Smart Throttle Control – Mid-Term Report

Name: Meghana Dumpala
Roll No: 240648
Project: Smart Throttle Control

Overall Summary

What I Learned So Far

During the first half of the Smart Throttle Control project, I learned the basics and practical aspects of control systems.

I started with open and closed loop systems, transfer functions, Laplace transform, bode plots and understood how physical systems like motors can be represented mathematically.

I learned PID control and the role of each term:

- Proportional control affects speed of response
- Integral control removes steady-state error
- Derivative control improves damping and stability

Then, I studied advanced control techniques such as:

- Lead compensators to improve transient response
- Lag compensators to reduce steady-state error
- Feedforward control for disturbance rejection
- MIMO systems and interaction between multiple inputs and outputs

I also learned how MATLAB Live Scripts are used to simulate systems, analyse responses, and compare performance.

I understood how control systems move from MATLAB simulations to real hardware using microcontrollers.

Tools and Techniques Used

- MATLAB Control System Toolbox

- MATLAB Live Scripts (.mlx)
- Step response and time-domain analysis
- PID tuning using pid() and pidtune()
- step(), lsim(), feedback() functions
- Performance metrics like rise time, settling time, and overshoot

---

Challenges Faced

- Understanding the effect of each controller parameter
- Handling steady-state error and overshoot together
- Visualizing interaction in MIMO systems
- Debugging MATLAB plots and simulation errors

My Contribution to the Project

I worked on:

- Designing and tuning PID controllers
- Analysing system performance using MATLAB
- Understanding compensators and feedforward control
- Preparing simulations and reports

Project Implementation Plan

Industrial Motor Speed Regulation

This system is commonly used in industries such as conveyors, robotics, and manufacturing machines.

Overall System Architecture

Block-Level Workflow

Reference Speed → Controller → Motor → Speed Sensor → Feedback

1. Desired motor speed is given as reference
2. Speed sensor measures actual motor speed
3. Controller compares reference and output
4. Control signal is sent to motor
5. Feedback loop corrects the error continuously

Key Components / Blocks

1. Sensor

- Measures motor speed
- Example: Encoder or tachometer

2. Controller

- PID controller implemented digitally
- Calculates control signal based on error

3. Actuator

- DC motor
- Converts electrical signal into rotation

4. Feedback Loop

- Sends output back to controller
- Enables error correction

5. Safety Mechanisms

- Actuator saturation limits
- Prevents damage due to high control signals

Control Techniques Used

PID Control

- Ensures accurate speed tracking
- Eliminates steady-state error
- Improves stability

Lead Compensator

- Speeds up the motor response
- Reduces rise and settling time

Lag Compensator

- Improves steady-state accuracy
- Useful when precision is required

Feedforward Control

- Handles known disturbances
- Reduces error before feedback reacts

Digital Implementation Approach

The controller is implemented on a microcontroller.

Digital Control Cycle

1. Read speed using ADC
2. Compute error
3. Run PID algorithm
4. Output control using PWM
5. Repeat at fixed sampling time

Digital control allows easy tuning, data logging, and flexibility.

Why This System is Relevant

Industrial motor speed control:

- Improves efficiency

- Reduces energy consumption
- Ensures smooth operation
- Is widely used in real-world applications

This makes it an ideal system for applying control concepts.

Conclusion

This mid-term phase helped me build strong fundamentals in control systems.

I learned how theory connects with simulation and hardware implementation.

The project improved my understanding of real-world control challenges and prepared me for advanced control topics in the future.