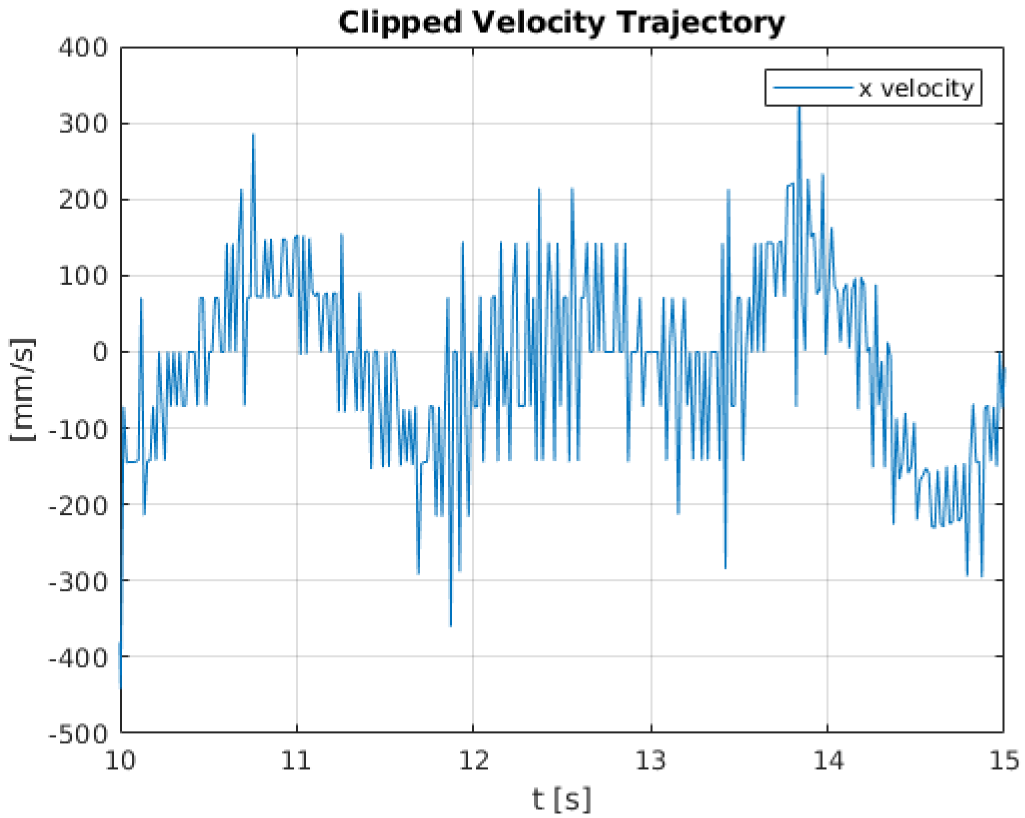
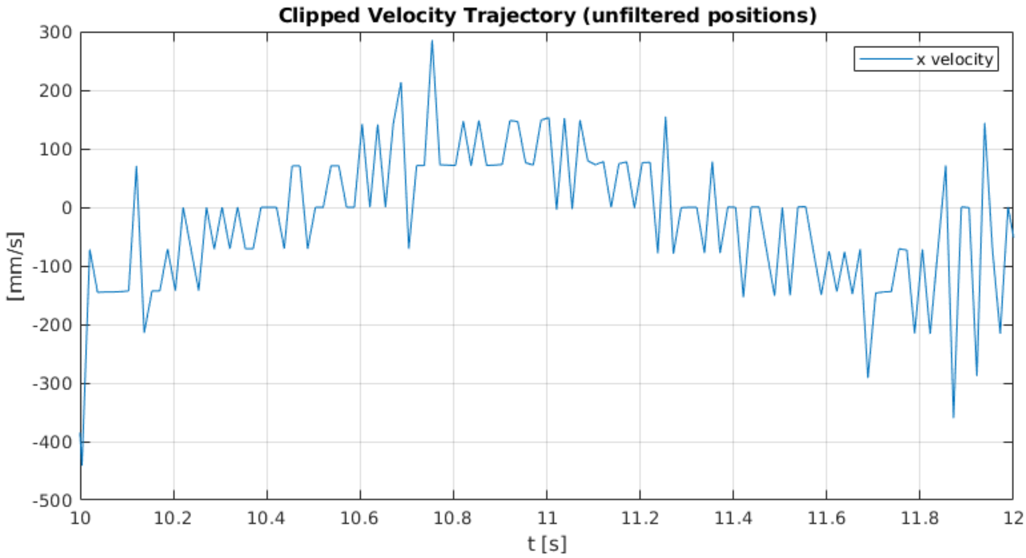
Lab 06 Group Tuesday 3_2

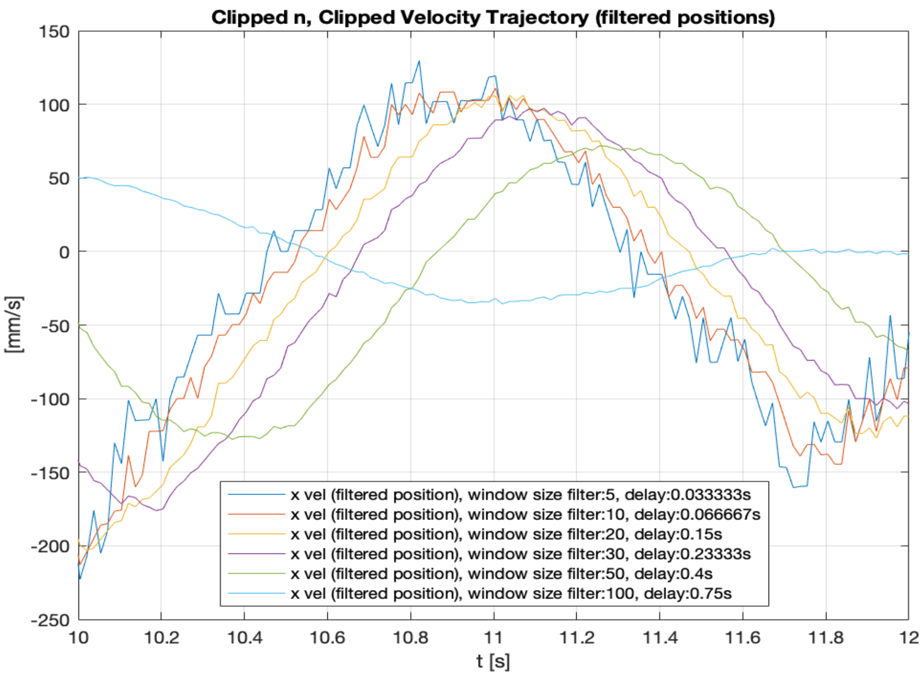
Q1



We can clearly see how the curve is not smooth but jumping around due to noise in the position estimations. This in turn would lead to "jumpy" behaviour when applied to our pid controller since our derivative term depends on the velocities: u_d[k] = v_ref[k] - v[k]. Therefore, we would prefer to use a smoothed version of the velocity estimates by e.g. removing noise by filtering either the positions or the velocities.

Q2





We can see how by increasing the window size the velocity estimation gets smoother. However, at the same time we gain more delay. Ideally, we would want a noise-free (smooth) and delay-free velocity estimation. Hence, we have to find the optimal trade-off between “smoothness” and time-delay.
Qualitatively, this makes sense, since when we have a window size of n, we:

- Get an average over the past n position readings -> reduces noise
- Get a time delay, which theoretically is: delay = dt * (n - 1) / 2

Given our observations and requirement of delay <= 0.16s, we choose the window size filter 20 from our experiments, since it is the smoothest out of all valid options which satisfy our requirement.

Q3

Done in code

Q4

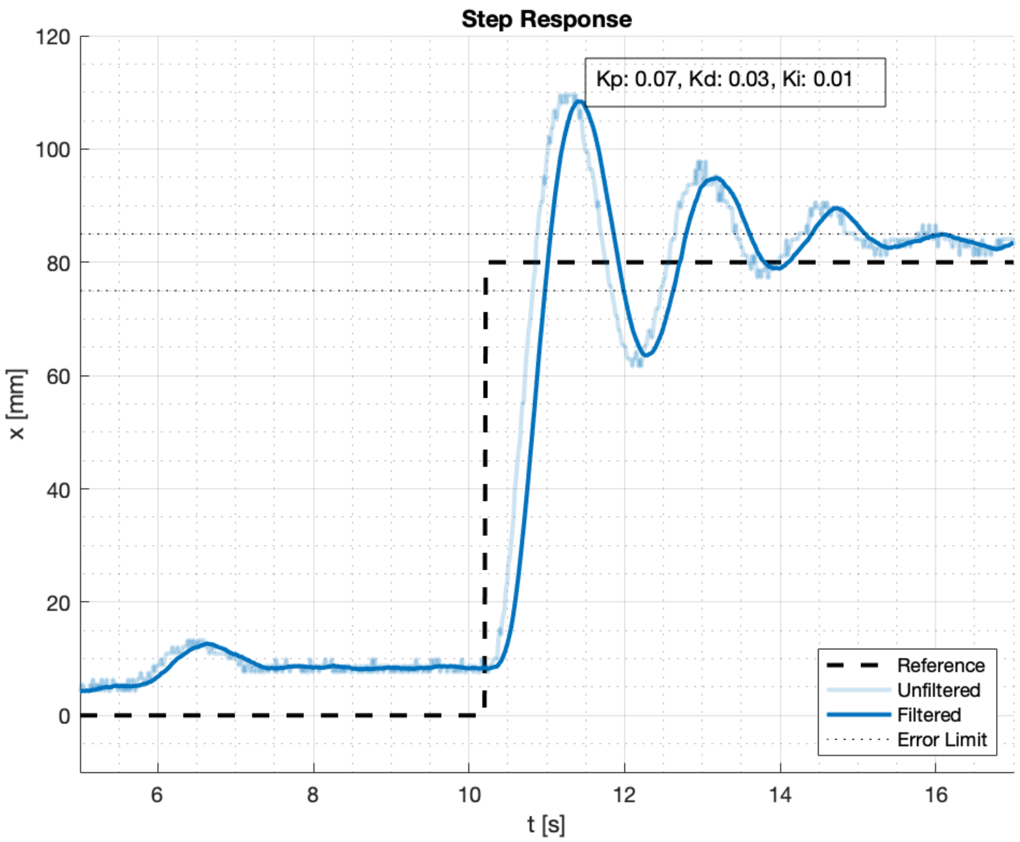
Done in code

VIDEO HAND IN OF BALL STABILIZED AROUND EQUILIBRIUM

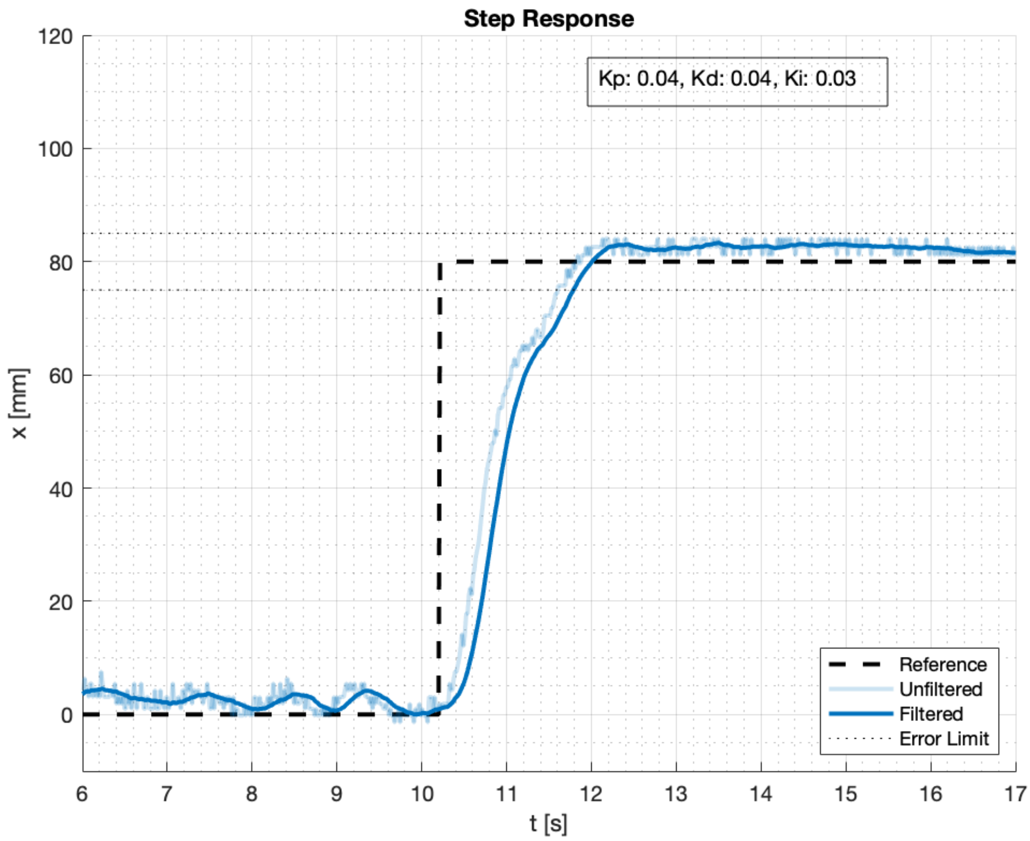
Q5

Done in code

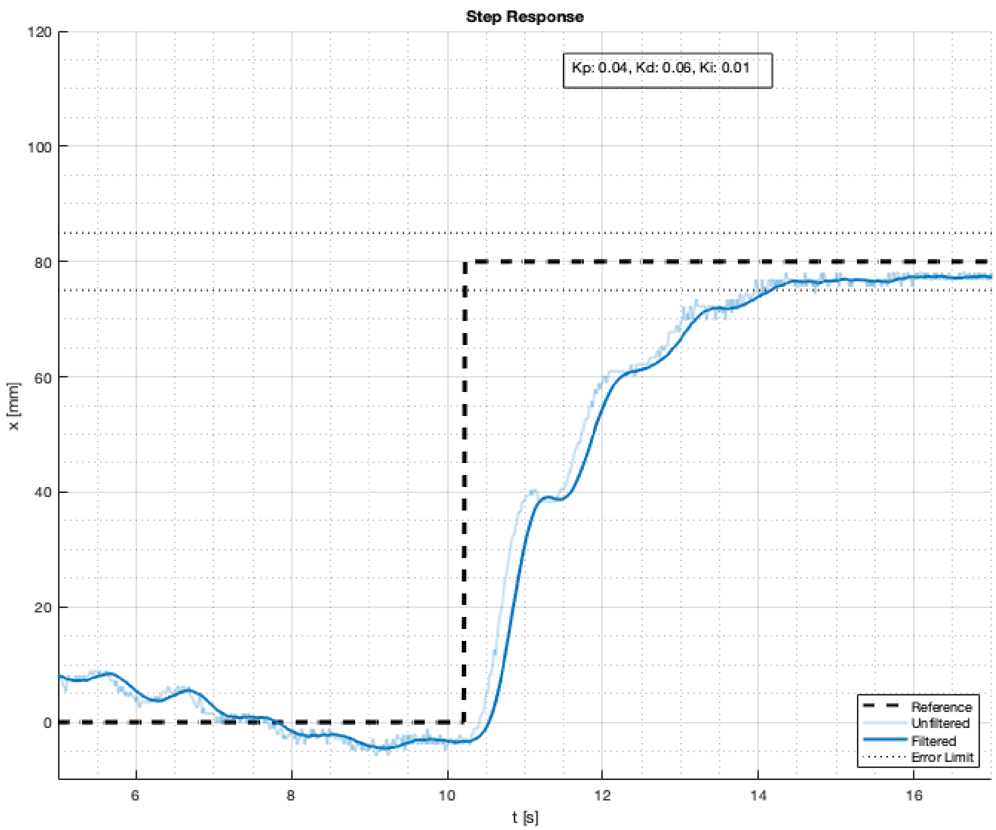
Underdamped System:



Critically damped System:

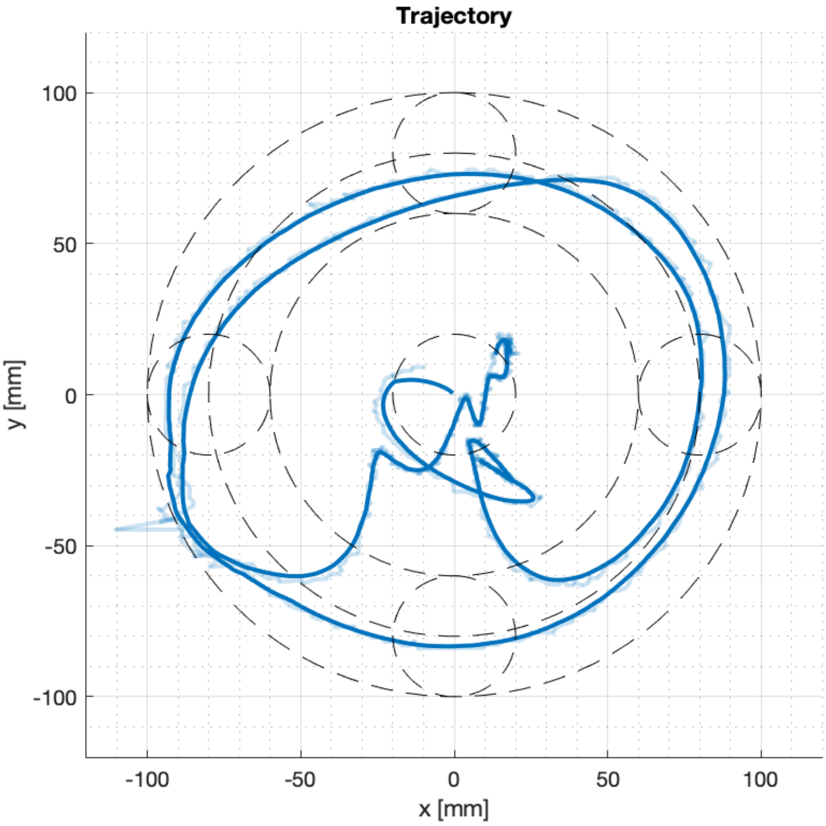


Over damped System:



Q6

Circular Trajectory two revolutions within margin:



Q7 bonus

Not implemented (butterworth filter)