EEE321-Lab2

Sait Sarper Özaslan

22002861

**Part 1)**

Starting off with SUMCS function the following code should be included so that one can comment according to it:

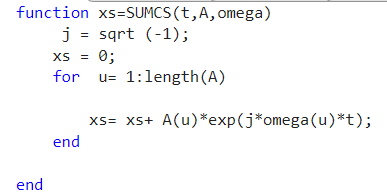


Figure 1: SUMCS code

To compute xs and following plots we need the following code:

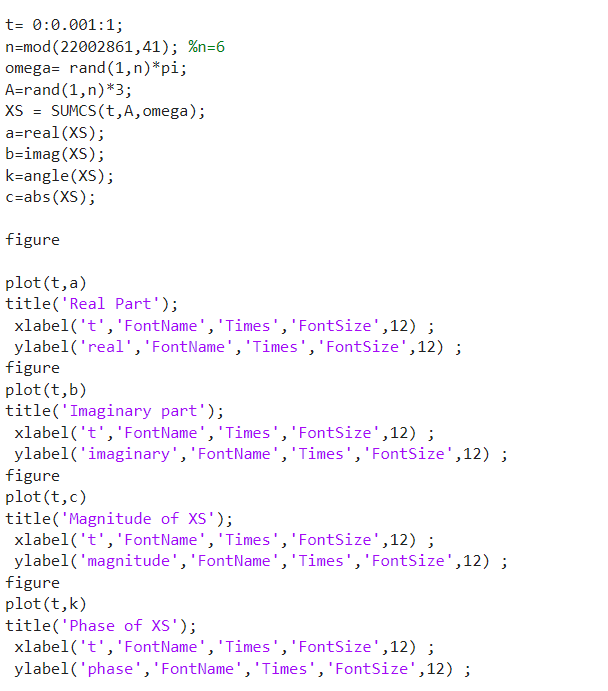


Figure 2: XS Calculation and Plots

Here are the following plots:

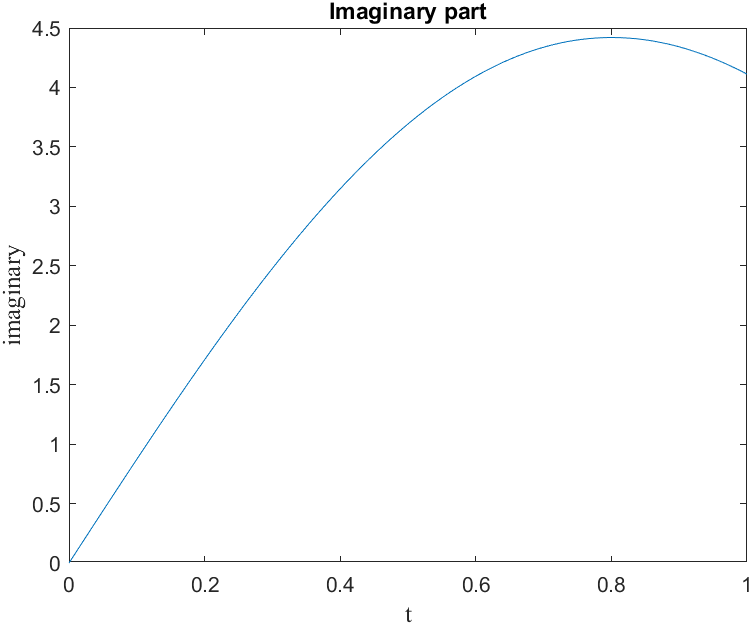
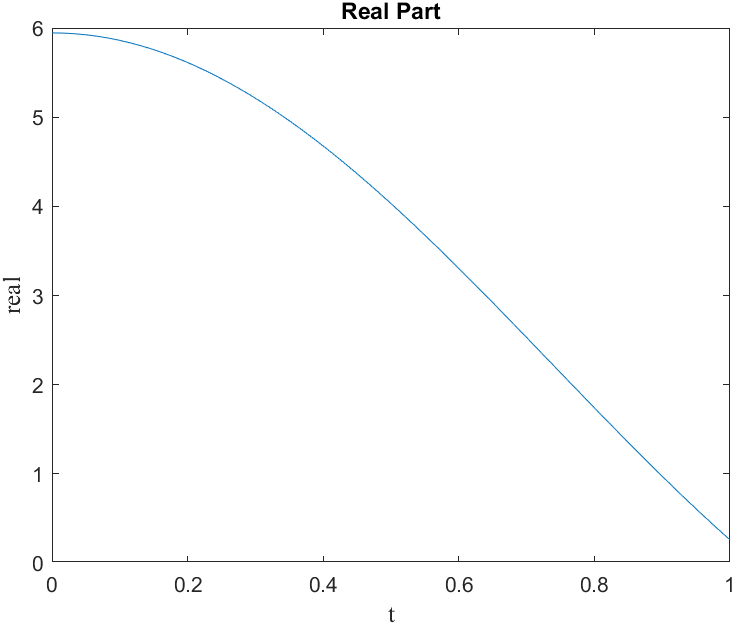


Figure 3 & 4: Real and Imaginary Parts of XS

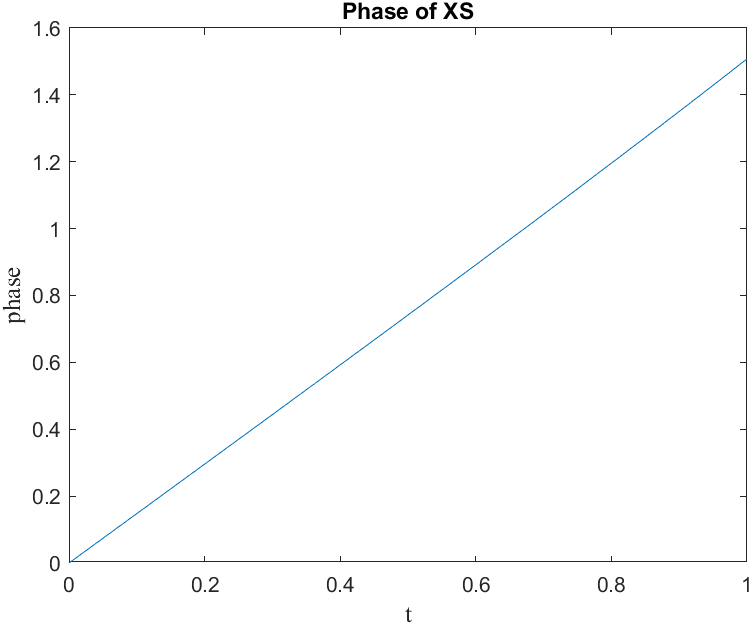
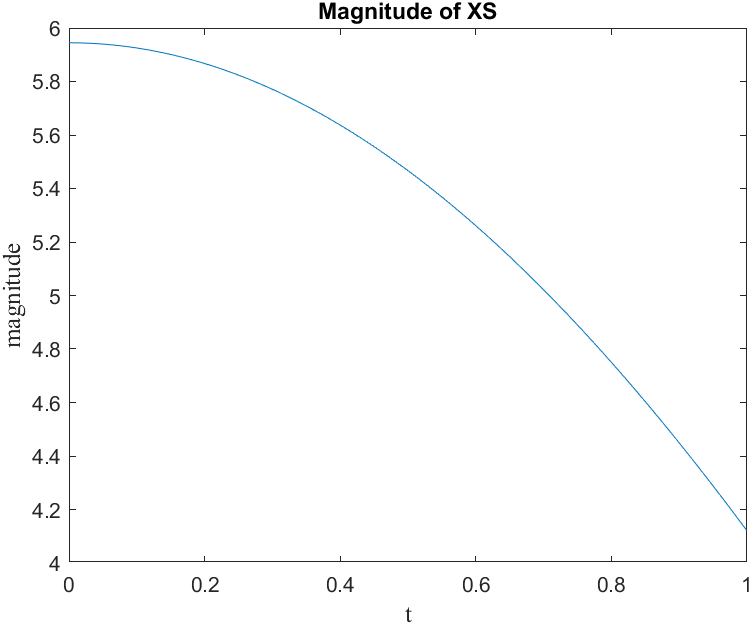


Figure 5 & 6: Magnitude and Phase of XS

**Part 2)**

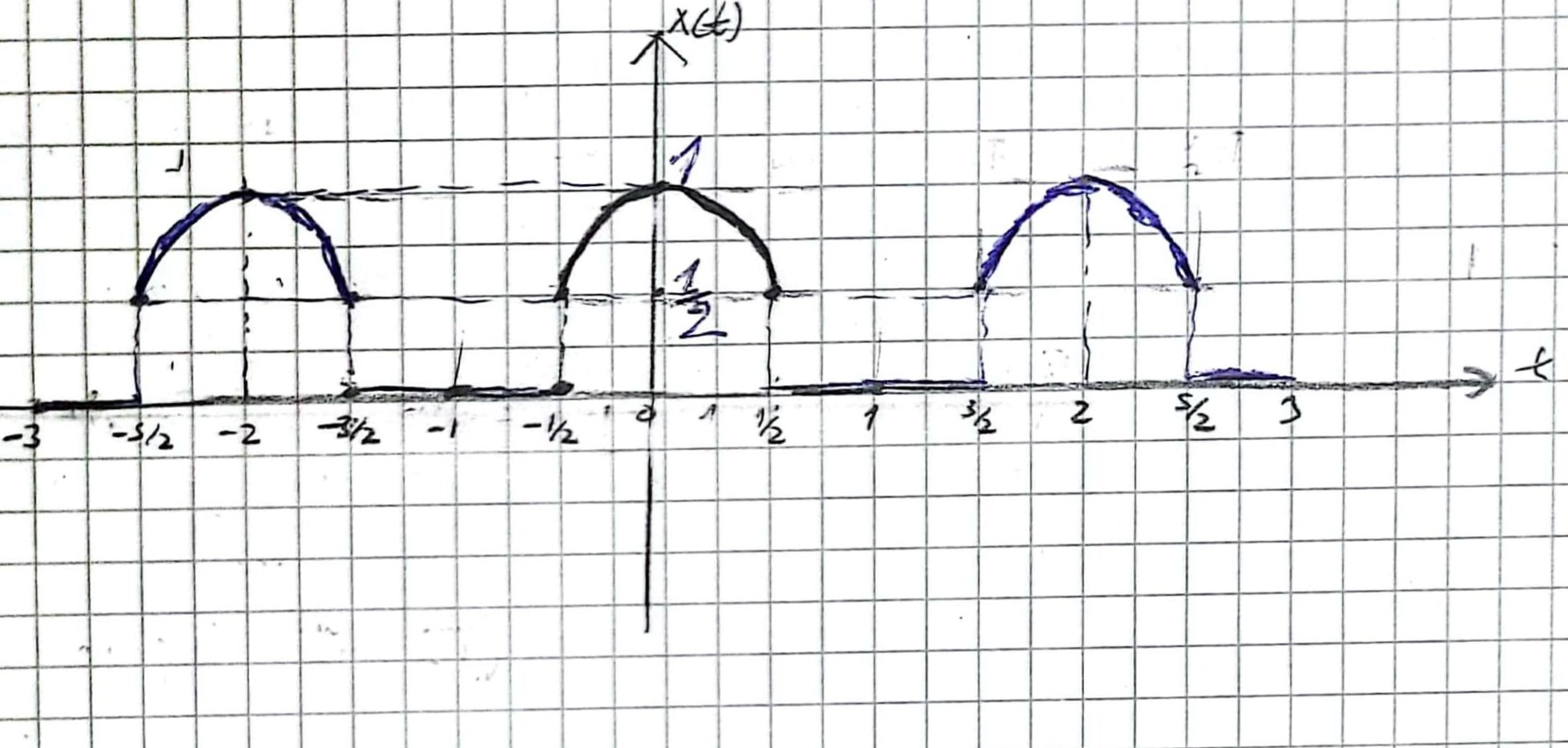


Figure 7: Plot of X(t) versus t

As instructed the plot is drawn as a piecewise periodic function.

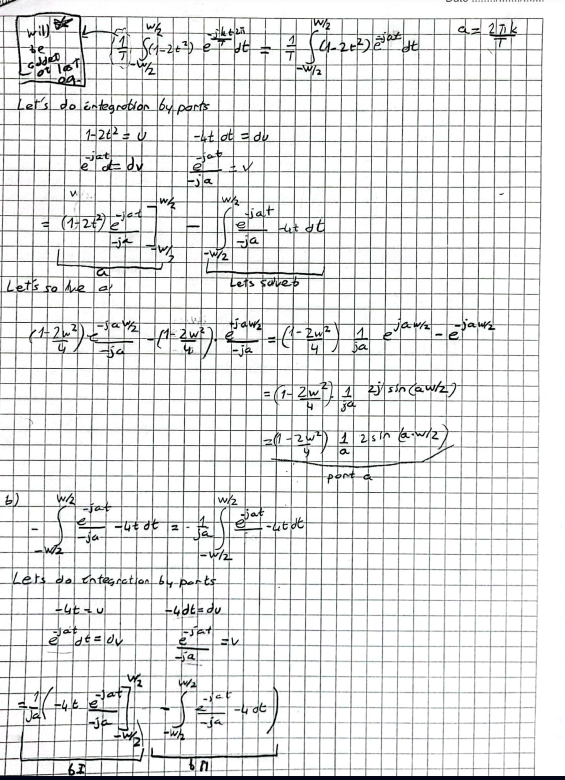


Figure 8: General Solution of Xk coefficients for W and T – 1

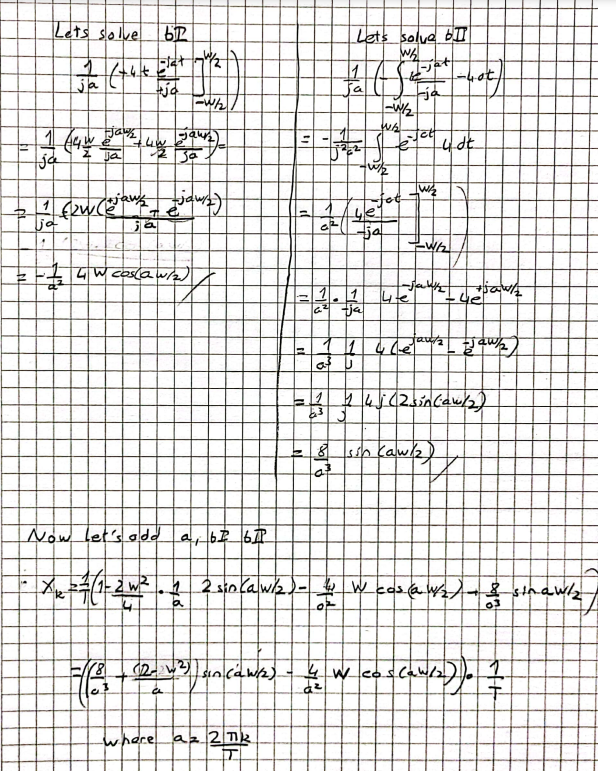


Figure 9: General Solution of Xk coefficients for W and T – 2

“a” is kept as itself to keep the result simple. It is used in the code as b where b = 2\*pi\*k/T.

* There is also the special case where k= 0, this will result in X**k=1/T\*(W-W^3/6)**

**Part 3)**

Starting off with the code for FSWave so that plots can be observed:

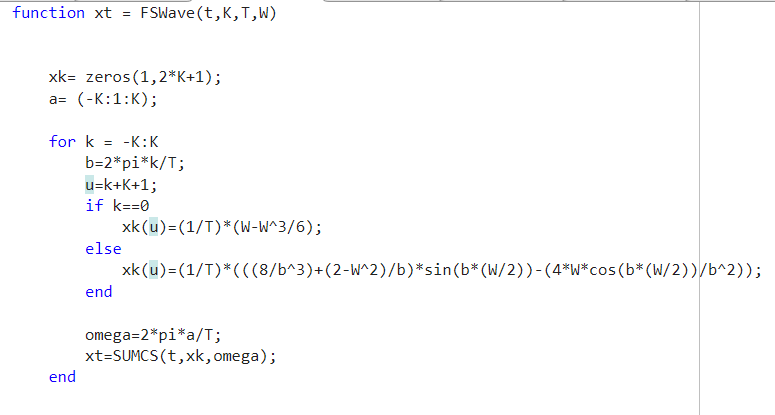


Figure 10: FSWave Function

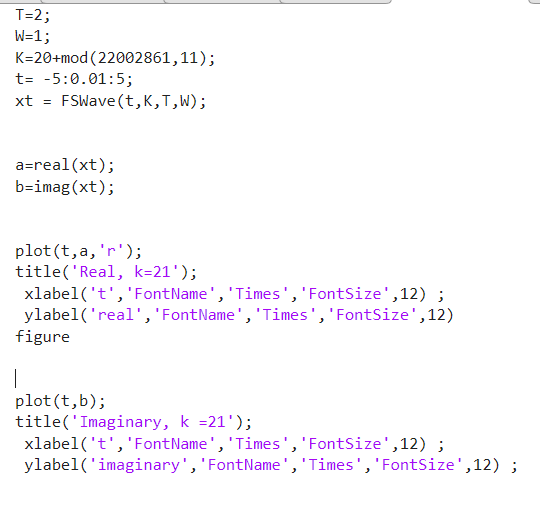


Figure 11: FSWave Real and Imaginary Parts

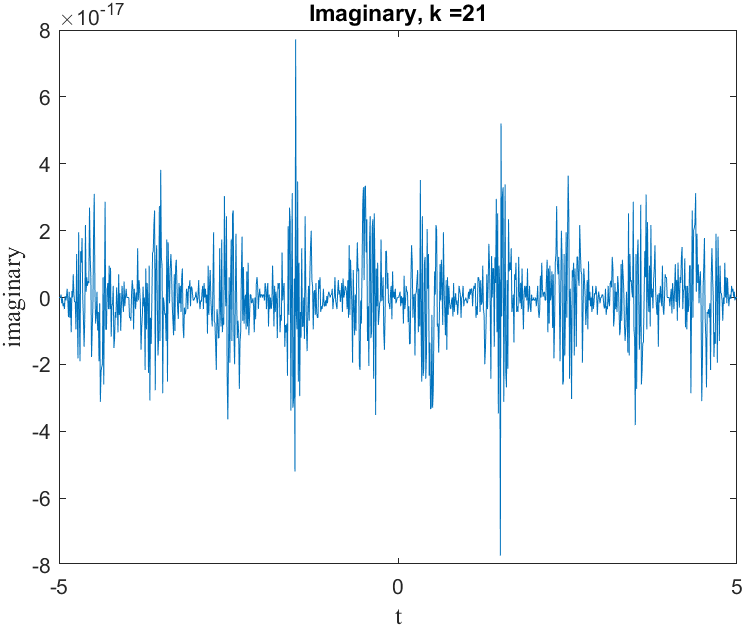
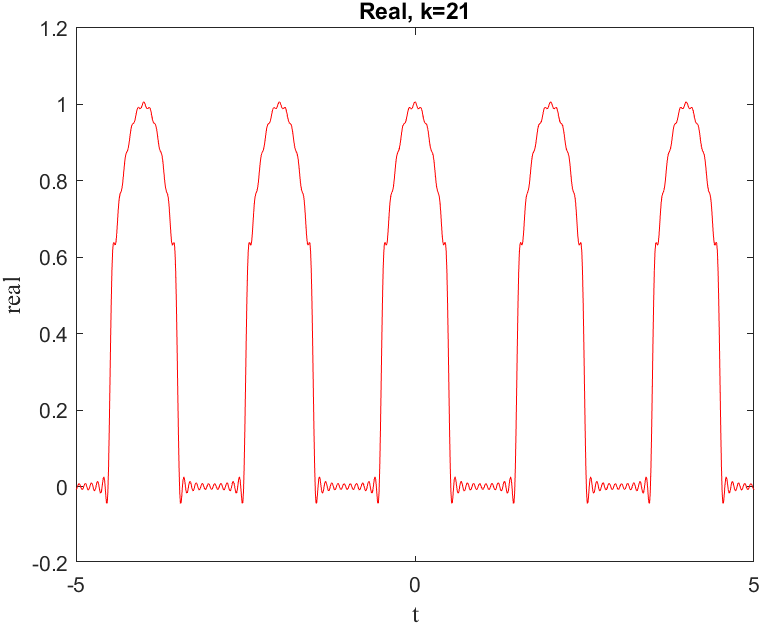


Figure 12 & 13: Real and Imaginary Part of xt

Examining the real part of xt we can see that the plot is similar to its sketch version. However, since we are doing rounding the values are not exactly as sketched. The maximum value found was x(t) = 1.000678 and the min value found was x(t)=-0.04348. We can still say that the values are close to the one in the sketch.

When looked at the imaginary part, the plot expected is 0 but due to MATLAB we see extremely small values that vary from +7.86\*10-17 to –7.85\*10-17. Though these values can practically be considered 0, the reason why it is not actually 0 is due to MATLAB’s rounding error.

MATLAB uses floating point precision for numbers and by default numbers are accurate 16 digits. As the precision itself can’t be infinite while calculating sin(pi/6)-0.5 one is likely to see a not exactly zero value.

To observe K’s behavior, we change D11= 1 with D5= 1 then we change K’s value according to it and get the following plots:

* K= 2 + D5

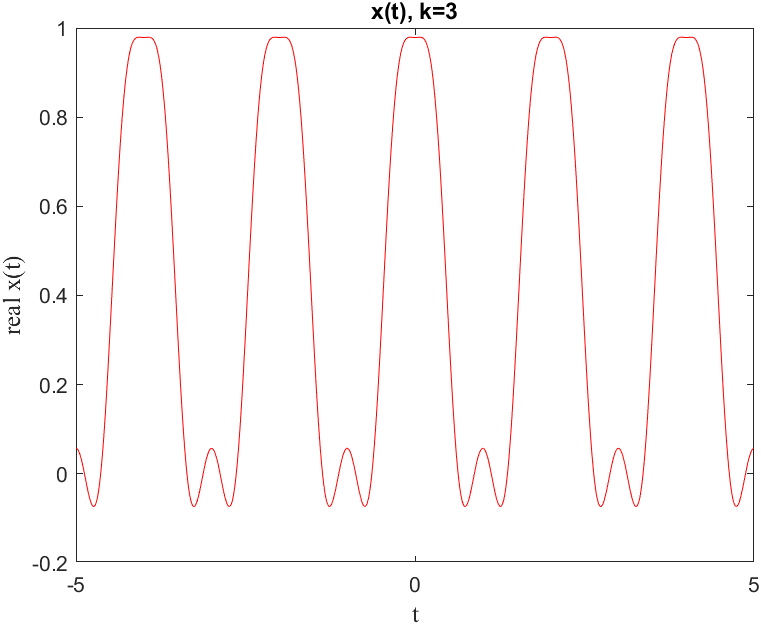


Figure 14: X(t) when K = 3

* K= 7 + D5

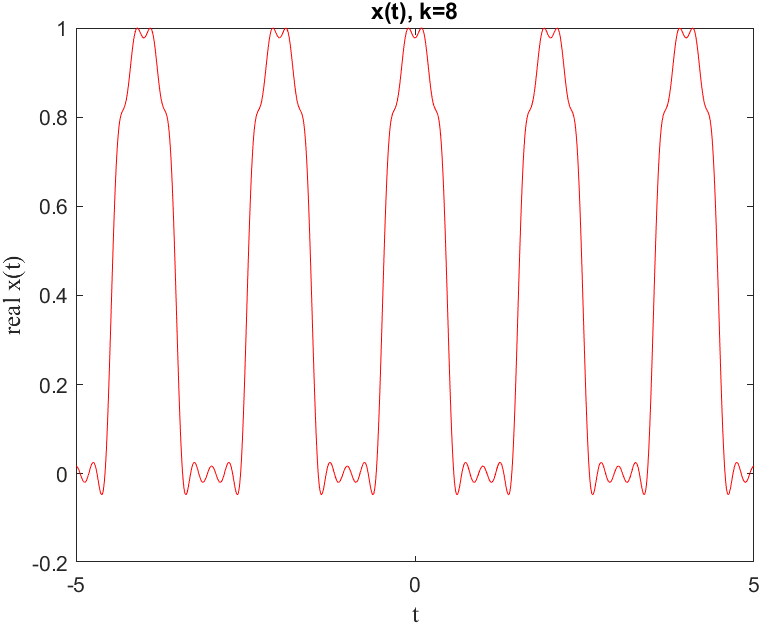


Figure 15: X(t) when K = 8

* K=15 + D5

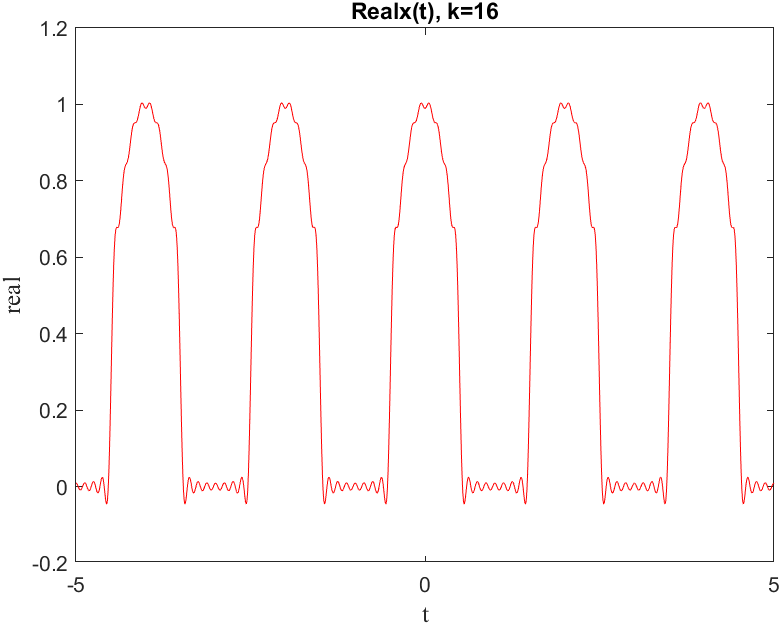


Figure 16: X(t) when K = 16

* K=50 + D5

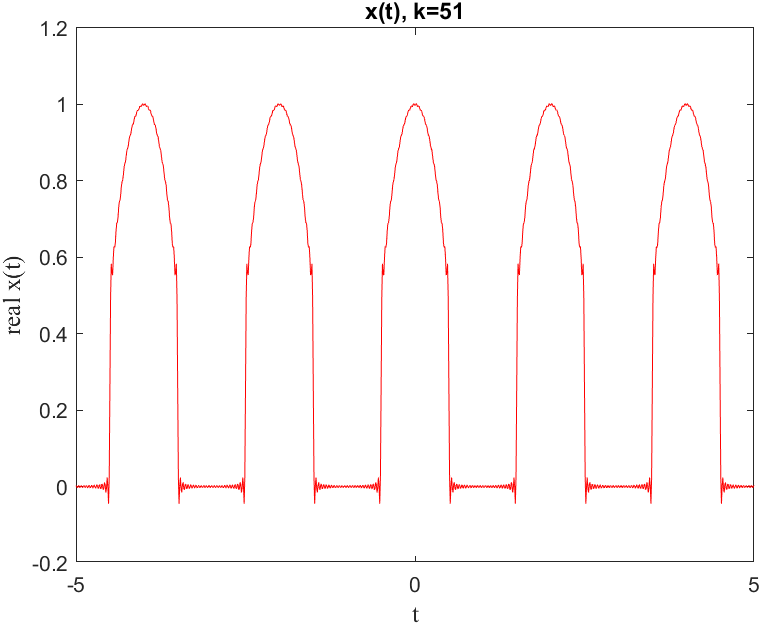


Figure 17: X(t) when K = 51

* K=100 + D5

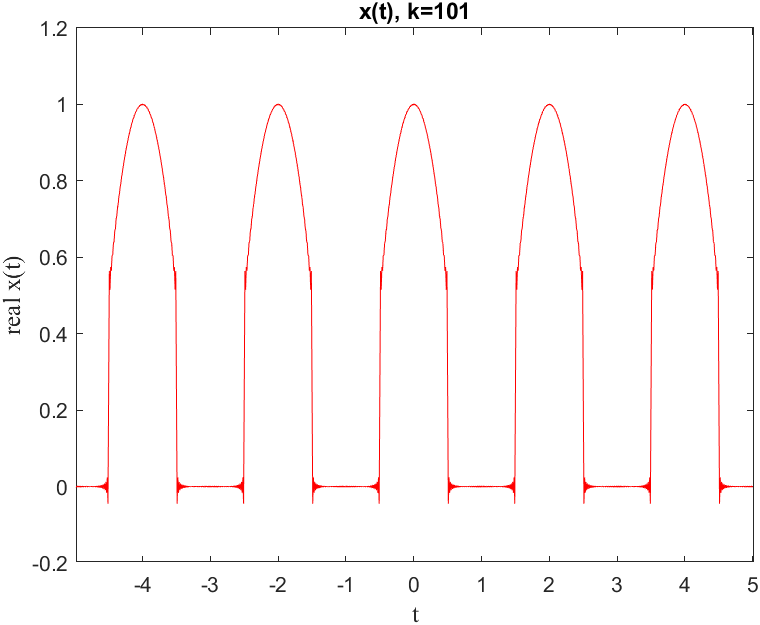


Figure 18: X(t) when K = 101

Looking at these plots we can say that as K, or the sampling size gets higher we get close to the actual graph of x(t). The distortions on the parts where 1-2t2 gets a value other than 0 lessens as K increases. If K goes to infinity, we expect to see the exact same graphic of x(t). However, the oscillation on where the values are zero does not ever completely disappear. This is due to the method used, being an approximation, which will eventually deviate from the actual result.

**Part 4 )**

1. Yk = -Xk

To change Yk = -Xk the following code Is changed:

From

* u=k+K+1;

to

* u=K-k+1;

Whose graph would be:

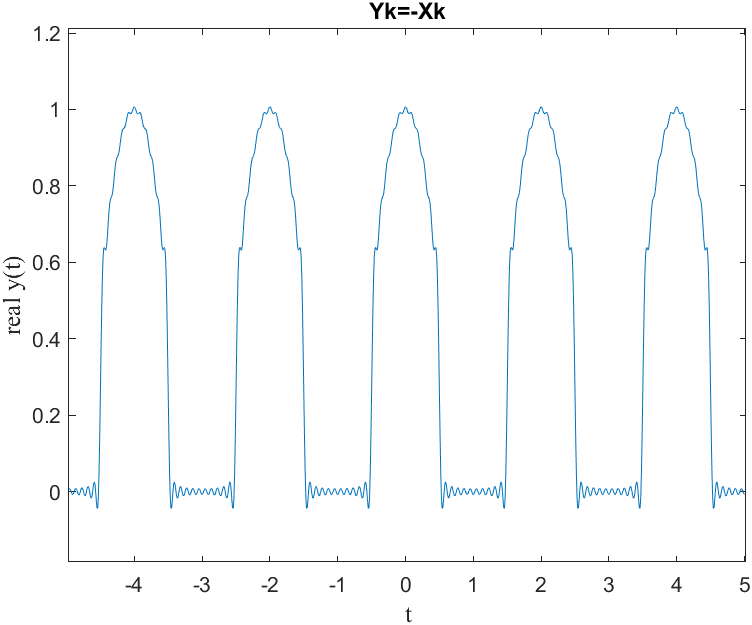


Figure 19: Plot of y(t) where Yk = -Xk

As the plot we drew is an even function making Yk = -Xk, shouldn’t cause any changes. This can be confirmed from figure 19. In here y(t) is similar to Re{x~(t)}. Normally we would be mirroing this in x axis

1. Yk = Xk\*exp(-j\*b\*t0)

To make the wanted change I did the following:

From

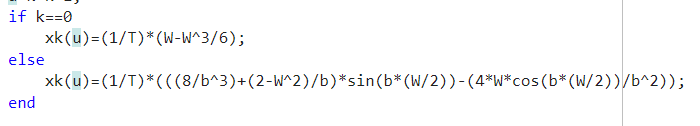


Figure 20: Original FSWave Function

To

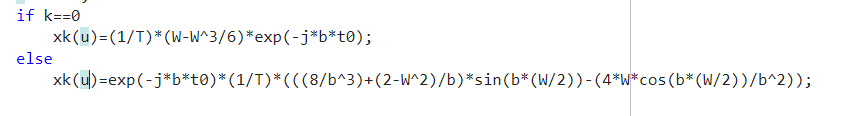


Figure 21: Changed FSWave

Essentially this is a phase shift to the right by t0. The value b= 2\*k\*pi/T as can be seen in earlier figures.

This results in the following plot when t0 =0.6:

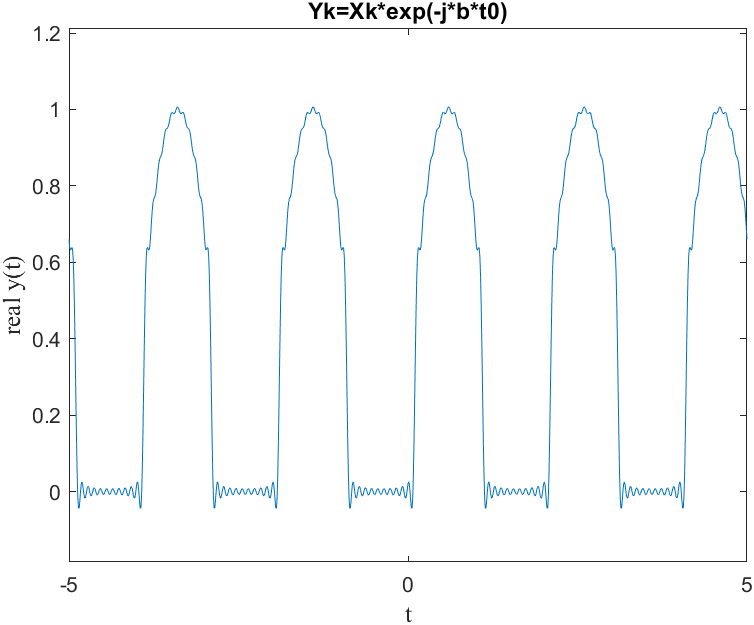


Figure 22: Re{Y(t)} phase shift to right by 0.6

As can be seen in the figure one can see that plot is shifted to the right by 0.6. To confirm this, the value used to be at center t=0 in the original figure, whereas Re{y(t)} = 1.0051 when t=0.59.

1. Yk = Xk\*j\*b

To make the specified change one needs to do the following:

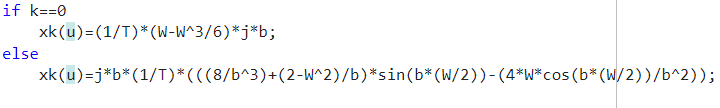


Figure 23: Changed FSWave

Essentially Xk coefficients were multiplied with j\*b = j\*2\*pi\*k/T. We can think of this as a differentiation of the general sum operation in figure 24.

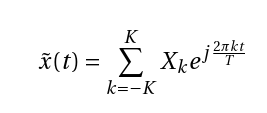


Figure 24: General FS Approximation Function

This results in the following plot:

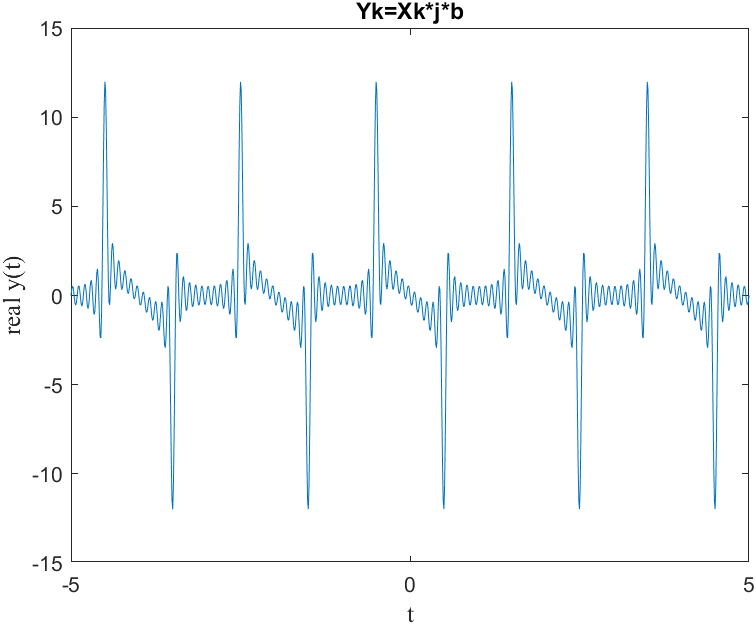


Figure 25: Differentiated Re{y(t)}

As one can see in figure 25, the peak in graph essentially correlates with the rises of the original plot in figure 12. Same case can be said for the dips in the plot as well.

1. Piecewise version for Yk

To make the required change the following code was implemented after Xk values were calculated.

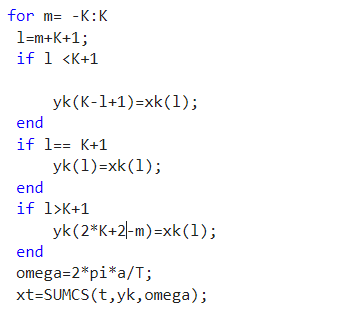


Figure 26: Added Code to FSWave

Essentially what was done in the code is to assign values to Yk after Xk’s values were determined. The placement of coefficient was determined with another loop and if statements which helped assing Xk[u]’s values to Yk. This resulted in the following graph,

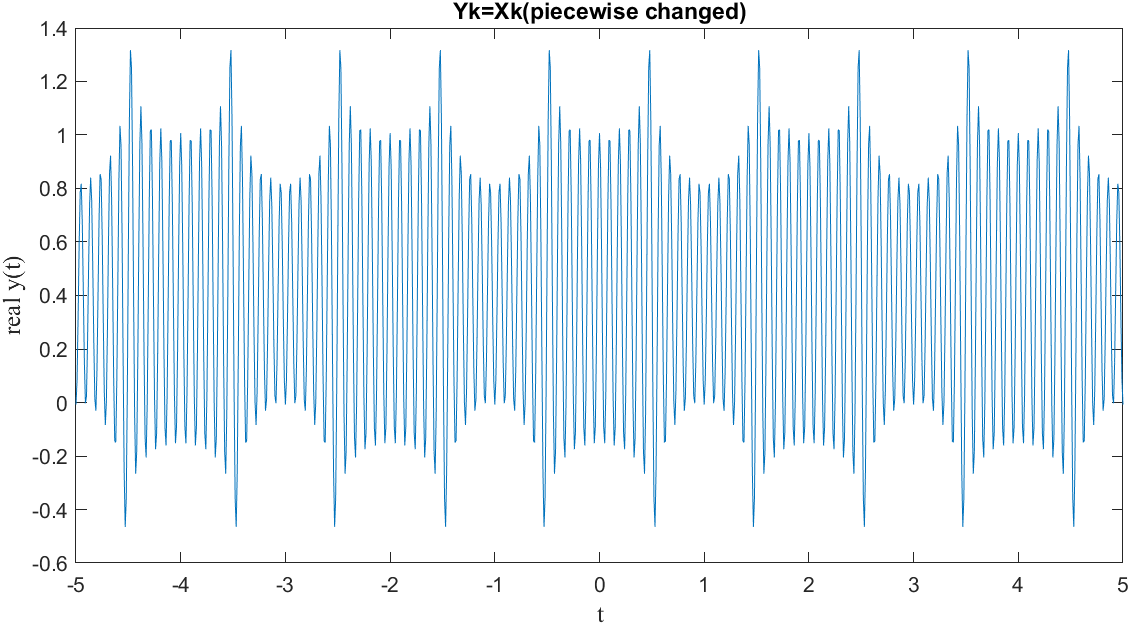


Figure 27: Piecewise Change Re{Y(t)}

I don’t really have any explanation for the last part as I don’t understand why we are doing this. Essentially all it does is to flip the places of positive and negative coefficents within itself.