PID Water Level Control System

The University of Iowa

Team Members: Cole Swanson, Jacob Arkin

Abstract

In this project will implement a PID control system that will keep a tank of water at a desired level controlled by the user. The water level will be displayed and plotted by a serial output and with an RPG the user can change what level of water is in the tank. The water level will be measured by a sonar depth sensor connected to the microcontroller. Also connected to the controller is two pumps. One pump will push water into the tank to raise the water level and the other will pump water out to lower it.

Introduction

This product was designed and build to model how a sonar sensor and PID controller could be used to control the level of liquid in a tank. The idea is that this could be something in a chemical production plant to make sure the containments stay at the appropriate level or in channel locks to make sure that the water doesn't overflow or go to low.

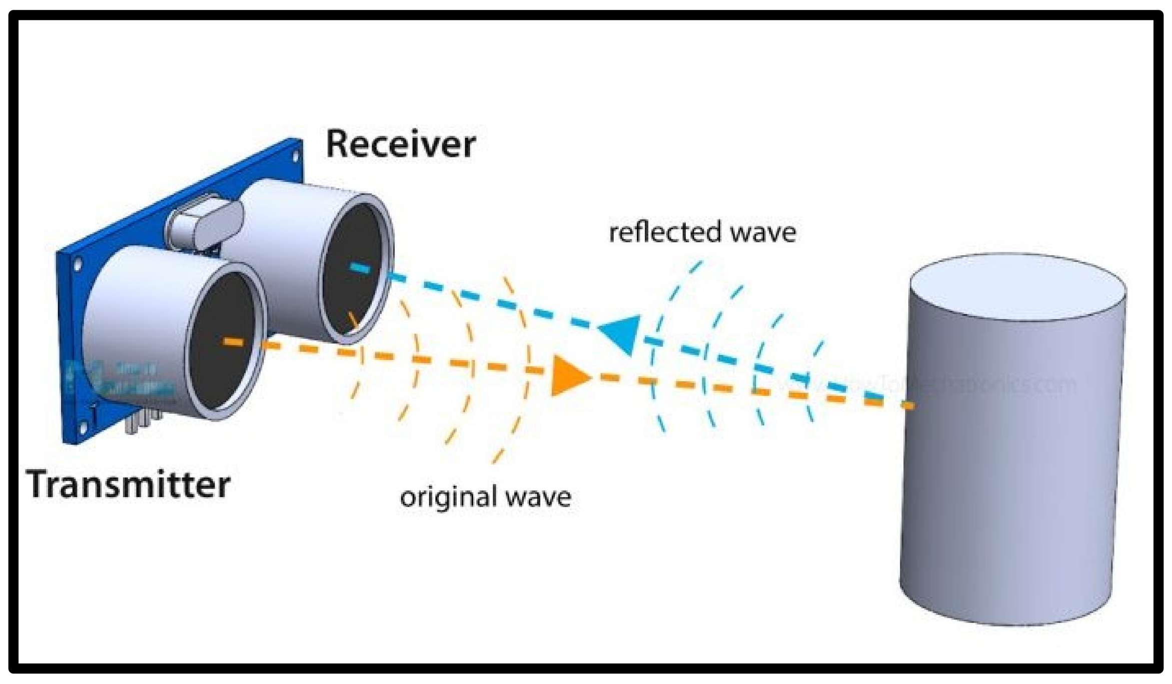
We built a water level controller that uses a sonar sensor to send and receive pressure waves.

Caption: Channel Lock (Going though boat locks)

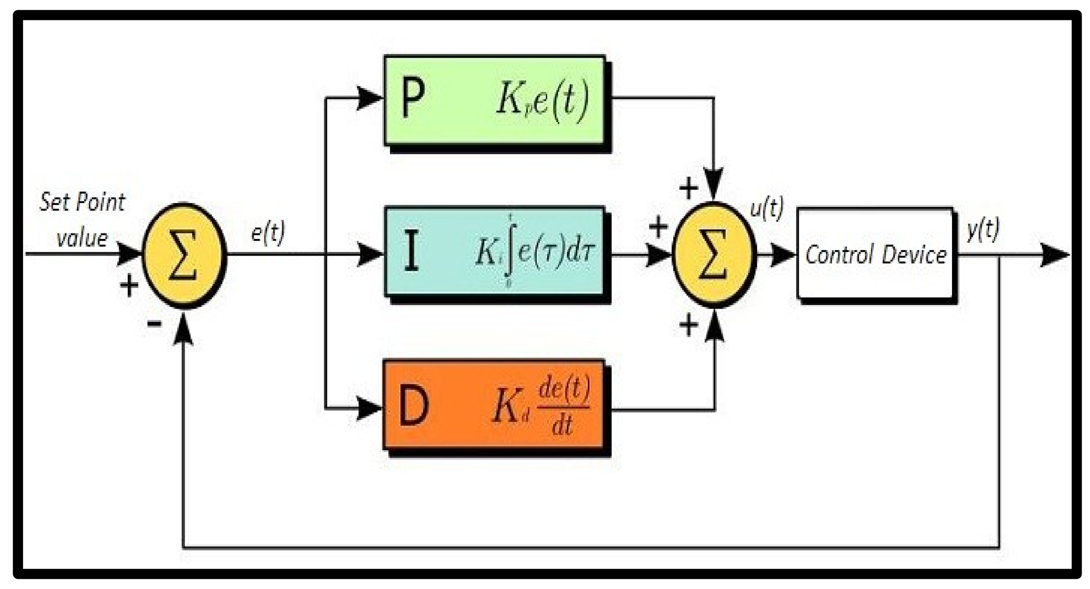
The time between sending and receiving is used to determine the water level inside a bucket. This is the relayed to a PID control system that preforms the calculation to determine which and how much the pumps need to run to keep the water level balanced. There is also a knob on the system which can be turned to change the desired water level. Lastly the height of the water level is communicated to a device through a serial connection and plotted.

System Description

The water control system operates by using a Sonar Sensor to measure the water level in the tank. Ultrasonic waves are emitted and reflected by the water. The difference between the time of transmittance and reception is multiplied by the speed of sound in the atmosphere and divided by two. This data is sent to a microcontroller, which implements a PID control system. The PID system adjusts pumps to maintain the desired water level, combining immediate response, error accumulation over time, and anticipation of future trends. This dynamic adjustment minimizes errors and ensures precise water level control. The output of the PID controller determines the timing for activating the pumps.



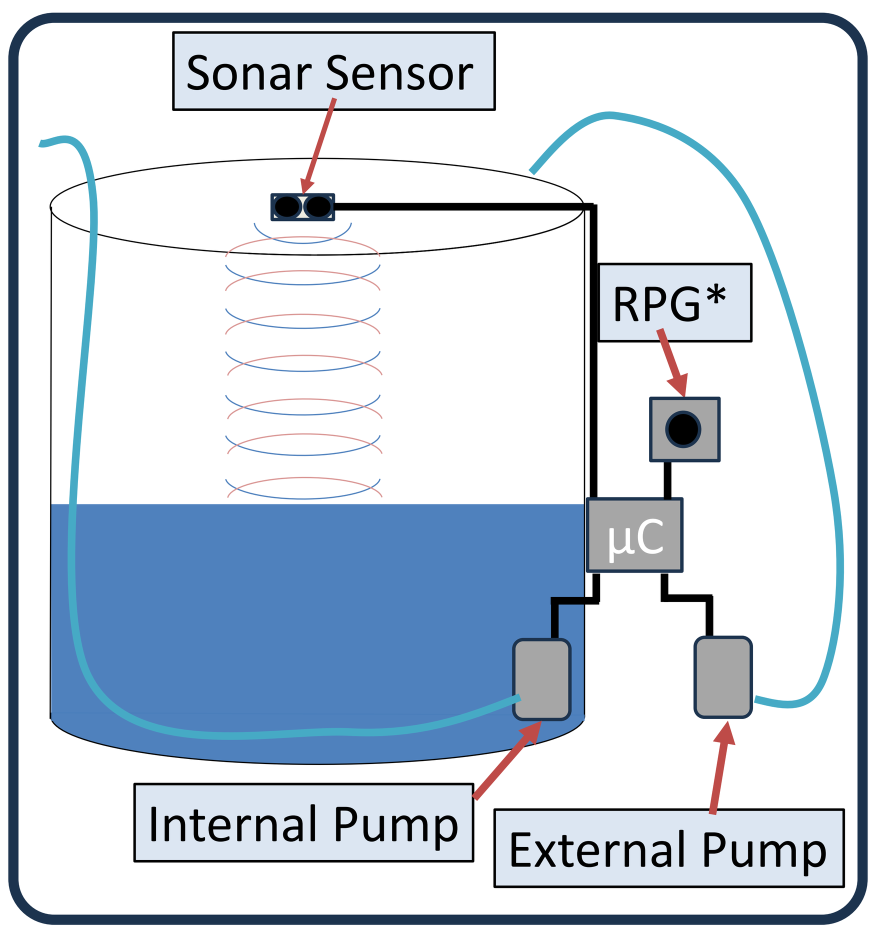
Caption: Sonar Sensor overview (Dejan, 2022)



Caption: PID Controller (Thorlabs)

Results

When the system is turned on and the bucket is filled with water. The initial set point of the PID system 80 mm of water depth. If the water in the tank is above the set point the internal pump will push water out of the tank at a rate of about 1 mm/s. Or if the water level is below the set point the external pump will push water into the tank from an external source at the same rate. When the RPG is rotated clockwise the set point is raised, and if the water level was currently at the set point the pumps will raise the water level. If the RPG is rotated counterclockwise the set point is lowered and the water level is lowered. Every click of the RPG will raise or lower the water 1 mm per tick. If water is removed or added to the tank with out the set point being changed it will the be sensed by the sonar sensor and the pumps will act accordingly to bring the water level back to the set point.



Caption: Model of PID controller
***Rotary Pulse Generator**

Caption: Photograph of PID controller in use

Lessons Learned

One thing we learned through this project is the importance of higher quality of products used within the system. For example, we used a relatively cheap sonar sensor, and it gave us somewhat inconsistent readings of the water level even when it wasn't changing. This led to us having to design a software solution to mitigate noise from the sensor.

Conclusion

This project is just a simple model of a system that could be implemented on a much larger scale. To do this we would recommend implementing a higher end sonar sensor that can be used for longer ranges with more accuracy, as well as stronger pumps that could output and input water faster. Lastly, we would like to thank Professor Krueger, and Teaching Assistants Zach and Tristan for their help with this project.

References

Salmony, P. (n.d.). PMS67/PID: PID controller implementation written in C. https://github.com/pms67/PID

Dejan. (2022, February 18). Ultrasonic sensor HC-SR04 and Arduino - Complete Guide. How To Mechatronics.

Thorlabs, Inc. Driver PID settings. (n.d.).

Going though boat locks with a houseboat. All About Houseboats. (n.d.).

Fleury, P. (n.d.). damadmai/pfleury: Peter Fleury's I2Cmaster, LCD, library, UART library. https://github.com/damadmai/pfleury