MATLAB-Based PID Control Design for Reactor Tank
Liquid Level Stabilization for teaching Control
Systems in Bioengineering.

Seniya Minsika Weerasooriya 28204

01
Introduction

Fluid Level Control Systems

Why do we need them?

- Ensures the **product quality** by precise control of fluid levels in operations
- Ensures **safety** by preventing of overflows and other equipment malfunctions and **overall stability of the system**.

Educational Significance

- Provides hands on experience with different control strategies.
- Study the effects of varying Proportional, Integral and Derivative gains.

Commercially available solutions

Purpose

- Used to teach and demonstrate P, PI and PID control systems.

Commercially available liquid level control trainers

- Gunt RT 10
- Amatrol Level/Flow Process Control Troubleshooting Training System

Drawbacks

- Prohibitive device cost
- Difficult to acquire for some institutions due to logistical complications.

Fig 1 Gunt RT 10

2
Objectives

Objectives

- Design of a practical fluid level control system, capable of implementing various control strategies. E.g. : P, I, PI and PID.
- Comparison of Performance with Simulation models.
- Contribution as an Educational Aid for Bioengineering.

Theoretical framework

Simulation model

- Torricelli's theorem and law of continuity for tank dynamics simulation
- Pump characteristic curve

Experimental Model

Toricellis theorem

- Used to find the ideal diameter for the orifice.

Control System Theory

In the design of the P,I,D wings of the controller.

Experimental model is validated against the simulation model.

3
Design and Implementation

The process :Experimental Setup

Goal

- Creation of an experimental setup for fluid level control.
- Easily sourcable electronics
- Control strategies must be variable.

Main Components

1. Tank
2. Controller
3. Pump
4. Sensor

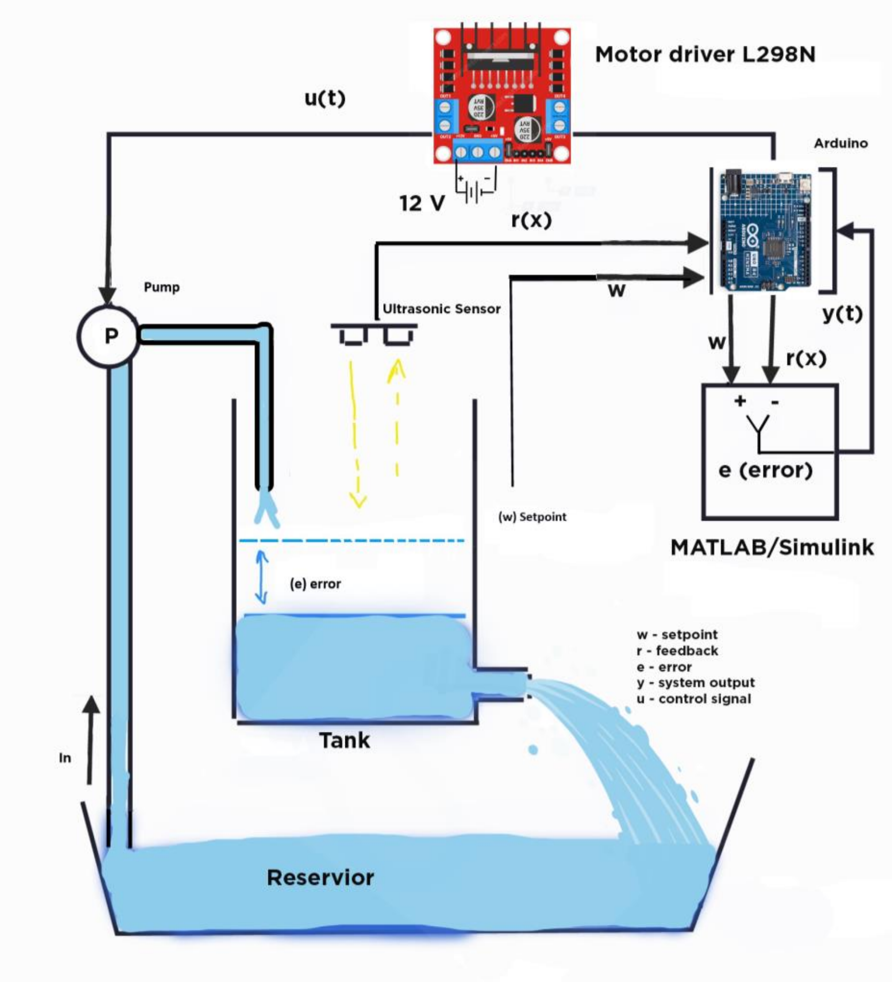
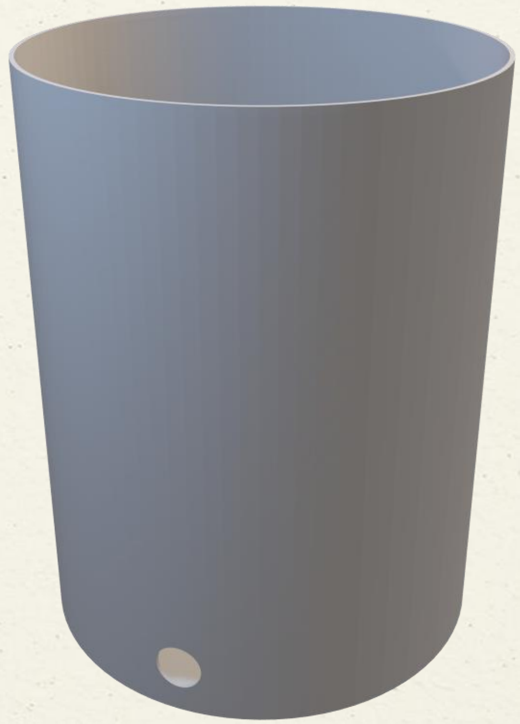


Fig 2: Illustration of experimental setup

Tank

- Glass tank with an orifice drilled on the bottom.
- Diameter = 15 cm
- Original orifice diameter = 14 mm
- Height = 0.2 m
- Printed orifice diameter = 4 mm (design), 3.5 mm (experimental)



Experimental Model (Hardware Design)

Custom Designed Outflow Pipe

Diameter : 3.5 mm (experimentally proven)

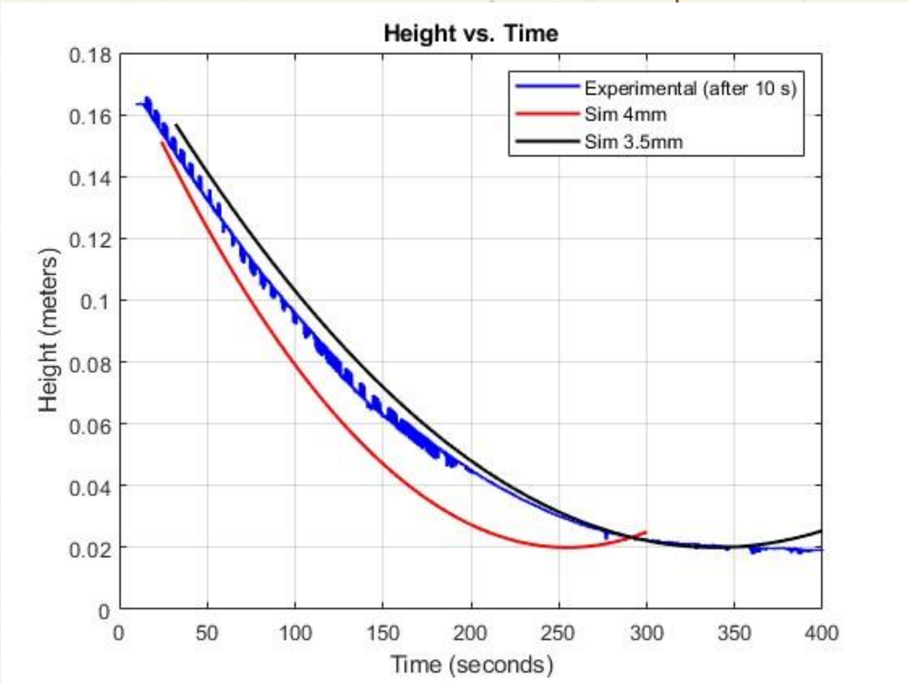
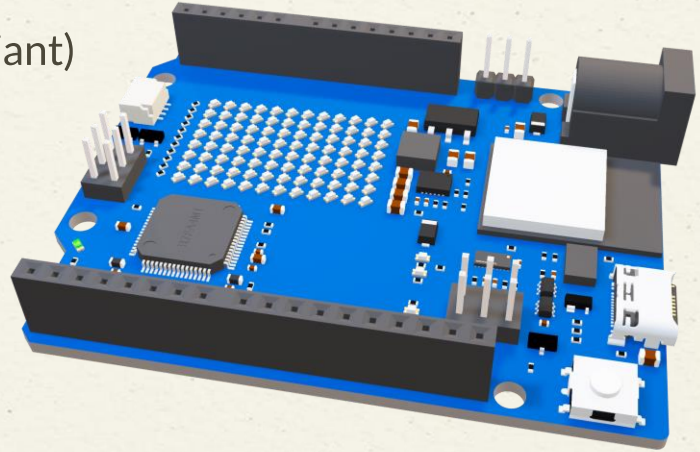
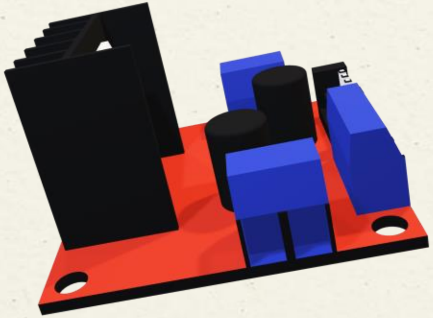


Fig 7 Free flow height vs time for designed outflow pipe diameter verification

Controller

- Utilizes Arduino UNO R4 WiFi (ESP32 Variant)
- Motor Driver – L298N





Pump

- R385 Diaphragm Pump 12V
- Max flow rate
  23 mL/sec experimental value
  *26 ml/sec in datasheet

Sensor

- SR04 Ultrasonic Sensor

Fig 4 R385

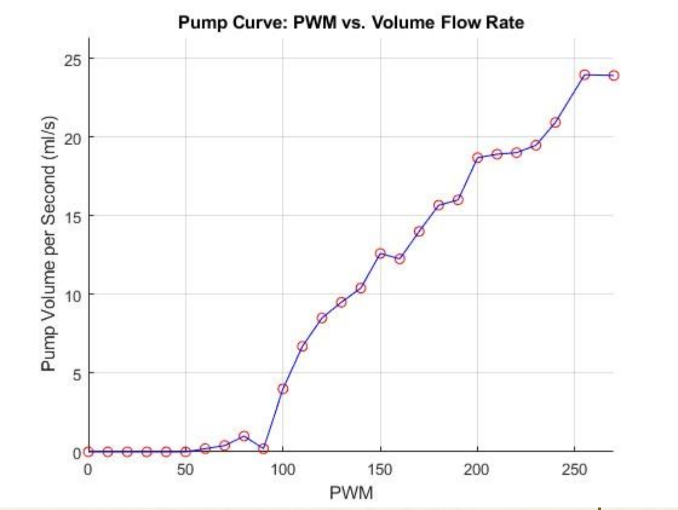


Fig 3 Pumping curve for R385

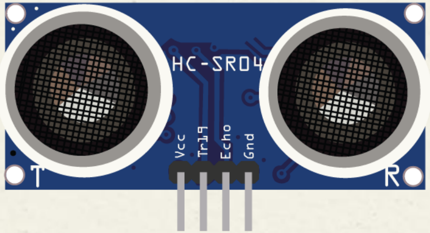


Fig 5 SR04 Ultrasonic Sensor

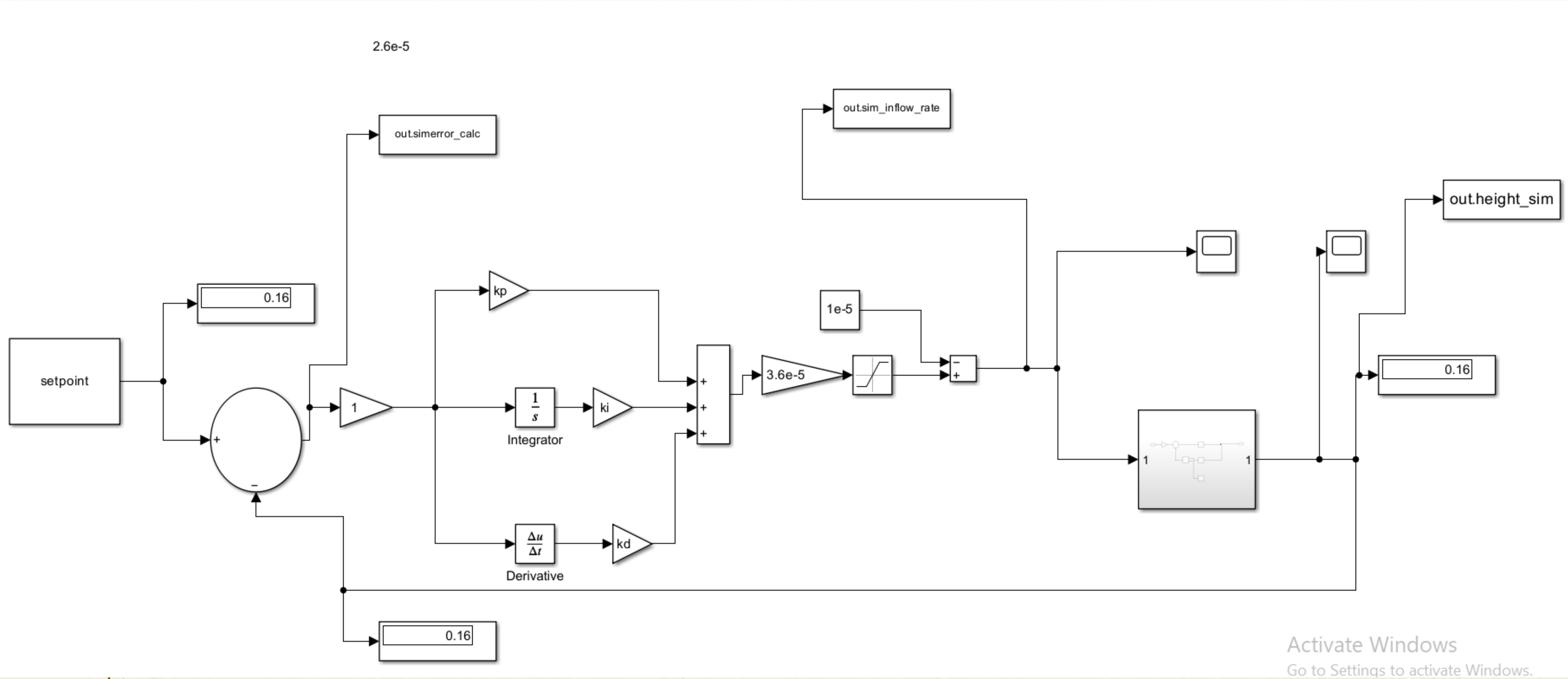
~4000 € vs ~50 €

Cost of Gunt RT 10

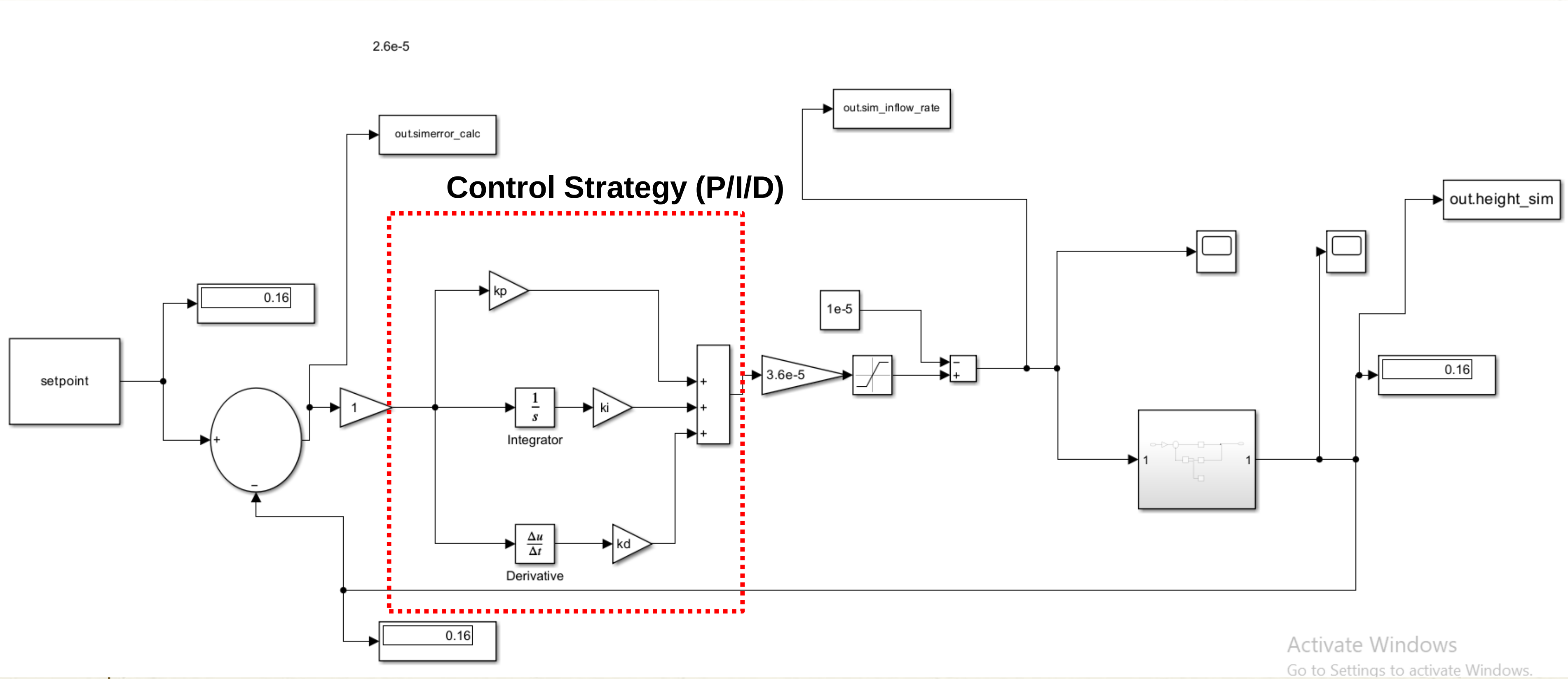
Cost of Proposed setup (excluding MATLAB)

4
Methodology

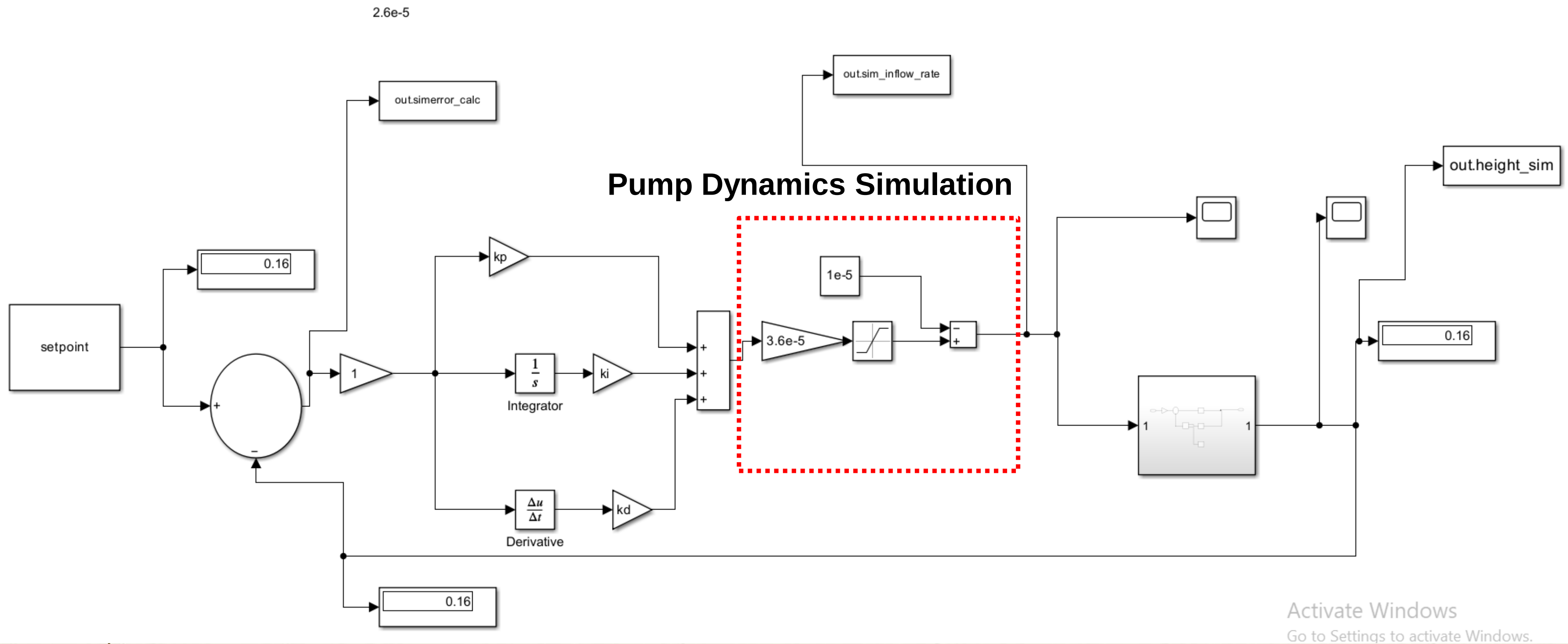
Nonlinear Simulation Model



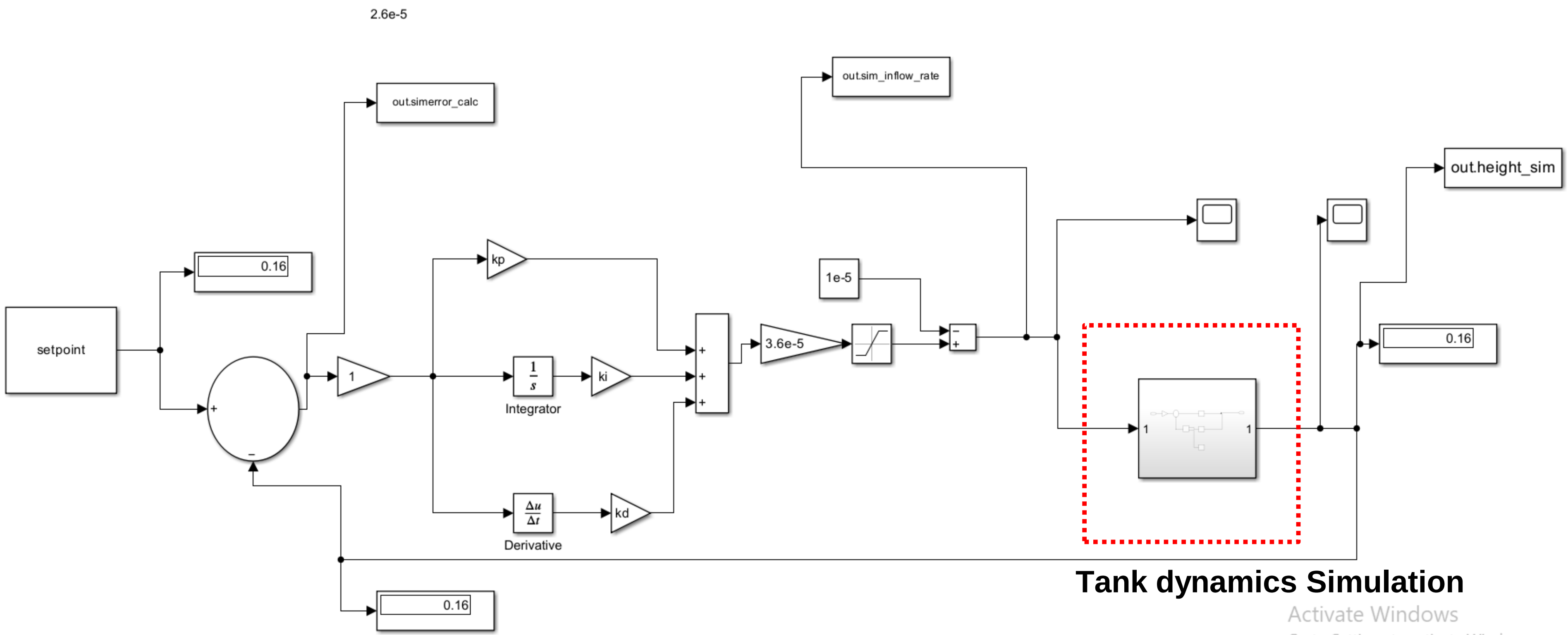
Nonlinear Simulation Model

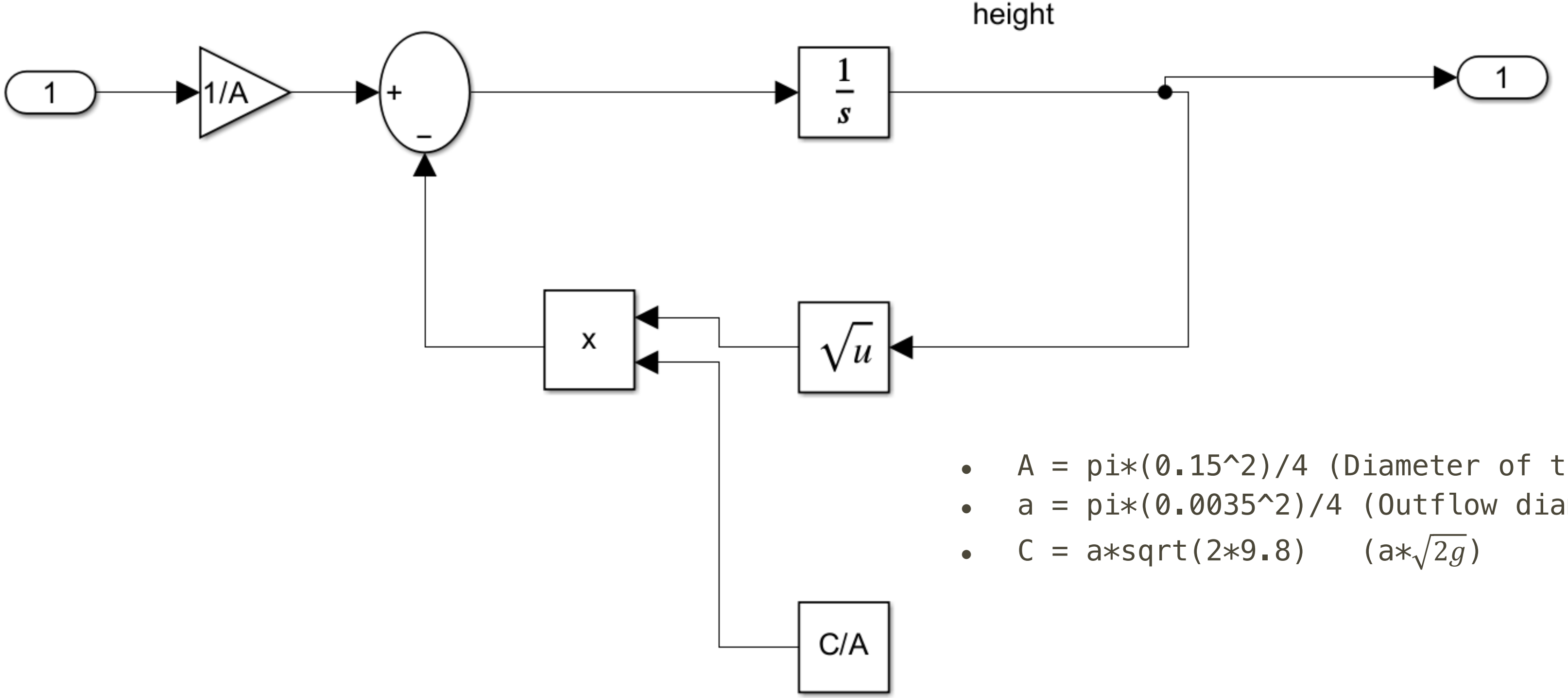


Nonlinear Simulation Model



Nonlinear Simulation Model





- A = pi*(0.15^2)/4 (Diameter of tank)
- a = pi*(0.0035^2)/4 (Outflow diameter)
- C = a*sqrt(2*9.8) (a*$\sqrt{2g}$)

The Experimental model

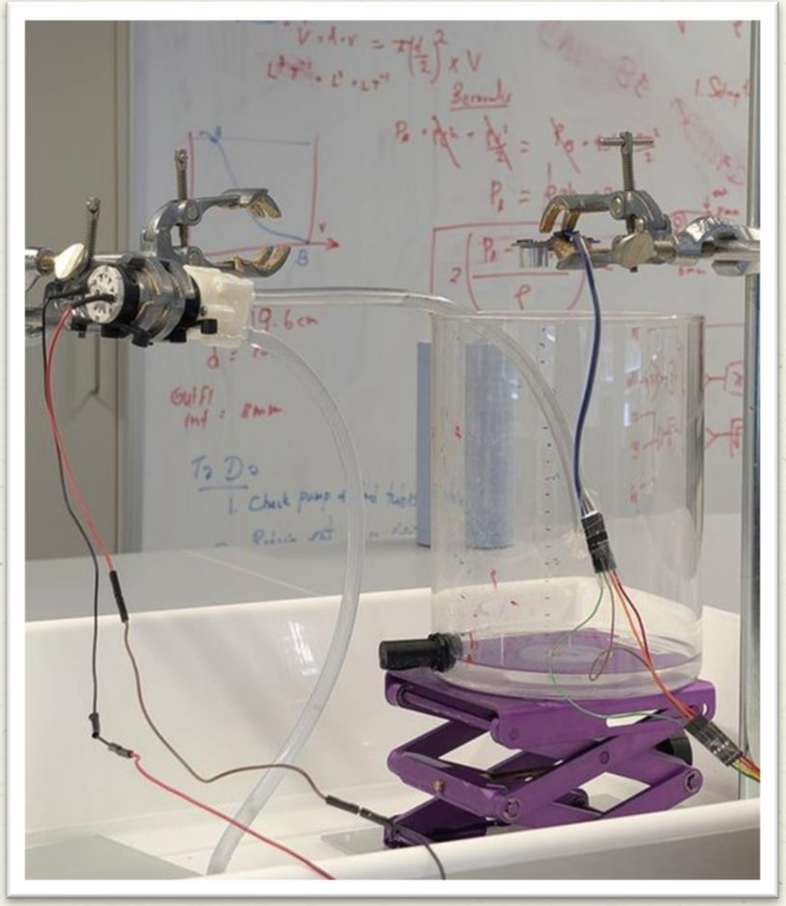
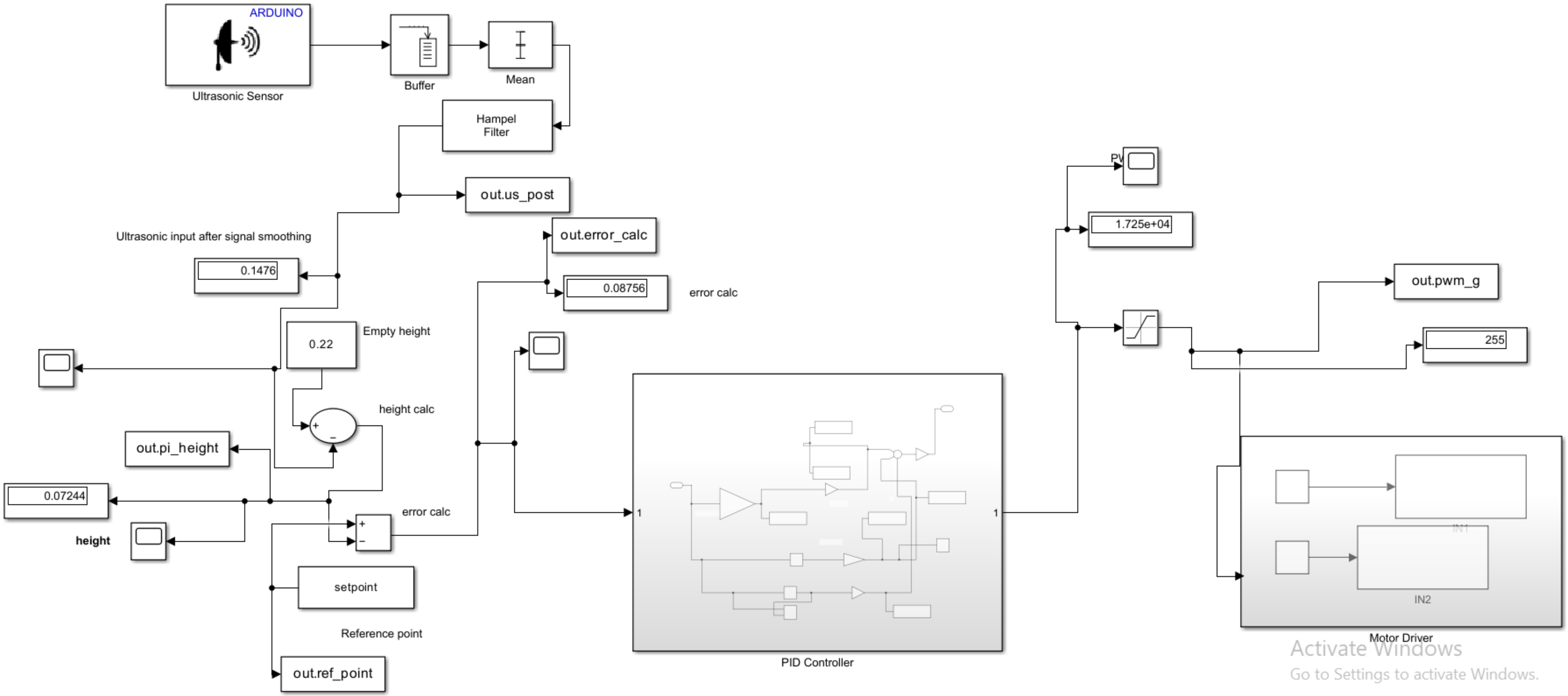


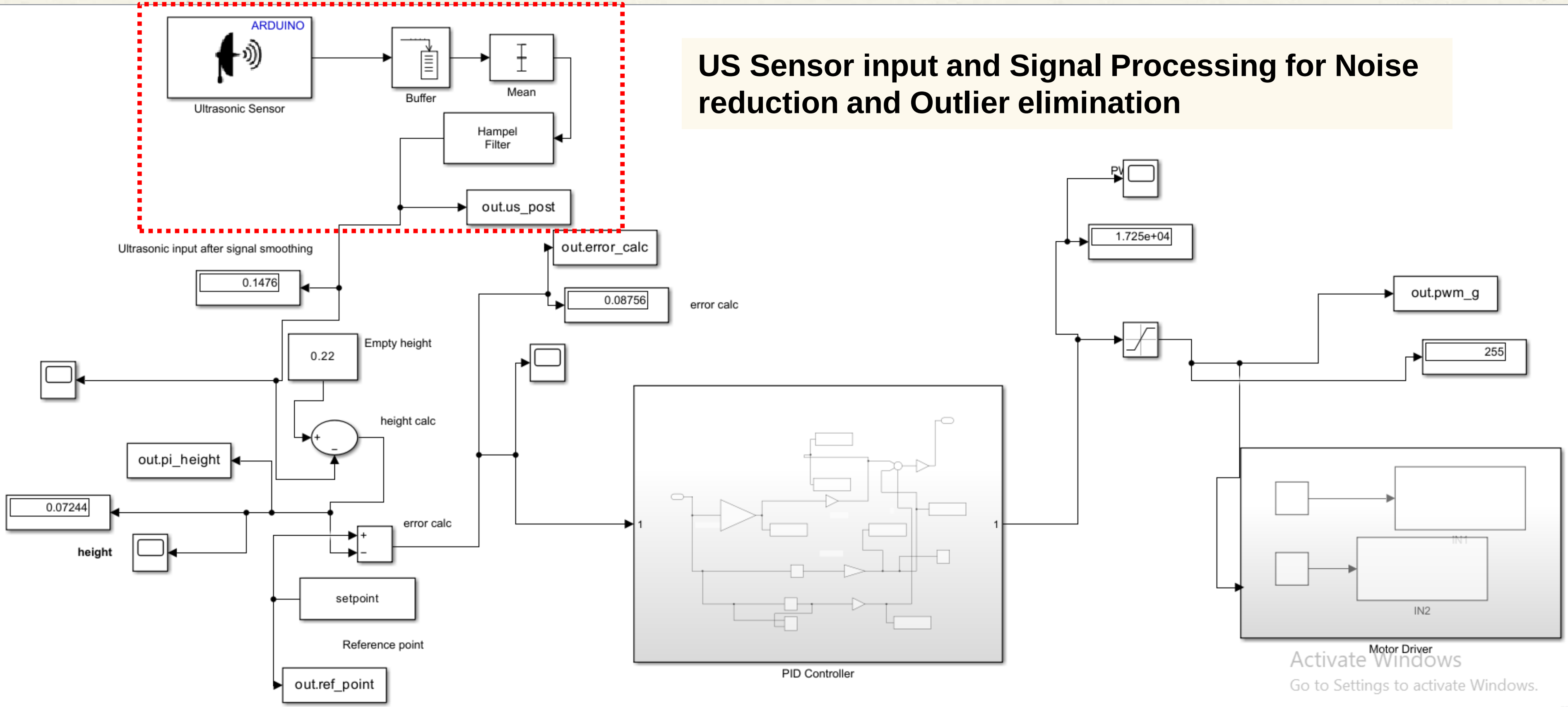
Fig 6 Experimental Setup

Experimental Model (Simulink Model)

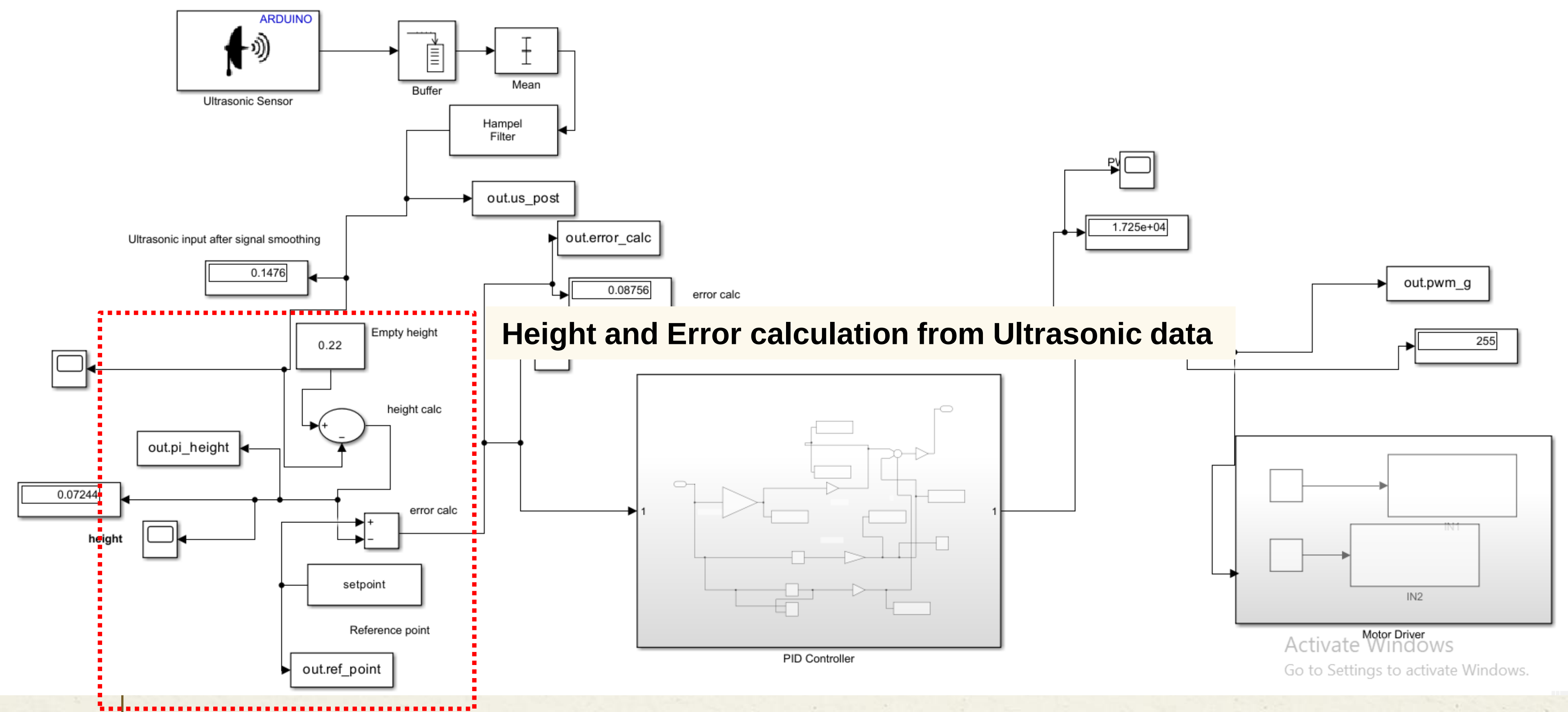


Experimental Model (Simulink Model)

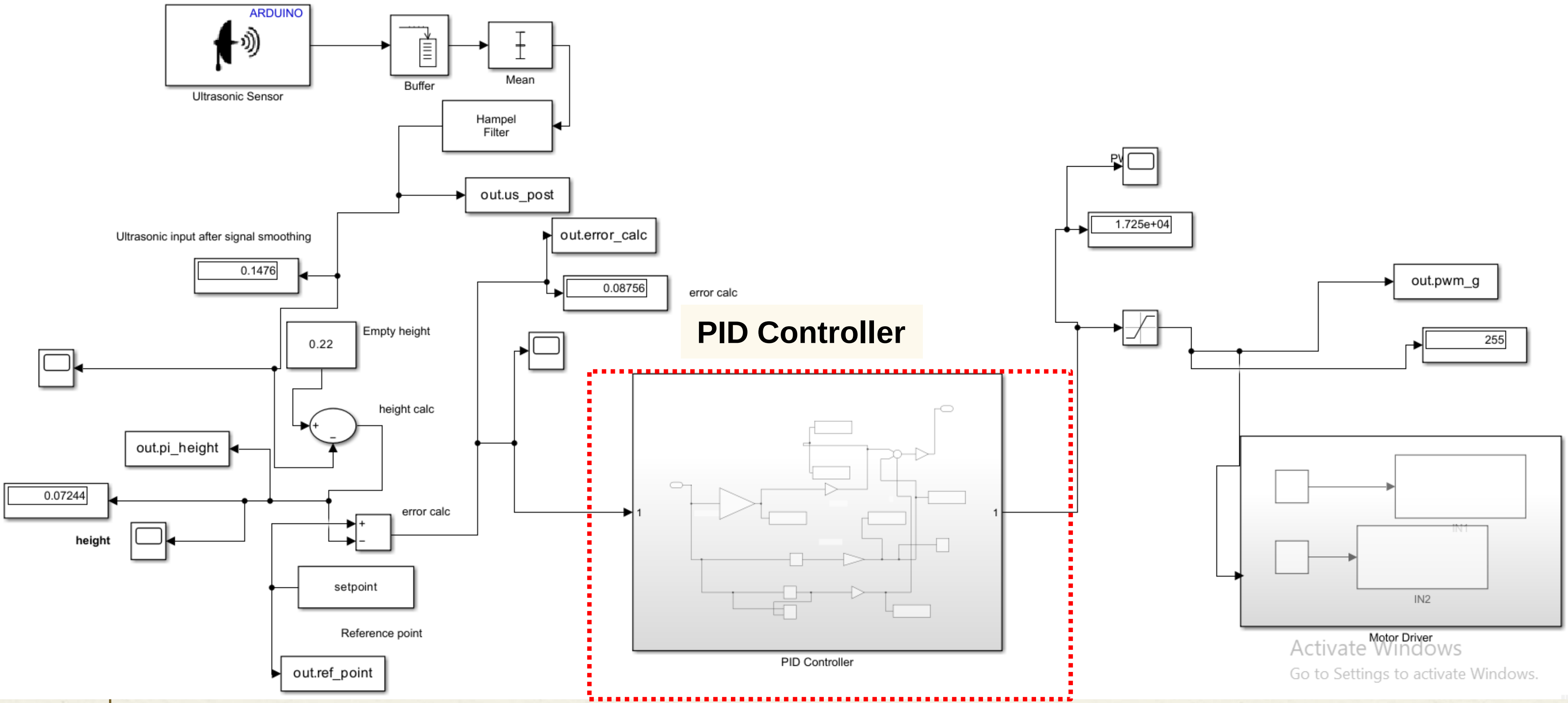
US Sensor input and Signal Processing for Noise reduction and Outlier elimination



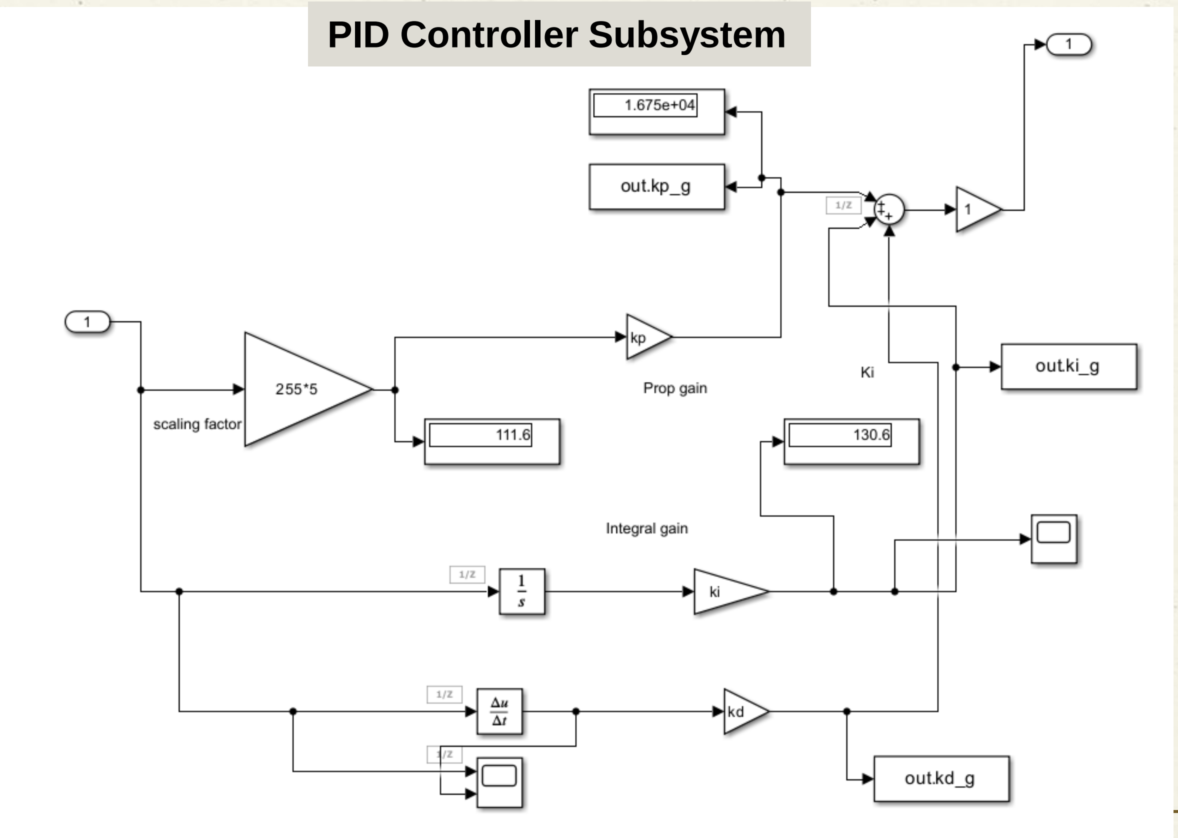
Experimental Model



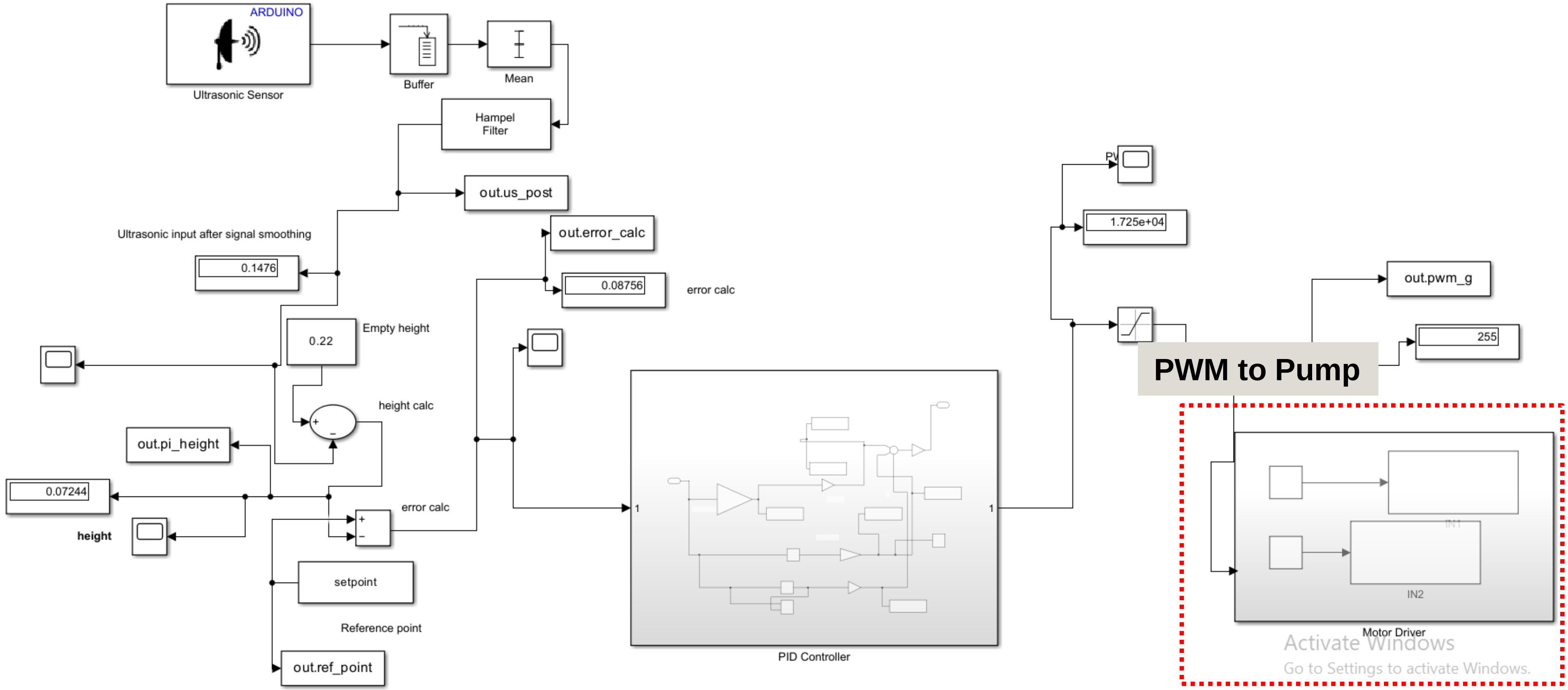
Experimental Model



Experimental Model



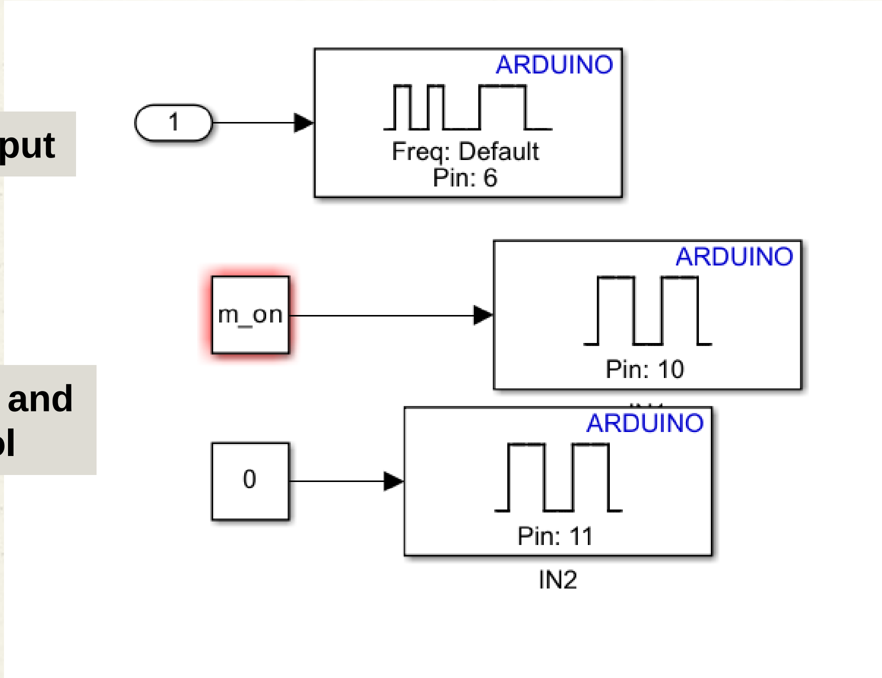
Experimental Model



Experimental Model

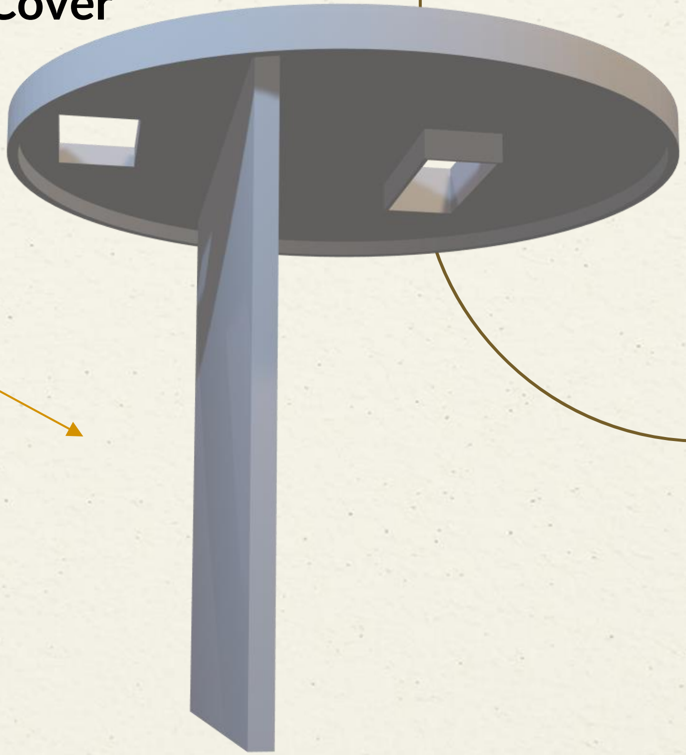
PWM Input

Motor ON/OFF and Polarity control



Experimental Model (Hardware Design)

Tank Cover



Wave breaker

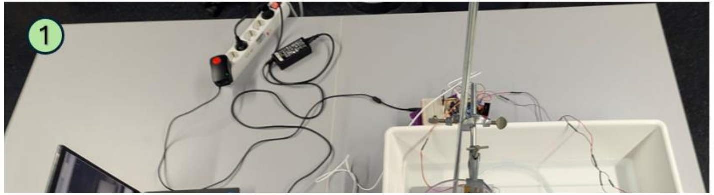
L298N Circuit Mount

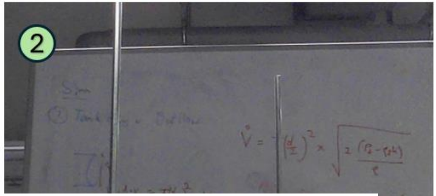
Automation of Experiments

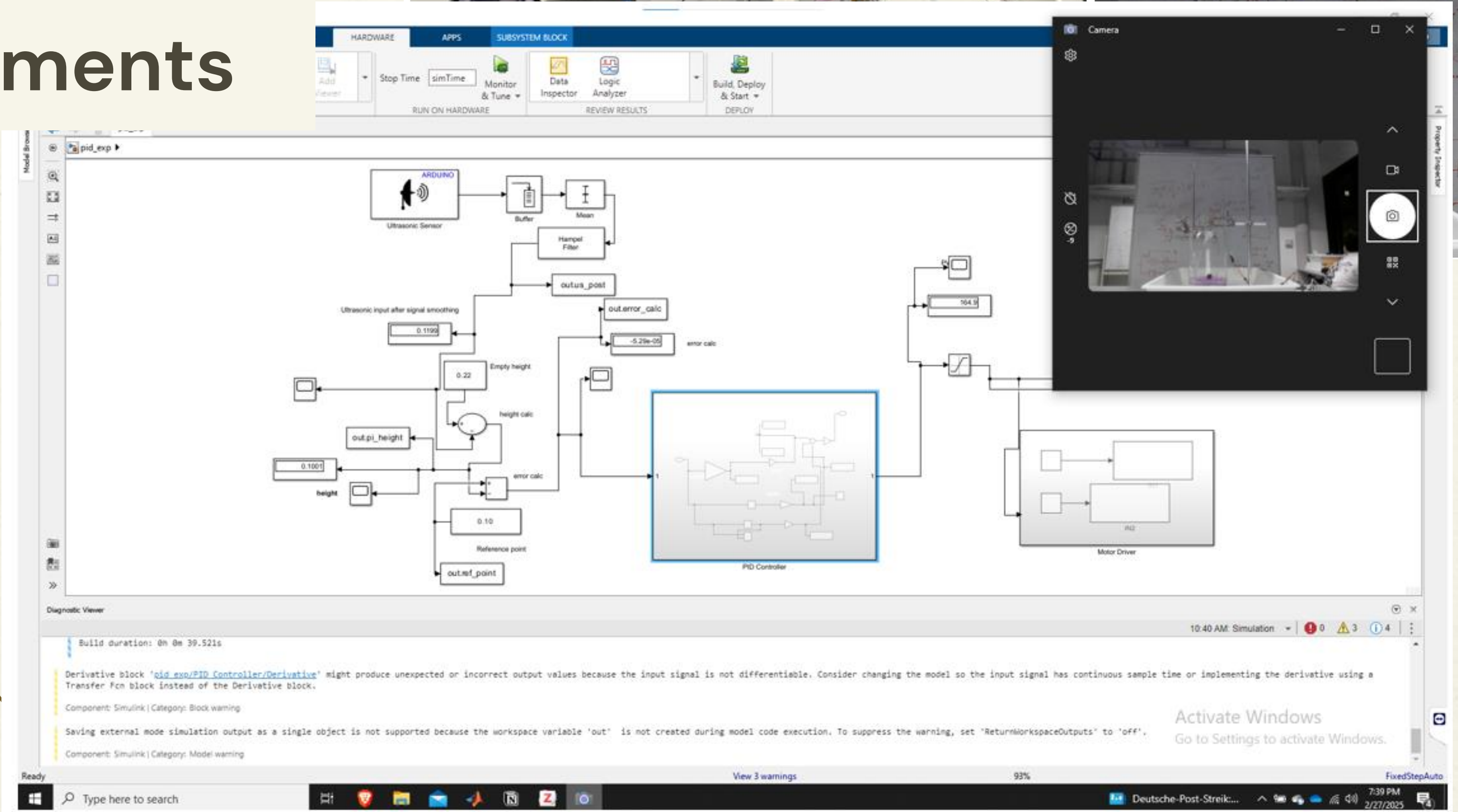
```matlab
%% Controller Parameters
simTime = 600;   % Simulation duration in seconds
ptime   = 300;   % Pause time between experiments to drain the tank

%controller gains (each row is one set: [Kp, Ki, Kd; Kp2, Ki2, Kd2;])
pidGains = [ 15, 0,0;
            15,  50, 10;
            15,   10, 2;
             ];


% Name of the Simulink model
model = 'pid_exp';

%% Open the Model and Set External Mode
open_system(model);
set_param(model, 'SimulationMode', 'external');

%% Loop Through Each PID Gain Set (each experiment handles one set)
numExperiments = size(pidGains,1);

for iter = 1:numExperiments
    fprintf('Experiment %d: Running with Kp = %g, Ki = %g, Kd = %g\n', ...
        iter, pidGains(iter,1), pidGains(iter,2), pidGains(iter,3));

    % Upload the PID gains to the base workspace for the Simulink model
    assignin('base', 'kp', pidGains(iter,1));
    assignin('base', 'ki', pidGains(iter,2));
    assignin('base', 'kd', pidGains(iter,3));

    % Set m_on = 1 to turn on the motor pump at the beginning of the experiment
    assignin('base', 'm_on', 1);
```

Fig 8 Automation Script code snippet

Remote Execution of Experiments

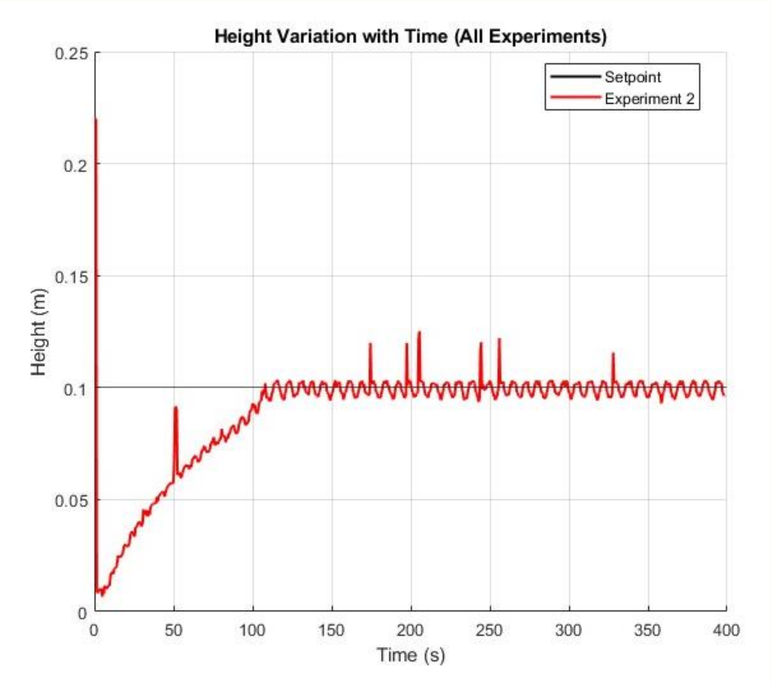






5
Results

ON OFF Control



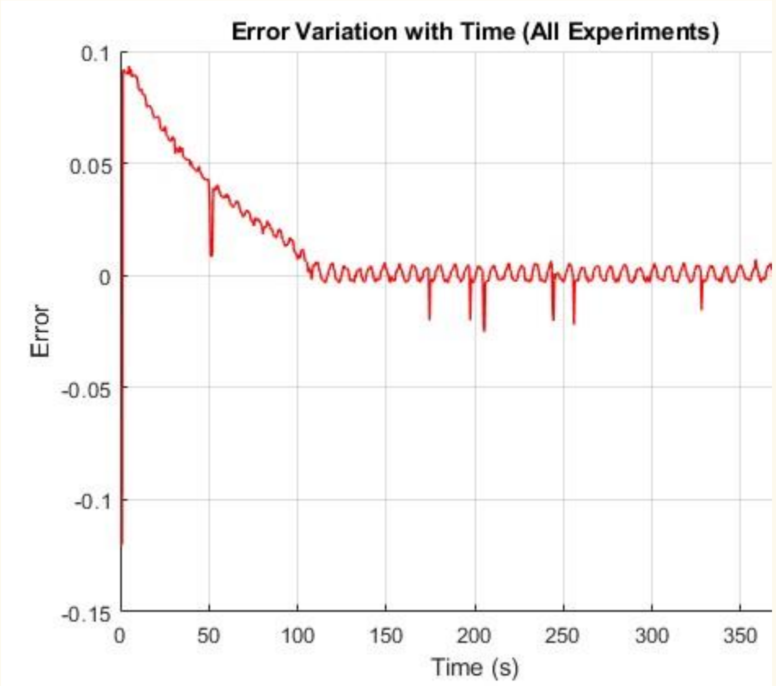
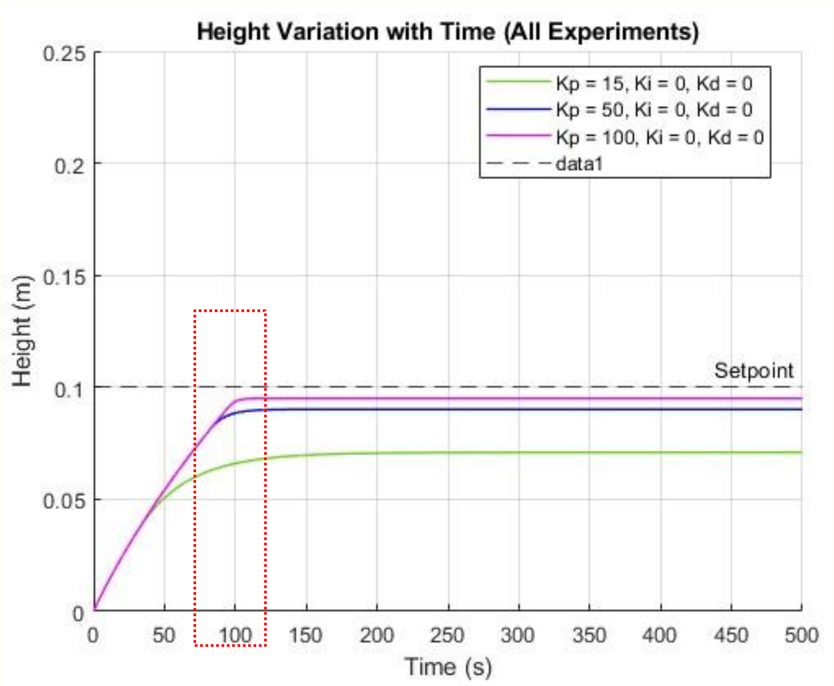


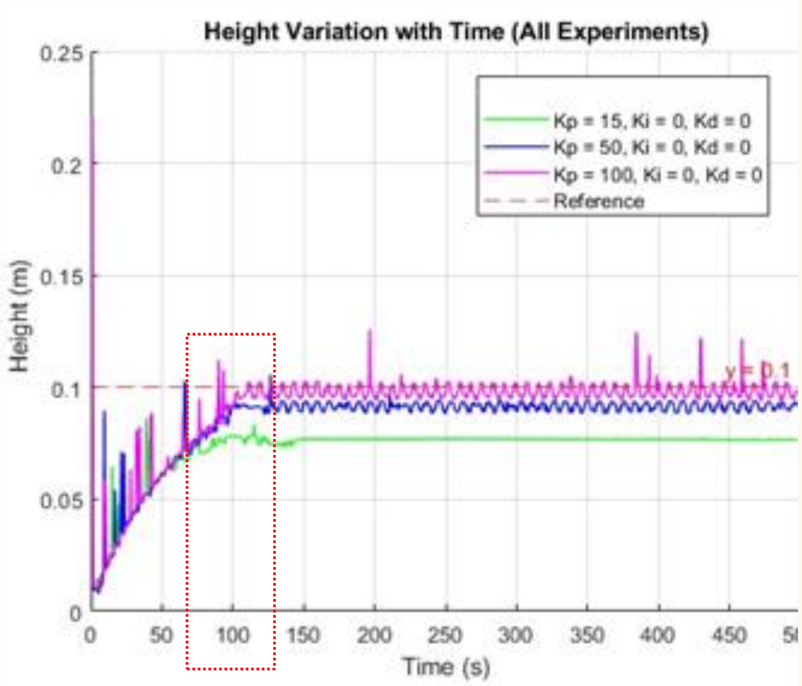
Fig 9(L),10(R) Height vs time; Error vs time. ON OFF Control setpoint = 0.1 m

P Control

Kp gains tested= 15, 50, 100



Simulation

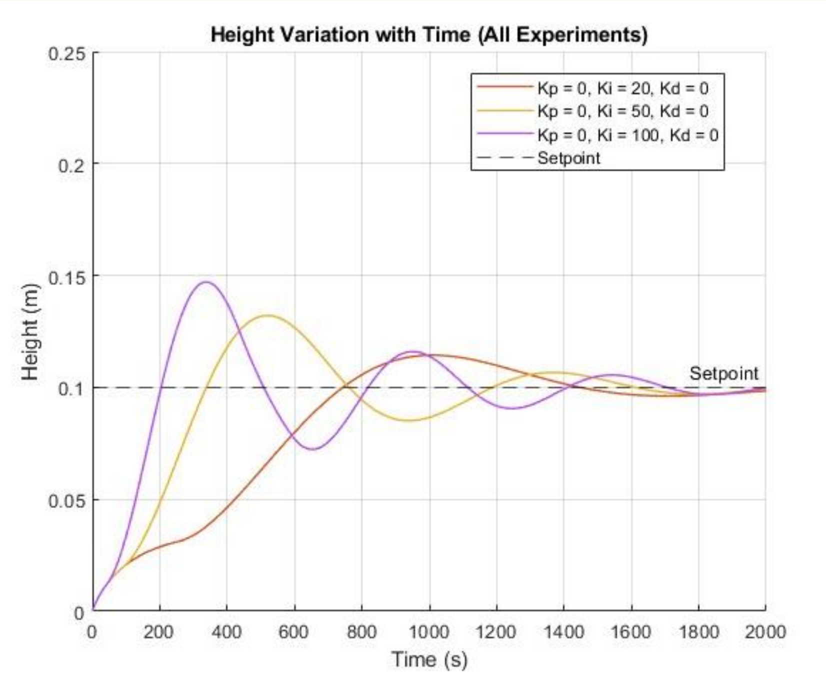


Experimental

Fig 11(L),12(R) Height vs time P Control setpoint = 0.1 m

I Control

Ki gains tested= 20, 50, 100



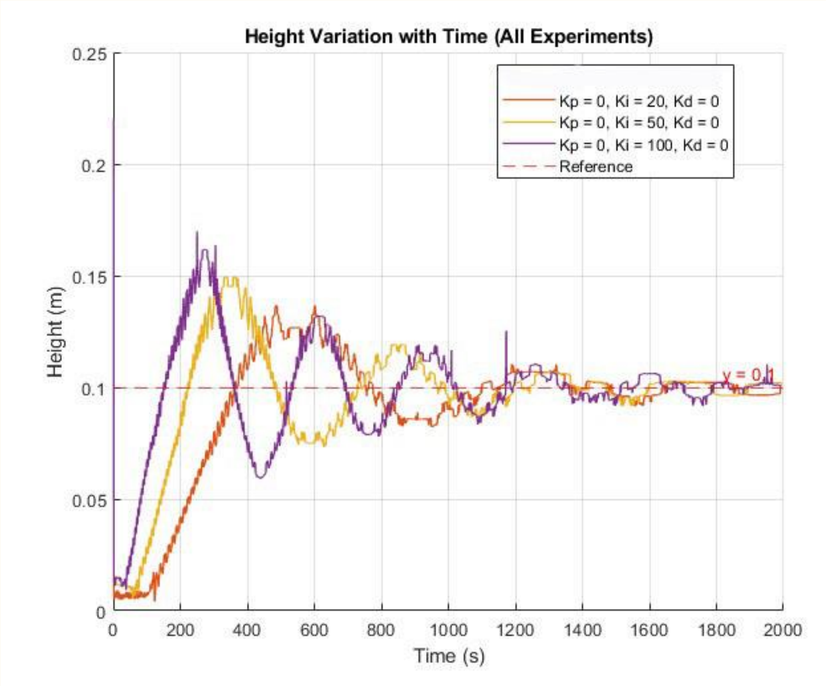


Fig 13(L),14(R) Height vs time: I control

I controller (Ki = 20)

- Oscillatory behavior.
- Eventual zero error steady state.

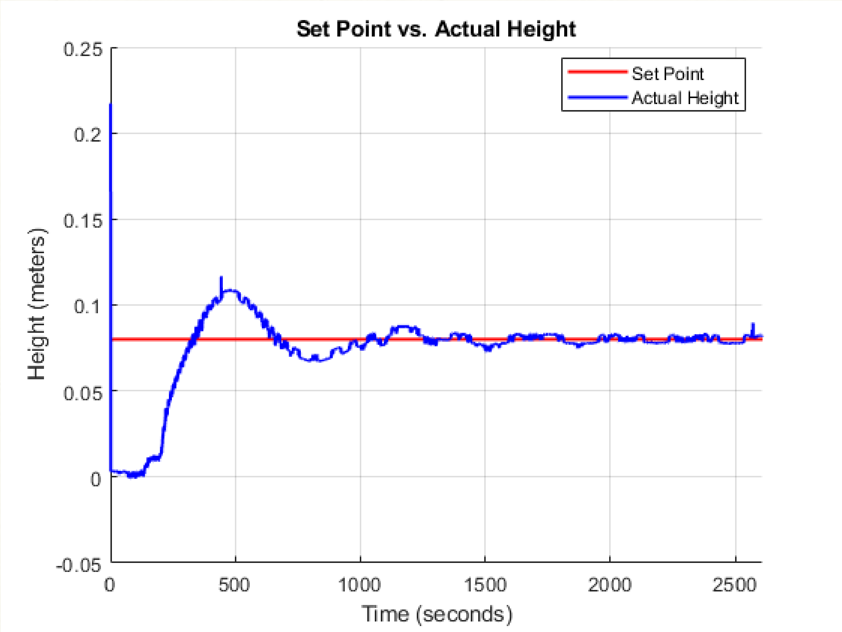
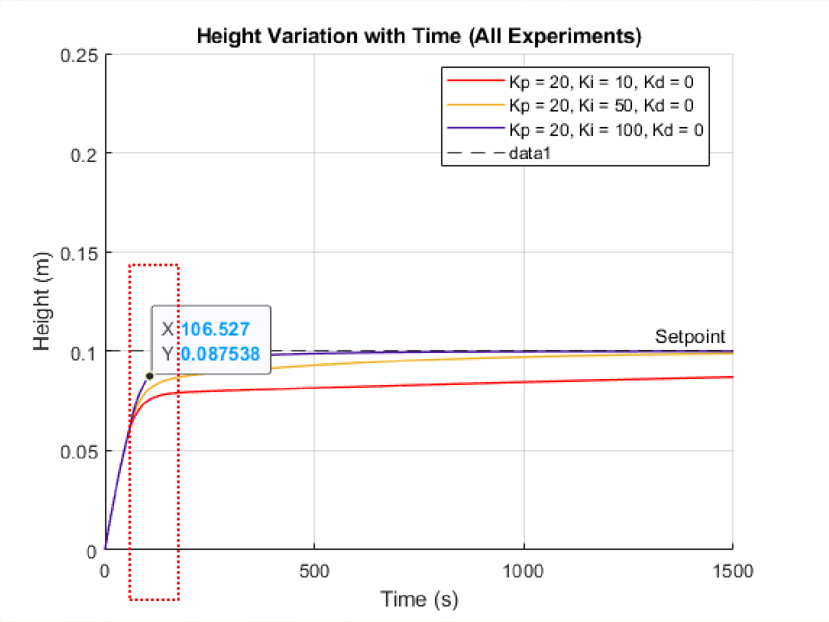


Fig 15 Height vs time: I control, setpoint: 0.08 m

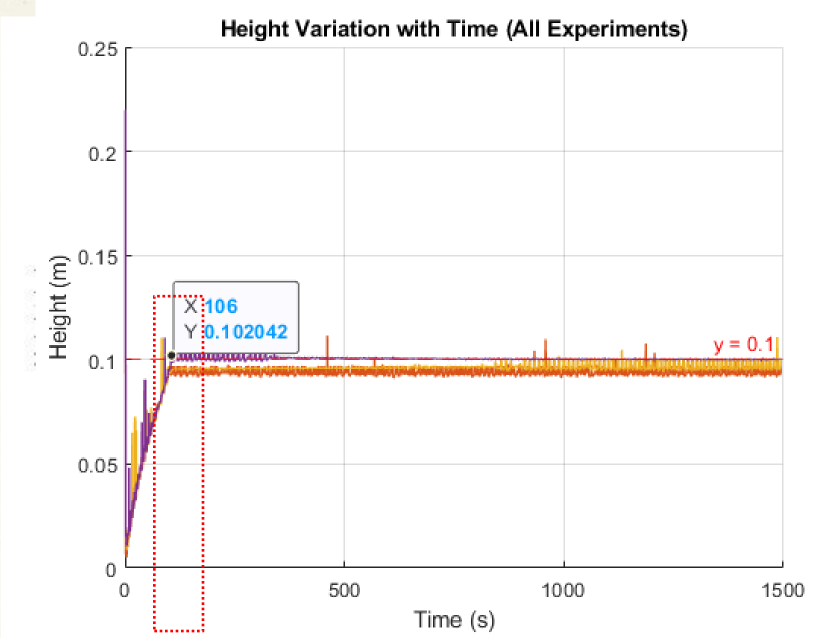
PI Control

Kp gain tested= 20

Ki gain tested= 10, 50, 100



Simulation



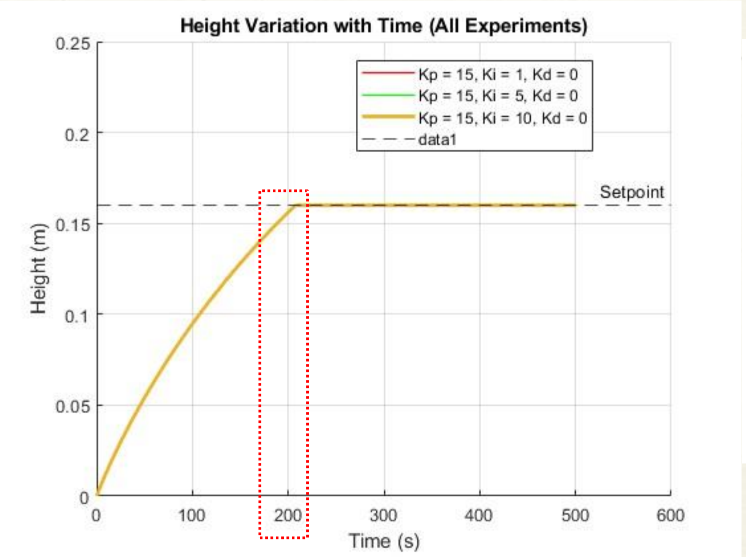
Experimental

Fig 16(L) Sim,17(R)Exp : Height vs time: PI control, setpoint = 0.1 m

Kp = 15, Ki = Varying

Kp gains tested= 15

Ki gains tested= 1, 5, 10



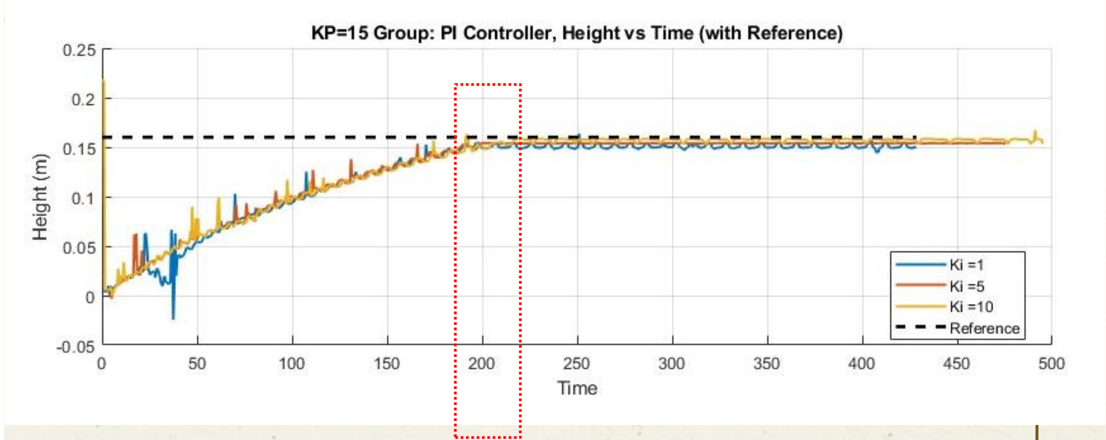
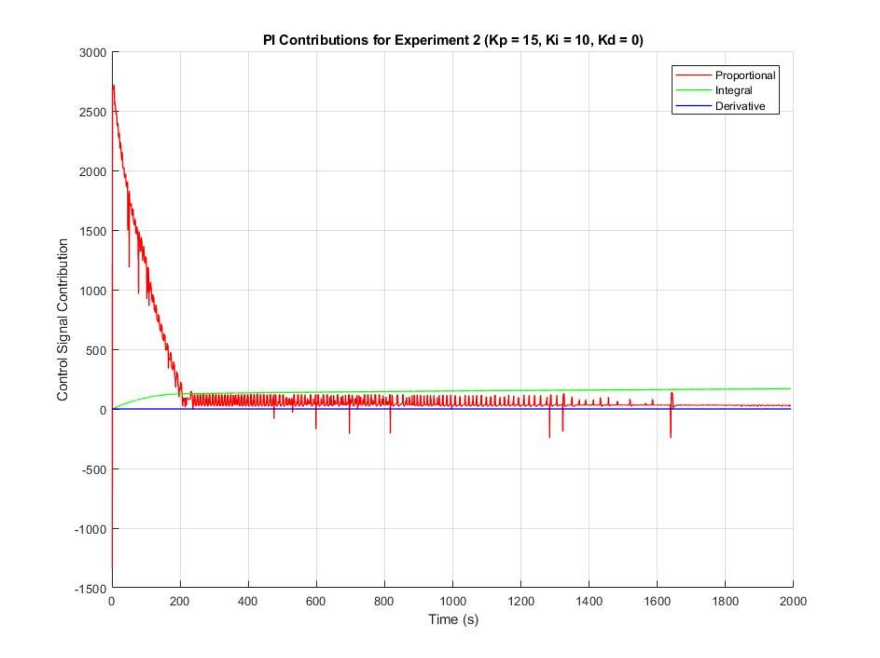


Fig 18(L) Sim,19(R)Exp : Height vs time: PI control, setpoint = 0.16 m

P & I contributions (Kp = 15, Ki = 10)



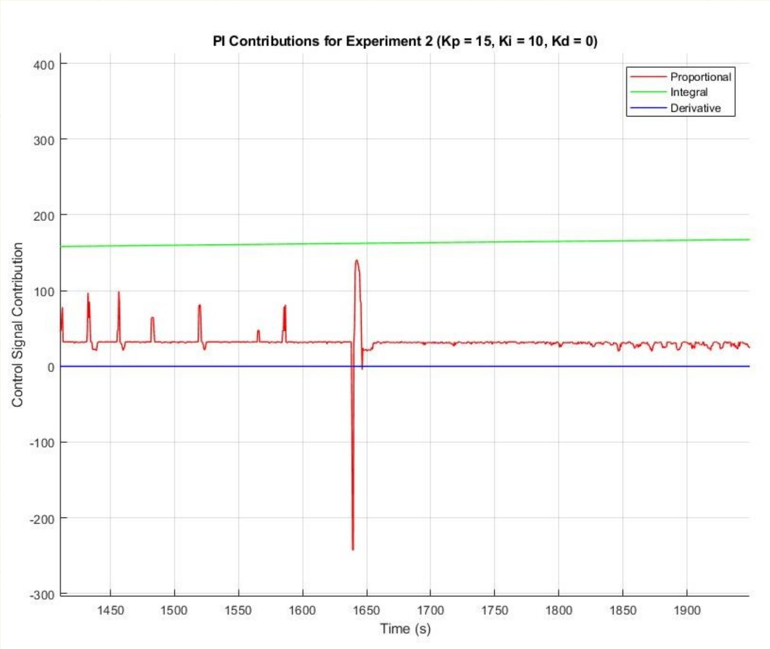


Fig 20: Proportional and Integral Control effort v time: PI control, setpoint = 0.15 m

PI controller Response Study

- 1 – setpoint = 0.16 m
- 2 – setpoint = 0.1 m
- 3 – setpoint = 0.08 m
- 4 – setpoint = 0.08 m, (Disturbance)
  500ml of water added

Kp = 5, Ki = 10

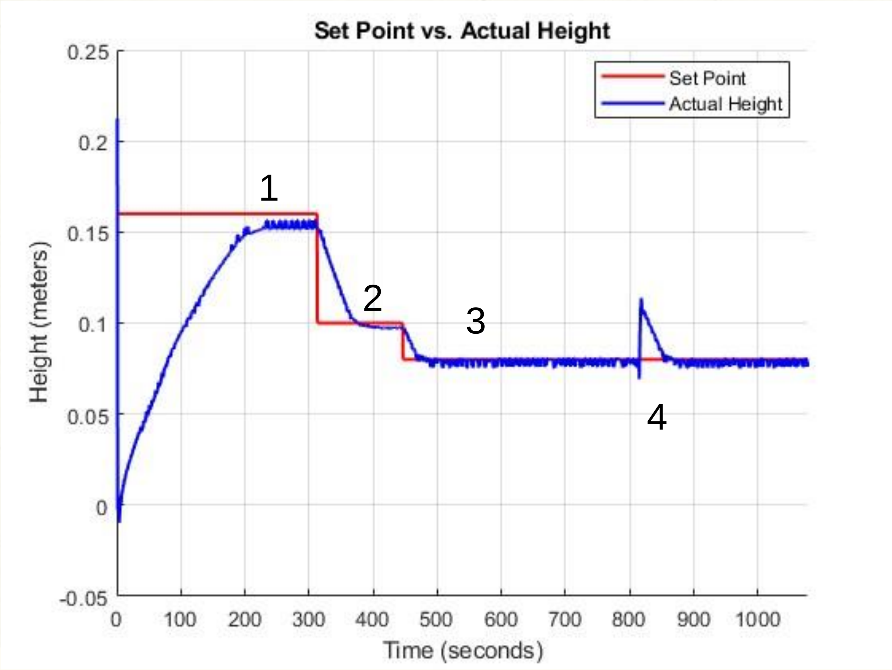
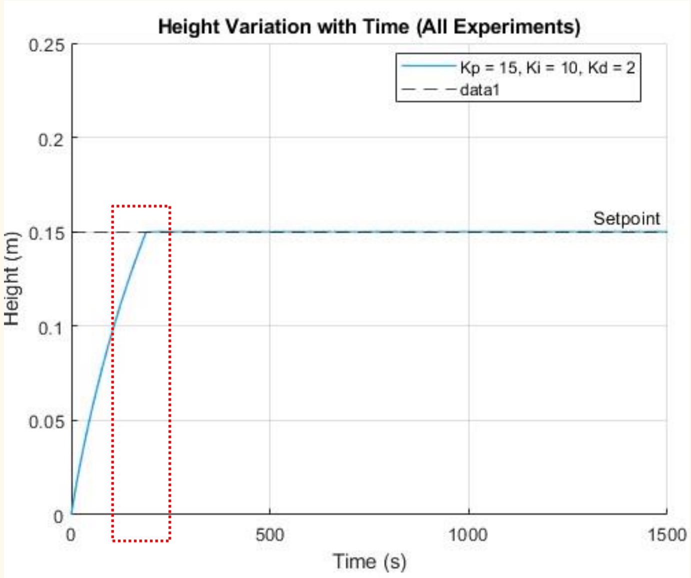


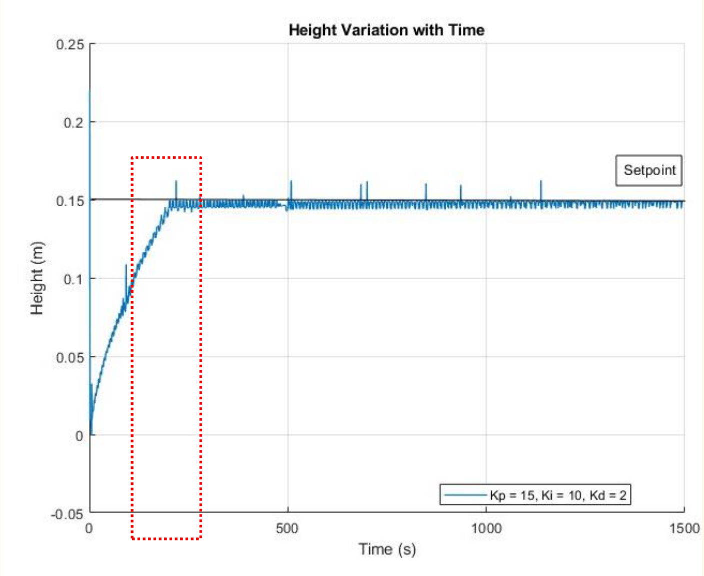
Fig 21: Height v time: PI control response study

setpoint (m)	**0.16** (1)	**0.1** (2)	3\4 - **0.08**
Actual (m)	0.155m	0.099m	0.08
Average Error %	3.125	1	0
Rise time (sec)	210	70	20

PID Control



Simulation



Experimental

Fig 22(L) Sim,23(R)Exp : Height vs time: PID control, setpoint = 0.15 m

6
Discussion

Overview of the Controllers

- ON OFF Controller: maintained around the setpoint although it had constant oscillations.
- P controller: reached the setpoint at higher Kp levels.
- I controller: Higher Ki, introduced more overshoot and oscillations.
- PI Controller: Best option
- PID Controller: Identical performance to PI controller as derivative action was not necessarily required for slow dynamics of fluid systems.

As a Teaching tool

- Provides a hands on method to visualize concepts of fluid mechanics and control theory in action.
- A modular setup
- Remote execution ability gives more accessibility.

Potential reasons for deviations in Simulations

- Ideal conditions were assumed for the simulations.
- Evidence of Turbulent flow

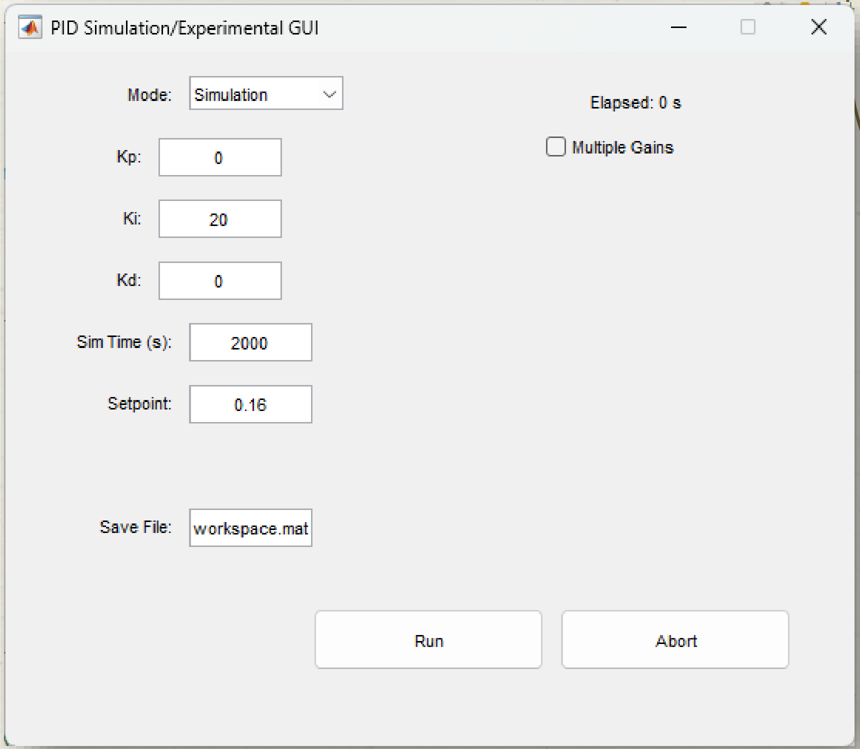
$$Re = \frac{\rho v \, L}{\mu} = \frac{1000 * \sqrt{2 * 9.81 * 0.1} * 0.0035}{0.0010.0010005} = 4900 \text{ (4000<Re : Turbulent flow)}$$

- The pumping curve was linearized to model its dynamics into the simulation.

GUI Interface (Beta)

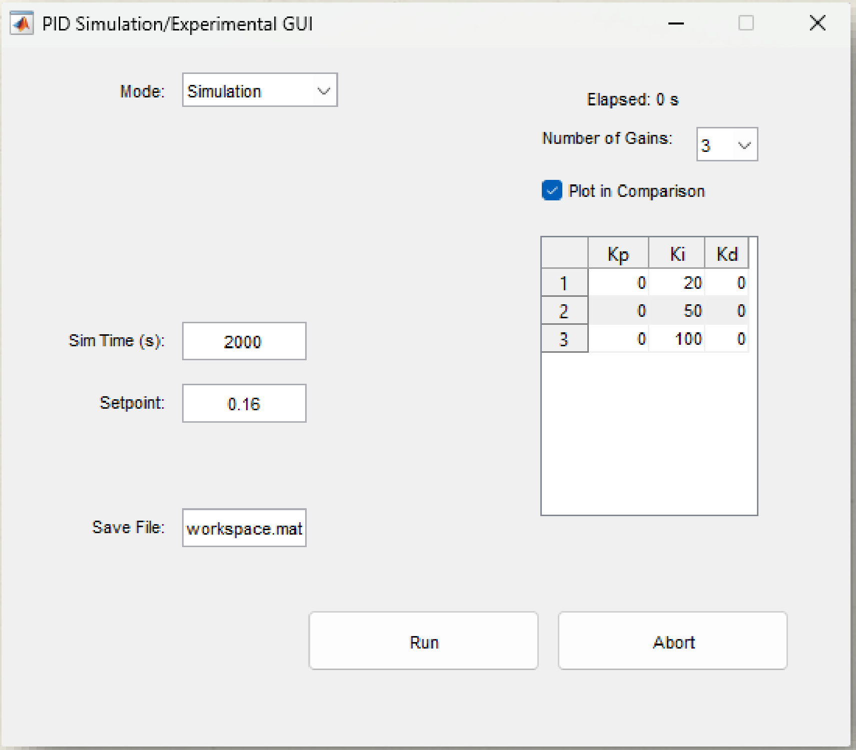
Simulation Mode

- Automation of the simulation experiments



GUI Interface

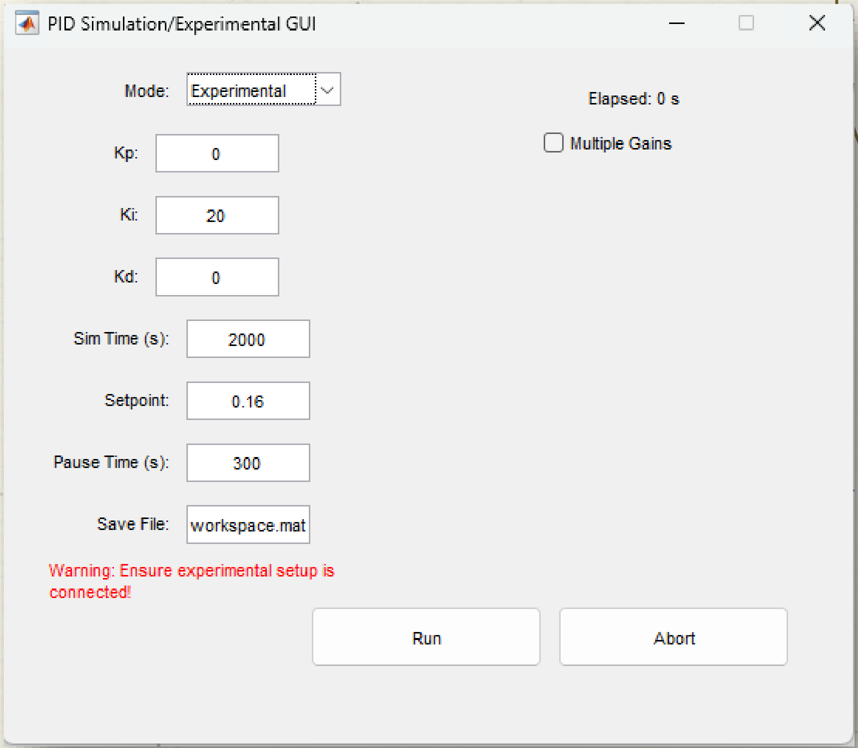
Simulation Mode:
Multiple Gain combinations



GUI Interface

Experimental Mode

- Automation of the experiments



Conclusion

- Simulation predictions closely match the experimental outcomes overall.
- Effects of various control strategies can be seen from the results.
- Cost effective design and provides a practical teaching tool.

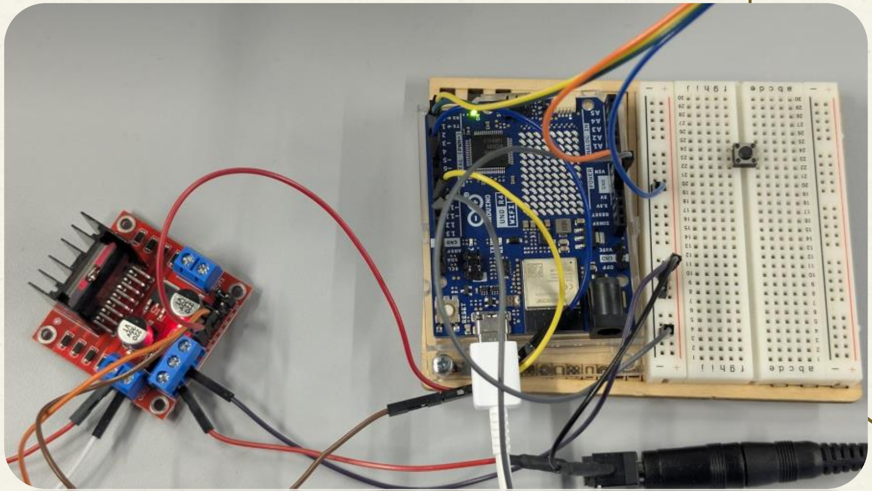
Future work

- Investigate LIDAR sensors for distance measurement, which could provide less noise.
- Implement the practical setup in a learning environment for student feedback.
- Incorporate adaptive control strategies.

%The End
Thank you!

7

Questions



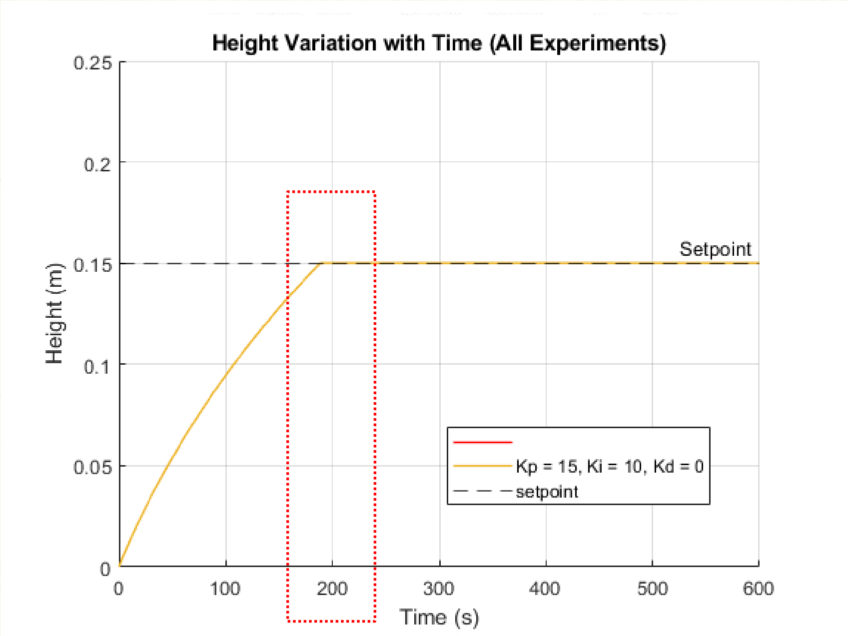
Literature review

- Åström, K. J., & Hägglund, T. (2006). *Advanced PID Control*. ISA.
- O'Dwyer, A. (2009). *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press.
- Seborg, D. E., Edgar, T. F., & Mellichamp, D. A. (2010). *Process Dynamics and Control* (3rd ed.). Wiley.
- Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2015). *Feedback Control of Dynamic Systems* (7th ed.). Pearson.
- Nise, N. S. (2011). *Control Systems Engineering* (6th ed.). Wiley.
- Ogata, K. (2010). *Modern Control Engineering* (5th ed.). Prentice Hall.

Resource Credits

- 3D models of Arduino, L298N: Printables.com
- Slidesgo for Slide template
- Fig 4 Amazon.de
- Fig 5 Funduino

Kp = 15, Ki = 10



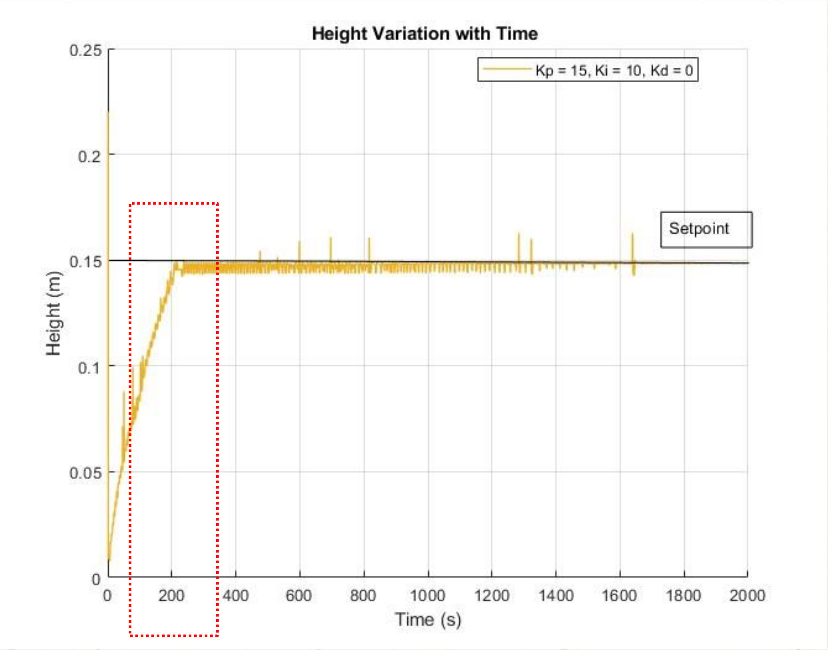


Fig 20(L) Sim,21(R)Exp : Height vs time: PI control, setpoint = 0.15 m

Damköhler number

- $Da_1 = \frac{\tau\ (residence\ time)}{T\ (chemical\ reaction\ time)}$
- $\tau\ (residence\ time)\ = \frac{V\ (Reactant\ Volume)}{Q\ (Volumetric\ Flow\ rate)}$