WAVEMAKING

From Discrete Samples

APRIL 29, 2014

DAVID HOUSTON

ENGR 267

Abstract

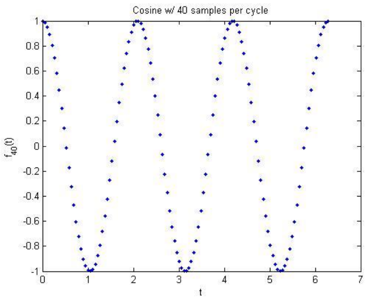
This lab aims at taking and utilizing the basic fundamentals of MATLAB by manipulating DSP. We exploited the various techniques into the MATLAB syntax in order to manipulated partials of previously generated waves and get the relative outputs. After achieving a mastery of these concepts we went about designing a square wave using the partials from the harmonics of a particular sinusoidal wave. Overall this lab produced an acute knowledge of the syntax as well as the operations that MATLAB can be used for.

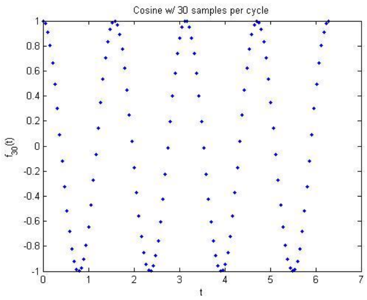
PROCEdURE

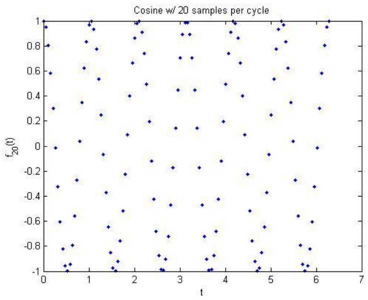
Task 1

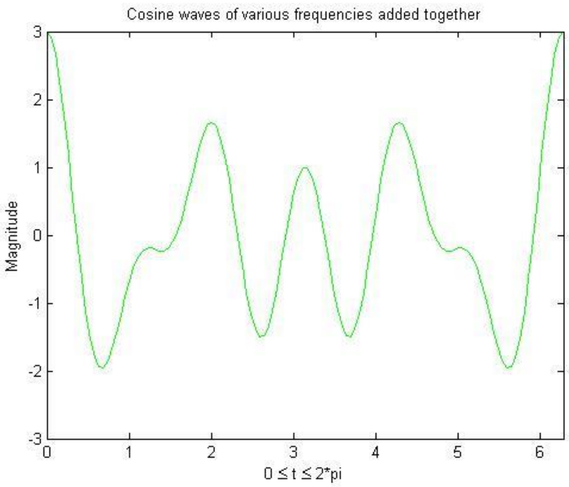
It was first observed to take a cosine function and generate it into three different waves, of which had periods of 40, 30, and 20 samples. It was then proposed to sum them together and display the output of all four equations.

```
t = linspace(0,2*pi,120);

f40 = cos(3*t);
f30 = cos(4*t);
f20 = cos(6*t);

fadd = f40 + f30 + f20;

plot(t,f40,'.')
xlabel ('t')
ylabel ('f_4_0(t)')
title ('Cosine w/ 40 samples per cycle')

plot(t,f30,'.')
xlabel ('t')
ylabel ('f_3_0(t)')
title ('Cosine w/ 30 samples per cycle')

plot(t,f20,'.')
xlabel ('t')
ylabel ('f_2_0(t)')
title ('Cosine w/ 20 samples per cycle')

plot(t,fadd,'g-')
xlabel ('0 \leq t \leq 2*pi')
ylabel ('Magnitude')
title ('Cosine waves of various frequencies added together')
axis([0 2*pi -3 3])
```





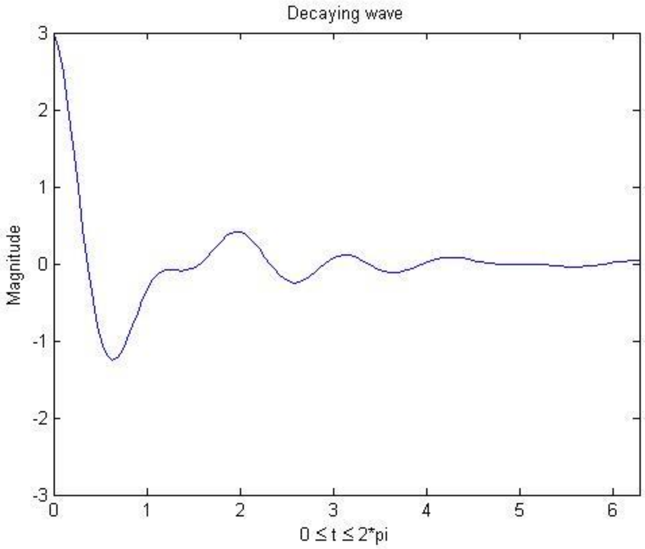




Task 2

After complete a magnificently mixed wave, it was then proposed by marketing that a decay function be implemented in the output of the mixed wave. The following formula was devised so as to give the half-life of this decay a value of 20 samples.

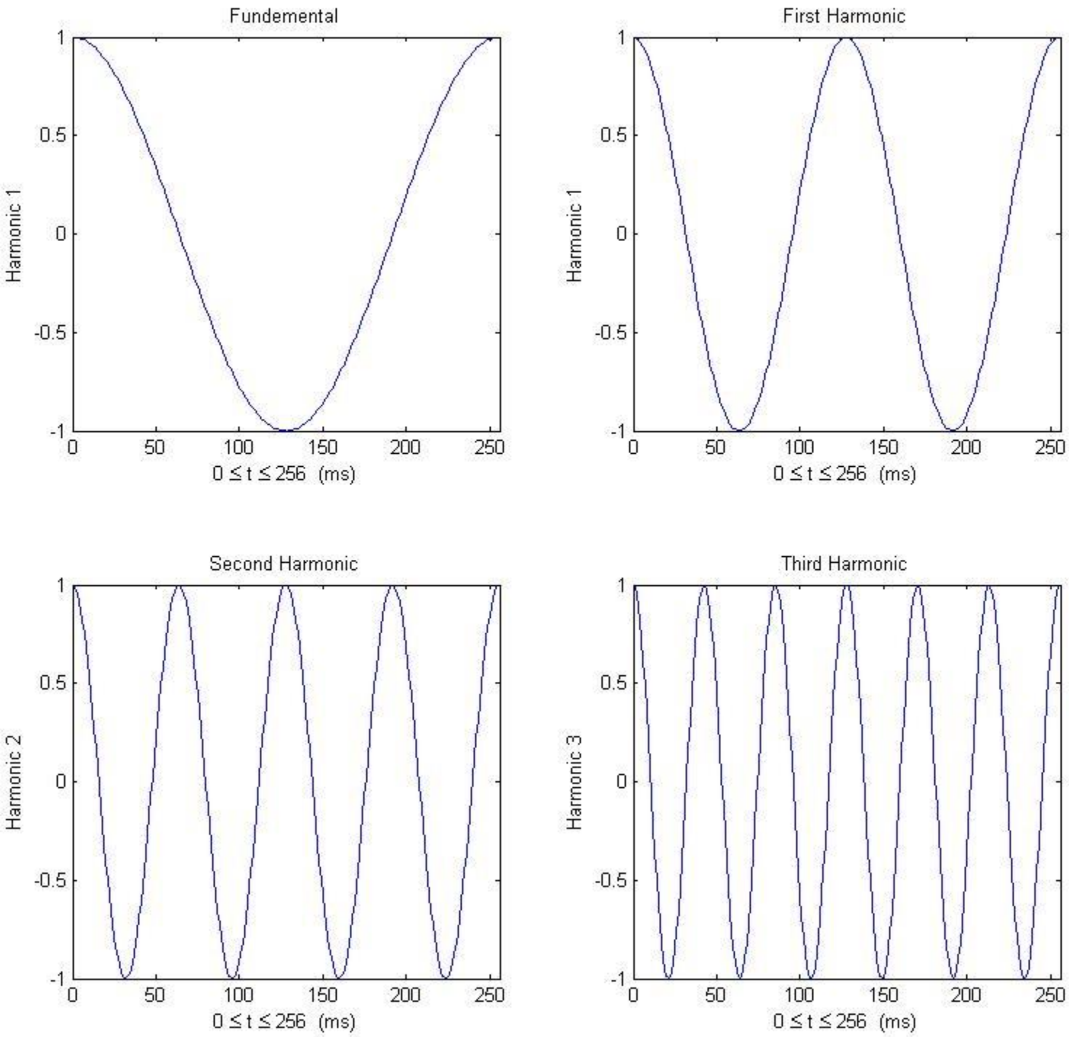
```
fdecay = (fadd).*1/(2^(n/20));

plot(t,fdecay,'b-')
xlabel ('0 \leq t \leq 2*pi')
ylabel ('Magnitude')
title ('Decaying wave')
axis([0 2*pi -3 3])
```



Tasks 3 & 4

Once again our fearless efforts, never taken for granted, are called upon to create another table of cosine values from zero to τ and display them dramatically. From that table, marketing, once again, decided that it would be in the best interest of the company to generate the next three harmonics of the original wave.

```
n = 1:256;
cos_val(n) = cos(2*pi*n/256);

subplot(221)
plot(cos_val)
title('Fundemental')
xlabel('0 \leq t \leq 256  (ms)')
ylabel('Harmonic 1')
axis([0 256 -1 1])

subplot(222)
plot(wave2)
xlabel('0 \leq t \leq 256  (ms)')
ylabel('Harmonic 1')
title('First Harmonic')
axis([0 256 -1 1])

subplot(223)
plot(wave3)
xlabel('0 \leq t \leq 256  (ms)')
ylabel('Harmonic 2')
title('Second Harmonic')
axis([0 256 -1 1])

subplot(224)
plot(wave4)
xlabel('0 \leq t \leq 256  (ms)')
ylabel('Harmonic 3')
title('Third Harmonic')
axis([0 256 -1 1])
```

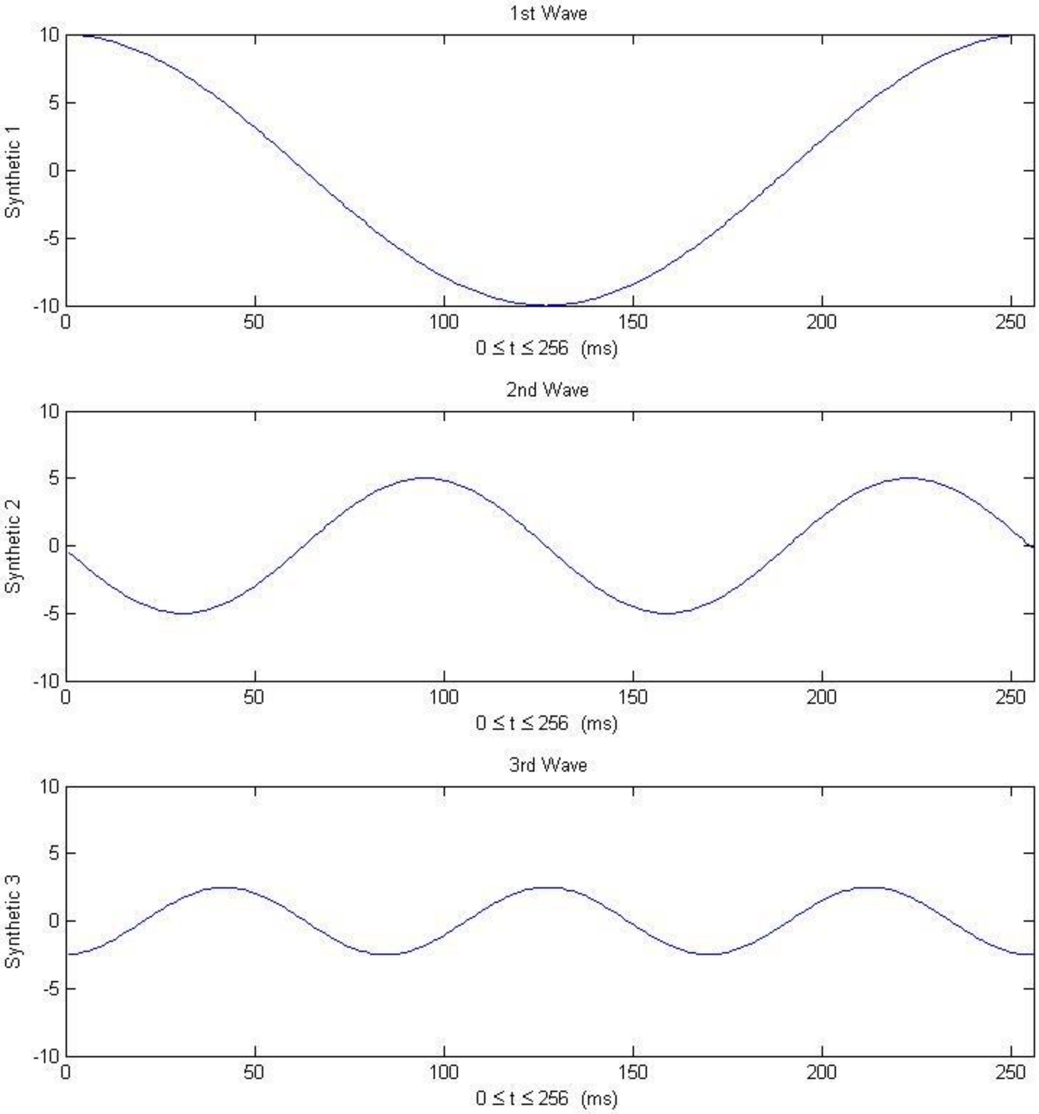


Tasks 6, 7, & 8

Thanks to marketing we were able to devise a plan to take the original cosine table and, from that, generate a specified wave using the synthesis equation.

```
synth_1(n) = 10*cos_val(mod(n,256)+1);
synth_2(n) = 0.5*synth_1(mod((2*n+64),256)+1);
synth_3(n) = 0.25*synth_1(mod((3*n+128),256)+1);

subplot(311)
plot(n,synth_1)
xlabel('0 \leq t \leq 256  (ms)')
ylabel('Synthetic 1')
title('1st Wave')
axis([0 256 -10 10])
```

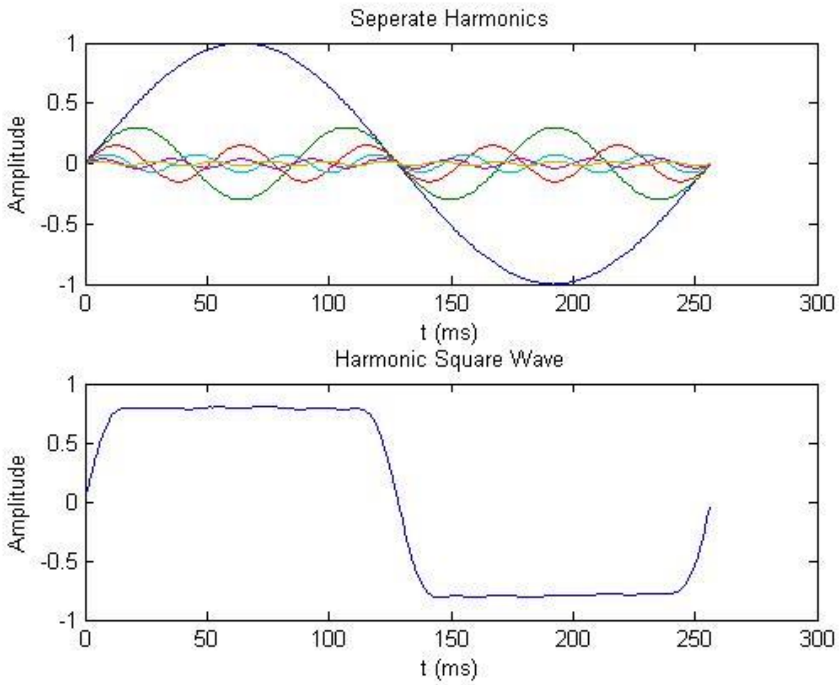
```
subplot(312)
plot(n,synth_2)
xlabel('0 \leq t \leq 256  (ms)')
ylabel('Synthetic 2')
title('2nd Wave')
axis([0 256 -10 10])

subplot(313)
plot(n,synth_3)
xlabel('0 \leq t \leq 256  (ms)')
ylabel('Synthetic 3')
title('3rd Wave')
axis([0 256 -10 10])
```



Task 9

After gaining an understanding of the use of the synthesis equations, we went about utilizing there magical powers to generate a square wave using the first five odd harmonics of a sinusoid.

```
sin_val(n) = sin(2*pi*n/256);

square_1(n) = sin_val(mod(n,257));
square_3(n) = .3*square_1(mod(3*n,257));
square_5(n) = 0.15*square_1(mod(5*n,257));
square_7(n) = 0.075*square_1(mod(7*n,257));
square_9(n) = .0375*square_1(mod(9*n,257));
square_11(n) = 0.01875*square_1(mod(11*n,257));

subplot(211)
plot(n,square_1,n,square_3,n,square_5,n,square_7,n,square_9,n,square_11)
xlabel('t (ms)')
ylabel('Amplitude')
title ('Seperate Harmonics')

subplot(212)
harmonic = square_1 + square_3 + square_5 + square_7 + square_9 + square_11;
plot(n,harmonic)
xlabel('t (ms)')
ylabel('Amplitude')
title ('Harmonic Square Wave')
```



CONCLUSION

This lab introduced the basic concepts of MATLAB by way of wave manipulation. For the first couple of tasks we identified the fact that we could indeed create waves of different frequencies from the same function and “mix” them together to create an oddly shaped wave. However, as things moved along, we encountered some interesting tasks.

When we created three harmonic partials out of one fundamental wave it was approximated that the maximum number of harmonic partials that could be created from a 256 sample wave were 128. This is because we need to have at least 2 samples per cycle in order to maintain wave shape preservation.

We then moved on to the synthesis equation and the manipulation of it. After successfully executing the commands necessary to create partial wave via the synthesis equation we were asked to create a square wave using the very same techniques. I found the task tedious using the original cosine wave as my base function, and therefore changed by base function to a sine wave which allowed me to use a zero degree phase shift for all counterparts.

Through all of the trials this lab has brought us we can clearly say that we are competent in the area of wave making and manipulation in the field of MATLAB. While other more complicated uses of MATLAB have not yet been utilized in this class, we hope to learn even more over the next couple weeks.