Smart Throttle Control Project

Mid-Term Report

SABEEH MUHAMMED MC
Roll No: 240898

1. Overall Summary

This mid-term report summarises my learning progress, technical understanding, and project involvement in the Smart Throttle Control project. The sessions and assignments have strengthened my foundation in control systems, particularly in modelling, response based analysis of system, stability assessment, and controller design. In a nutshell project allowed me to build intuition on control system design.

Key Concepts Learned

- Relationship between system poles and time-domain behaviour
- First and second order step response characteristics
- Final Value Theorem
- how to position poles and zeros for optimal speed,accuracy and stability.
- plant based and response based modelling
- PID Control
- use of system type to predict error in different input classes
- Feedback and Feedforward loops
- MIMO
- Hardware implementation of simulated system

Tools and Techniques Learned

- Modelling systems in laplace domain
- MATLAB Control System Toolbox
- Simulink

- General response analysis ”lsim()”
- Step response analysis & stepinfo()
- Response comparison across controller designs
- Simulation-based verification before real implementation

Personal Contribution to the Project

My active contributions include:

- Solving analytical questions related to:
- Running simulations in matlab and designing basic controllers
- Building intuition to understand the design process
- Designing a quadcoptor system.

These reinforced both conceptual understanding and problem-solving ability.

Challenges Faced

- debugging matlab code in matlab livescript,as there were many sub-questions for a single plant model in assignments.
- struggled a little with building intuition on pole/zero locations in transfer function and their implications.
- while plotting bode plotes for 2nd order transfer function could’nt understand the intuition of slight overshoot at natural frequency.

I addressed these problems by using online resources for learning, used LLMs as a tutor. And kept in touch with my project mentors.

Overall, the learning experience so far has made me more confident in interpreting dynamic behaviour using both intuition and mathematics.

Project Implementation Plan: Quadcopter Drone Control System

System Overview and Design Philosophy

The objective of the proposed system is to control a quadcopter drone using remote user commands specifying the desired altitude, direction of motion, and speed. The fundamental control challenge is to appropriately regulate the RPM of four independent rotors such that these three objectives are achieved simultaneously in a stable and coordinated manner.

The design philosophy adopted is based on the following ideas:

- Altitude control is primarily achieved through symmetric thrust, where all four rotors operate at the same nominal RPM.
- Directional motion is achieved by intentionally creating a controlled tilt of the drone through differential RPM between paired rotors.
- Speed regulation is achieved by maintaining the required tilt angle while modulating thrust magnitude.
- Environmental disturbances are handled proactively using feedforward compensation, while user commands are handled using feedback control.
- Since altitude, speed, and direction are inherently coupled, the overall system is treated as a multivariable (MIMO) control problem rather than independent single-input single-output loops.

This approach closely mirrors how practical quadcopter systems are structured.

Overall Workflow and System Architecture

The quadcopter control system is structured as a layered closed-loop architecture that transforms high-level user commands into low-level actuator control while maintaining stability, robustness, and safety. The overall workflow follows a clear signal flow from sensing to actuation, with feedback and feedforward paths integrated to handle both user intent and real-world disturbances.

System Architecture

The system operation begins with the **remote inputs**, where the user specifies the desired altitude, direction of motion, and speed. These commands are interpreted by the **command interpreter**, which converts them into physically meaningful reference signals while enforcing safety limits such as maximum allowable tilt angles or thrust levels.

Simultaneously, the drone's onboard **sensor suite** measures the current state of the vehicle. This includes inertial measurements (gyroscope and accelerometer), altitude information from barometric or LiDAR sensors, velocity estimation from GPS or optical flow,

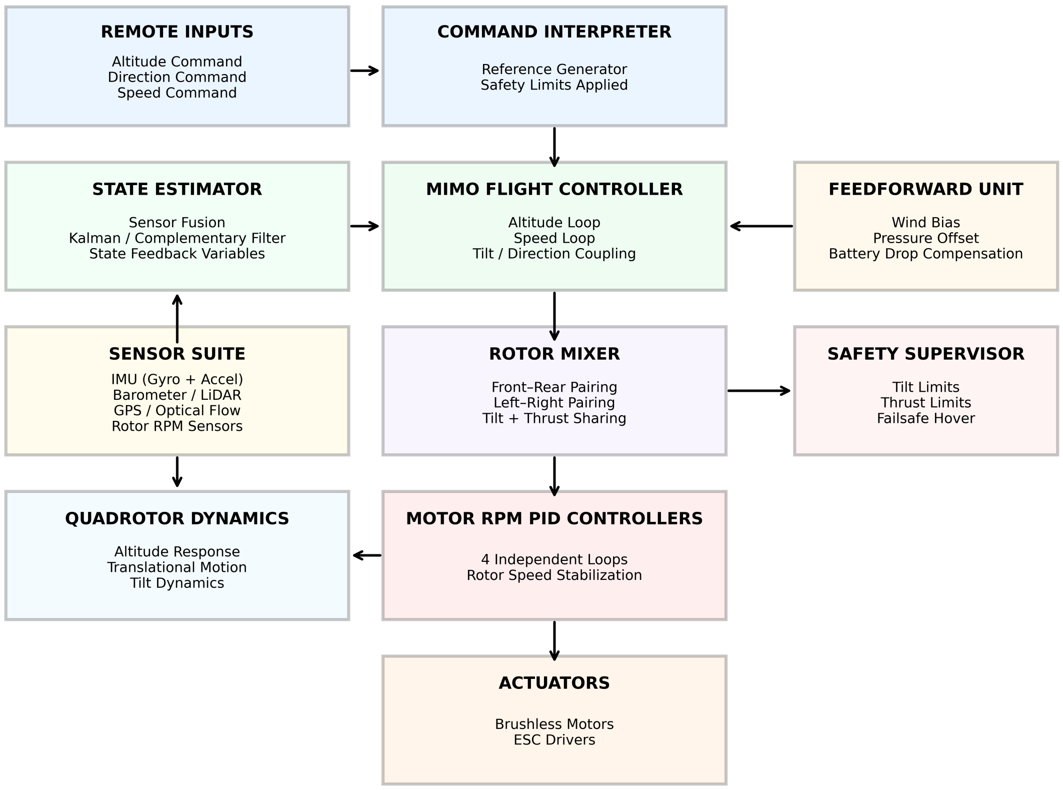

Figure 1: High-level block diagram of the quadcopter control system showing sensor inputs, control layers, actuator interfaces, and feedback paths.

and rotor speed feedback. Since raw sensor data is noisy and incomplete, it is processed by a **state estimator**, which performs sensor fusion and filtering to produce reliable estimates of altitude, velocity, and orientation.

These estimated states, along with the user-defined references, are fed into the **MIMO flight controller**. This controller simultaneously regulates altitude, speed, and direction while explicitly accounting for the coupling between these variables. For example, when the drone tilts to achieve horizontal motion, the controller compensates for the resulting loss of vertical thrust to maintain the commanded altitude.

To improve performance under predictable disturbances, a **feedforward unit** injects compensation terms based on factors such as wind bias, pressure variation, and battery voltage drop. This proactive compensation reduces the burden on the feedback controller and improves response speed.

The controller outputs are passed to the **rotor mixer**, which converts high-level thrust and tilt commands into individual rotor RPM references. This block implements the coupling logic between rotor pairs (front–rear and left–right) to generate controlled tilt while preserving overall thrust balance.

Each rotor is regulated by an **inner-loop motor RPM PID controller**. These fast inner loops ensure accurate tracking of the commanded rotor speeds and isolate the higher-level controller from actuator dynamics. The resulting control signals are applied to the **actuators**, consisting of brushless motors and electronic speed controllers (ESCs).

The physical response of the drone is captured by the **quadrotor dynamics** block, which includes altitude response, translational motion, and tilt dynamics. The resulting motion is sensed again by the onboard sensors, closing the feedback loop.

Finally, a **safety supervisor** continuously monitors system behavior and enforces constraints such as tilt limits, thrust saturation, and failsafe hover modes. This supervisory layer operates independently to ensure safe operation even under unexpected conditions.

Overall, this architecture reflects a practical and scalable quadcopter control system, combining feedback control, feedforward compensation, and multivariable coordination in a structured and modular manner.

Key Components and Functional Blocks

Sensor and State Estimation Block The system uses a combination of sensors such as an inertial measurement unit (IMU), altitude sensors (barometer or LiDAR), velocity estimation (GPS or optical flow), and rotor RPM feedback. These measurements are filtered and fused to obtain reliable estimates of altitude, velocity, and orientation. Accurate state estimation is critical, as all control decisions depend on these values.

Reference and Command Processing User inputs from the remote controller are converted into reference signals for altitude, speed, and direction. This block also applies basic constraints to ensure commands remain within safe operational limits, preventing excessive tilt angles or thrust demands.

Multivariable Flight Controller The core controller coordinates altitude, speed, and directional motion. Rather than treating these objectives independently, the controller accounts for their coupling. For example, when the drone tilts to move forward, the controller compensates for the resulting loss of vertical thrust to maintain altitude. This coordination reflects a MIMO control perspective.

Rotor Mixing and Coupling Logic Directional motion is achieved by coupling specific pairs of rotors. For forward or backward motion, the front and rear rotor pairs are differentially controlled. For lateral motion, the left and right rotor pairs are adjusted. This block ensures that tilt is generated in a controlled manner while preserving overall thrust balance.

Inner Motor RPM Control Each rotor is equipped with a fast inner-loop PID controller that regulates its RPM. These inner loops ensure that commanded rotor speeds are achieved quickly and accurately, isolating the higher-level controller from motor and actuator dynamics.

Feedforward Compensation To improve performance under real-world conditions, feedforward terms are used to account for disturbances such as wind gusts, pressure variations, and battery voltage drop. By compensating for these effects before significant errors develop, the burden on the feedback controller is reduced. The inputs for feedforward control can be obtained real time through sensor,for eg: windspeed can be detected through Anemometer.Also we could use microcontrollers programmed with wind data from that region to predict wind gusts at different altitudes to avoid the lags feedback cycles may cause due to clockspeed limitations.

Safety and Supervisory Mechanisms A supervisory layer enforces safety constraints such as maximum tilt angles, thrust saturation limits, and emergency hover or descent modes. This layer operates independently of the main controller to ensure safe operation even in the presence of sensor faults or unexpected disturbances. .