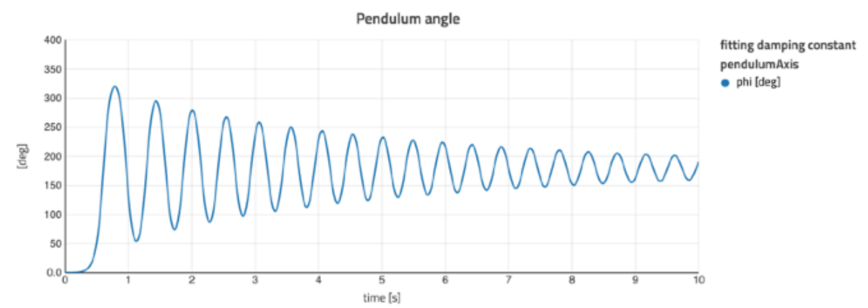
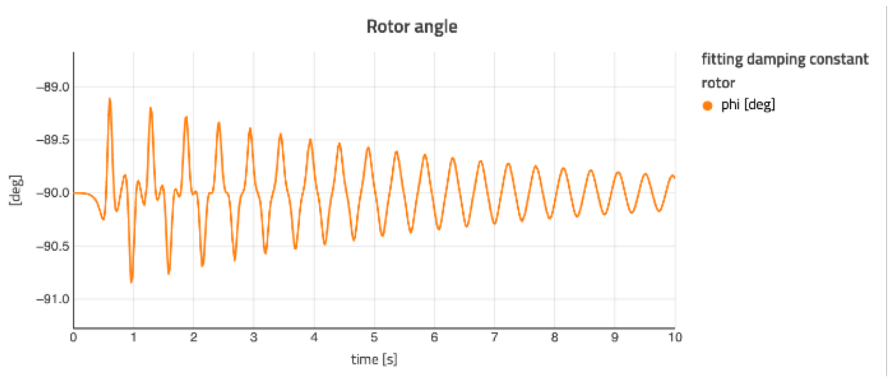
Task 1.1:
I fitted the two damping constants by iteratively testing different values and examining how they compared real experiment plots. I fitted the pendulum damping by comparing the amplitude of the sinusoidal of the simulation with that of the real process and lowered the constant if the amplitude decreased too fast, and vice versa. I figured it was more important to examine amplitudes at the latter part of the simulation since it is easier to distinguish the effect of the dampening envelope in this part. The rotor damper was more difficult to fit, since the realisation was less stable and therefore not very similar to the simulation. My approach was to match the amplitude of the first rotation as good as possible. The resulting constants I used was:

Pendulum damper	0.000003
Rotor damper	0.015
Second pendulum damper	0.0000015

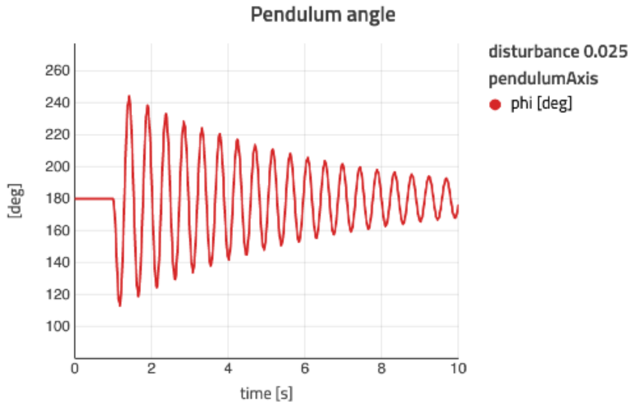
where the second pendulum damper was used in task 2.

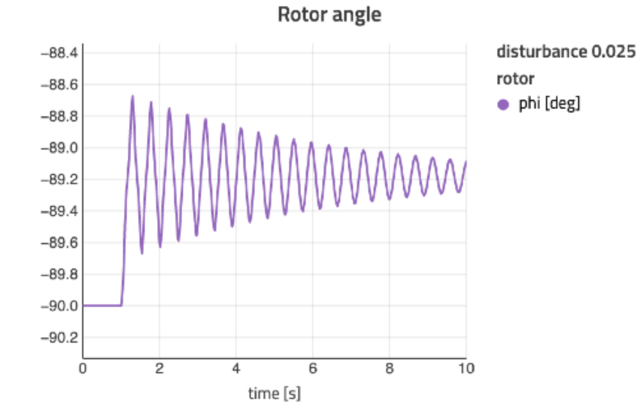
The resulting graph of the pendulum and rotor angles:

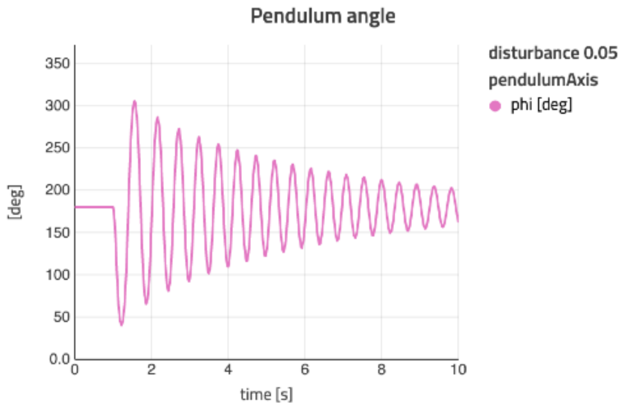


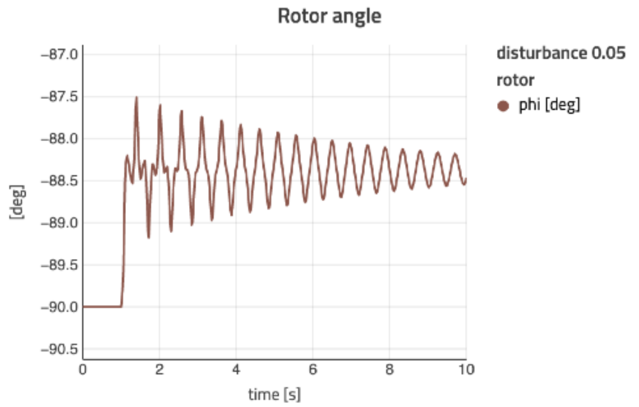


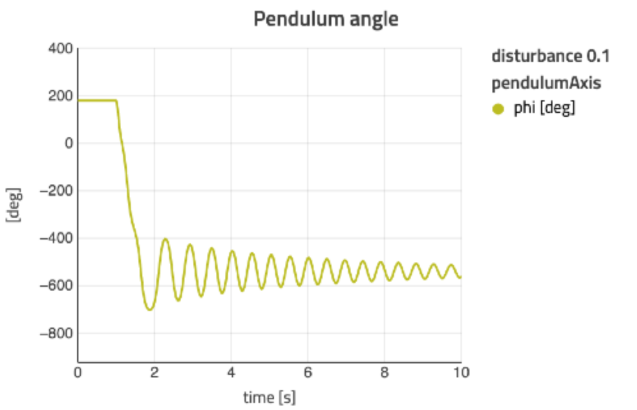
Task 1.2:
The pendulum and rotor angles for rectangle pulse disturbances one second into the simulation with amplitude 0.025, 0.05 and 0.1 are illustrated below. Note that in the last case, the powerful disturbance causes the pendulum to rotate two full revolutions, which is why it stabilises at -540 degrees. Worth noting is also that for a small disturbance, 0.025, the rotor is more stable and does not twitch as much as it does for greater disturbances.

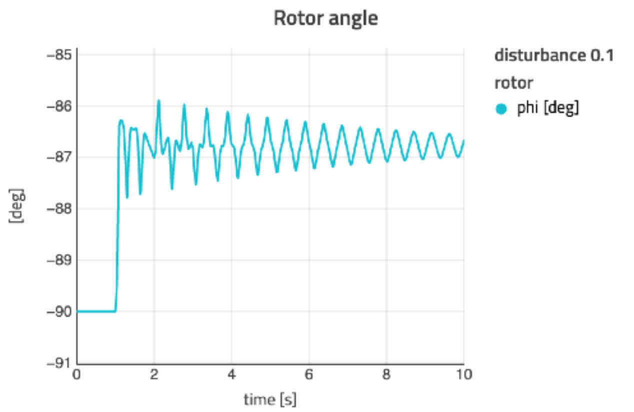




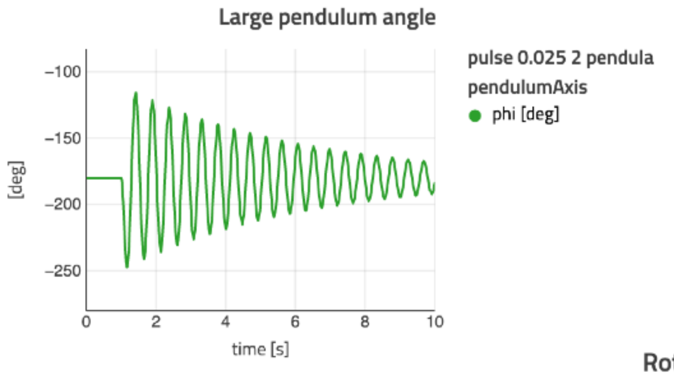


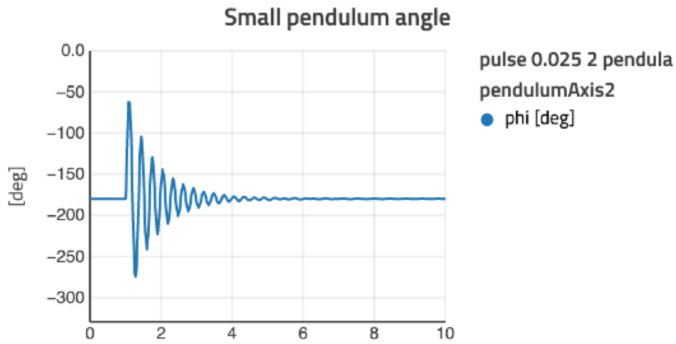


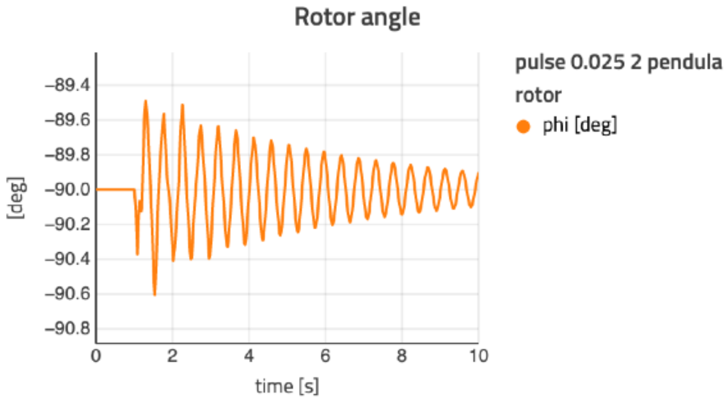


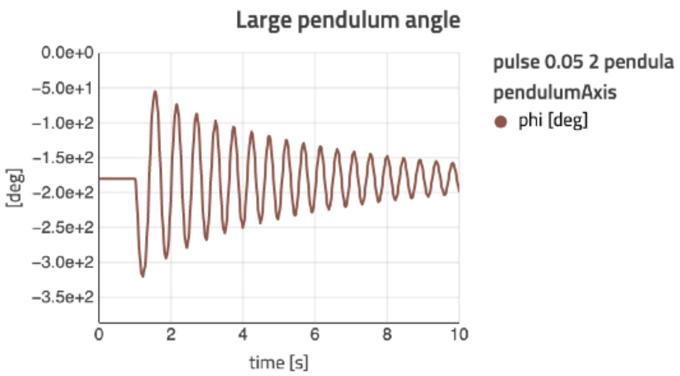


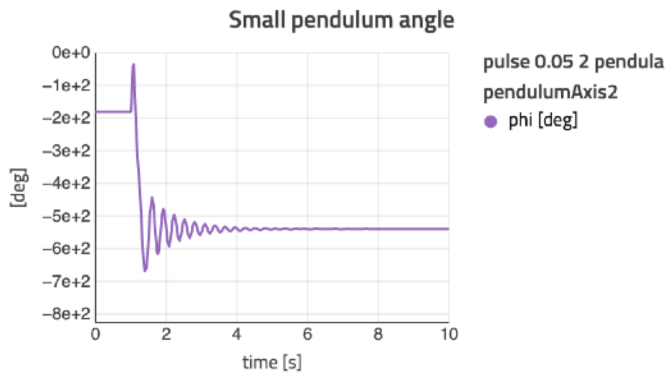
Task 2.1:
When testing with two pendula, it becomes apparent that the smaller pendulum has less inertia and is in motion for a shorter period of time. It also performs full revolutions at smaller disturbances than the larger pendulum. An interesting takeaway is that the small pendulum does not perform a full revolution for the greatest disturbance. This could have to do with the fact that the large pendulums rotation stabilises it.

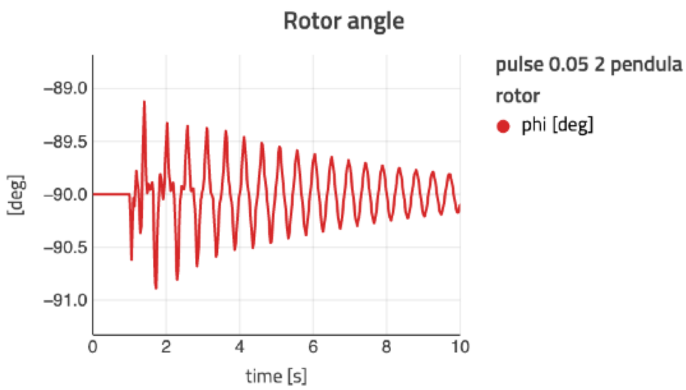


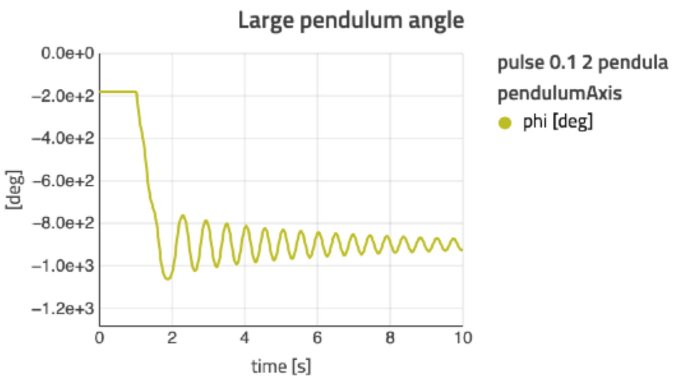


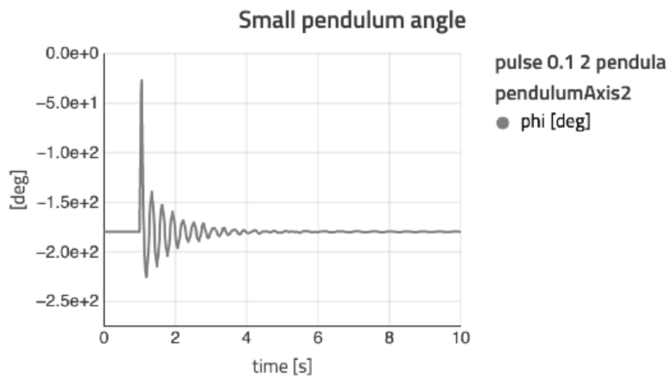


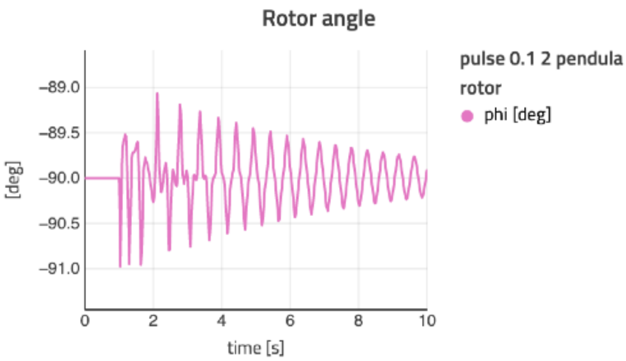




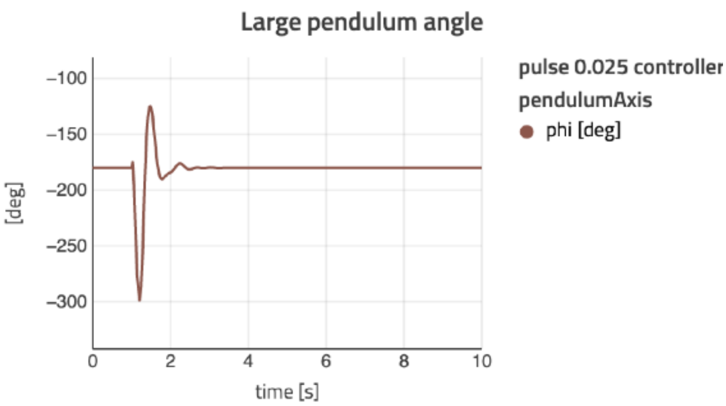


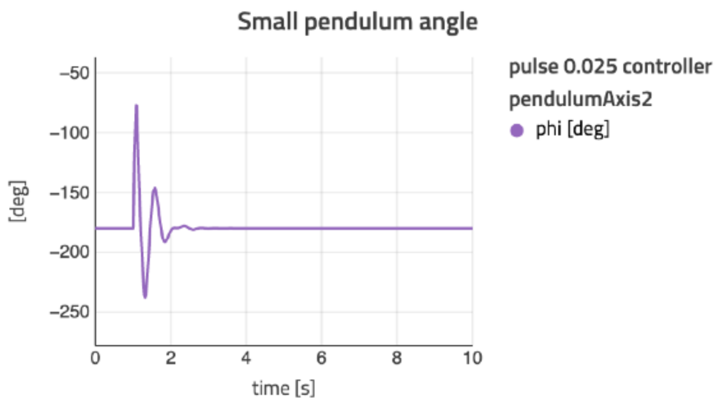


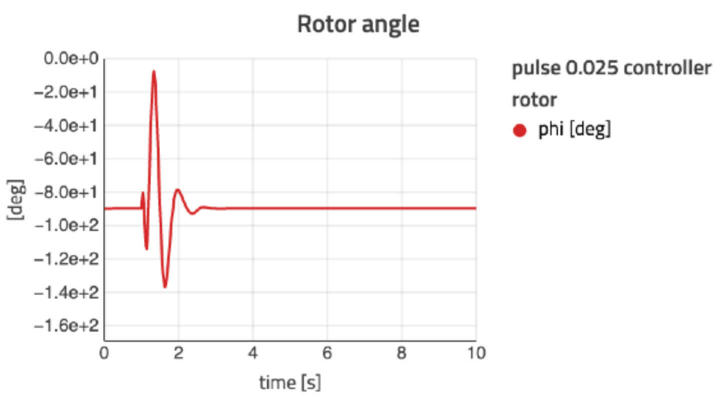


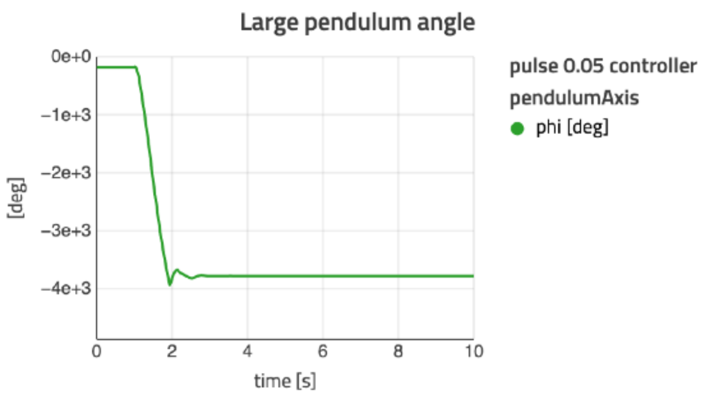


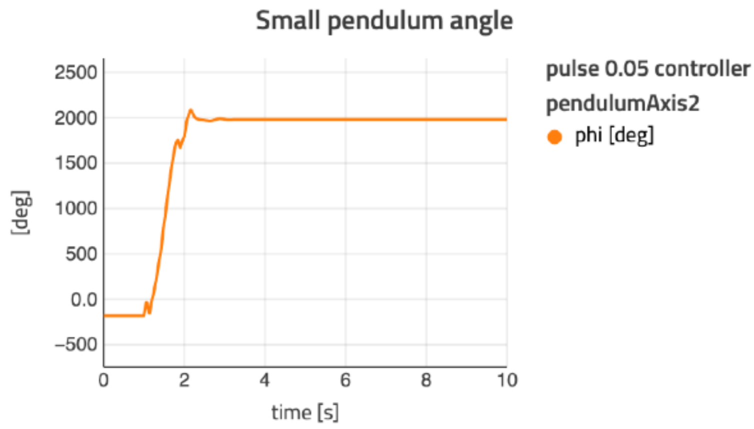
Task 3.
After implementing the controller according to the instruction I tested it for the same disturbances as before. The resulting motions are displayed in the graphs below. For small disturbances the approximation of a linear system is accurate enough to work for the controlling equation. For greater disturbances, the system loses linearity which leads to inaccurate controlling. However, even for the greater amplitudes, the controller finds a stable state faster than the uncontrolled system, although being after several revolutions.

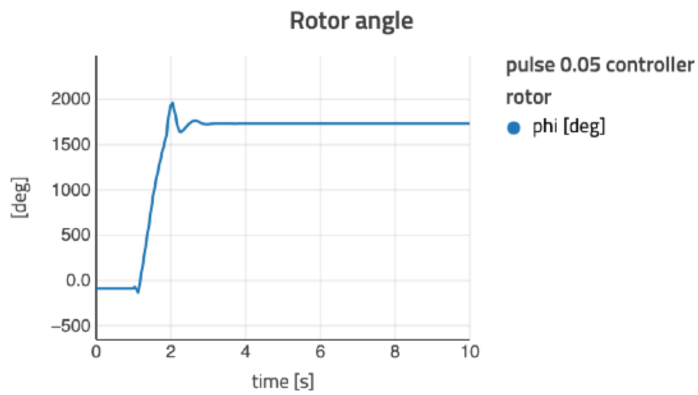


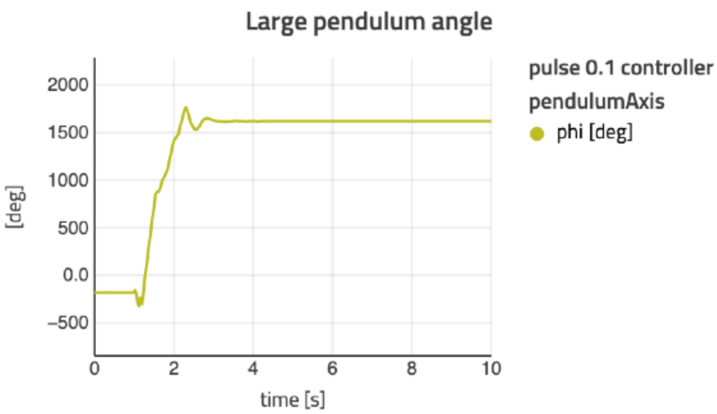


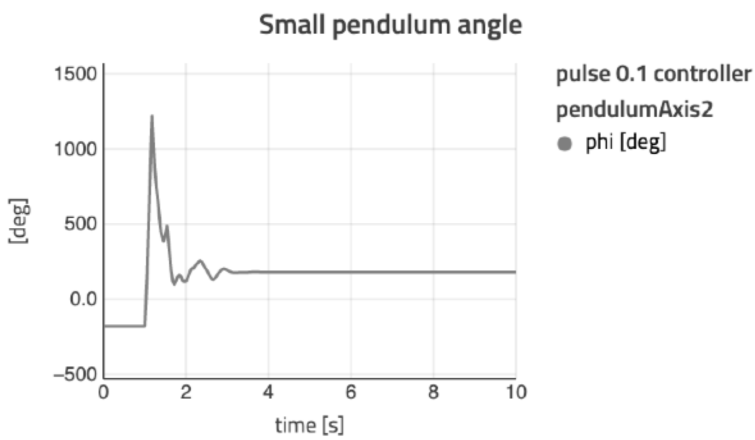


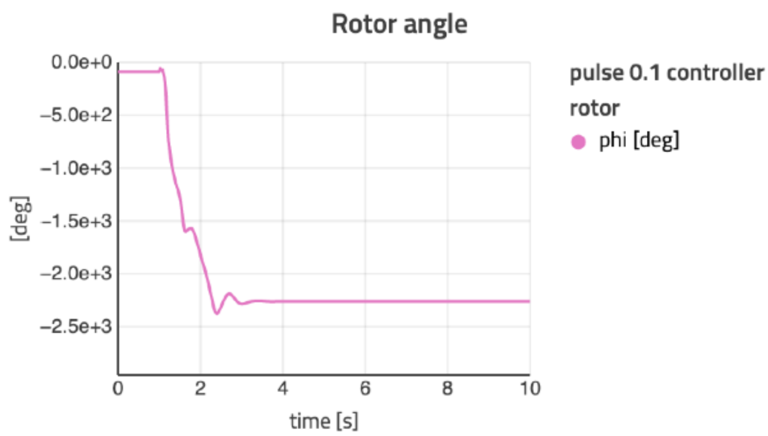

Large pendulum angle
pulse 0.05 controller
pendulumAxis
phi [deg]
[deg]
time [s]
0e+0
−1e+3
−2e+3
−3e+3
−4e+3
0
2
4
6
8
10


Small pendulum angle
pulse 0.05 controller
pendulumAxis2
phi [deg]
[deg]
time [s]
2500
2000
1500
1000
500
0.0
−500
0
2
4
6
8
10

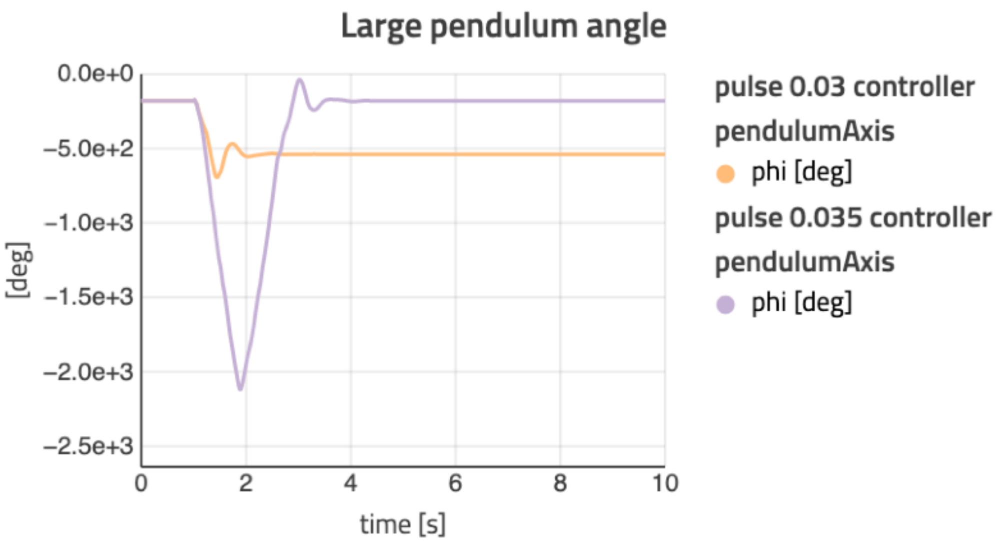

Rotor angle
pulse 0.05 controller
rotor
phi [deg]
[deg]
time [s]
2000
1500
1000
500
0.0
−500
0
2
4
6
8
10

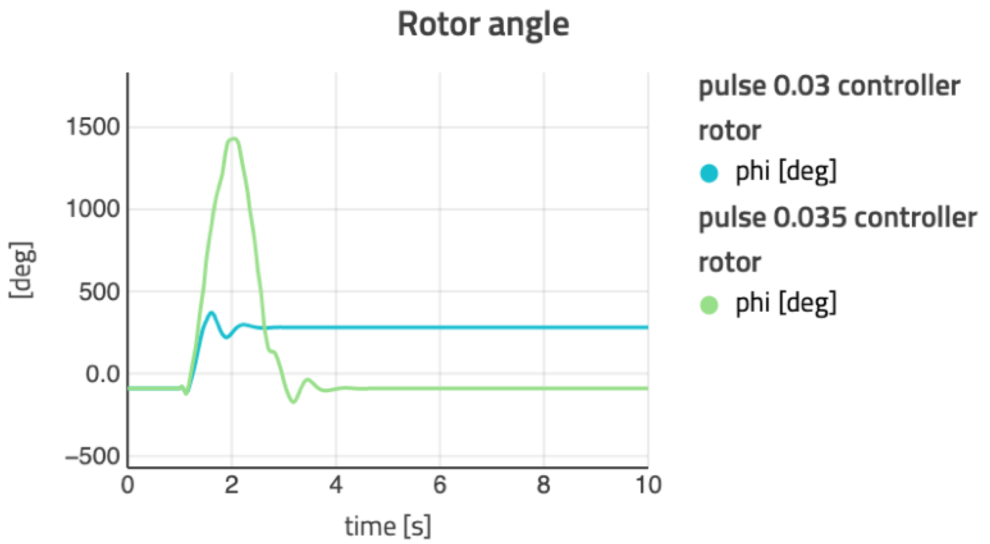

Large pendulum angle
pulse 0.1 controller
pendulumAxis
phi [deg]
[deg]
time [s]
2000
1500
1000
500
0.0
−500
0
2
4
6
8
10

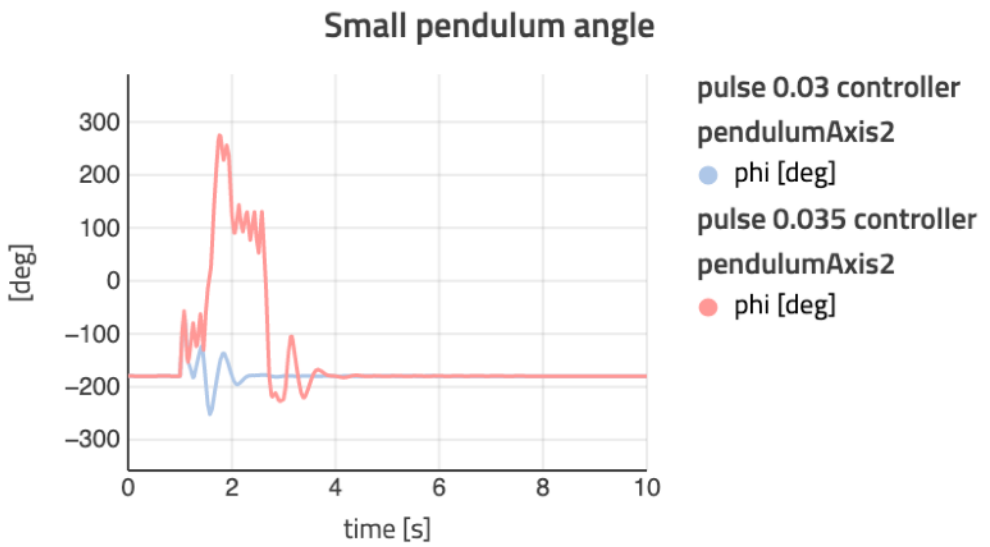

Small pendulum angle
pulse 0.1 controller
pendulumAxis2
phi [deg]
[deg]
time [s]
1500
1000
500
0.0
−500
0
2
4
6
8
10


Rotor angle
pulse 0.1 controller
rotor
phi [deg]
[deg]
time [s]
0.0e+0
−5.0e+2
−1.0e+3
−1.5e+3
−2.0e+3
−2.5e+3
0
2
4
6
8
10

I wanted to find the region where the linear approximation does not suffice anymore, and I therefore tested disturbance amplitudes slightly greater than 0.025 to find the boundary. Somewhere between 0.03 and 0.035 the controller goes slightly crazy before finding the stable region as seen in the plots below:







Extra:
I wanted to optimise the Q-vector in order to faster stabilise the pendulum, so I tried different values to penalise velocity more/less and angle more/less. None of them were successful but a funny thing happened when I removed the penalty for rotor angle, and raised the penalty for pendulum angular velocity. This resulted in a first stage of chaos and then an equilibrium where the pendula got fixed in a midway-position at 90 degrees, spinning around the y-axis. Se plots and 3D animation below. The small pendulum spinns 187 revolutions and is then stuck at 90 degrees. The large pendulum spinns a slightly more modest 16 laps and is then stuck at 90 degrees. This was for an L-vector {0, 7.07106781, 60.3939673, 9.87026104, 229.774066, 2.14595225}.