deeply intertwined with each other. Since the LF model was designed specifically to model the glottal source derivative, we need to separate the spectrum of the glottal source from that of the vocal tract as best as we can.

The Iterative Adaptive Inverse Filtering method as first described in [21] does this by iteratively making estimates of the glottal source spectrum and vocal tract spectrum and alternatively inverse filtering with one to get better estimates of the other. The final estimate of the glottal source spectrum is then far better than could otherwise have been obtained with simpler inverse filtering methods. The specific technique we use is more or less identical to the setup described comprehensively in Raitio et al.[2] but in the interests of keeping the paper as self-contained as possible, we provide a block diagram that describes the process in Figure 2.

The result of Iterative Adaptive Inverse Filtering is a set of LPC coefficients (which we convert to LSPs) that provide an estimate of the vocal tract spectrum and the glottal source function (the outputs of the two shaded blocks in Figure 2).

While it could be argued that the IAIF procedure might not produce a glottal source spectrum that is perfectly separated from vocal tract effects, we were more interested in getting a glottal source estimate that could be modeled well for speech synthesis. For that purpose, we believe that this method is sufficient.

5. Fitting LF Model parameters

Automatic extraction of LF model parameters from running speech has been approached in a variety of ways and with varying degrees of success [11]. We tried a variety of time- and frequency-domain methods for fitting LF parameters to the IAIF residual. The one reported here is the best performing for this particular task. While we found that spectral fitting worked well in some cases, it had a tendency to be quite unstable and would fail to find a proper fit in other cases. In contrast to this, one time-domain method was found to produce consistent and reasonable estimates for the LF Model parameters.

Our analysis windows were 70 msec wide and stepped in 5 msec increments through the IAIF-estimated source waveform while fitting LF model parameters. The LF parameters obtained for the single pitch pulse that spanned the center of each analysis window were output as the parameters for each 5-msec frame. Thus, the time-domain fitting, which is inherently pitch-synchronous, was used to obtain parameter estimates for a uniform frame rate. Fitting the LF model was a two-stage process in which we first located a probable instance of maximum glottal closure rate ($T_c$ in the LF model), and then optimized parameters for a pitch pulse around that closure event. We did this by generating a model LF pitch pulse with $T_0$ set to something close to the speaker’s maximum pitch period and default shape parameters. This pitch pulse was then convolved with the windowed signal. The magnitudes, locations, and spacing of peaks in the convolved signal were used to estimate voicing, pitch period locations, and $F_0$ within the window. If a pitch period was located in a region that spanned the center of the window, that period was then passed to a second stage optimization where the $T_p$, $T_e$, $T_a$, and amplitude parameters were iteratively adjusted to reduce an RMS error metric until no further improvement was found.

The second stage optimization was quite simple. The pitch period duration $T_0$ was held constant and the three temporal parameters, $T_p$, $T_e$, and $T_a$ (quantized to the sampling interval for the digital waveform) were adjusted by one sample time forward or backward on each iteration. Similarly, the amplitude term was adjusted by 1dB on each iteration. A new pitch pulse was generated and the RMS error term was reevaluated following the adjustment of each parameter. If the adjustment of a parameter resulted in a reduction in the error term, the new parameter value was kept, otherwise the adjustment was discarded.
and the direction of change for that parameter on the next iteration was reversed. Iteration was stopped when no parameter adjustments in any direction led to further reduction in the RMS error.

This process, while very inefficient, seemed to be quite robust when supplied with reasonable starting estimates of $T_0$ and $T_e$, despite moderately strong F1-second harmonic influence on the source waveform.

We must remember that the LF model is only an approximation of the shapes in the glottal flow derivative. The glottal flow derivative contains a lot of high frequency content that the LF model fails to capture. Without these high frequency components, the synthesized speech tends to sound hollow and muffled. To model these components, we subtract the fitted LF model from the glottal derivative. A low order LPC (we used a 4th order LPC) is then fit to the remaining components that are not captured by the LF model.

This approach also avoids the need to make a voicing decision. We had originally tried making voicing decisions and switching between the LF model and white noise as was appropriate. However, we found it difficult to make accurate voicing decisions; and imperfect voicing decisions made the synthesized speech sound jarring or extremely hoarse. These problems can be mitigated by modeling the ‘residual’ of the LF model.

![Figure 3: Fitting to the residual: the raw glottal flow derivative is in black, the estimated LF model is in blue and the residual error is in red](image)

5.1. Results

We applied our techniques to a statistical parametric speech synthesizer built using the ClusterGen framework[22] trained with data from the CMU Arctic database[23]. The database consists of data from multiple speakers, each of whom provided one hour of speech. While our methods were validated using data from multiple speakers, in the interest of simplicity, we will only discuss results from the system trained with the data from a single speaker, RMS (male, American English). 20-order LSPs were used for the vocal tract model and 4-order LPCs for the residual of the LF model.

To test the quality of our models, we conducted listening tests on Amazon’s Mechanical Turk using the Testvox framework[24] where listeners were asked to choose between two systems. The first system was a synthesizer that used Line Spectral Pairs and the above described (LF) models for the residual. The second system was a baseline system that used an identical vocal tract model but a mixed excitation (ME) residual[25]. Listeners were asked to pick the system which they thought sounded more natural. 19 listeners were asked to choose between 10 utterances generated by each system resulting in 190 tests between the two systems. In 116 of those tests, listeners judged our system to be more natural. In 70 of those tests, listeners judged the baseline system to sound more natural. In 4 of the tests, listeners did not have a preference.

A Generalized Estimating Equation (GEE) model was used with listeners repeated over sentences to test the hypothesis that the model intercept is zero, in other words, that the odds of a listener selecting the LF version were equal to the odds of the listener selecting the ME version as more natural sounding. The model coefficient for the intercept was 0.505 with a standard error of 0.2525, a Wald Chi-Square of 4.002 with 1 degree of freedom and a p-value of 0.045. We can thus reject the hypothesis of equal odds and conclude that listeners preferred to LF model versions of the sentences.

6. Improving the Fit

The LF model fitting process described in a previous section is essentially a form of gradient descent and so is highly dependent on starting from a good initial position. Statistical Parametric Synthesis[26] provides us with an elegant way of getting good initial estimates. We start by fitting the LF model to frames of speech at 5 msec intervals, as described earlier, with a window large enough to contain at least two or three glottal pulses. We then use these parameters along with the vocal tract model to build a synthetic voice in the ClusterGen framework. ClusterGen involves building a set of Classification And Regression Trees (CARTs)[27] that learn a mapping between the phones (with context) and the feature vectors, LSPs and LF parameters in this case.

One of the most unique parts of our approach is this: we use these CARTs to predict the LF parameter values of our entire database. The LF fitting process then uses these predicted parameters as seed values. These are used to create the model shape that is convolved with the frame to detect the glottal pulse and also as one of the multiple possible initializations in the gradient descent. By iteratively using CARTs to initialize the LF fitting and feeding the result of fitting process back into the CART training, we are able to get a very good final estimate of the true LF parameters. This final estimate is better than the first estimate for two reasons. Firstly, the prediction process using the CARTs has a smoothing effect on the parameter estimates which helps to remove outliers. Secondly, the use of multiple appropriate initializations greatly improves the chances of the gradient descent process finding the optimum parameter values.

![Figure 4: Iterative estimation of LF parameters](image)
eter Gen could model well. The second metric was the RMSE and Correlation of Fitting process works well. For example, if a bug in the code caused the parameters of every single pitch period in the database to be identical, then the ClusterGen predictions would be perfect all the time. However, we must not assume that optimizing the fitting metric alone is sufficient. The fitting process is worthless to us unless we can predict the values from the text. Therefore, our modeling technique must be able to do well under both metrics. Table 1 shows us that both the metrics indicate that our models get better as the iterations progress. Even in the case where the fitting RMSE starts to increase a little, the correlation still keeps improving.

<table>
<thead>
<tr>
<th>Iteration number</th>
<th>LSP Fitting</th>
<th>Prediction</th>
<th>RMSE</th>
<th>Corr</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>406.89</td>
<td>0.482</td>
<td>4.840</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>405.94</td>
<td>0.479</td>
<td>6.909</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>395.17</td>
<td>0.518</td>
<td>5.611</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>391.37</td>
<td>0.534</td>
<td>5.061</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>389.98</td>
<td>0.543</td>
<td>4.836</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>389.65</td>
<td>0.547</td>
<td>4.722</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>390.06</td>
<td>0.550</td>
<td>4.661</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>390.56</td>
<td>0.551</td>
<td>4.636</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. Optimizing the Synthesizer itself

While we use CMU’s ClusterGen Statistical Parametric Speech Synthesizer to model the LF parameters, there is nothing specific to ClusterGen that could not be done in any other parametric synthesizer such as HTS [29]. The only reason we used ClusterGen is because we are more familiar with our own system.

As our parameterization of speech in this model is with LSPs and LF parameters and the original phonetic and HMM state labeling is carried out with MFCCs (through EHMM [30]), we know that the labels will not be optimized for this alternative parameterization. We therefore make use of our move_label algorithm [31] which moves phoneme and HMM state boundaries based on how well the parameters at either side of the boundary can be predicted. This technique typically produces models better than the equivalent of doubling the data.

This technique has been used to optimize MCEP-based models but the LSP and LF have more varied magnitudes thus causes unintended weighting of the importance of each parameter. Thus we convert all the parameters to Z-scores (number of standard deviations from the mean), even though not all parameters are actually Gaussian. This allows a more equal optimization of the parameters. We measure LSP distortion as root mean squared difference between predicted LSPs and a held out set (in the un-Z-scored domain). We multiplied this by 1000 to give us a cosmetically nice number. For LF distortion, we again use root mean squared difference between prediction and held out data (non-Z-scored domain) but no scaling was necessary. We did 20 iterations each.

<table>
<thead>
<tr>
<th>Optimization</th>
<th>Pass</th>
<th>LSPD</th>
<th>LFD</th>
<th>Duration</th>
<th>RMSE</th>
<th>Corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>7.472</td>
<td>4.829</td>
<td>0.907</td>
<td>0.425</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSP</td>
<td>16</td>
<td>7.245</td>
<td>5.015</td>
<td>0.957</td>
<td>0.305</td>
<td></td>
</tr>
<tr>
<td>LF</td>
<td>20</td>
<td>7.961</td>
<td>3.702</td>
<td>0.961</td>
<td>0.310</td>
<td></td>
</tr>
<tr>
<td>LSP+LF</td>
<td>20</td>
<td>7.379</td>
<td>4.657</td>
<td>0.946</td>
<td>0.313</td>
<td></td>
</tr>
<tr>
<td>LSP+LF+Dur</td>
<td>5</td>
<td>7.418</td>
<td>4.777</td>
<td>0.878</td>
<td>0.479</td>
<td></td>
</tr>
</tbody>
</table>

Standard Deviation for the baseline system was 0.385 for LSPD and 0.589 for LFD. All other SDs are of similar magnitude.

The LSP+LF, which gave the lowest LFD, produced speech which we informally identified as being more smooth than the baseline, but the durations (in text to speech) were different enough to be less desirable. When we added the duration optimization constraint to our move_label optimization it converges quicker, but the spectral quality of the signal is not perceptably different from the baseline (though the durations are better).

8. Conclusions

This work shows a novel method to derive LF parameters from appropriately extracted residuals. Addressing excitation modeling is currently one of the more important issues in statistical parametric synthesis that should offer the brightness and naturalness that high quality (and costly-to-develop) unit selection synthesizers offer. Also more importantly this will allow us to address issues in modeling different speech styles efficiently. This work also continues our direction in investigating non-standard parameterizations of speech and complementing our Statisticl Phrase Accent Model for F0 [32] where the trajectory over accents is parameterized, and articulatory feature use in statistical parametric synthesis [33].
9. References


