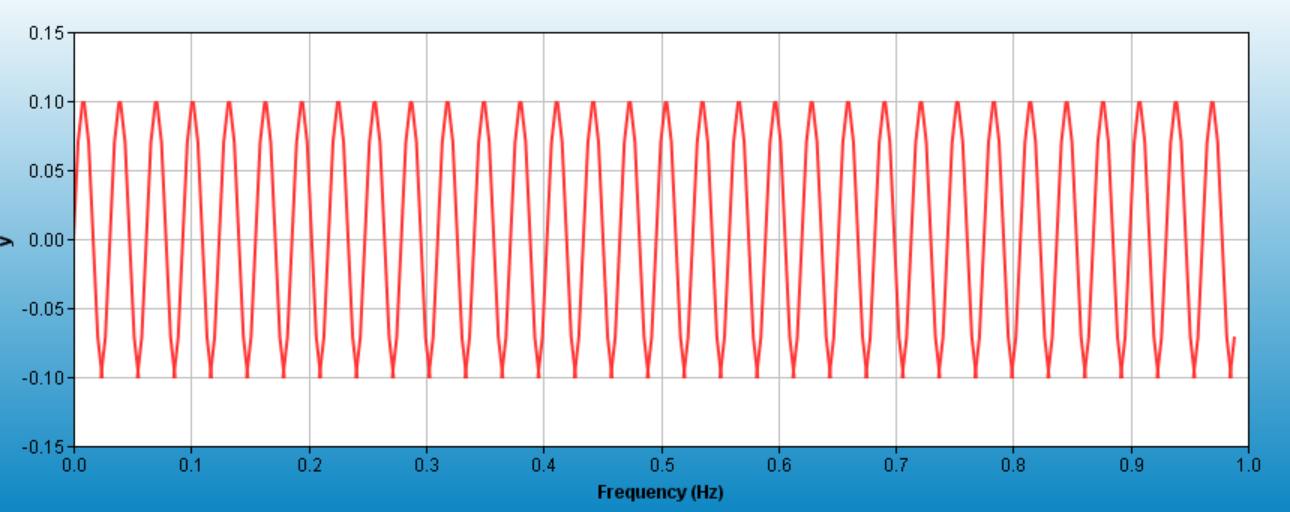
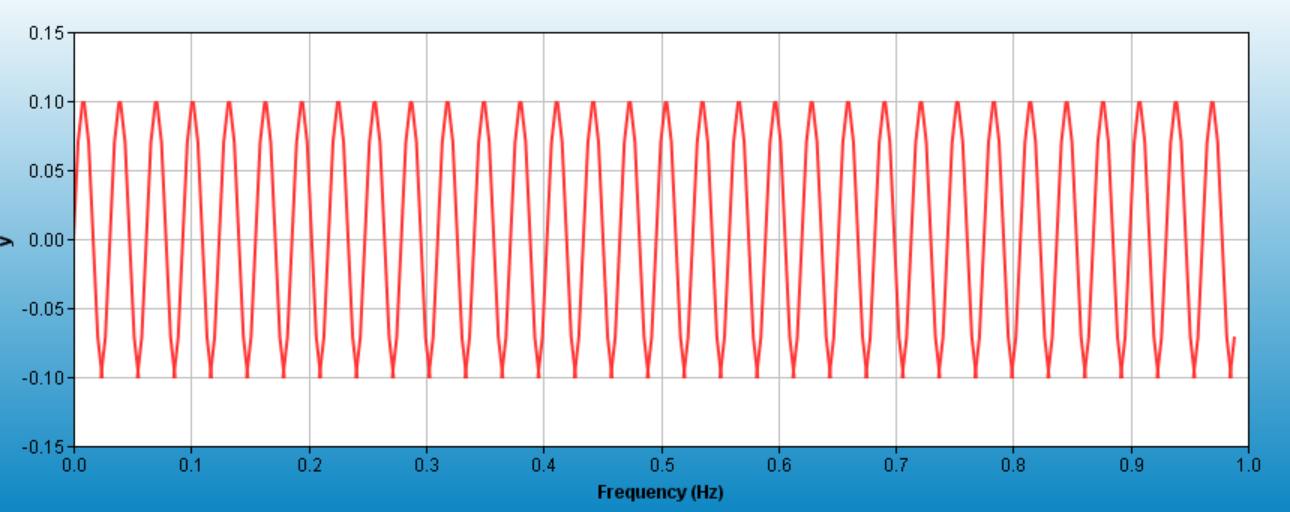
**B. Implementation of digital anti-aliasing filters on a DSP**

1. **Plot of the sample signal x(n) = 0.1 \* sin(0.25 \* 1.0 \* n);**

The Time domain representation of the signal has been ploteed in CCES as the figure below, where its x axis is time(s) and y axis is amplitude.



The frequency domain representation of the signal is shown in the figure below, where the x axis is frequency(Hz), y axis is amplitude.



1. **Write a c-program which generates two sequences of 256 samples of x(t), using the sampling frequencies F1 = 1.2Hz and F2 = 4.8Hz.**

**The code used to generate the 2 signals are attached below.**

#include <stdio.h>

#include <math.h>

*// Globals*

#define N 256

#define PI 3.1415

float x1[N]; *// Sample at T1*

float x2[N]; *// Sample at T2*

int main(void)

{

int i;

float omega1 = 0.25 \* PI, omega2 = 1.9 \* PI;

float T2 = 1/4.8;

float T1 = 1/1.2;

float a=0.12;

float alpha1 = 0.593, alpha2 = 0.464;

x1[0] = exp(-a\*0\*T1)\*cos(omega1\*0\*T1) + 0.1\*sin(omega2\*0\*T1);

x2[0] = exp(-a\*0\*T2)\*cos(omega1\*0\*T2) + 0.1\*sin(omega2\*0\*T2);

**for** (i = 0; i < N; i++)

{

x1[i] = exp(-a\*i\*T1)\*cos(omega1\*i\*T1) + 0.1\*sin(omega2\*i\*T1);

x2[i] = exp(-a\*i\*T2)\*cos(omega1\*i\*T2) + 0.1\*sin(omega2\*i\*T2);

}

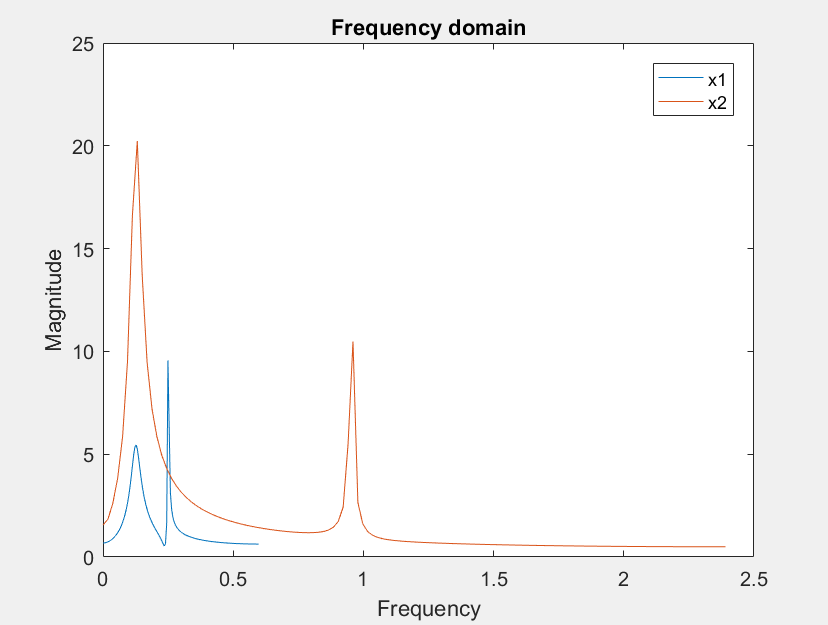
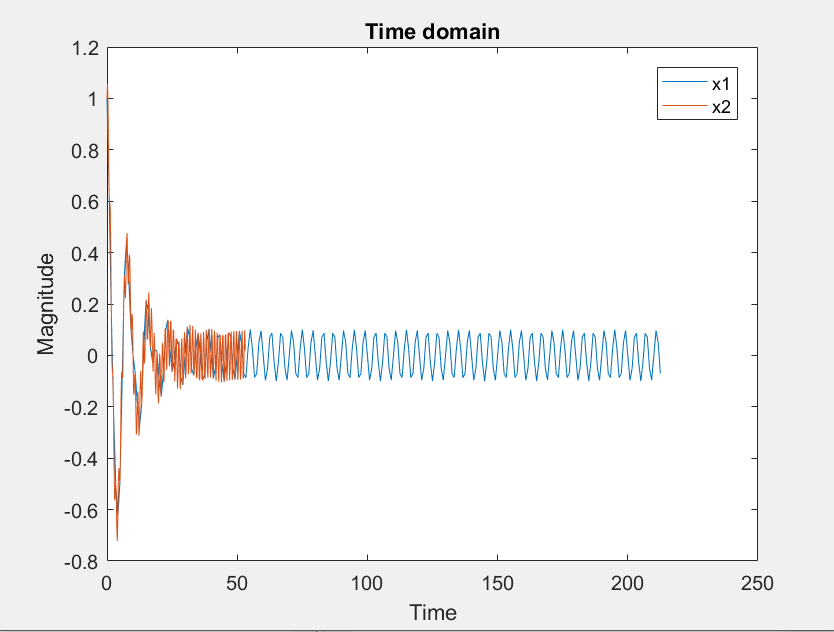
printf("Done.**\n**");

**return** 0;

}

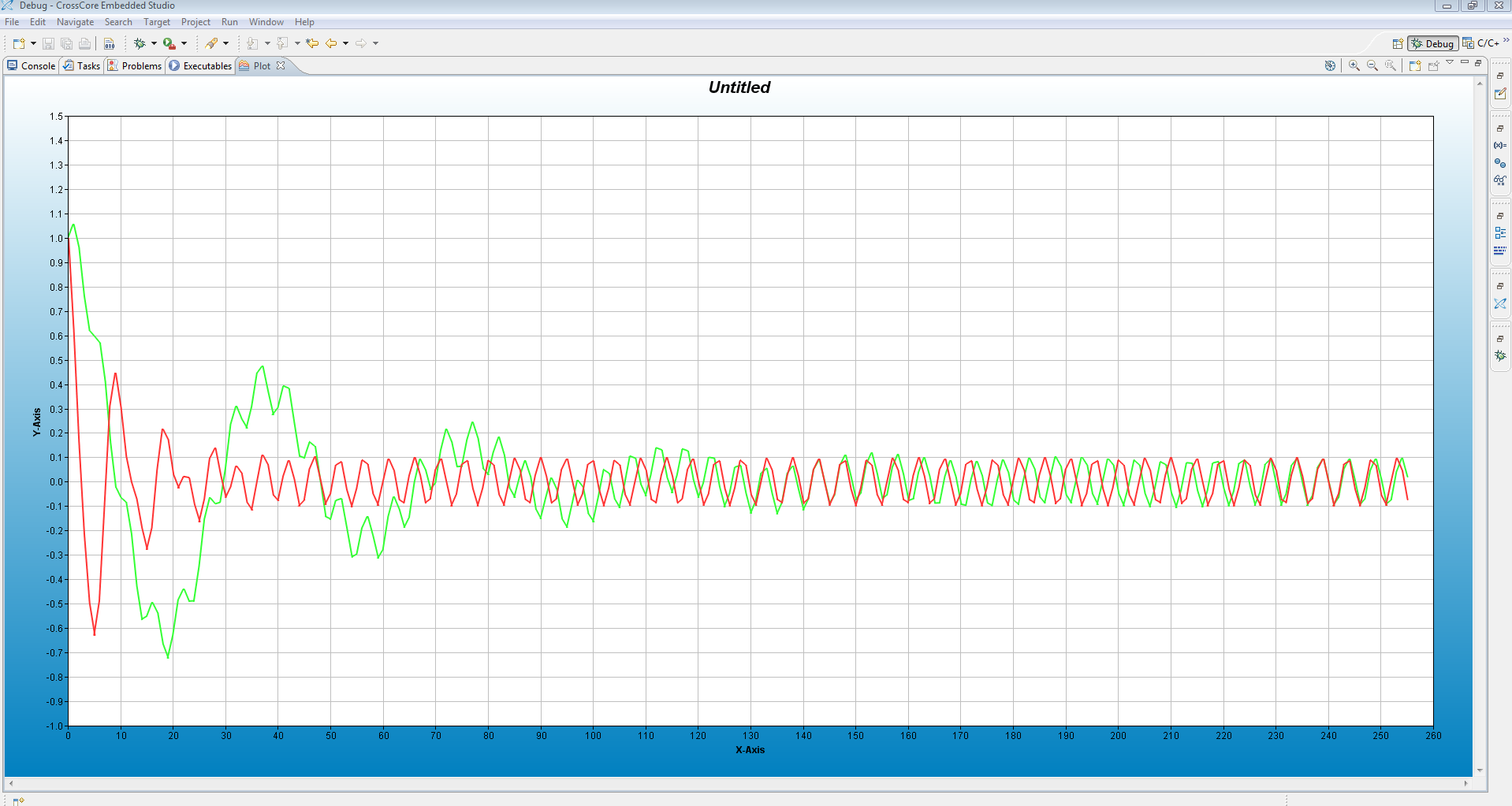
1. **Use the plot facility within CCES to plot the sampled signals in the time domain and in the frequency domain. Comment on the results.**

To predict how the signal is going to behave, Matlab was used to plot x1 and x2 in time and frequency domain. (Code attached in appendix).

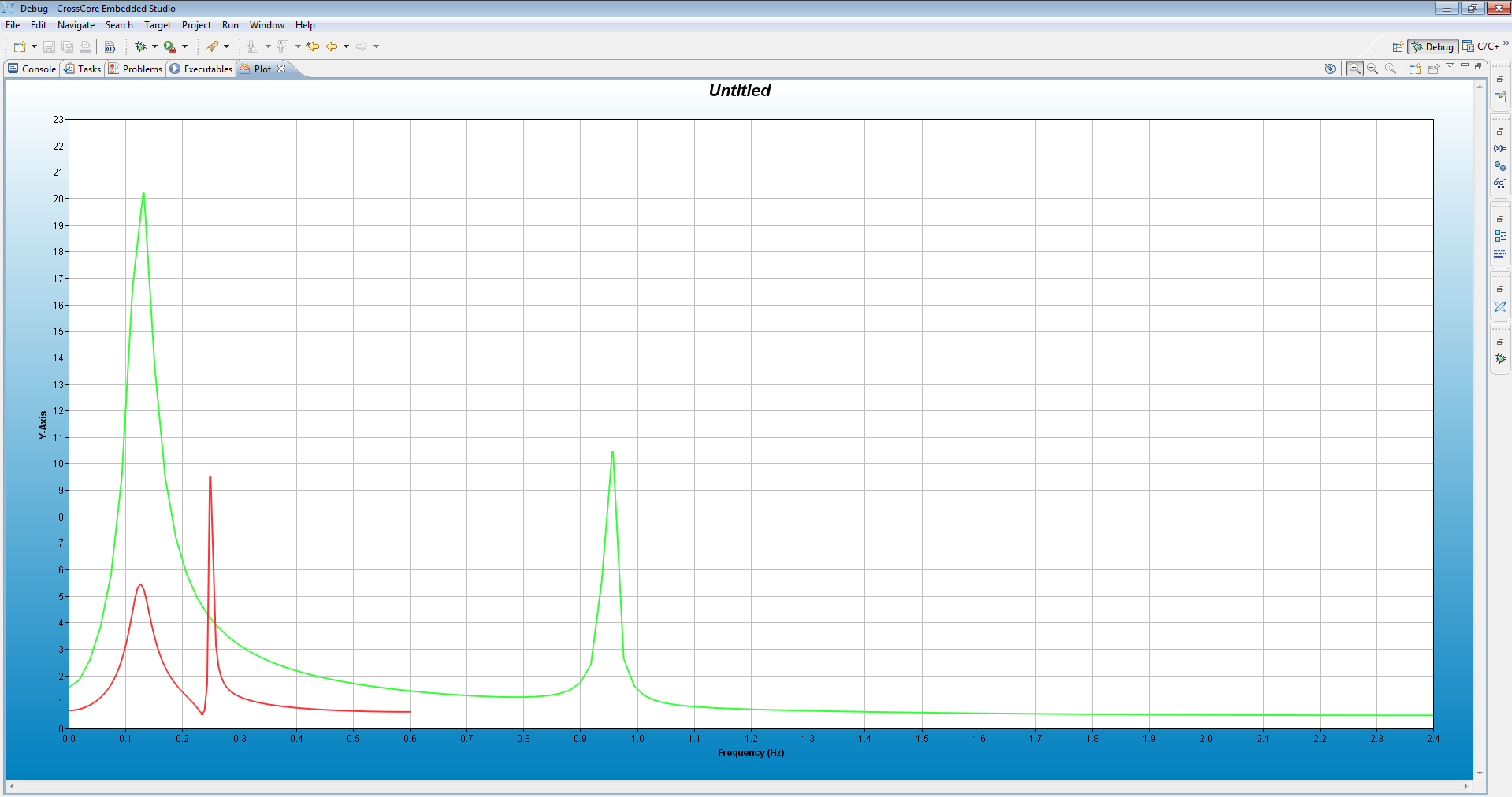


**The result generated by CESS are shown as below**

**Time domain plot. x axis: sample time, y axis: magnitude(linear scale)**



**Frequency domain plot. x axis: frequency(Hz), y axis: magnitude(linear scale)**



**As the plots above show, the signal generated in the DSP board are identical to that simulated in Matlab.**

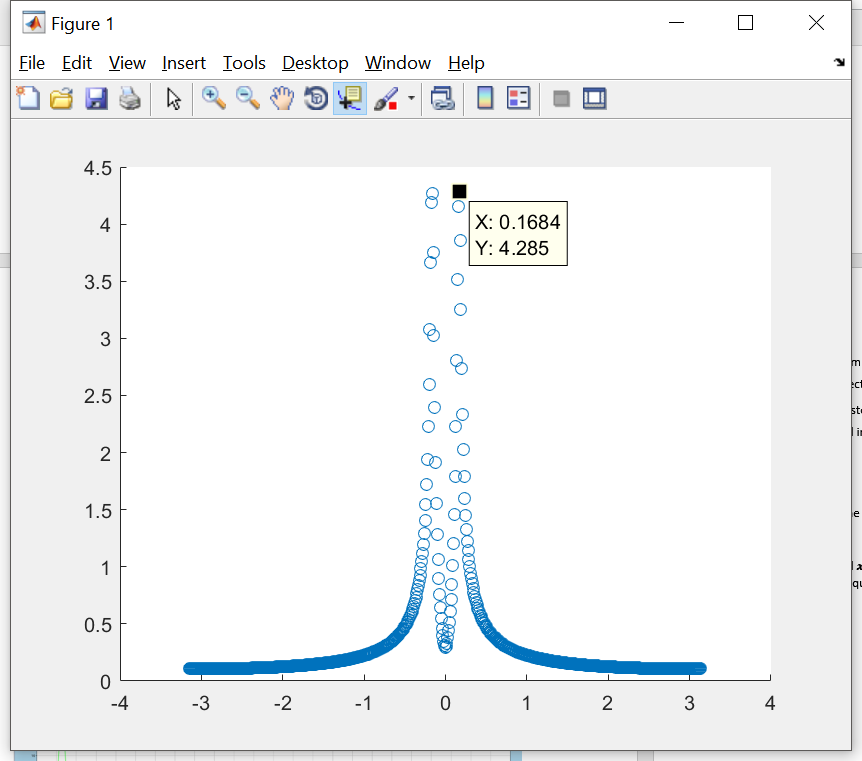
**Comment on the result:**

**Theoretically speaking, the signal should generate two peaks at , and . Which is matching up with what signal x2(the green trace) in the CCES FFT plot.**

**However, there are two facts about the signal x1’s FFT plot were noticed in the FFT plot: 1. The 0.95Hz peak was missing, 2. There is an “unpredicted” peak between the region of 0.2-0.3Hz. Fact 1 might due to the sampling angular velocity . When the sampling frequency is less than of the signal, high frequency component of the signal will be distorted while sampling. Fact 2 is due to the fact that the sampling frequency is too low and it cause the folding effect around The frequency component from is distorted and folded back to 0.6-(0.95-0.6) = 0.25Hz, which fell into the 0.2-0.3Hz region we mentioned in the last paragraph.**

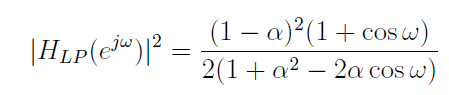
1. **Analysis of the first order and second order filter**

**After DTFT, the signal , has been plotted in frequency domain in question part A(c), and can be obtained from the plot.**



1. **Find of the first order filter**

**It has been proven that**



**The gain(magnitude)=0.95 of this filter at ,**

**can be calculated by**

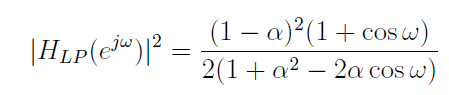
**Solving this equation at Wolfram Alpha gives us**

**As the filter is designed to filter out 5% of the gain of x1, now we need to evaluate the gain of the signal after the filter.**

**Recall in discrete frequency domain, there will be a peak at rad/s.**

**As the requirement of the filter is to get a gain less than 0.25 at , the calculation above has shown that this filter won’t satisfy the design requirement.**

1. **Find of the second order filter**



**Since ,**

**The gain(magnitude)=0.95 of this filter at ,**

**can be calculated by**

**Solving this equation at Wolfram Alpha gives us .**

**As the filter is designed to filter out 5% of the gain of x1, now we need to evaluate the gain of the signal after the filter.**

**Gain = 0.207 is less than 0.25 therefore this filter will satisfy the design specification.**