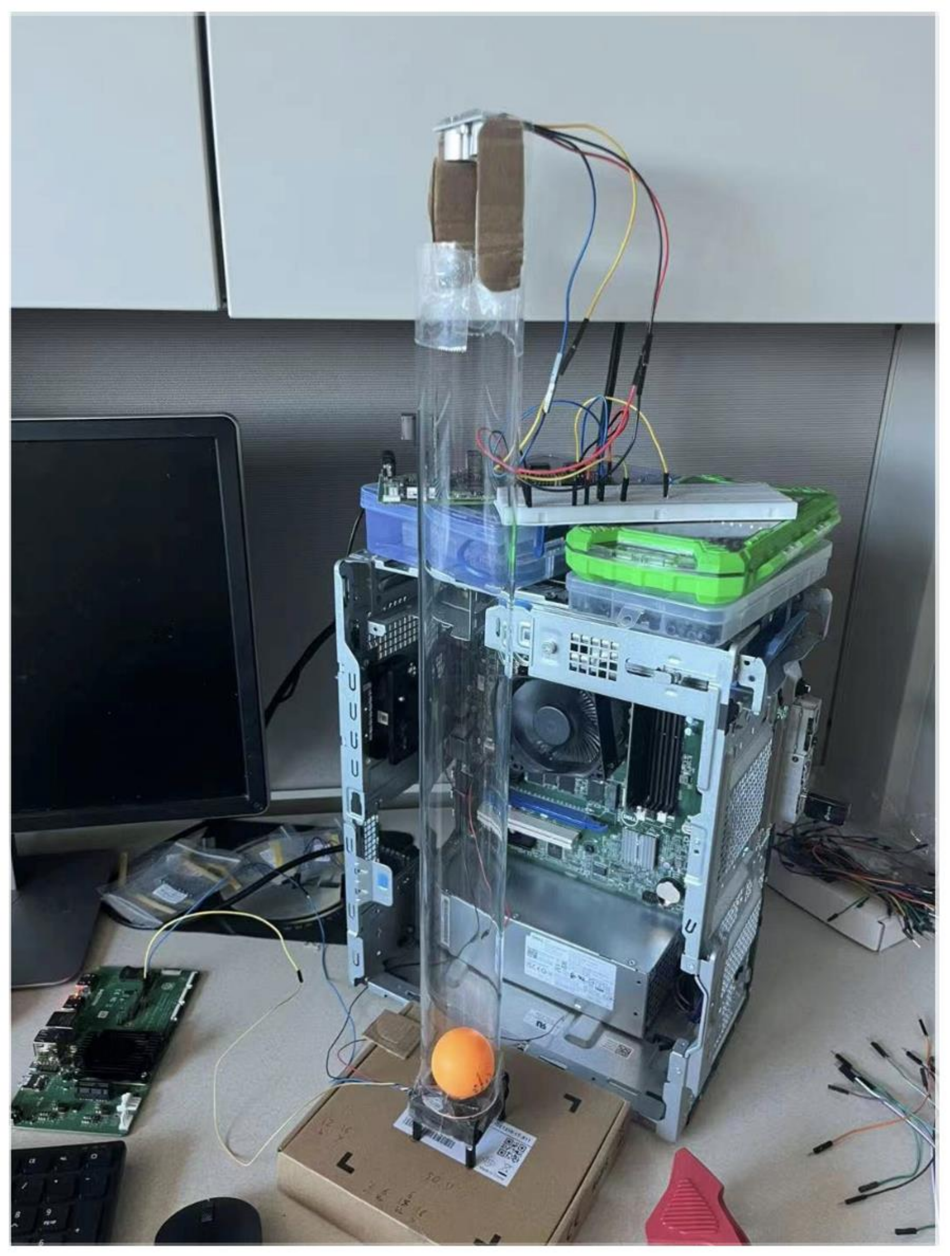
“Experimental Performance Evaluation of Real-Time Traffic Scheduler in Control Systems”

Progress Report I

SE5402/CSE5312: ARCHITECTURE OF INTERNET OF THINGS

FALL 2025

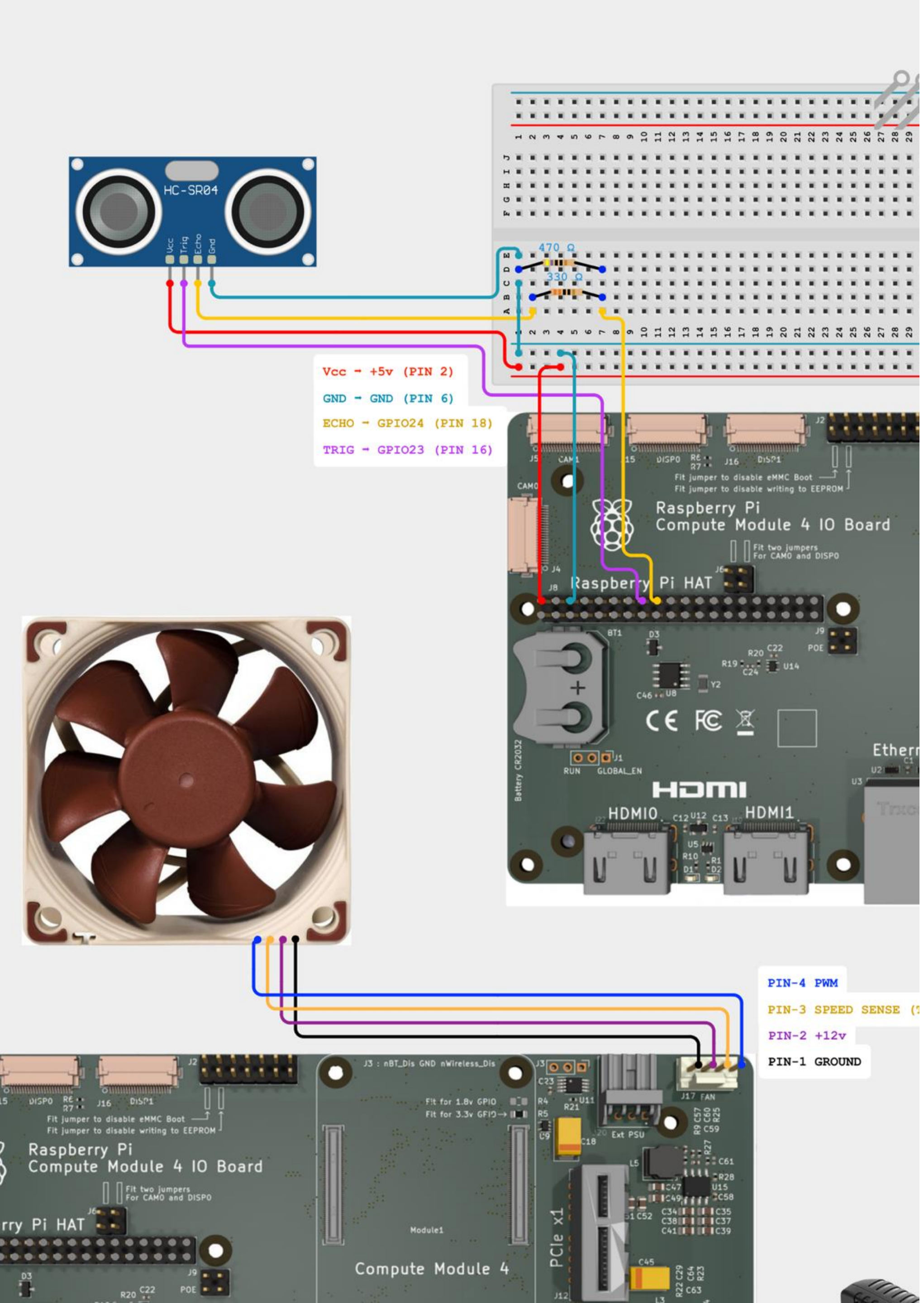
ABBY HORNING & JAKE THURMAN



Project Goal & Deliverables

- Project Goal (The “Why”)
  - The primary goal is to experimentally evaluate how applying a real-time traffic scheduler improves the quality of control for a physical system under heavy network congestion.
- Deliverables (The "What")
  - The project is divided into four phases, culminating in a full analysis:

Phase 1 & 2 (Baseline)	• Establish reliable communication between the **sensor node** and the **fan controller** over Wi-Fi. Implement the PID controller and successfully demonstrate performance degradation under network congestion.
Phase 3 (Evaluation)	• Implement and apply a real-time traffic scheduler (simulated via Linux **tc**) and collect improved performance data.
Phase 4 (Final)	• Analyze collected data, evaluate trade-offs (cost, performance), and deliver the final report and presentation.



Hardware Process & Challenges

Power	· Exchanged external fan power with direct IO board connection. · Lost an initial power supply, causing delays.
CM4 Availability	· Stability issues with original device; had to exchange for new.
Driver/Device Support	· Initially struggled with the PWM vs EMC2101/2301 support on the CM4 (likely exasperated by original problem device).

Software Architecture & Implementation

Control Loop (Sensor/PID Controller Node)

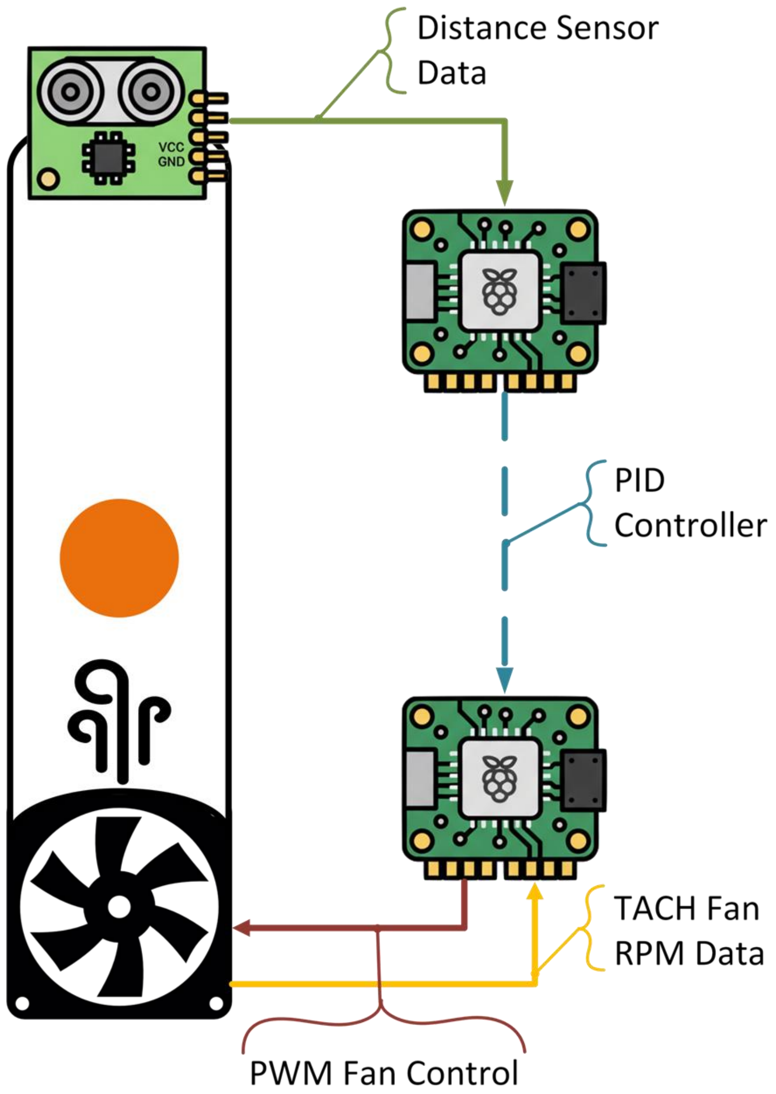
- Reads ball height and runs PID control loop
- Sends control output (fan speed %) over UDP

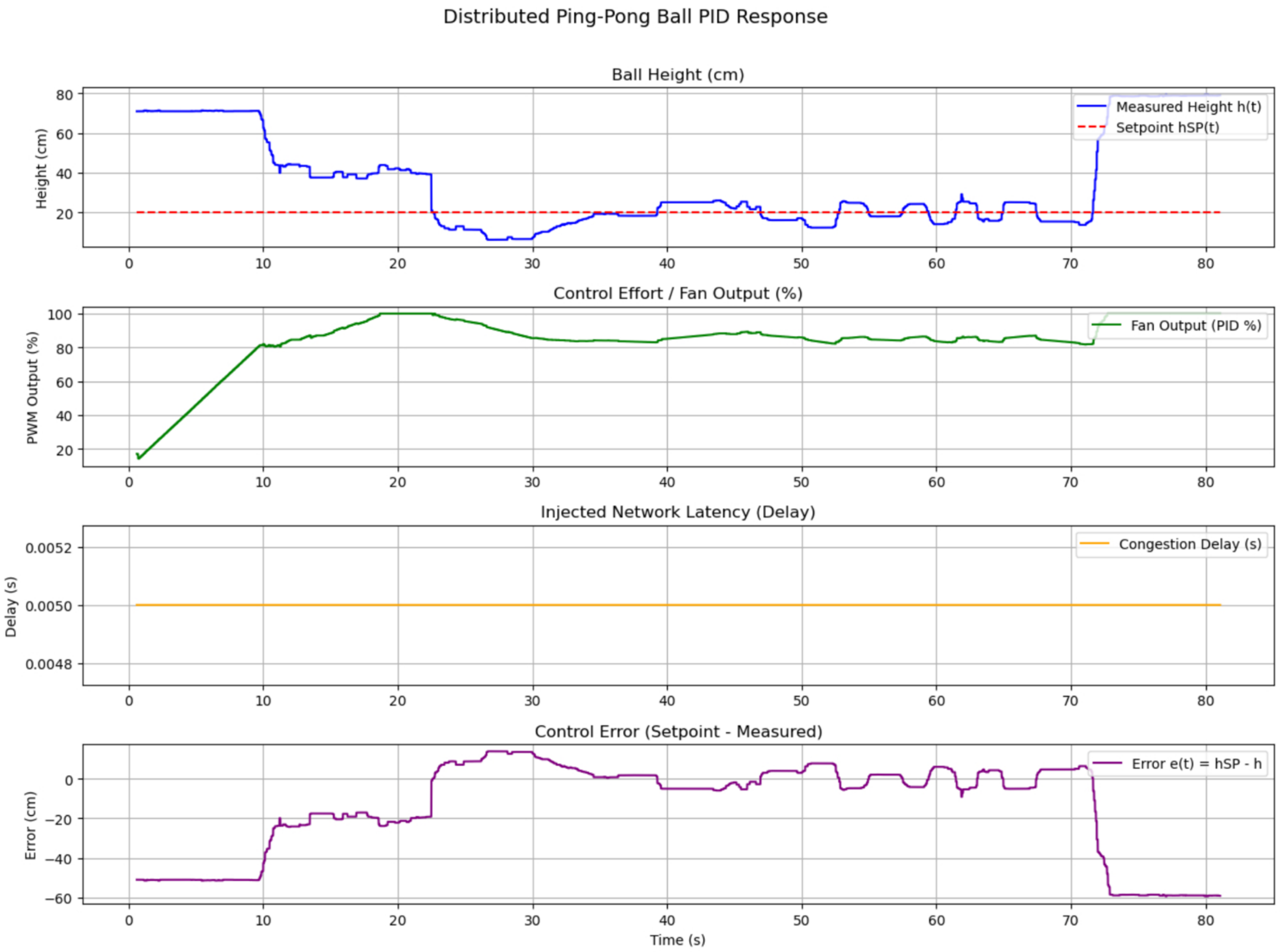
Fan Management (Fan Node)

- Receives control value and applies PWM to fan

Utility & Stability Scripts

- Configures static IP addresses for both Raspberry Pis
- Exchanges SSH keys to enable passwordless communication
- Injecting delay / jitter / bandwidth load for experiments





Real-Time Scheduling (Network Congestion)

Goal: Evaluate Control Performance

How does network latency affect the PID loop's ability to maintain a stable temperature?

Implementation:

- Linux Traffic Control (*tc*)

Congestion Test

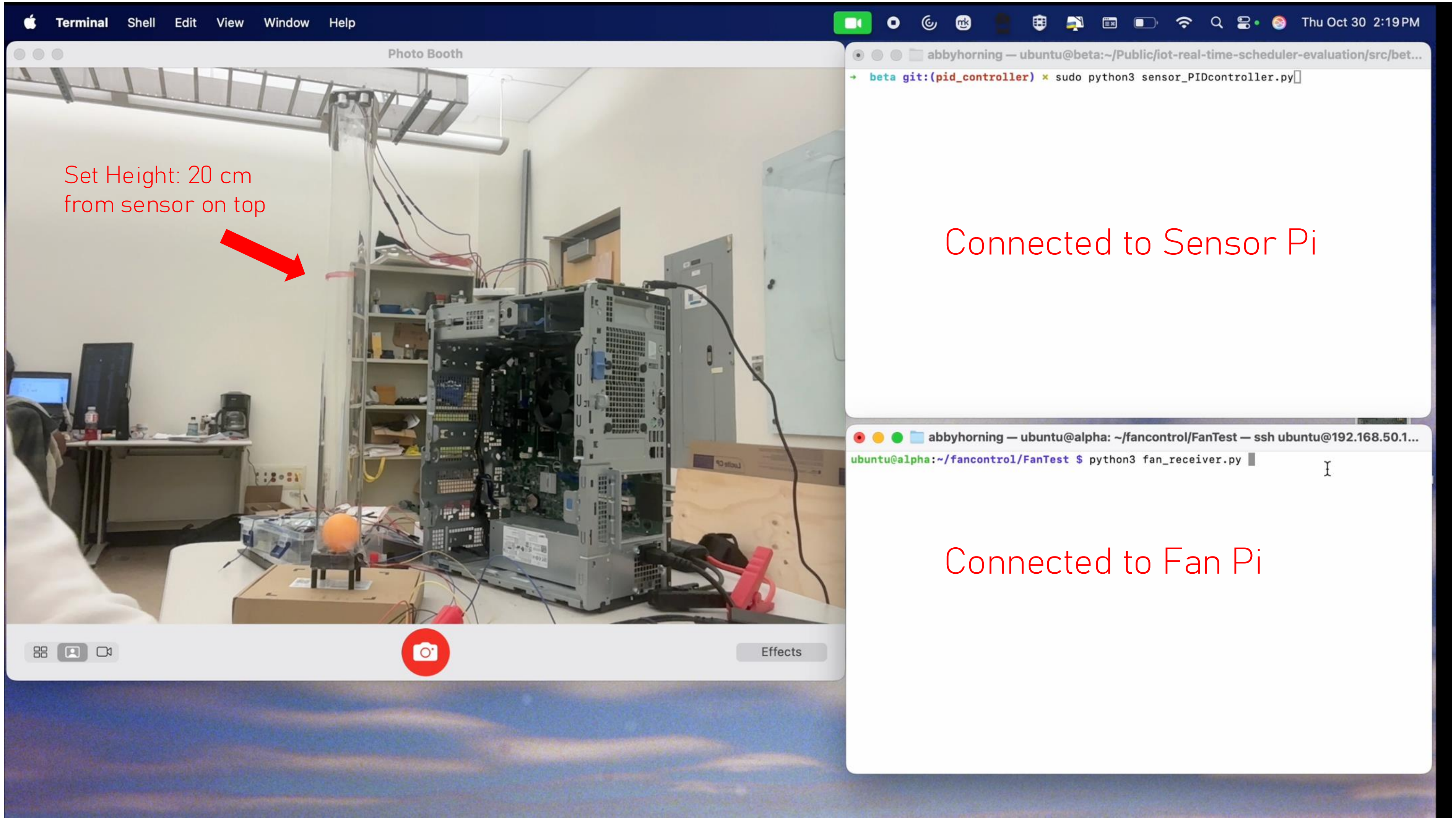
1. Degraded Baseline:
   - No *tc* rules applied.

2. Improved Control:
   - *tc* priority rules applied to the network interface.
   - UDP packets on port 5005 are filtered into a High-Priority Queue.
   - Bulk background traffic is relegated to a Low-Priority Queue.

Next Steps

- Final PID Tuning: Complete the final, precise tuning of the K_p, K_i, and K_d parameters to achieve optimal steady-state error and overshoot.
- Baseline Data Collection: Collect the clean performance data for our initial, non-congested run.
- Execute Congestion Experiments: Run the core experiment comparing Degraded vs. Prioritized network conditions using utility scripts.
- Final Analysis: Analyze the logged data (RPM, Duty Cycle, Time) and plot the results to visually demonstrate the benefit of the real-time scheduler.

Demo


Terminal Shell Edit View Window Help
Thu Oct 30 2:19 PM
Photo Booth
Set Height: 20 cm
from sensor on top
Effects
abbyhorning — ubuntu@beta:~/Public/iot-real-time-scheduler-evaluation/src/bet...
beta git:(pid_controller) ✗ sudo python3 sensor_PIDcontroller.py
Connected to Sensor Pi
abbyhorning — ubuntu@alpha: ~/fancontrol/FanTest — ssh ubuntu@192.168.50.1...
ubuntu@alpha:~/fancontrol/FanTest $ python3 fan_receiver.py
Connected to Fan Pi

References

- Salzmann et al. (2025): Hovering a ping-pong ball: A demonstration setup for teaching PID control
  - (https://doi.org/10.26434/chemrxiv-2025-328tk)
- An example of how Linux traffic scheduler improves control under traffic congestion:
  - (https://github.com/NXP/dds-tsn)
- Depth/Distance Sensors:
  - (https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/)
- Tuning PID parameters:
  - https://davidr.no/iiav3017/papers/Ziegler_Nichols_%201942.pdf
- PWM Fan Control:
  - PWM basics: https://www.arduino.cc/en/Tutorial/Foundations/PWM
  - PWM fan control: https://github.com/folkhack/raspberry-pi-pwm-fan-2

https://github.com/jthurm11/iot-real-time-scheduler-evaluation