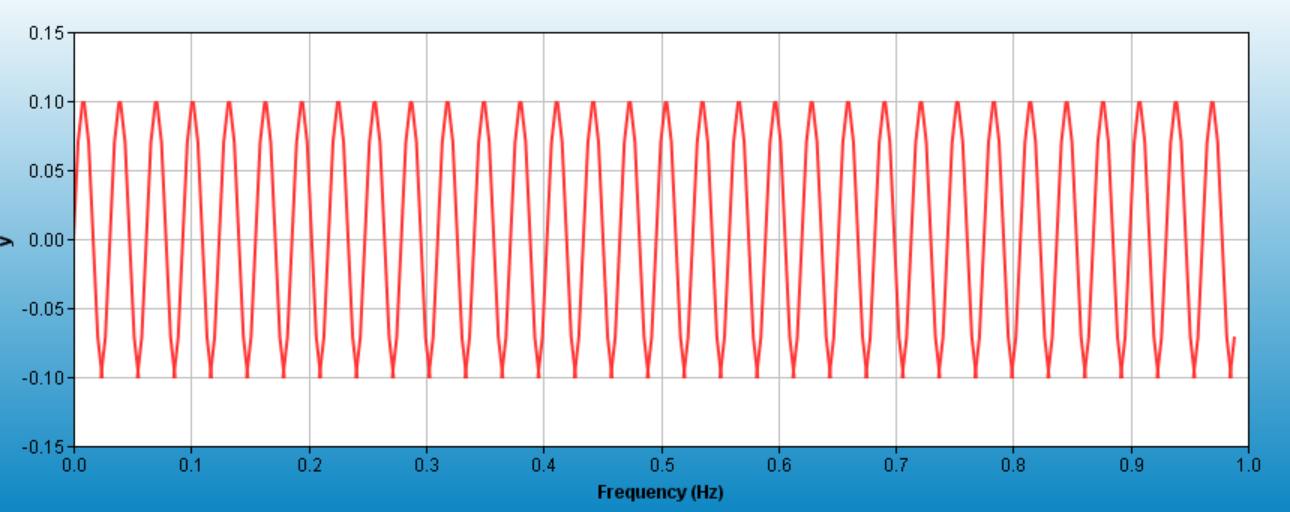
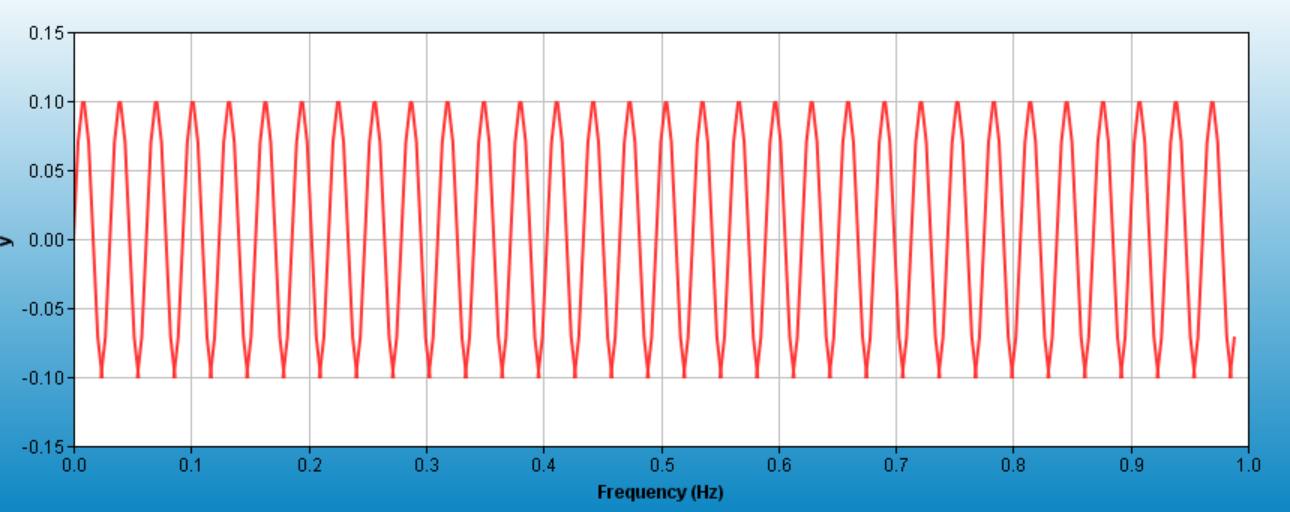
**B. Implementation of digital anti-aliasing filters on a DSP**

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];

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)

