ECE380 Project Task 1

Anthony Zhelnakov and Ahsen Qureshi

June 2024

1 Introduction

A mobile robotic system, with an unknown mathematical model, can be probed to observe its response to different inputs. The robot position output of the system, however, is noisy. A third order low-pass filter must be designed to to filter out this noise. The python script "task1.py" implements this, and is referenced in this document.

2 Task 1

2.1 Item 1: Inputs and Outputs

Constant input signals u_1 and u_2 with amplitude 1 must be applied to the robotic system over 15 seconds. The inputs are created on line 24 of "task1.py", and the unfiltered outputs y_1 and y_2 are obtained on line 29 using the provided SystemModel's "sim" method.

2.2 Item 2: Unfiltered Plots

The plots of the inputs and unfiltered outputs are shown in Figure 1. They are produced on lines 41-59 and 70-79.

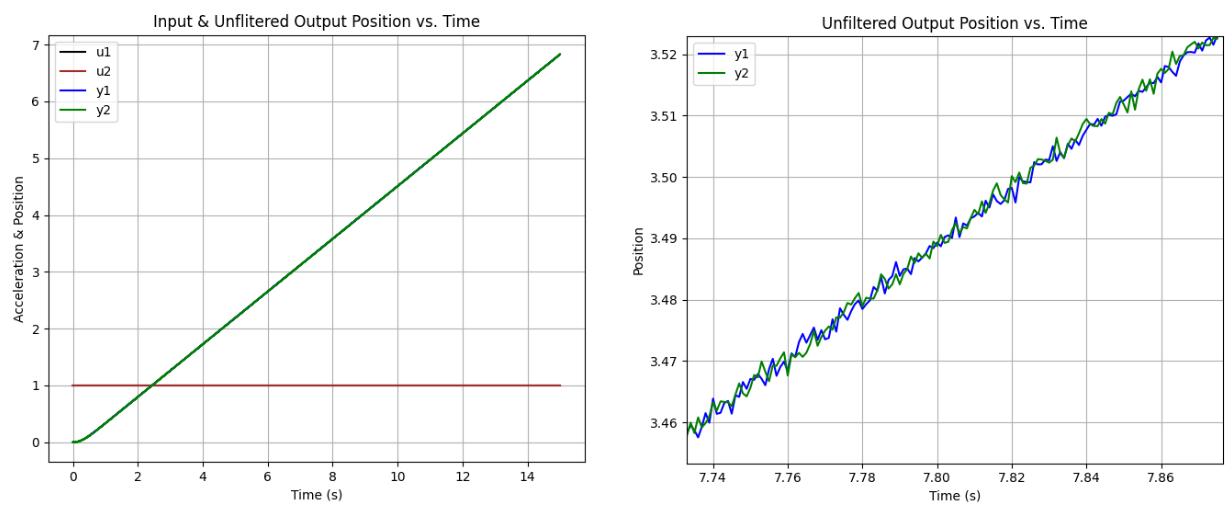

Figure 1: Plots of u_1, u_2, y_1, y_2 Signals and Close-Up of y_1 and y_2 Noise

2.3 Item 3: Filter Design

The requirement is to design a third order low-pass filter with cut-off frequency $\omega_c = 10$ and real and coincident poles. Pass-band gain is not needed. A filter transfer function which satisfies this criteria is

$$G_f(s) = \frac{1}{(\frac{s}{10} + 1)^3}$$

2.4 Item 4: Applying the Filter

To apply the filter to the output signals, whilst only being provided discrete output values of the robotic system, a discrete convolution can be done with the filter's impulse response and the output signals. The inverse Laplace transform of $G_f(s)$, and hence the filter's impulse response, is

$$h(t) = 500e^{-10t}t^2u(t)$$

This function is implemented on lines 10-11, and is used on line 20 to obtain the discrete impulse response.

The filtered output $y_f[\tau]$, where τ is a discrete value of time (in seconds) spaced apart by 0.001s, is then obtained from the convolution:

$$y_f[\tau] = 0.001 \sum_{k=-\infty}^{\infty} y[k]h[\tau - k]$$

where $y[x]$ is the system's position output, $h[x]$ is the filter impulse response in discrete time, and the scaling factor of 0.001 is the spacing between samples. In essence, a convolution is a sum of the areas of rectangles, and in the discrete case each area is $y[k]h[\tau - k]$ multiplied by 0.001 seconds. This is implemented using numpy's "convolve" function on lines 35 and 36 to obtain the filtered outputs.

Figure 2 below shows unit step input (u_1 & u_2), unfiltered output (y_1 & y_2) and filtered output (y_{1f} & y_{2f}) signals. The plots were produced on lines 81-90.

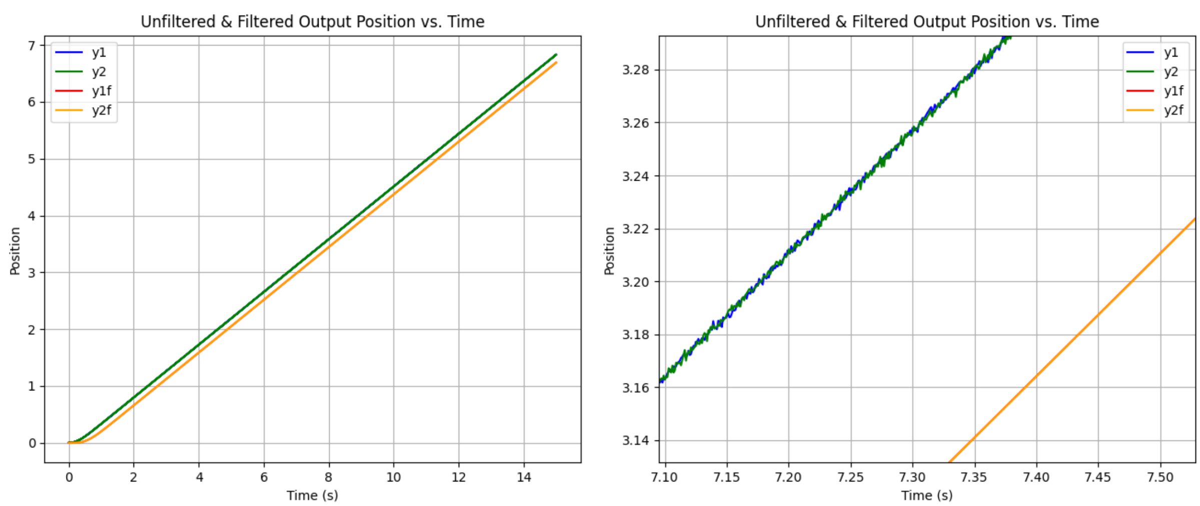

Figure 2: Entire and Close-Up Plot of y_1, y_2, y_{1f}, y_{2f} Signals

3 Conclusion

The filtered outputs can be observed to be slightly delayed from the original outputs, however, they are very steady with no visible fluctuations. The filter seems to have effectively reduced the noise of the robot position output signals.