Haptic Human-Robot interfaces: Lab 3

Rovina Hannes Kaspar
hannes.rovina@epfl.ch
247575

Srinath Halvagal Manu
manu.srinathhalvagal@epfl.ch
280594

Assistants:
Romain Baud
Jacob Hernandez Sanchez
Philipp Hörler

April 17, 2018

1 Introduction

The goal of this lab was to build a simulink model of the haptic paddle in order to safely implement a PID regulator on the hardware. Once the model was completed, it was compared to the real setup in order to validate it. The results from the model were used to tune a PID regulator on the hardware to reach a specified performance and the influence of the different PID parameters on the step response was studied.

2 Method

Using the haptic paddle the dry friction of the paddle was estimated and using this value, a model for the static friction was added to the simulink model of the paddle. Furthermore, the parameters for Simulink's inbuilt PID regulator were tuned in order to have an estimate of the parameters needed for the PID regulator on the actual haptic paddle. Once a stable and desired performance was achieved in simulation, the PID regulator was implemented on the hardware. With the help of the graphical interface along with a few additions and adjustments through the gui, the PID regulator was tuned using an experimental approach as well as the Ziegler-Nichols method. Since the haptic paddle has an optical encoder as well as a hall sensor for position measurement, both sensors were used separately for the PID regulator in order to compare their individual performance for position control.

3 Results

3.1 Simulation

3.1.1 Modeling the dry friction

The dry friction arises due to many factors, the main one being the friction of the cable on the screw which is strongly influenced by the tension in the cable. The tension can change over time. Therefore the estimation needs to be updated every time before using the paddle.

The estimation of the dry friction was achieved by applying an incremental motor torque on the paddle until it started to move. This experiment yielded the result that motion started from a motor torque of $T_0 = 0.001Nm$.

Modeling the dry friction can be very tricky, and we needed to make quite a few heuristic adjustments. The first basic condition to check is if the torque is high enough to overcome the value T_0 recorded before, and only allow motion if this condition is true. Once the paddle has started moving for the first time, the dry friction has a different behavior and a memory block is needed to remember that motion has already started.

Once the paddle has started moving, the dry friction has a constant value equal to T_0 and a direction always opposing the paddle motion. To model this, we look at the sign of the paddle velocity and apply a dry friction in the opposite sense. Since we relied on the sign function, we faced problems when the velocity was close to zero (i.e, at the extreme positions). The velocity would sometimes switch sign in an unpredictable way around zero, possibly because of the numerical differentiation used for calculating the velocity. To avoid this instability, at velocities lower than a certain threshold, we set the friction to zero. Figure 1 shows a close up of the dry friction model we implemented for our Simulink model of the paddle and the complete model can be seen in Figure 2.

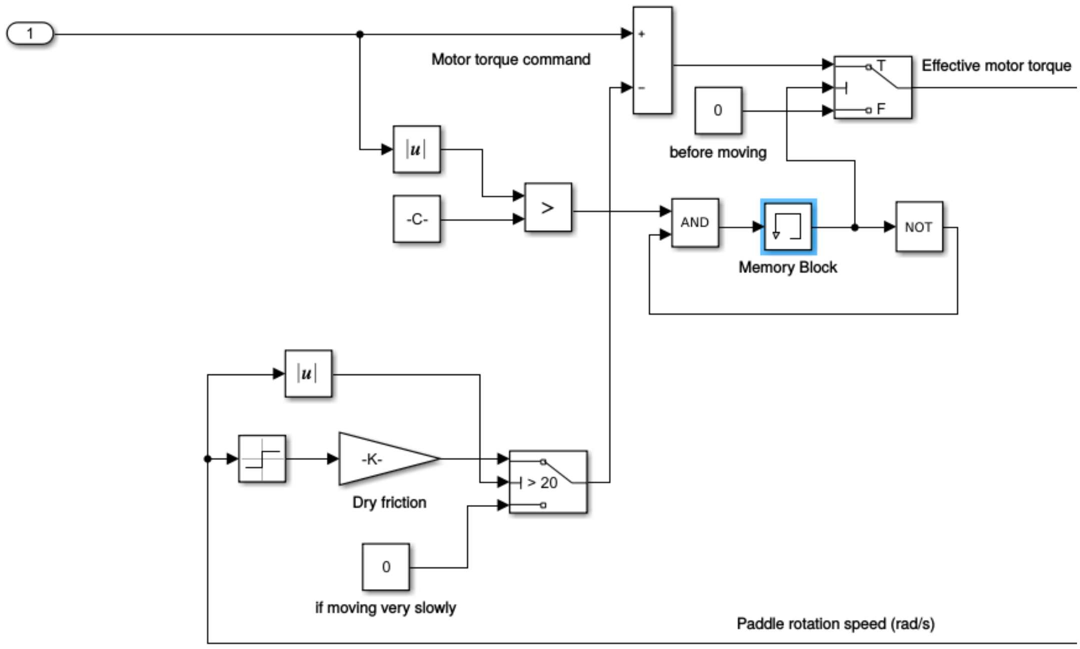


Figure 1: Close up of the dry friction model

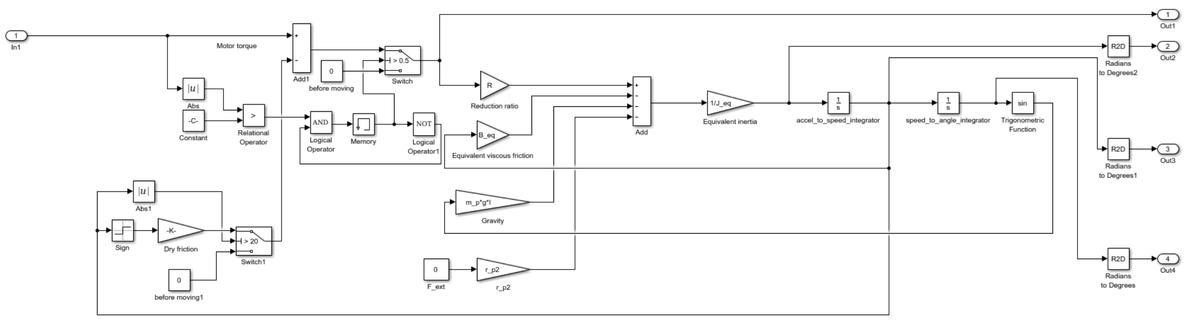
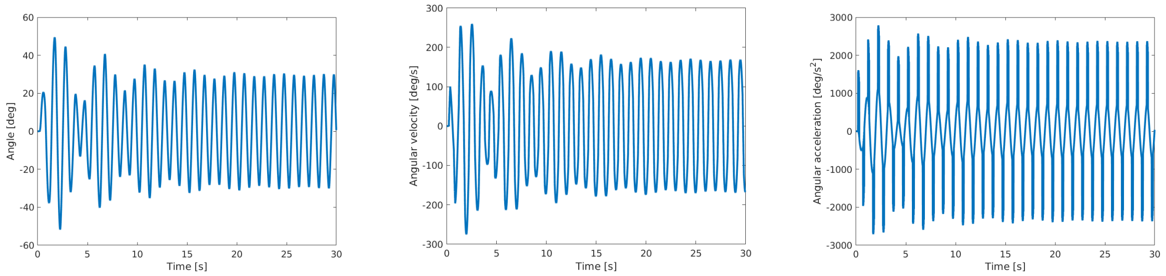


Figure 2: Simulink model of the paddle with dry friction

3.1.2 Model validation

To validate the correctness of our model, trajectories for several different sinusoidal torques were simulated. As can be seen in Figure 3a to 3c, the paddle does not swing out of bounds for the applied sinusoidal motor torque of 0.00101 Nm. Nevertheless nonlinear effects due the friction model can be seen at the turning points of the paddle where there are large spikes in the acceleration. It also takes some time for the amplitude of motion to settle down to the steady value that we expect. When applying a slightly higher motor torque of 0.0011 Nm, it can be observed in Figure 4 that the paddle drifts initially out of bounds up to an angle of 45 degrees since there are no stoppers at ± 35 degrees implemented in the model. Further increasing the sinusoidal torque yields to angles in the order of 100 or even 1000 degrees.



(a) Paddle angular displacement (b) Paddle angular velocity (c) Paddle angular acceleration

Figure 3: Analysis of the simulink model for a sinusoidal torque of 0.00101 Nm (slightly higher than the dry friction 0.001 Nm) and visualization of the nonlinear effects at the turning points. We see that a sinusoidal torque of amplitude 0.00101 Nm gives a position within bounds of the paddle

3.1.3 Comparison between simulation and haptic paddle

Figure 4 compares the modeled displacements with the actual paddle positions in response to a sinusoidal torque of an amplitude of 0.0011Nm. We see that the model requires some time ($\sim$ 12s) for its estimates to settle down. We also observe a similar, but less pronounced transient variation in the amplitude even in the case of the real paddle. Apart from a slight mismatch in the amplitudes due to modeling errors, overall the modeling seems to be reasonably accurate.

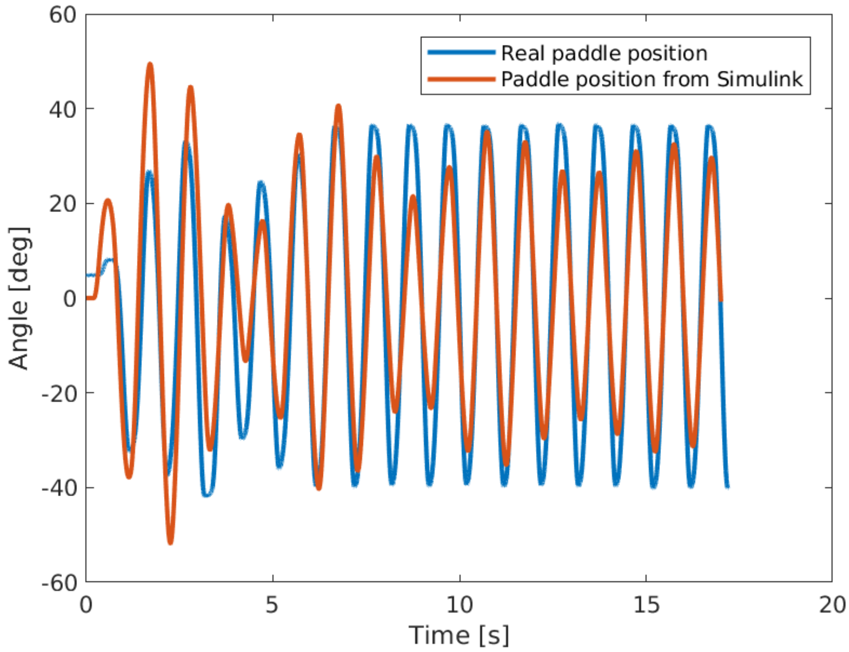


Figure 4: A time-aligned comparison of the displacement predicted by the simulation and the displacement actually observed on the paddle in response to a sinusoidal torque of 0.0011Nm. The displacement goes out of bounds initially in case of the Simulink model demonstrating the inexactness of our friction model

3.1.4 PID regulator

After the model validation, a PID regulator was added to the model (Figure 5) in a feedback loop in order to control the position. After some experimental tuning, the following parameters for the PID regulator were found:

$$\tau_M = K_P \times error(t) + K_I \int_0^t error(t)dt + K_D \times \frac{d}{dt} error(t)$$

$$K_P = 0.01 \qquad K_I = 0.1 \qquad K_D = 0.0001$$

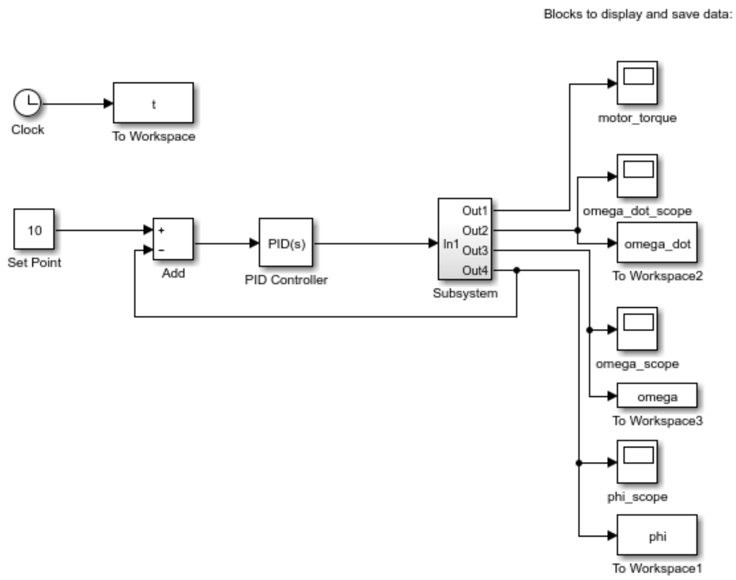


Figure 5: Simulink model with a PID regulator controlling the position of the paddle in a feedback loop

Using this PID regulator with the parameters given above, the step response for a step of 10 degrees was simulated (Figure 6). As can be seen in the figure, the overshoot is 1.3°, the rise time is under 0.05 seconds and the settling time is 0.23 seconds. This satisfies the specifications given for the regulator, namely:

- Maximum overshoot: 25% (2.5°).
- Maximum settling time to reach an error band of ±0.2°: 350 ms.
- Maximum rise time: 100 ms.

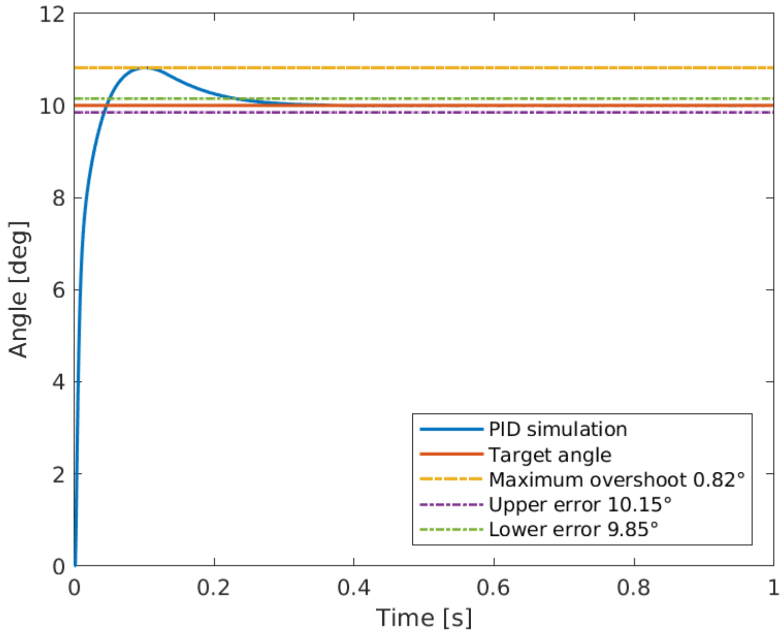


Figure 6: Simulated step response for a target angle of 10°, with overshoot of 0.82°, rise time of 46 ms and a settling time of 228 ms for an error band of ± 0.15 °

3.2 PID tuning on hardware

The strategy used to tune the PID regulator on the hardware consisted of changing one by one the gains of the regulator starting with the derivative gain and taking the order of magnitude of the gains in the simulated regulator as a reference. The tuning was done using the hall sensor for position estimation. The derivative gain K_D was increased until a noticeable feedback was recognized when moving the paddle. This corresponded to a gain $K_D = 0.0003$.

Then the proportional gain was increased to get the desired rise time of the step response. As can be seen in Figure 7a using a proportional gain of 0.01 the regulator never reaches the target angle but has a very direct transition. Therefore K_P was further increased until the rise time and settling time were in the specified range. For $K_P = 0.05$ this is the case but this gave a maximum overshoot of 4.42°while our target was to keep it below 2.5°as illustrated in Figure 7b.

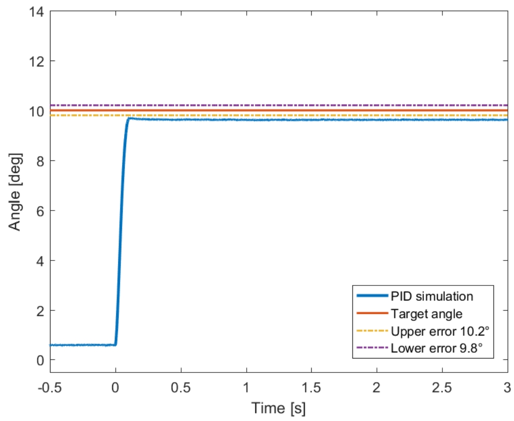
To decrease this overshoot and the initial oscillations, the derivative gain was increased and the desired specifications could be met with a derivative gain of 0.0008. The result of this regulator configuration is plotted in Figure 7c.

Because the K_I gain was not used until this point, we would observe a small but finite steady state error. To eliminate this, the integral gain K_I was increased from 0 steadily until 0.28 at which point a further increase would lead to the overshoot and settling time increasing too much. Figure 7d shows that with the integral gain of 0.28 the overshoot is increased to 1.59°but the steady-state error nearly vanished as expected due to the integral gain.

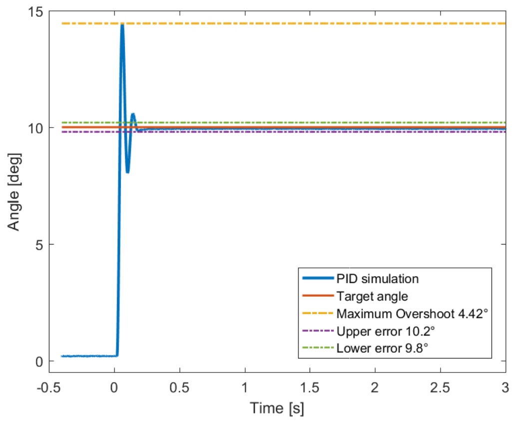
Comparative plots of the step responses for the same regulator gains as before but using the optical encoder for position measurement are illustrated in Figure 8.

3.2.1 Results

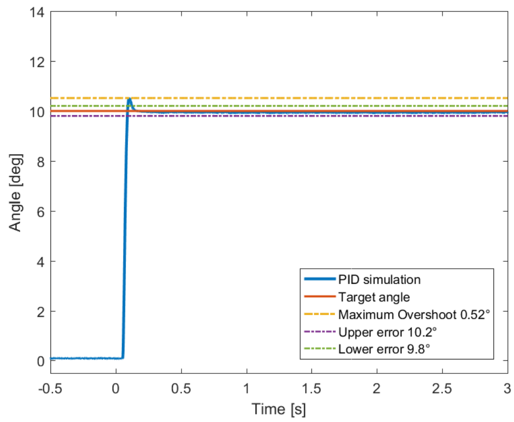
A summary for the tuning process as well as the comparative results when using the optical encoder is shown in Table 1.



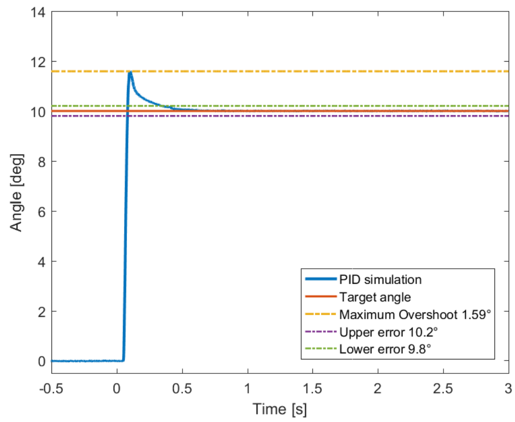
(a) $K_P = 0.01$ $K_D = 0.0003$
t_rise=infinity t_settle=infinity



(b) $K_P = 0.05$ $K_D = 0.0003$
t_rise=22.05ms t_settle=135.80ms



(c) $K_P = 0.05$ $K_D = 0.0008$
t_rise=36.05ms t_settle=66.85ms



(d) $K_P = 0.05$ $K_I = 0.0001$ $K_D = 0.0008$
t_rise=31.15ms t_settle=296.45ms

Figure 7: 4 responses of the PID regulator with different gains corresponding to the tuning process for a step input of 10°and using the hall sensor for position measurements

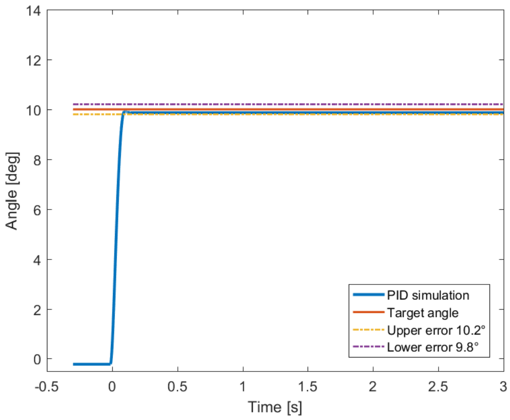
	K_P	K_I	K_D	Overshoot	Rise time(ms)	Settling Time(ms)
Hall sensor	0.01	0	0.003	0°	∞	∞
	0.05	0	0.003	4.42°	31.15	296.45
	0.05	0	0.0008	0.52°	36.05	66.85
	0.05	0.0001	0.0008	1.59°	31.15	296.45
Optical Encoder	0.01	0	0.003	0°	∞	9.85
	0.05	0	0.003	4.69°	25.90	∞
	0.05	0	0.0008	0.65°	40.95	75.60
	0.05	0.0001	0.0008	1.82°	35.00	325.50

Table 1: Performance of the regulator for different values of the PID coefficients

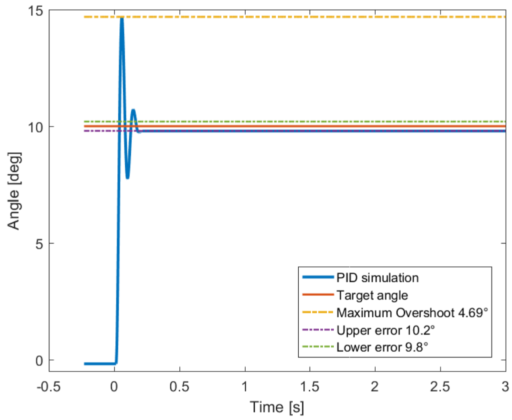
As can be seen in the summary of the step responses in Table 1, the results are quite similar for the hall sensor and optical encoder position feedback loops. Since the optical encoder gives position measurements in steps, the regulator is a little bit slower and therefore a slightly worse performance because it has these delays in the position measurement due to the quantization error.

3.2.2 Importance of filtering

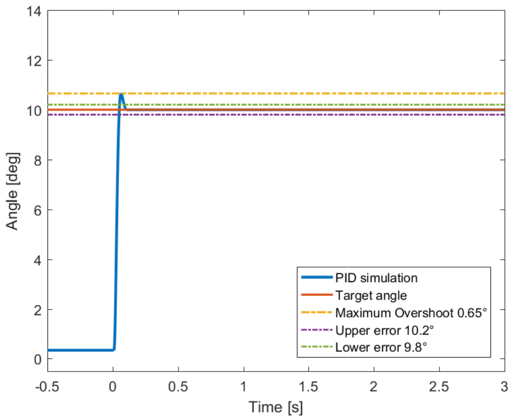
The position input to the PID regulator was filtered to smooth the derivative of the error and to reduce amplification of noise peaks. One could also have filtered the error derivative again but this was not necessary since the performance of the regulator was already sufficient without this step. Noise peaks in the position signal would have been amplified upon differentiation (for using the differential gain) and would have degraded the step response. Due to the filtering, this is avoided and larger gains can be used. Therefore filtering the position measurement is very important to obtain a good regulator.



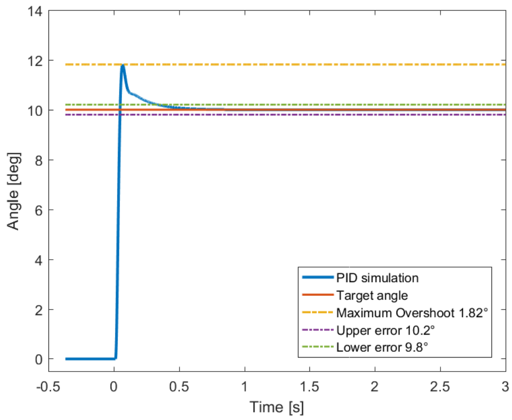
(a) $K_P = 0.01$ $K_D = 0.0003$
t_rise=infinity t_settle=9.85ms



(b) $K_P = 0.05$ $K_D = 0.0003$
t_rise=25.9ms t_settle=infinity



(c) $K_P = 0.05$ $K_D = 0.0008$
t_rise=40.95ms t_settle=75.60ms



(d) $K_P = 0.05$ $K_I = 0.0001$ $K_D = 0.0008$
t_rise=35.00ms t_settle=325.5ms

Figure 8: 4 responses of the PID regulator with different gains corresponding to the tuning process for a step input of 10°and using the optical encoder for position measurements

3.2.3 Ziegler-Nichols tuning

The first step of the Ziegler-Nichols tuning procedure is to keep increasing K_P with the other gains at zero until sustained oscillations are observed. The corresponding gain K_u and oscillation period T_u were found experimentally to be 0.05 and 86.7ms respectively. Next the gains for the PID regulator are calculated as follows:

$$
\begin{aligned}
K_P &= \quad 0.6 \times K_u \quad = \quad 0.03 \\
K_I &= \quad \frac{K_P}{T_u/2} \quad = \ 0.692 \\
K_D &= K_P \times T_u/8 = 0.000325
\end{aligned}
\tag{1}
$$

These values are in the same order of magnitude as we obtained with the manual tuning procedure. The step response with these parameters (on the real paddle) are shown in Figure 9. We see that while the rise time and settling time are within the specifications, the overshoot is much too high (10.2°in this case). This can be countered by reducing the integral gain and increasing the derivative gain.

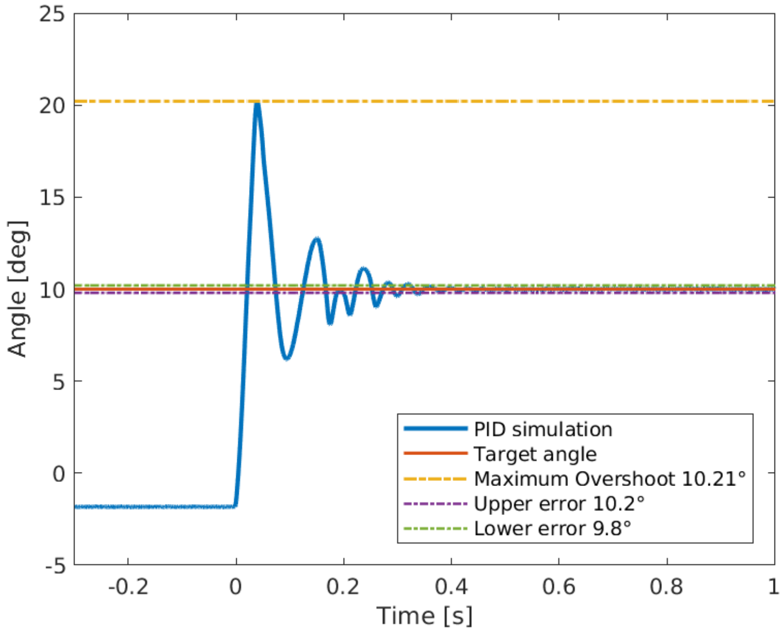


Figure 9: Step response with the parameters given by Ziegler-Nichols tuning, with an overshoot of 10.2°, rise time of 21ms and a settling time of 341.25ms for an error band of ± 0.2°

4 Conclusions

In this laboratory session the use of a simulink model enabled us to safely tune a PID regulator on the hardware of the haptic paddle. The simulation gave a good starting point for tuning the PID parameters on the hardware and the final gains on the actual paddle were in the same order of magnitude as in the simulation. Furthermore it was realized that the hall sensor and the optical encoder both work well for position control with the right gains of the controller. With the Ziegler-Nichols tuning method, the paddle could also be controlled but the performance of the regulator was worse compared to the one that was hand-tuned and did not meet the performance criterion. Nevertheless, it could also serve as an alternative first hint for the tuning of the parameters instead of trial and error from the start.