

FCS: Template for PID control.

Student: Johnson Domacasse
Date: 11 Nov 2023
Student#: 4471709
Teacher: Suzana Andova

Introduction:

The purpose of this document is to justify the decisions I have made throughout making the template for the Feedback Control System (FCS) for my ES project. Bear in mind, that this is not the final design but a start to designing the FCS for the project.

Functions:

The task was to make a Proportional integral derivative (PID) controller with the help of multiple functions. The most common looking PID loops look something like the one you see in figure 1.

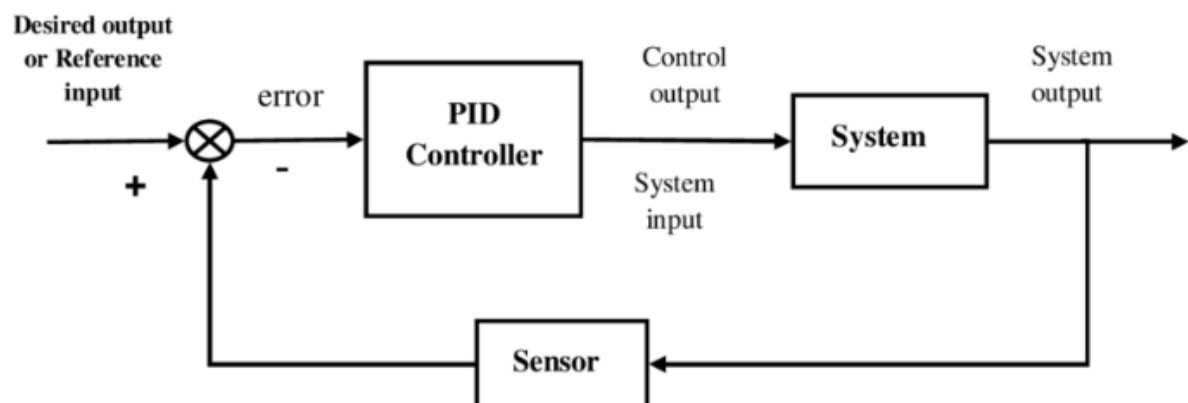


Figure 1: PID controller loop.

I am going to assume the reader knows what a PID controller is and how it functions, so the next section will mostly talk about how I structured my loop with different functions and how they are related to the figure above.

PID Controller:

The PID controller function is where the PID controls are set and an output is returned. The error variable is calculated within this function to use as our proportional variable. The integral is calculated to avoid a situation where the actuator needs to hold its current output signal. The derivative is calculated in order to avoid a situation where overshoot can occur. The output is then calculated by multiplying these variables with their designated K_x values.[1][2]

Servo Controller:

We are making a PID controller for our robot platooning project. In order to control this robot we use Servo motors. This will act as our plant (system). In order to control this servo, we have this function that converts the desired speed into PWM signals to the servo.

Ultrasonic Controller:

The servo motors themselves will not be giving feedback to the entire loop but an alternative sensor. The ultrasonic distance sensor (sensor) will act as our feedback giver. Our feedback in this case would be the distance from our setpoint. We don't want the robot to get any closer than 10 centimetres to the object in front of it.

This function is to simply read the distance from the front of the robot to the object in front of it.

Conversion:

This conversion function acts as a barrier between the PID controller and the plant. It takes the output of the PID controller and converts it into a desired speed variable to be used later on.

NOTE: It is still uncertain whether this function is necessary or not.

PID Loop:

Finally, we simulate what you can see in figure 1 in this function. We read the current distance from the ultrasonic sensor (for feedback). We have a check within our system to see if the distance is below 10 (stop servo motors in this case).

We then pass the setpoint (determined at the beginning) and our current sensor readings to our PID controller function. This output signal is then passed to the conversion function (again, not certain if needed). The return of this functions will be our desired servo speed which is then passed to our servo controller function to make the motors work.

References:

- [1] - *Improving the Beginner's PID – Introduction*. Project Blog RSS. (n.d.).
<http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/>
- [2] - *An Introduction to PID Controllers*. Gillard, G. (July 2017).
<https://georgegillard.com/resources/documents>