

ELEC-E8101 Group project:

Lab B report

Group 11

HOUILLE Florent, MONTE SASTRE Enrique, SOULARD Yoann, WANG Tong

November 12, 2019

Reporting of Task 5.1

We implemented our PID with the values we obtained in lab A. Unfortunately, the sampling time (42ms) was not working at all. The robot's wheels were just stuck, making a nearly imperceptible back and forth movement. If the sampling time would have been too high, the system would have been unstable and the command would have maxed the limit out right at the beginning. So we didn't find a good reason to explain this behaviour. We tried to bend the robot to force the wheels in one particular direction, but still they are alternating quickly between forward and backward movement. So instead we used the minimum sampling time (5 ms) and tuned the PID's parameter to finally obtain :

$$K_p = -250; K_i = -500; K_d = -0.8$$

Actually these parameters were difficult to tune because the robot's behaviour is highly depend of its initial angle. For instance sometimes it could stay up for more than 5 seconds and other times it fell in only 1 second with the exact same parameters set. We decreased K_p to get a smoother behaviour around the null angle (i.e. improve stability, prevent jitters). We decreased integral term to prevent overshoot (once the robot reached the null angle, it was falling the other side). We increased derivative term so that the robot is better "aware" that it is going to fall down and can act before it's too late. Here is a result we recorded with Simulink :

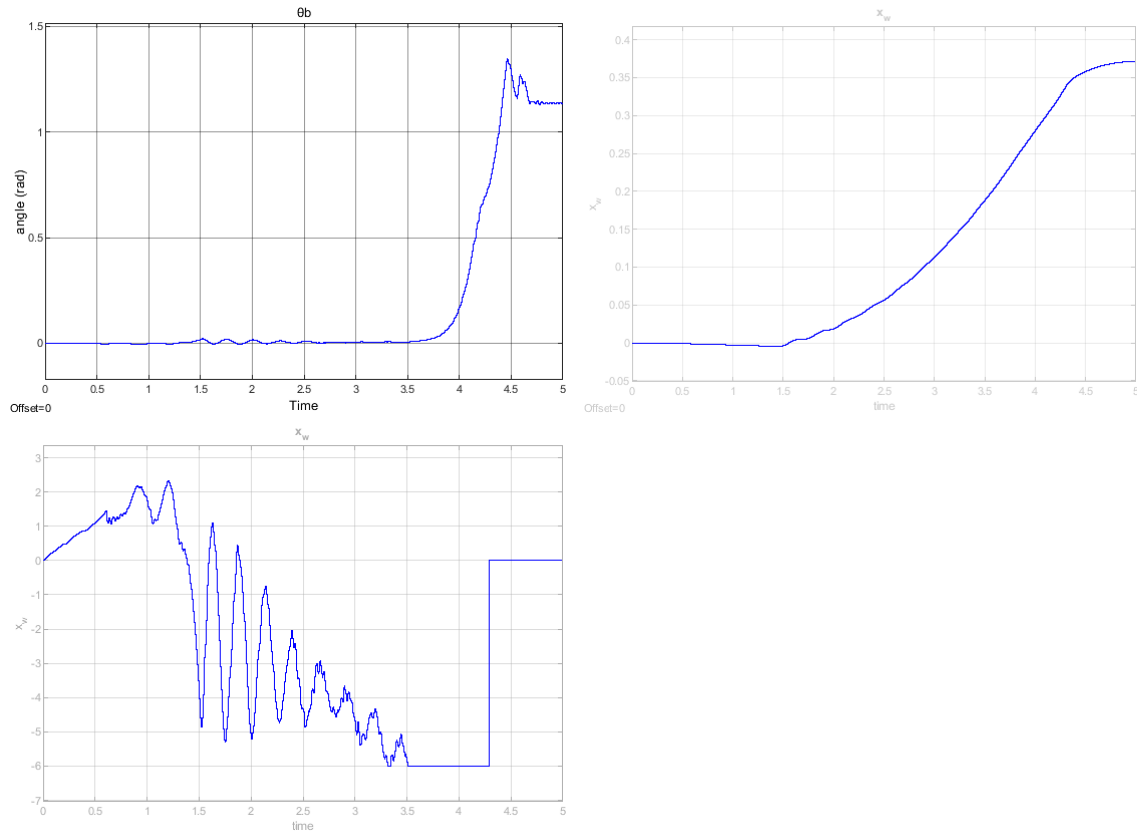


Figure 1: Simulation results (the robot falls after 4 seconds)

One can see that the command is oscillating. This might be due to high frequency noises coming from the sensors and then transmitted to the PID's derivative action (we didn't use any filter).

Reporting of Task 5.2