a)

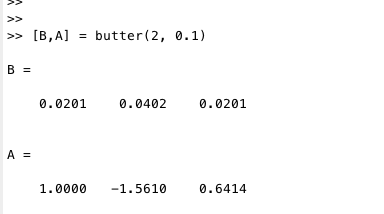
[B,A] = butter(N,Wn)

where N means Nth order lowpass digital Butterworth filter, and Wn means the cutoff frequency.

In this question, we use second order Butterworth filter, so N = 2.

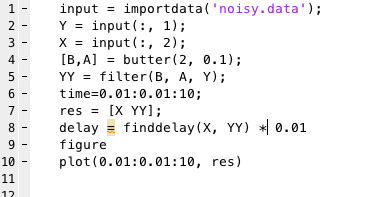
Sampling frequency is 100Hz, and cutoff frequency is 5Hz, so Wn = 5/(100/5) = 0.1.

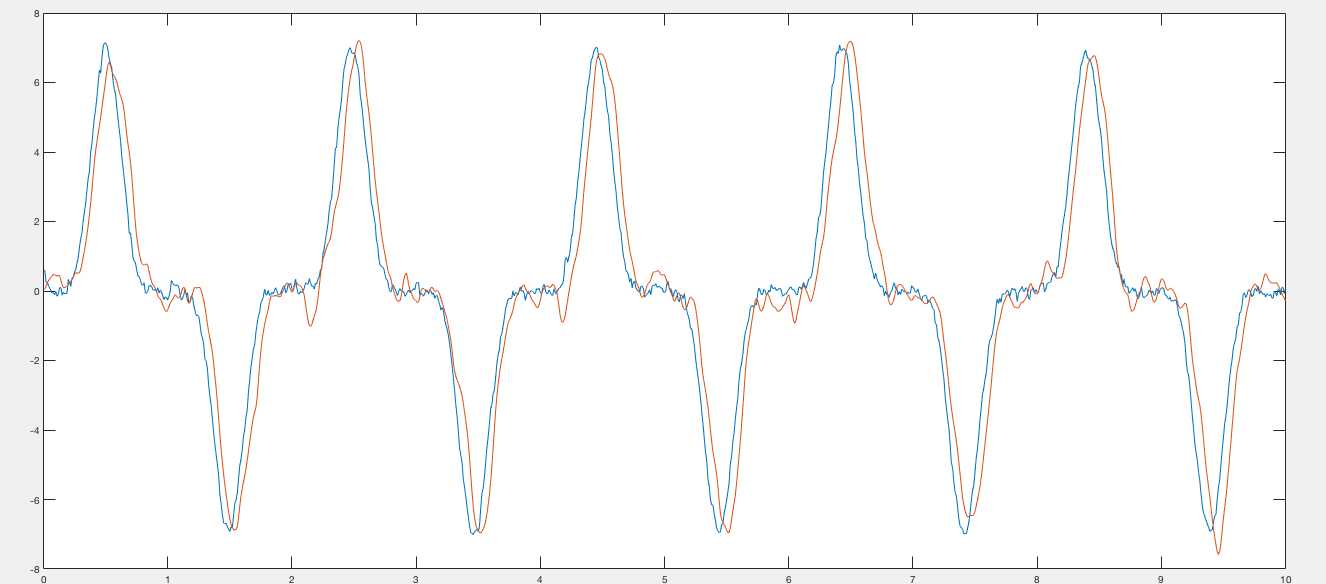
Thus, [B,A] = butter(2, 0.1)



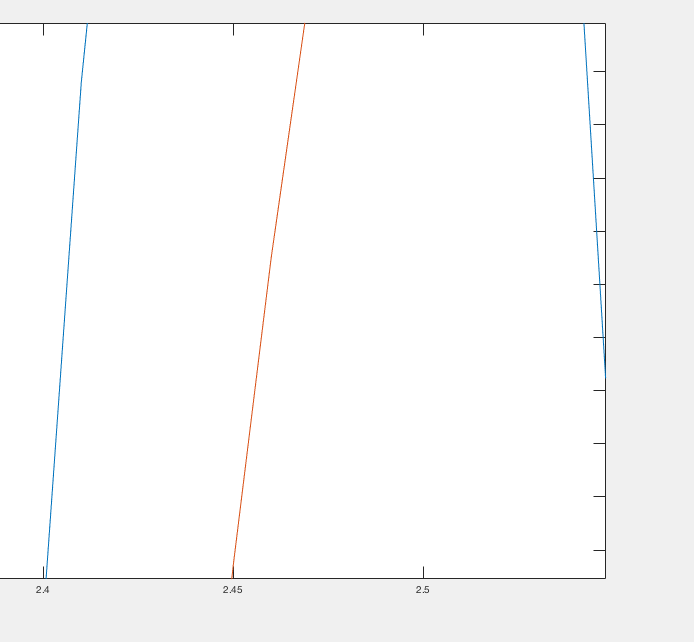
= 0.0201 = 0.0402 = 0.0201 = -1.5610 = 0.6414

b)

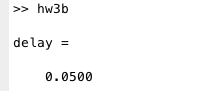




The filter works pretty well, basically get the value we want, but has some delay, and shake a lot more when position near 0.

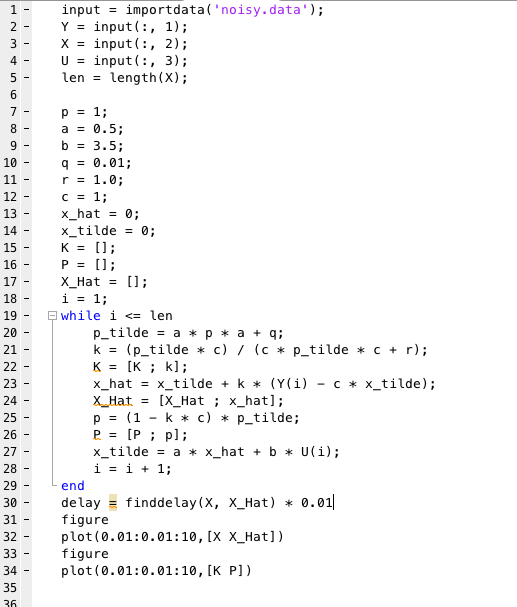


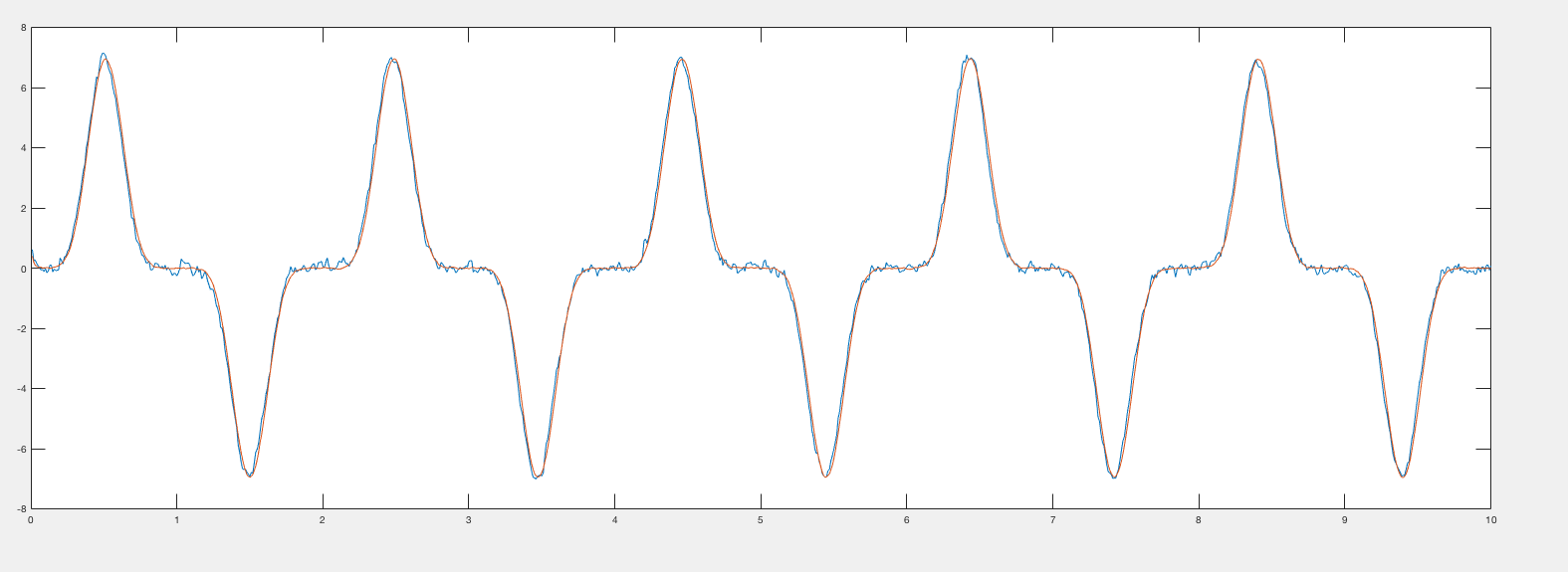
When we zoom in the plot, we can find the delay is about 0.05.

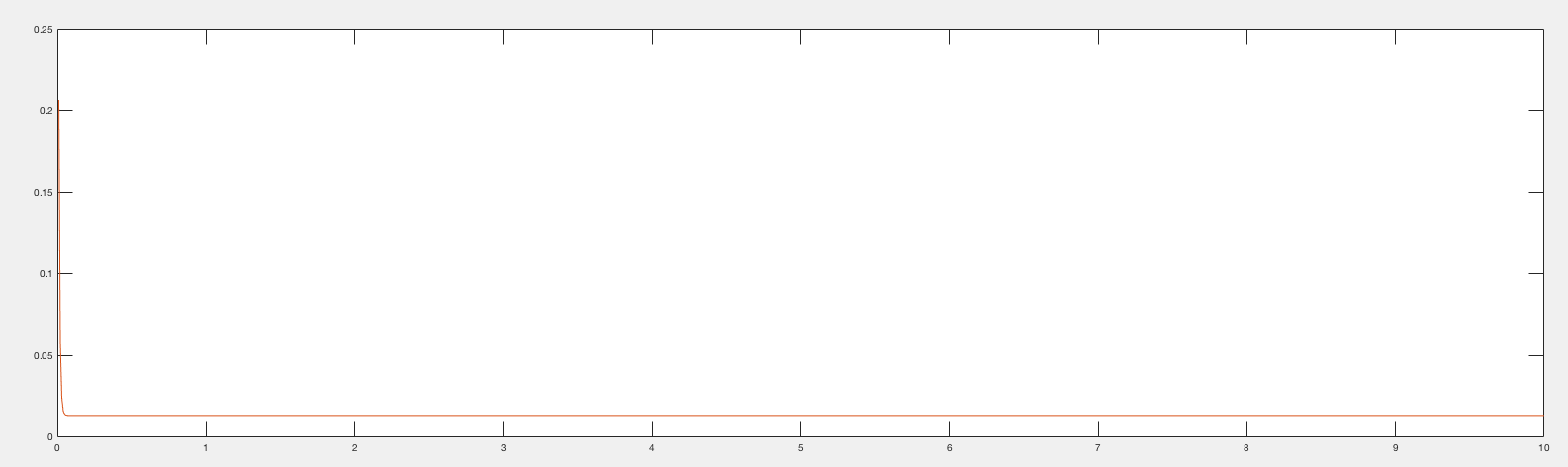


Besides from the plot, we can also get delay from script by “finddelay” function. In conclusion, delay is 0.05

c)

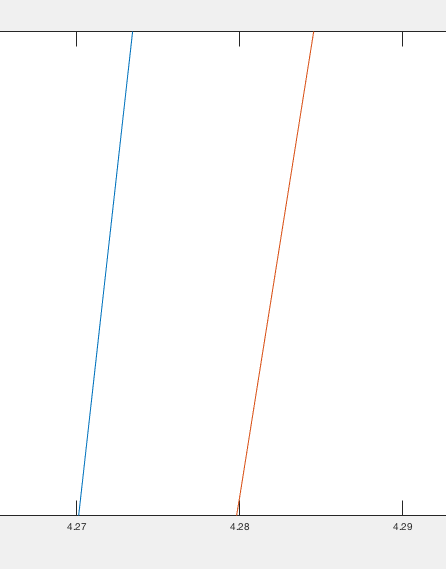




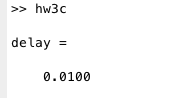


This filter works awesome, and much better than Butterworth filter. The line is almost coincident with line for X. This filter has less delay than Butterworth, and more smooth than Butterworth when position near 0. The line of this filter is even smooth than line for X.

Because at the beginning, we initialize , but we don’t know what exactly is, so we have some noisy at beginning, so we initialize .



When we zoom in the plot, we can find the delay is about 0.01.



Besides from the plot, we can also get delay from script by “finddelay” function. In conclusion, delay is 0.01.