Group 4: Performance of a simple PID Controller for controlling the depth of an underwater vehicle

1st Asan Adamanov
Wet Whale
TUHH
118 614 29
Asan.Adamanov@tuhh.de

2nd Sven Dierfeld
Wet Whale
TUHH
215 959 67
Sven.Dierfeld@tuhh.de

3rd Jonathan Hellwig
Wet Whale
UHH
738 119 4
jonathan.hellwig@studium.uni-hamburg.de

4th Tolga Dirikgil
Wet Whale
TUHH
211 584 03
Tolga.Dirikgil@tuhh.de

***Abstract*—In this report we show how a PID controller is used to control the depth of a remotely operated vehicle (ROV), in this case the BlueROV2 by BlueRobotics. Method used to tune the controller gains is the "Ziegler-Nichols method". The investigation will show the limitations of controlling with a simple controller regarding overshoot and settling time.**

***Index Terms*—depth control**

I. ROV Introduction

The underwater vehicle used in this class is the BlueROV2 by BlueRobotics in its heavy configuration. This remotely operated (ROV) is equipped with four vertically and four circularly arranged thrusters. For additional buoyancy it is fitted with pieces of foam hidden under blue covers as one can see in Figure 1.

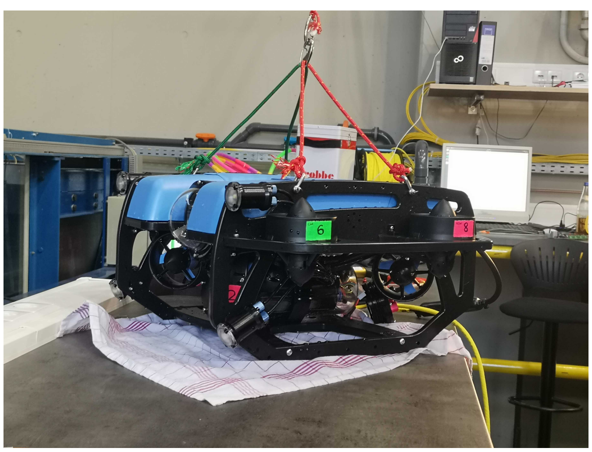

Fig. 1. BlueROV2 as used in this assignment

Figure 2 shows the ROVs thruster configuration. There are six thrust actions, that can be done using this configuration. They are thrust, lateral thrust, vertical thrust, pitch, roll, yaw. Since we are interested in depth control, only the thrusters related to vertical thrust will be adressed in table I. The number in this table gives information wether the thruster is to be used and its algebraic sign in which direction of rotation for its related action. In this case "1" means the thruster is used in mathematically positive direction of rotation and "-1" in negative positive direction of rotation, whereas "0" means it isn't used at all.

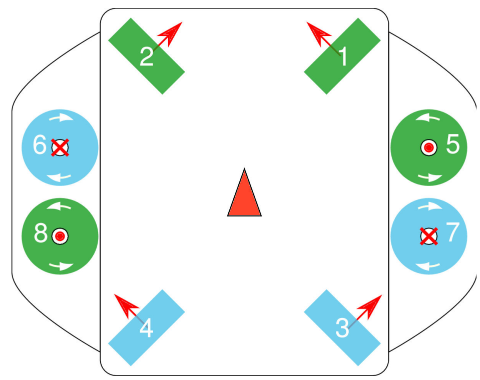

Fig. 2. Thruster configuration of BlueROV2 (topview)

TABLE I
For vertical thrust actuated thrusters

	Vertical thrust
Thruster 1	0
Thruster 2	0
Thruster 3	0
Thruster 4	0
Thruster 5	1
Thruster 6	-1
Thruster 7	-1
Thruster 8	1

II. Method of PID parameter tuning

Here we discuss on how we calculate depth and use it to tune our PID Parameters for depth control.

A. Depth calculation

The calculation of this ROVs depth d in m is done by measuring pressure p in pa underwater by an integrated sensor and measuring the atmospheric pressure $p_{atmosphere}$ in pa at the water surface resulting in this equation

$$d = -1 * \frac{p - p_{atmosphere}}{100000 \frac{pa}{m}} \tag{1}$$

B. Method of PID Tuning

The method we used to tune our PID paramters is called the "Ziegler–Nichols method" by which one raises the gain K_p until stable observations can be observed. This ultimate gain K_u is to be multiplied with different constants to get your derivative gain K_i and your integral gain K_d. By that we calculated the gains seen in table II

TABLE II
PID PARAMTERS

	Gain value
Kp	7.5
Ki	0.0012
Kd	0.013

C. Method used for performance testing

The performance of this PID controller will be tested by changing depth setpoints in a sinusoidal way and changing depth setpoints in a stepped slope. Observing the output signal we will observe the overshoot and steady-state error to determine the performance of this controller.

D. Safety features

To not jump out of the water or bump against the floor of the water tank, the ROV needs to stay in a safe zone, which lays between the boundaries $d_1 = -0.1\ m > d > d_2 = -0.8\ m$. To achieve this there is a check, if the desired setpoint of depth is in this safezone. If not, the setpoint will be rejected. If the ROV gets out of the safezone due to overshoot while controlling, there will be only positive control effort, if $d_2 > d$ to get it back up into the safezone quickly and only negative control effort, if $d_1 < d$ to get it back down into the safezone quickly.

III. Results

In Figure 3 a plot shows the response of the system to two alternating depth setpoints resulting in a step response. The orange line shows the setpoints and the blue line the measured depth.One can observe the overshoot $\Delta d_{s2} = 0.18\ m$ and it's steady-state error $\Delta d_{s2,error} = 0.03\ m$ for the setpoint $d_{s2} = -0.6\ m$ and the overshoot $\Delta d_{s1} = 0.13\ m$ and it's steady-state error $\Delta d_{s1,error} = 0.03\ m$ for the setpoint $d_{s1} = -0.2\ m$.

In Figures 4 and 5 one can see two plots showing setpoints in a sinusoidal shape and the ROVs related response. The blue line shows the measured depth and the orange line the setpoints. One can observe the overshoot $\Delta d = 0.04\ m$ for the setpoint functions minimum and no overshoot for its maximum. However one can observe nonsmooth behaviour where the ROV doesn't follow the sinusoidal setpoint shape at the setpoint functions maximum.

IV. Conclusion

The results show a certain overshoot while controlling the ROV which is a result of the "Ziegler-Nichols" tuning method, which aims for a fast response time. Since the safety of the vehicle is more important underwater than the speed of the response, there should be another tuning method used for future assignments, where a small overshoot is the aim. The

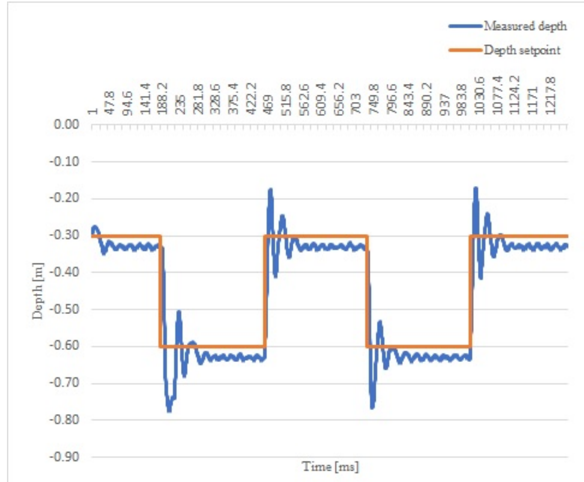

Fig. 3. Step response of depth control

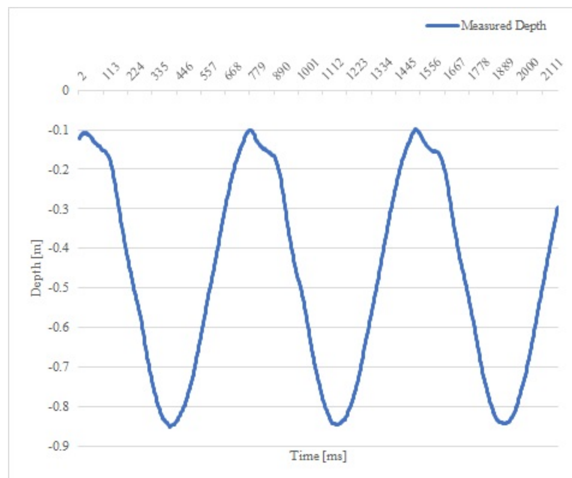

Fig. 4. Depth measurements to sinusoidal depth setpoints

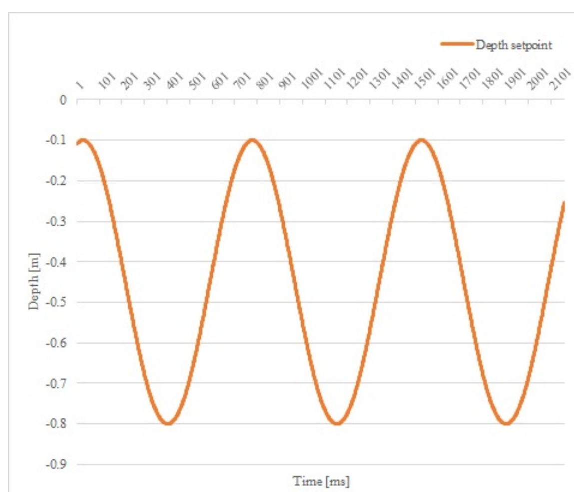

Fig. 5. Sinusoidal depth setpoints

steady-state error can be corrected by using a larger amount of integral gain.

The bump around $d = -0.1\ m$ seen in figure 4 may result from the ROV being out of the safezone for a short amount of time resulting in negative only control effort for that amount of time which leads to full thrust in the direction of gravity. The absence of this bump around $d = -0.8\ m$ in figure 4 where the ROV is actually below the safezone may result from the thrusters force acting against the direction of gravity. One can verify (or falsify) this assumption by testing depth control with a smaller maximum and higher minimum for the sinusoidal setpoint shape, so the ROV does not get out of the safezone. Other than that the PID controller performs well and stays stabe. Adding a model based observer would improve the performance but one would need to identify the system parameters. Here the PID control shows a good trade off between the little effort of implementing this controller and it's performance.

We weren't able to measure the drift of the rover in x or y direction, since there was no measurement unit for that, but one could easily see a drift while controlling the depth if pitch or roll angles where unequal to zero. To avoid this drift it will be useful to control these angles with a PID controller as well.