Lab 2
Analyzing Sinusoids with Varying Frequency and Convolution in Smoothing

Aneesh Malhotra
G00844135

January 24, 2017

1 Introduction

First Objective

The first objective of this lab was to analyze sinusoids with varying instantaneous frequency. This was done by creating a function $\theta(t)$ such that $\frac{d\theta}{dt} = \omega(t)$. Using this we created signals that varied in frequency both linearly and quadratically, and confirmed our results using a spectrogram.

Second Objective

The second goal of this lab was to analyze how the convolution operation can smooth out signals. We started with a simple DC signal and saw how through several convolutions the signal became closer to a Gaussian distribution, a smooth function. We did this again with a noise signal and obtained the same results.

2 Main Body

2.1 Constant Signal

We began by defining our sample frequency as $f_s = 14400$ and our fundamental frequency $f_0 = 440$. Our time vector t consisted of time samples at the sample period $T_s = \frac{1}{f_s}$ on the interval $[0, 1]$. We obtained the following plot of the first 200 samples of a sinusoid with frequency f_0.

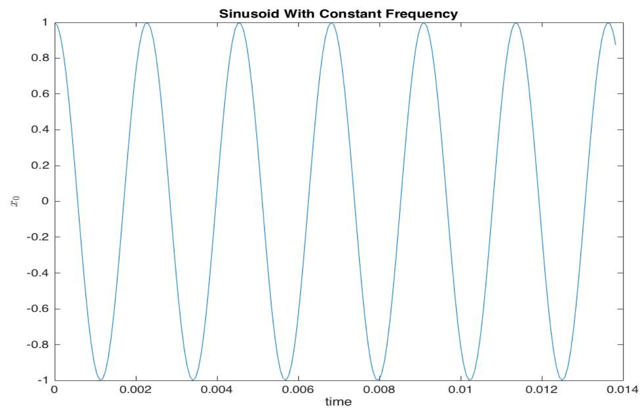

Figure 1: First 200 Samples of $\cos(2\pi 440t)$

2.2 Frequency-Varying Signals

Next we created signals with varying frequency. In order to do this, we needed to identify a function $\theta(t)$ such that its derivative is the desired function of angular frequency. We initially created a signal that starts at 4400Hz and ends at 5500Hz in the duration of one second. In order to do this, we calculate the the angular frequency function $\omega(t)$.

$$\frac{\Delta y}{\Delta t} = \frac{5500 - 4400}{1} = 1100$$
$$\omega(t) = 2\pi(4400 + 1100t)$$

Now that we have $\omega(t)$ we can integrate to obtain $\theta(\tau)$.

$$\theta(t) = 2\pi \int_0^t 4400 + 1100\tau \, d\tau.$$

Therefore

$$\theta(t) = 2\pi(440t + 1100\frac{t^2}{2}).$$

This calculation was repeated for signals with linearly decreasing, triangular, and quadratically increasing frequencies. We obtained the following spectrograms:

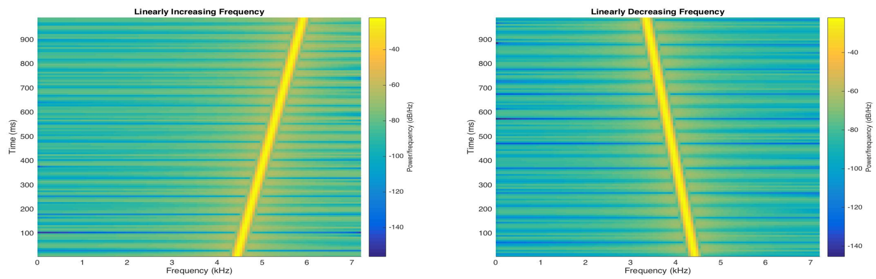

Figure 2: Linearly Increasing and Decreasing Frequencies

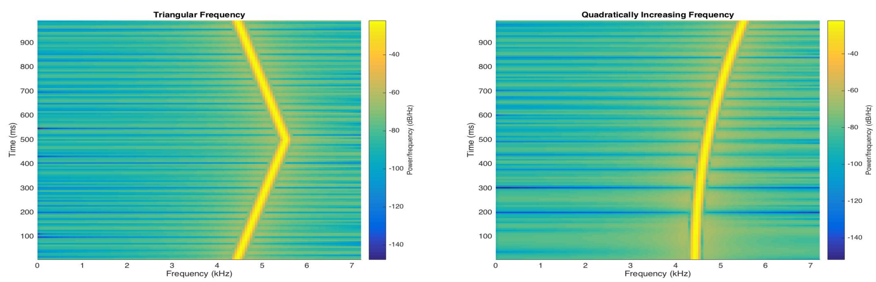

Figure 3: Triangular and Quadratic Frequencies

2.3 Convolution and Smoothing

In this section we looked at the convolution operation and how it tends to smooth functions when applied continuously. We began with a square signal $p_0(t)$ that takes the value of 1 when $0 \leq t \leq 1$ and is 0 otherwise. We applied the convolution operation such that $p_1(t) = p_0(t) * p_1(t)$, $p_2(t) = p_1(t) * p_1(t)$, and so forth. Each of the outputs was normalized by the maximum value of the signal and scaled in time so that we can analyze each signal on the same time axis.

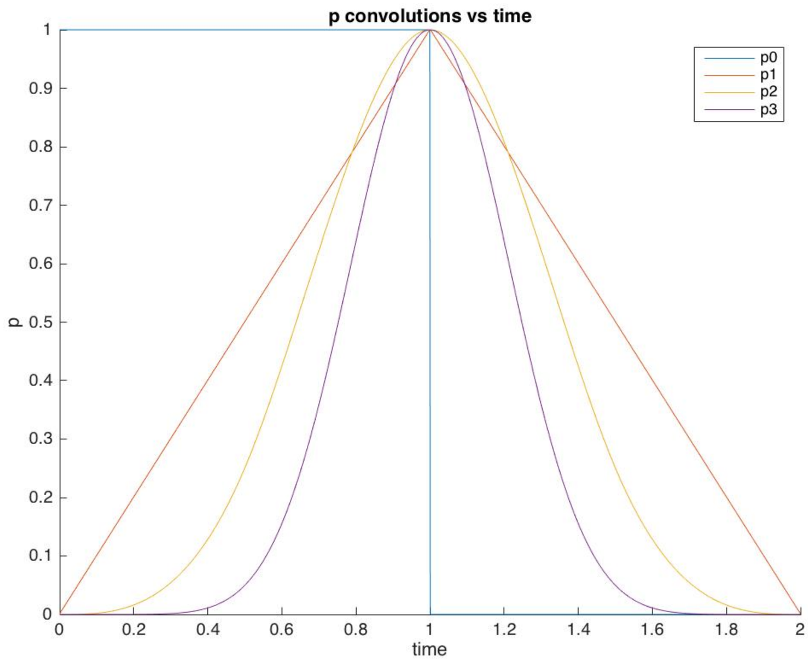

Figure 4: Plots of a Square Signal With Multiple Convolutions

This was repeated for a noise signal obtained using the command rand(2000,1) and the following plot was obtained

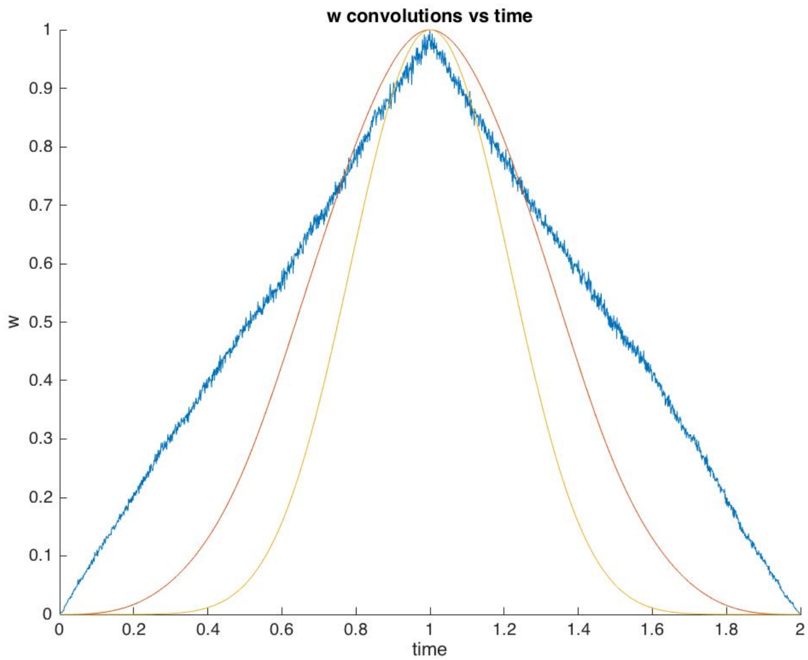

Figure 5: Plots of a Noise Signal With Multiple Convolutions

We see that in both cases, the signal tends to become more bell-shaped. We confirm this by seeing the traditional bell curve given by the Gaussian function

$$g(t) = e^{-\frac{1}{2}(\frac{t-1}{a}^{2})}.$$

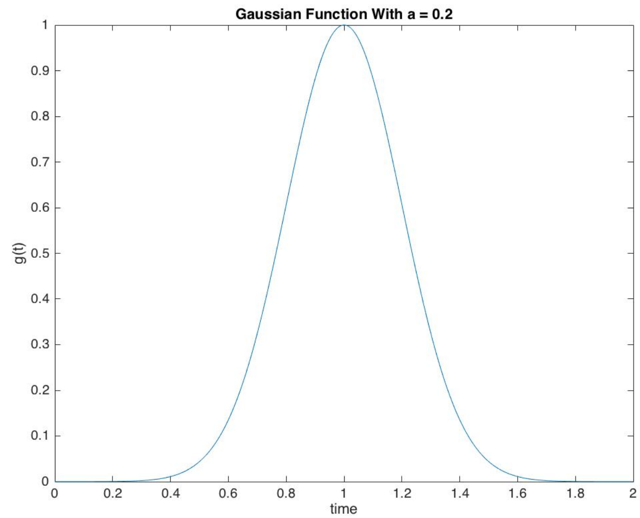

Figure 6: Plots of a Gaussian Distribution

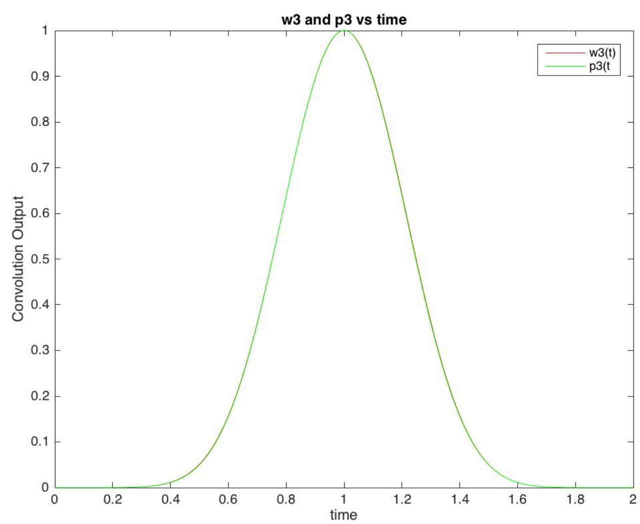

Figure 7: Plots of p_3 and w_3

We see that the plot of the Gaussian distribution is very similar to those of p_3 and w_3.

3 Conclusion

3.1 Sinusoids with Varying Frequency

In this section we plotted and analyzed sinusoids whose frequencies varied with time. In order to do this we algebraically solved for a function that describes the angular frequency instantaneously in time, and integrated to find the input argument to the cosine function. I chose to integrate from a starting value of $t = 0$, although we could technically choose any constant to add, as it would only affect the phase of the signal and not the instantaneous frequency. The most difficult frequency function to plot was the triangular as it required splitting the time domain into an increasing and decreasing part and combining them (see Appendix for code). We also listened to each of these signals to see the qualitative effect of varying the frequency. As expected, the linearly increasing and decreasing signals were chirps whose pitches varied. The triangular frequency did as expected and increased in pitch and then decreased back to its original pitch. The quadratic signal was interesting as the pitch started off having a very low rate of change and obtained a higher rate in time.

3.2 Convolution and Smoothing

In this section we applied the convolution operation several time to both a square wave and a noise function. Both, however, resulted in a Gaussian distribution. I explain this phenomenon using probability. The expected value of the sum of two probability density functions is the convolution between them. By the central limit theorem, if we look at each distribution as an experiment, the sum of the data should approach a Gaussian distribution. This is what we saw with these signals. Both signals started off as being very sharp, and contained discontinuities, but approach a Gaussian when applied to an LTI system. This is interesting because we are taking a function with discontinuities and making it continuously differentiable for all values in the domain.

4 Appendix

4.1 Part 2 Code

```
fs = 14400;
Ts = 1/fs;
f0 = 440;
t = 0:Ts:1;
x_0 = cos(2*pi*f0*t);
figure
plot(t(1:200),x_0(1:200))
xlabel('time')
ylabel('$x_0$','Interpreter','LaTex')
title('Sinusoid With Constant Frequency')
```

```
x_1 = cos(2*pi*(750*t.^2 + 4400*t));

%soundsc(x_1,fs)

x_2 = cos(2*pi*(-550*t.^2 + 4400*t));

t1 = t(:); %defines a new time vector for x_3
t1(7201:end) = 0;
t2 = t(:);
t2(1:7200) = 0;

x_3 = cos(2*pi*(4400.*t1 + 1100*t1.^2) + 2*pi*((6600).*t2 - 1100*t2.^2));



% Quadratic
x_4 = cos(2*pi*(4400*t + 1100*t.^3/3));




figure

spectrogram(x_1, 256,158,256,fs)
title('Linearly Increasing Frequency')


figure
spectrogram(x_2, 256,158,256,fs)
title('Linearly Decreasing Frequency')

figure
spectrogram(x_3, 256,158,256,fs)
title('Triangular Frequency')

figure
spectrogram(x_4, 256,158,256,fs)
title('Quadratically Increasing Frequency')
```

4.2 Part 3 Code

```
Lab 2 Part 3

t = linspace(0,2,1000); %time vector
tp0 = linspace(0,2,2000);

p0 = [ones(1,1000), zeros(1,1000)]; %1000 ones then 1000 zeros
```

```
p1 = conv(p0,p0); % convolution
p1 = downsample(p1,2); %scaled time
p1_max = max(p1); %max value
p1 = p1/p1_max; %scaled output

p2 = conv(p1,p1);
p2 = downsample(p2,2);
p2_max = max(p2);
p2 = p2/p2_max;

p3 = conv(p2,p2);
p3 = downsample(p3,2);
p3_max = max(p3);
p3 = p3/p3_max;

figure;
hold on;
plot(tp0,p0)
xlabel('time')
ylabel('p')
title('p convolutions vs time')

plot(t,p1(1:1000))

plot(t,p2(1:1000))

plot(t,p3(1:1000))

legend('p0','p1','p2','p3')
hold off;
%part g

t = linspace(0,2,2000); %time vector

w0 = rand(1,2000);
soundsc(w0,2000); % sounds like wind

w1 = conv(w0,w0);
w1 = downsample(w1,2);
w1_max = max(w1);
w1 = w1 / w1_max;

w2 = conv(w1,w1);
w2 = downsample(w2,2);
w2_max = max(w2);
w2 = w2/w2_max;

w3 = conv(w2,w2);
w3 = downsample(w3,2);
```

```
w3_max = max(w3);
w3 = w3/w3_max;

figure
hold on

plot(t,w1)
xlabel('time')
ylabel('w')
title('w convolutions vs time')

plot(t,w2)

plot(t,w3)


hold off;

%part j
tp = linspace(0,2,1000); % Scales t so that p is nonzero
figure
plot(t,w3,'r',tp,p3(1:1000),'g')
xlabel('time')
ylabel('Convolution Output')
title('w3 and p3 vs time')
legend('w3(t)', 'p3(t')

%part l
a = 0.2;
g = @(t) exp(-.5*(((t-1)/a).^2));
figure
plot(t,g(t));
title('Gaussian Function With a = 0.2')
xlabel('time')
ylabel('g(t)')
% plot of Gaussian looks the same as w3 and similar to w3 and p3
% p3 is shifted, but looks identical to w3
```

5 References

Lathi, B. P. Linear Systems and Signals. New York: Oxford UP, 2005. Print.