ELEN 4810 Final Project Report
Software based FoF voice synthesizer

Shravan Siddartha Nagam (sn2597) and Anish King (awk2123)
Department of Electrical Engineering
Columbia University

***Abstract*—This report presents work done as part of final project for the course ELEN 4801. We present a FOF (Forme d'onde Formantique) based voice synthesizer. This technique was developed in 1980's to synthesize rich human voices. We demonstrate the technique by using Matlab. The script allows the user to input a sound of their choice, which is first discretized and then processed to be played back at the end.**

I. Introduction

The FOF technique was developed to synthesize human voice ([1]). The human vocal tract can be considered as a filter which takes impulses generated by the human body and spits out voice. This concept is illustrated in Fig. 1 shown below.

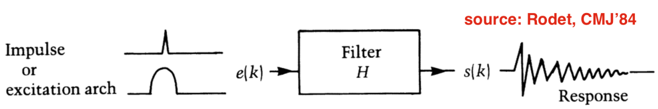

Fig. 1. Human voice synthesis concept

The filter characteristic is however highly complicated with high frequency selectivity. The bode plot of this filter transfer function has very sharp peaks at various frequencies. To emulate sound, we must first design this filter. To design such a complicated filter in a single stage is not advisable. Therefore the filter is designed as a parallel bank with multiple resonators. Each of this resonator resonates at a particular frequency and has a transfer function with a peak at that specified centre frequency. Now by clubbing multiple resonators at different centre frequencies, in parallel, we get a transfer function which has multiple peaks as shown in Fig. 2

With this in mind, the next step is to set the various parameters that define the frequency response of a resonator (formant). The parameters are the peak gain, bandwidth(width of the peak), roll off and centre frequency. Let us now define a time varying signal $s(t)$ which is the impulse response of the formant. The equation for $s(t)$ is given below

$$
\begin{aligned}
s(t) &= 0 \quad \text{for} \quad t \leq 0 \\
s(t) &= \frac{A}{2}(1-\cos(\beta t))\,e^{-\alpha t}\sin(\omega_c t+\phi) \;\; \text{for } 0 \leq t \leq \frac{\pi}{\beta} \\
s(t) &= A\,e^{-\alpha t}\sin(\omega_c t+\phi) \;\; \text{for} \quad t > \frac{\pi}{\beta}
\end{aligned}
$$

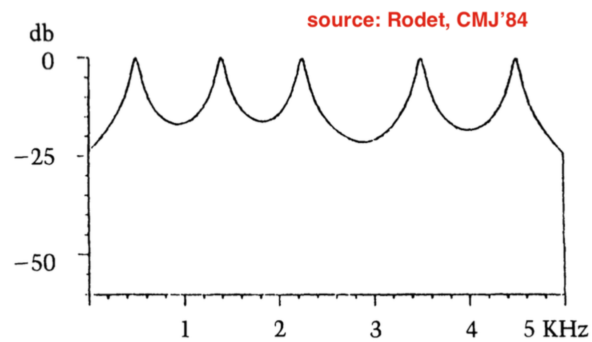

Fig. 2. Human voice synthesizer transfer function

The fourier transform of s(t) is given by

$$S(\omega) = A\,\frac{\beta^2}{2}\,\frac{e^{(\alpha+j\omega)\frac{\pi}{\beta}}+1)}{(\alpha+j\omega)\left[(\alpha+j\omega)^2+\beta^2\right]}$$

The magnitude response of $S(\omega)$ for varying values of β is shown in Fig. 3

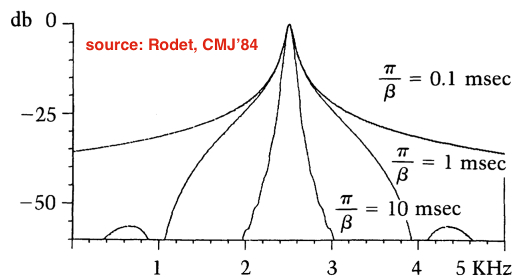

Fig. 3. Magnitude response of a formant

In [1] it has been shown that a human voice can be emulated by having 5 formants with parameters given in Fig. 4

The parameters for the 5 formants as described in [1] are shown in Fig. 5

II. Matlab Implementation of FoF synthesizer

A 5-parallel FoF synthesizer has been realized in Matlab. Here we are dealing in discrete signals. So care has to be taken while choosing the sampling frequency so as to avoid aliasing. The impulse response of the second formant listed in Fig. 5 is shown in Fig. 6 The impulse response of the parallel bank FoF synthesizer is shown in Fig. 7

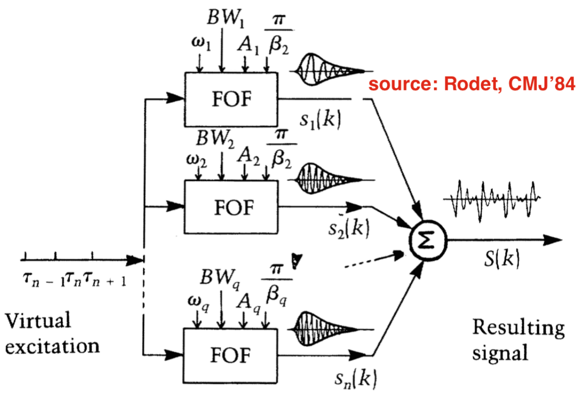

Fig. 4. FoF synthesizer

$\frac{\pi}{\beta_1}$ = .002 sec	F_1 = 260 Hz A_1 = .029 BW_1 = 70 Hz pu = .002 sec
$\frac{\pi}{\beta_2}$ = .0015 sec	F_2 = 1764 Hz A_2 = .021 BW_2 = 45 Hz pu = .0015 sec
$\frac{\pi}{\beta_3}$ = .0015 sec	F_3 = 2510 Hz A_3 = .0146 BW_3 = 80 Hz pu = .0015 sec
$\frac{\pi}{\beta_4}$ = .003 sec	F_4 = 3090 Hz A_4 = .011 BW_4 = 130 Hz pu = .003 sec
$\frac{\pi}{\beta_5}$ = .001 sec	F_5 = 3310 Hz A_5 = .00061 BW_5 = 150 Hz pu = .001 sec

source: Rodet, CMJ'84

Fig. 5. Parameters for the FoF synthesizer

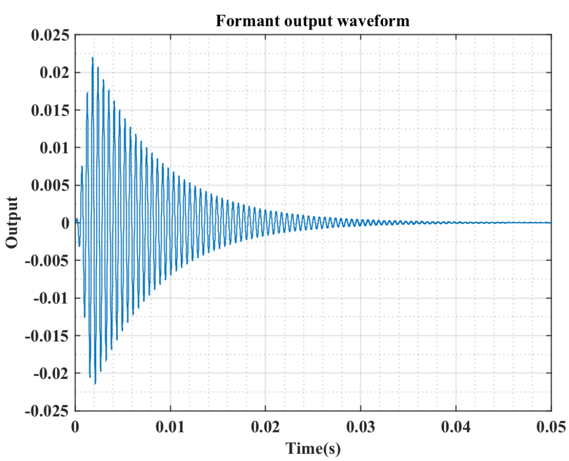

Fig. 6. Formant impulse response

III. Voice generation using the FoF synthesizer

Now we have the FoF synthesizer built in matlab. The next step is to produce meaningful voices. To do this we

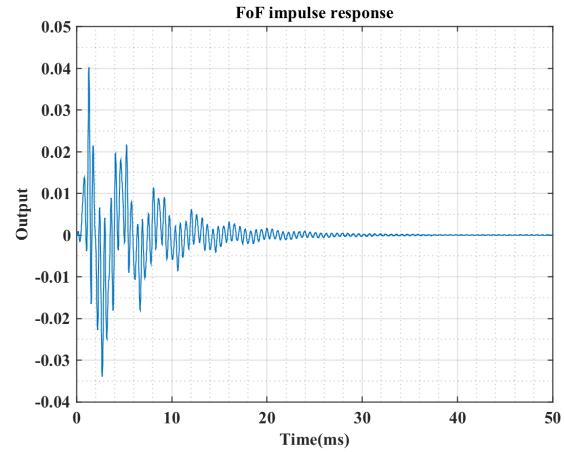

Fig. 7. FoF synthesizer impulse response

need to find the correct input for the synthesizer. To show the robustness of this system, we chose to give the user the option to input a voice. The input is then sampled and the correct impulse train for the synthesizer is deduced.

The algorithm to extract the impulse train heavily relies on the fact that the pitch of human voice is around 100 Hz. Because of this low pitch, the response due to an impulse dies out before the next impulse comes. This means, we can deduce the time stamps of the impulse train by just looking at the time stamps of the peaks in the input waveform. This sparsity of the input waveform allows us to easily extract the impulse train.

The input waveform, impulse train (this should be a stem plot but it is plotted as a line as there are too many points and the stem plot gets messy) and output waveform are shown in Fig. 9

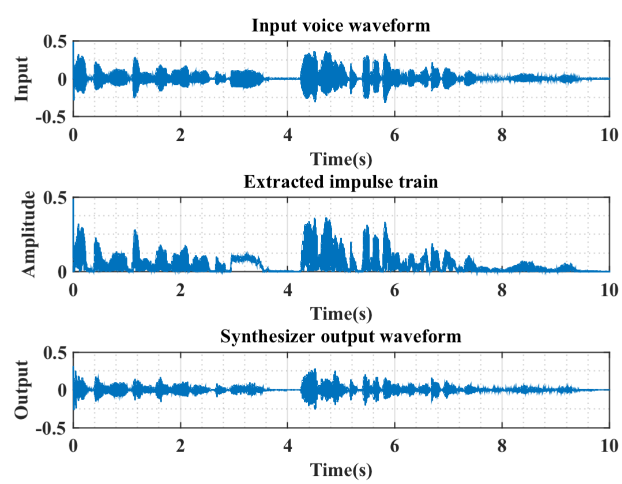

Fig. 8. Input output waveforms of the Fof synthesizer

IV. Running the code

Please open all files in matlab and navigate to to fof.m file. After executing the script, the CW will show START SPEAKING in the command line. At this point please start speaking so that your voice can be recorded. The duration of recording time can be set in the fof.m file, specifically line 17 by setting the parameter record time. After the the CW shows END OF RECORDING, you may stop. The code will take about 20 seconds to execute. After the execution, the synthesizers output is played.

V. Acknowledgments

We would like to thank Prof. John Wright and TA Yenson Lau for valuable discussions and comments.

References

[1] X. Rodet, "Time-domain formant-wave-function synthesis," *Computer Music Journal*, vol. 8, pp. 9–14, Autumn 1984.