Preparation of Report for Speech Processing, (ELC 470)

Vowel Synthesis Project

Daniel Funke*

Date Submitted: 4/18/2019

* Department of Electrical and Computer Engineering, The College of New Jersey, (*email: funked1@tcnj.edu*)

Abstract:

The generation of human speech is a well-understood process that lends itself to replication through electrical means. This exercise examines a method for vowel synthesis based on the acoustic theory of speech production. Here, MATLAB is used to generate and modulate a pulse train with the intention to synthesize English vowel sounds. The process described here did elicit a waveform whose frequency characteristics matched those of an actual speech sample. However, when this waveform is used as a speaker input signal, the resultant sound in no way resembled human speech.

Introduction:

Background and Theory:

Human beings produce speech by altering the acoustic properties of air expelled from their lungs. Air passes through the trachea where it is transformed into a series of pulses by the vocal folds. This pulse train, also known as a glottal pulse train, then passes through the remainder of the vocal tract where its frequency characteristics are altered by the physical configuration of the speaker's mouth, tongue, nose, and vellum. In this way, the speaker's vocal tract acts as a temporary linear, time-invariant filter as it produces individual sounds.

Human speech can be synthesized by creating a system that mimics the action of the vocal tract. The acoustic properties of an individual's speech can be extracted using quantitative methods and used as system specifications to define filter characteristics. Passing a glottal pulse through this filter produces an output signal with characteristics similar to the measured speech fragment. This laboratory exercise examined this process for the purpose of synthesizing English vowels.

Objective:

The purpose of this exercise is to demonstrate the feasibility of using a linear, time-invariant filter to modulate a simulated glottal pulse train to synthesize individual English vowels. Both the generation and modulation of a glottal pulse train is performed using MATLAB. The modulation of the input signal is accomplished using a sixth-order transfer function, also developed using MATLAB.

Materials and Methods:

Materials:

1) MATLAB R2017b

Nomenclature:

Variable Name	*Description*
F	Resonance Frequency
B	Bandwidth of Resonant Frequency
F_s	Sampling Frequency
ω_0	Normalized Angular Frequency
r	Radius of the Poles

Methods:

The first three formant frequencies for the traditional set of spoken English vowels were previously collected and are listed in ***Table 1***. A second order transfer function, the general form of which is shown in Equation 4, was developed for each formant frequency. The bandwidth for each formant frequency was limited to 100Hz. Given the second order transfer function for each of the three formant frequencies of a vowel, a sixth-order transfer function was generated using Equation 5. All these calculations were performed using MATLAB.

Table 1: Experimentally-Collected Vowel Formant Frequencies

Vowel	**First Formant Frequency (Hz)**	**Second Formant Frequency (Hz)**	**Third Formant Frequency (Hz)**
A	530	2150	3000
E	150	2690	3450
I	900	2900	5000
O	690	3250	5500
U	315	2350	3300

$$a_1 = -2r\cos\omega_0$$ **Equation 1**

$$a_2 = r^2$$ **Equation 2**

$$r = e^{-\frac{\pi B}{F_s}}$$ **Equation 3**

$$H_i(z) = \frac{1}{\left(1 - re_{0_0}^{\,j\omega}\right)\left(1 - re_{0_0}^{\,-j\omega}\right)} = \frac{1}{1 - 2r\cos\omega_0 z^{-1} + r^2 z^{-1}} = \frac{1}{1 - a_1 z^{-1} + a_2 z^{-1}}$$ **Equation 4**

$$H(z) = H_1(z)H_2(z)H_3(z)$$ **Equation 5**

A glottal pulse was then generated to serve as an input signal for this transfer function. This was performed using the GenGP function described in the appendix. This function takes a sampling frequency and a fundamental pitch frequency as input parameters. Using these inputs, the GenGP function produces a simple pulse train with an amplitude of 1 and a frequency equal to the pitch frequency. Random noise is then added to the signal to simulate a real-life glottal pulse train.

Finally, the MATLAB filter function was used to synthesize the vowel signal. This function takes the numerator and denominator coefficients of the sixth-order transfer function and the glottal pulse signal as inputs and produces a new signal vector as output. A fast Fourier Transform (FFT) was performed on this output, and the positive portion was plotted using MATLAB. The resultant signals are shown in Figures 1 through 5.

Results:

Figures 1 through 5 show the resultant signal produced after modulating a glottal pulse with a sixth-order transfer function. Each signal was modulated using a transfer function developed using that vowel’s observed formant frequencies.

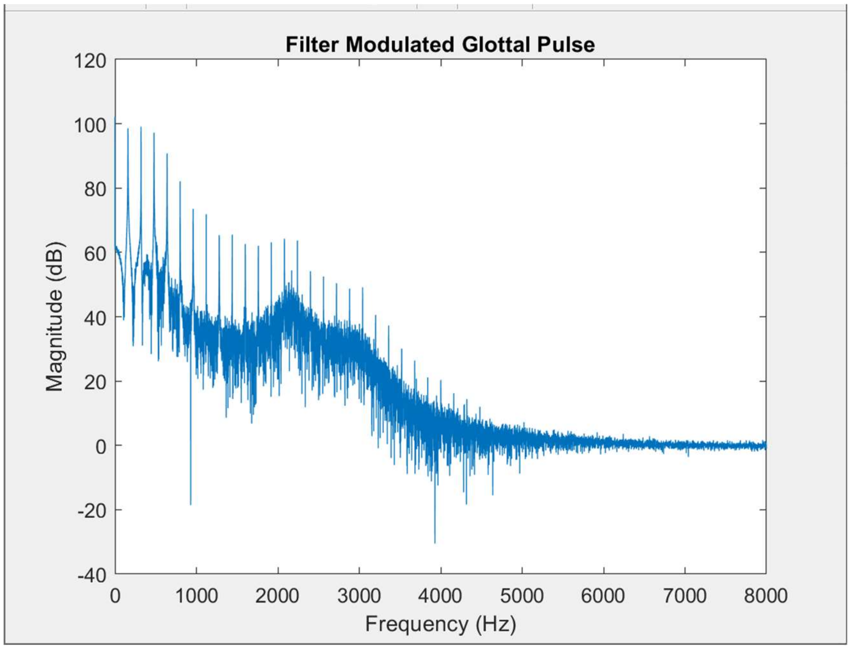

Figure 1: Filter Modulated Glottal Pulse for the Vowel "A"

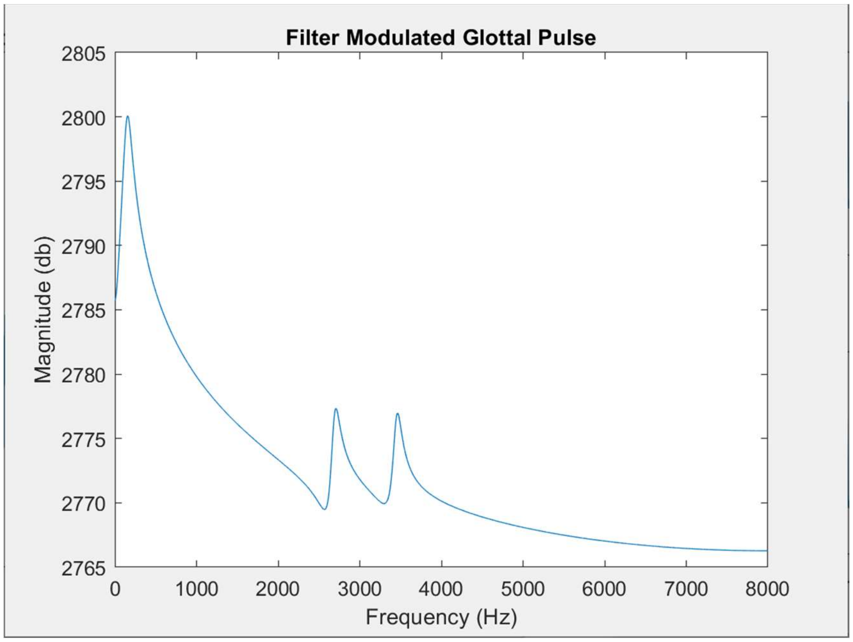

Figure 2: Filter Modulated Glottal Pulse for the Vowel "E"

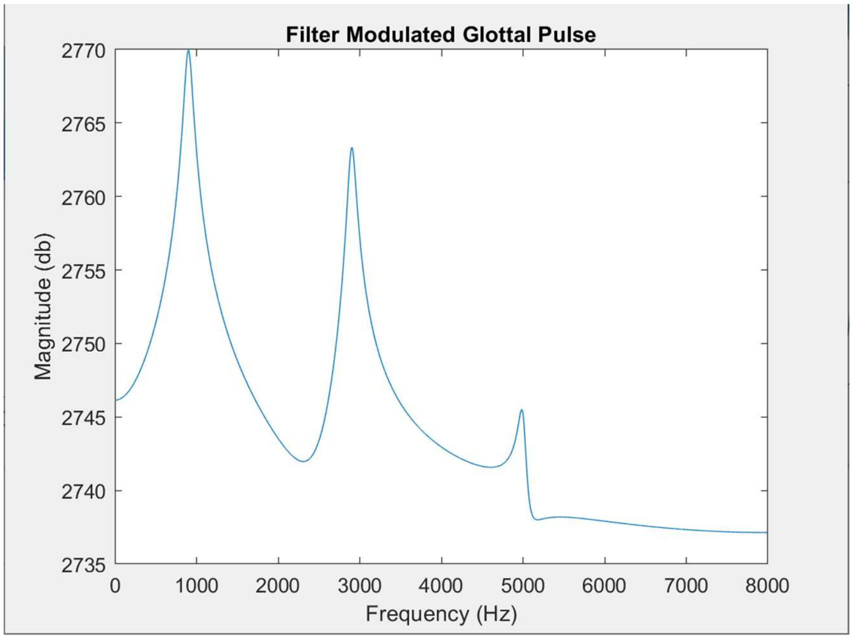

Figure 3: Filter Modulated Glottal Pulse for the Vowel "I"

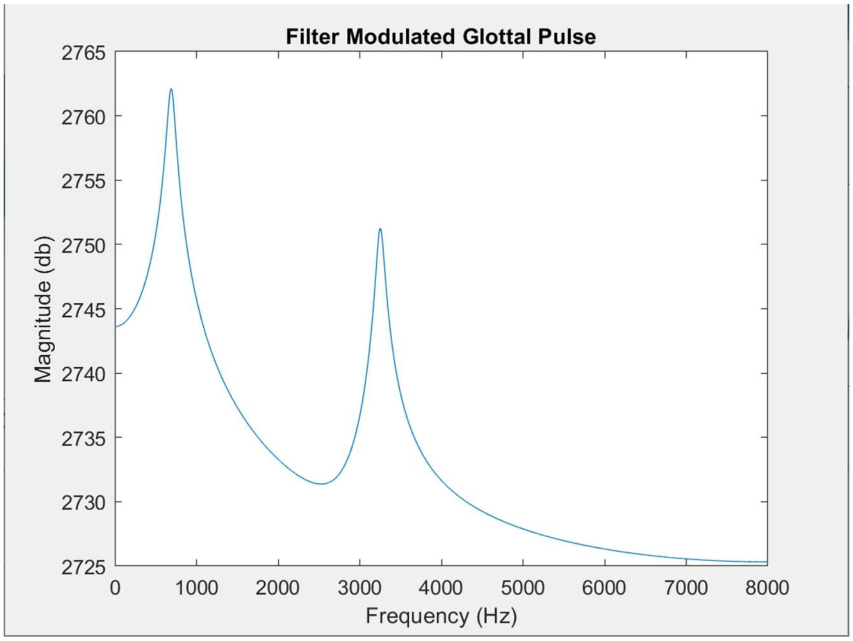

Figure 4: Filter Modulated Glottal Pulse for the Vowel "O"

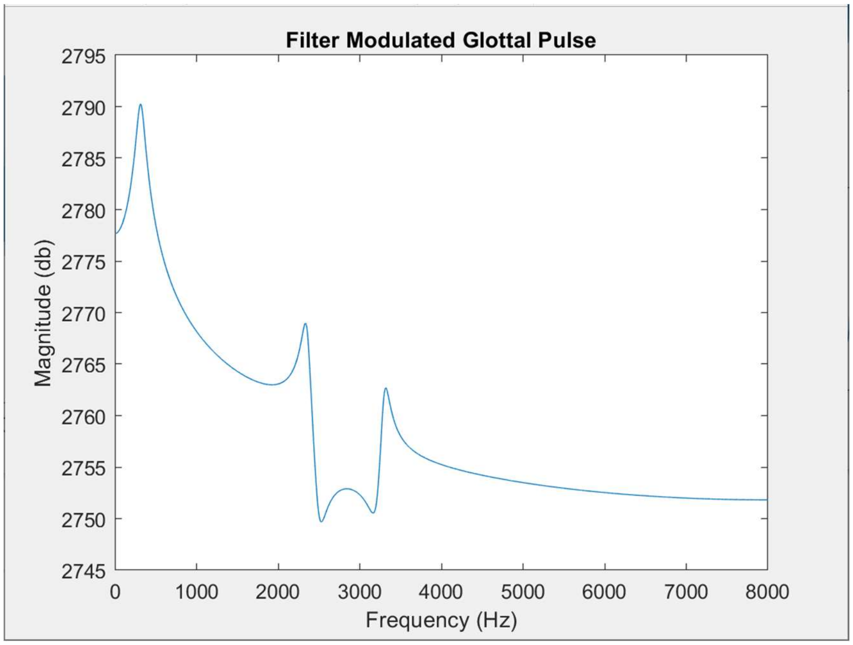

Figure 5: Filter Modulated Glottal Pulse for the Vowel "U"

Discussion:

This method of synthesizing spoken vowels appeared to produce signals whose frequency characteristics closely resembled actual values. Figures 1 through 5 show these output waveforms. Peaks can be readily observed at the appropriate formant frequencies for the respective vowel, which are listed in Table 1. However, the actual sound produced by computer speakers when using these signals as an input source in no way resembled actual human speech.

Additionally, there appeared to be a significant departure between the results obtained for the vowel “A” and the rest of the vowel set. For instance, Figure 1 shows significant noise contamination in frequency plot of the vowel “A,” whereas the plots of the remaining vowels appear smooth and free of noise. However, the amplitude of this plot is much more practical than the amplitudes of the signals depicted in Figures 2 through 5. The maximum amplitude of the synthesized “A” signal is below 100 dB, while the remaining vowels have peak amplitudes close to 3000 dB. These amplitudes were influenced by the filter’s bandwidth. Decreasing the bandwidth reduced the output signal’s amplitude, while increasing it slightly pushed the signal’s amplitude toward infinity. This likely indicates that there was a problem with the synthesis algorithm employed.

Conclusion:

The process utilized during this exercise produced a waveform whose frequency characteristics closely matched those of an actual speech sample. However, when this waveform is used as a speaker input signal, the resultant sound in no way resembled human speech. A better methodology needs to be employed in order to synthesize an output signal that will produce comprehensible speech.

Appendix:

MATLAB Code:

Normalize Frequency Function

```
function y = NormFreq(f, Fs)
    y = (2 * pi * f) / Fs;
end
```

Generate Glottal Pulse Function

```
function [glottal] = GenGP(fs, f0)
    % Glottal Pulse Limits
    N1 = round(0.0025 * fs);
    N3 = round(0.003 * fs);
    N2 = N3 - N1;

    % Create Glottal Pulse
    glot = zeros(1, N3);
    for i = 1:N1
        glot(i) = 0.5 * (1 - cos((pi * i)/N1));
    end
    for i = N1 + 1 : N3
        glot(i) = cos(pi * (i - N1) / (2 * N2));
    end

    % zero pad glot signal
    pad= zeros(size(glot));
    glot= cat(2, pad, cat(2, glot, pad));

    % Creates a Pulse Train
    pulse= zeros(1, 1*fs);
    for i= 1: length(pulse)
        if mod(i, f0) == 0
            pulse(i)= 1;
        end
    end

    % Creates complete glottal signal
    glottal= conv(glot, pulse);
    noise= randn(size(glottal));
    noise= noise ./ (10*max(abs(noise)));
    glottal= glottal + noise;
end
```

Find Filter Coefficient Function

```
function [a1, a2] = Findfc(form_f, Fs, HPF, LPF)
    omega = NormFreq(form_f, Fs);
    Bw = LPF - HPF;
    k = (-pi * Bw) / Fs;
    r = exp(k);
    a1 = (-2 * r * cos(omega));
    a2 = (r * r);
end
```

Vowel Synthesis Program

```
F0 = 100;           % Fundamental pitch frequency
Fs = 16000;         % Sampling frequency (samp/sec)

ff1  = 530;         % first formant frequency
HPF1 = 450;         % high-pass cutoff frequency (Hz)
LPF1 = 610;         % low-pass cutoff frequency (Hz)

ff2  = 2150;        % second formant frequency
HPF2 = 2050;        % high-pass cutoff frequency (Hz)
LPF2 = 2260;        % low-pass cutoff frequency (Hz)

ff3  = 3000;        % third formant frequency
HPF3 = 2800;        % high-pass cutoff frequency (Hz)
LPF3 = 3200;        % low-pass cutoff frequency (Hz)

% Calculate Filter coefficients
[a11, a12] = Findfc(ff1, Fs, HPF1, LPF1);
[a21, a22] = Findfc(ff2, Fs, HPF2, LPF2);
[a31, a32] = Findfc(ff3, Fs, HPF3, LPF3);

% Generate 6th order transfer function
sos = [1 0 0 1, a11 a12; 1 0 0 1, a21 a22; 1 0 0 1, a31 a32];
[b, a] = sos2tf(sos)

% Generate glottal pulse input signal
gp = GenGP(Fs, F0);

% Apply filter to input signal
output = filter(b, a, gp)
```

```
soundsc(output, Fs)

%%% PLOTS %%%

% Plot Glottal Pulse
figure (1);
subplot 211
t= linspace(0, length(gp)/Fs, length(gp));
plot(t, gp)
xlabel('Time (sec)')
ylabel('Amplitude')
title('Glottal Pulse    ')

% Plot Glottal Pulse Train
subplot 212
t= linspace(0, ((1/F0) * 23), length(gp(Fs/F0: ((1/F0) *
23)*Fs)));
plot(t, gp(Fs/F0: ((1/F0) * 23)*Fs))
xlabel('Time (sec)')
ylabel('Amplitude')
title('Glottal Pulse Train    ')

% Plot Complete Glottal Pulse
figure(2)
t= linspace(0, Fs/2, length(gp)/2);
G= (abs(fft(gp)));
input=db(G(1: length(G)/2));
plot(t,input)
xlabel('Frequency (Hz)')
ylabel('Magnitude dB')
title('Complete Glottal Pulse    ')

% Plot output signal
figure(3)
t = linspace(0, Fs/2, length(output)/2);
o2 = (abs(fft(output)));
o3 = db(o2(1: length(output)/2));
plot(t, o3)
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
title('Filter Modulated Glottal Pulse')
```