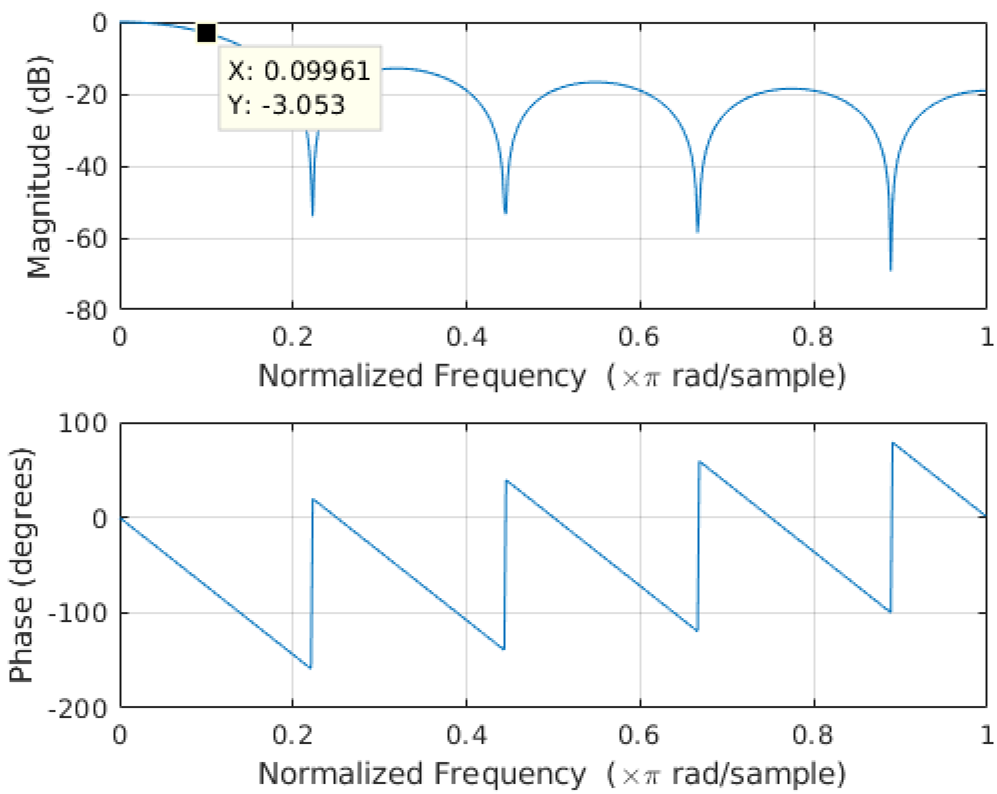
HW10

Since the sampling frequency is 100 Hz, this means that a Nyquist frequency of '1' corresponds to 50 Hz. Thus, to find a -3 dB cutoff frequency at 5 Hz, we need to use 10% of 1 = 0.1 Nyquist frequency

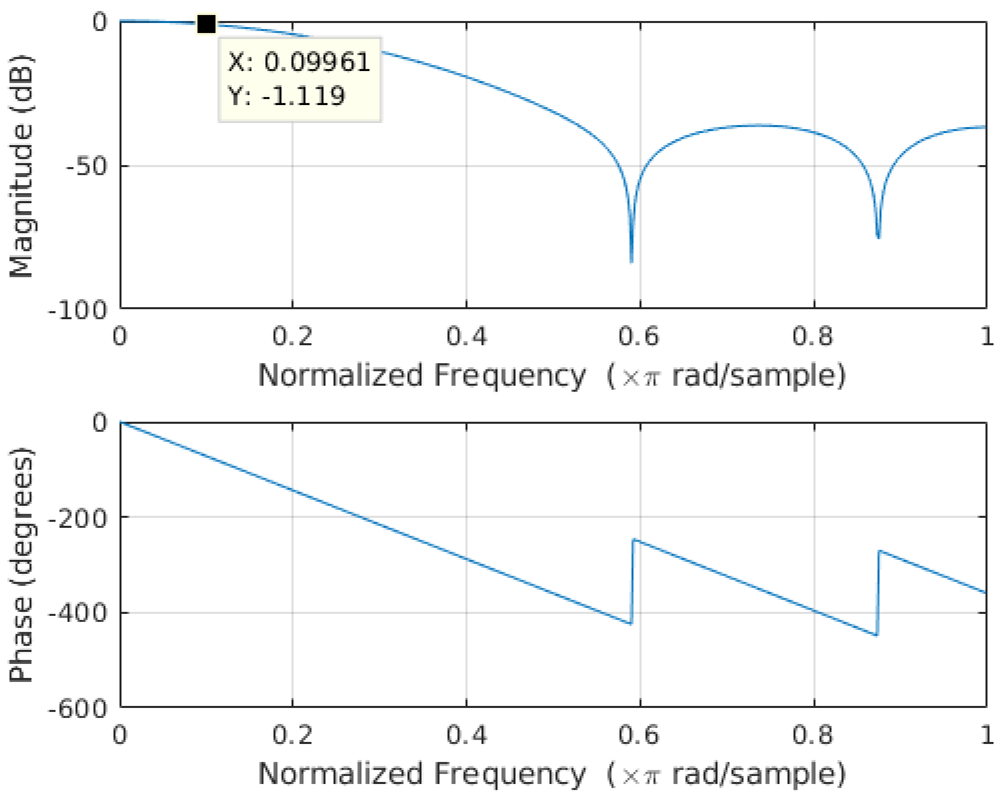
MAF

To achieve a 5 Hz cutoff frequency, I just increased the order of the system until -3 dB at 0.1 Nyquist frequency was reached. In this case, the order was 9. So the 'b' vector is [1 1 1 1 1 1 1 1 1]/9. The graph for this is shown below. This filter also significantly attenuates signals occurring at 11 Hz or multiples of it.



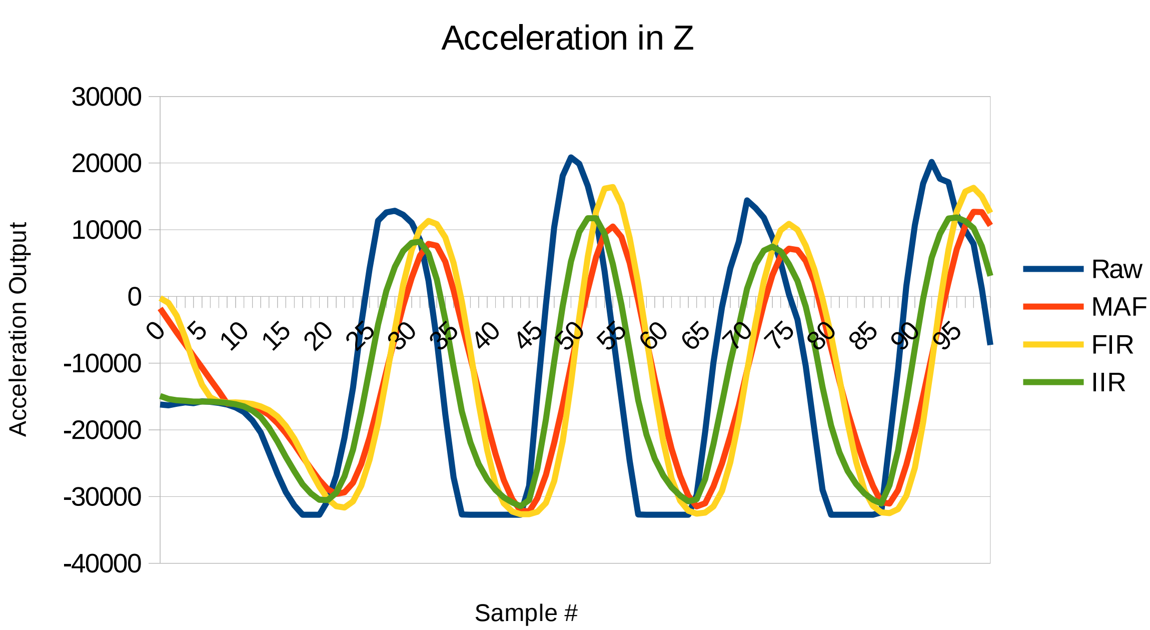
FIR

To achieve a 5 Hz cutoff frequency, I used the same order system as the MAF (easier to compare) at 0.1 Nyquist frequency. The inputted code was 'b = fir1(8,0.1)'. This gave me a 'b' vector of: {0.0144, 0.0439, 0.1202, 0.2025, 0.2380, 0.2025, 0.1202, 0.0439, 0.0144}; For some reason, the outputted magnitude plot did not show the -6 dB cutoff frequency at 0.1 of the Nyquist frequency (-6 dB is apparently the default setting in MATLAB). This issue was discussed with Nick.



IIR

For this filter, I randomly decided to make my equation: 0.7 * OldOutput + 0.3 * rawInput
After running the program, I plotted the output using Linux's LibreOffice Calc, and got this chart:



To me, it seems that the FIR filter worked the best as the amplitude of the wave was preserved more so than the others. Additionally, it looks pretty smooth. However, the IIR one is is a close second since it has the least lag and looks smooth, but it also sacrifices amplitude.