Flight Controller Firmware Development and Simulation
Blending Digital Control and Embedded Implementation

Author: Souleymane Dembele

Advanced Embedded Systems / TECE 512

Instructor: Dr. Jie Sheng

March 15, 2025

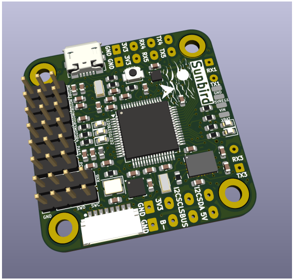
Project Objectives and Requirements

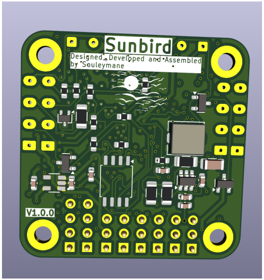
- Design and analysis of digital controllers using state-space techniques (LQR, LQG, MPC)
- System identification from experimental data and state-space model synthesis
- Digital controller implementation on an Arduino/embedded platform
- Application of Kalman filtering to fuse sensor data for improved performance
- Comprehensive simulation environment for control strategy validation
- Embedded system integration with real-time sensor processing and control feedback

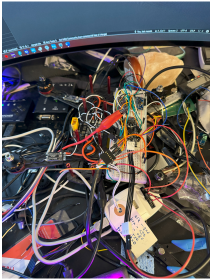
Hardware Design and PCB Development

- PCB Version 1: STM32F405
  - Power regulation: 3.3V and 5V power lines
  - Sensors: Accelerometer, Magnetometer, Barometer, Gyro
  - External flash: 128Mb (initially 16Mb)
  - Radio communication module
  - Completed and tested
- STM32F411 Development Board for software testing

Hardware Design and PCB Development (Images)







Firmware Directory

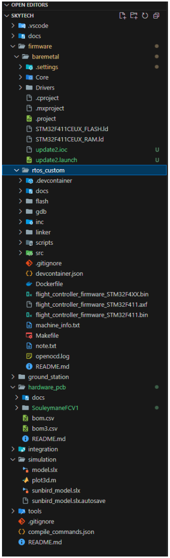


Figure: Firmware Directory Structure

Sensor Integration, Quaternion Computation & Sensor Fusion

- **Integrated Sensors:**
  - **ICM20602:** High-accuracy accelerometer and gyroscope data via SPI/I2C.
  - **BNO080:** 9-DOF sensor with onboard sensor fusion for reliable orientation via SPI.
- Real-time quaternion computation (Yaw, Roll, Pitch) supports flight attitude control.
- **Kalman Filter:** Fuses data from both sensors to reduce noise and enhance estimation accuracy.

Communication Interfaces

- CRF Protocol Implementation (Express LRS LoRa)
- UART for debugging and telemetry
- UART data parsing with a state machine (interrupt-driven)
- Reliable UART data transmission confirmed

Timers, PWM, and RTOS Integration

- **PWM Implementation:**
  - Motor control via ESC
  - Buzzer sound patterns
- Timer configuration for precise module cycles
- Custom lightweight RTOS for task scheduling and inter-task communication

ESC Protocol Comparison

- PWM Protocols:
  - Standard PWM (500Hz)
  - **OneShot125**, OneShot42, Multishot
  - Digital Protocols: Dshot, Proshot

Digital Controller Design & Embedded Implementation

- **Controller Design and Analysis:**
  - Utilized state-space techniques (LQR, LQG, MPC) to design robust controllers.
  - Analyzed system stability and performance through simulation.
- **System Identification:**
  - Collected experimental data to develop accurate state-space models.
  - Enabled controller synthesis based on real-world dynamics.
- **Embedded Implementation:**
  - Deployed the digital controller on ARM cortex
  - Integrated sensor data processing and real-time control feedback.
- **Kalman Filtering:**
  - Applied to fuse data from the ICM20602 and BNO080.
  - Enhanced accuracy of state estimation and improved control

Simulation Environment Design

- Developed a complete simulation of the flight controller system.
- **Key Features:**
  - Dynamic modeling of flight dynamics and system response.
  - Emulation of sensor outputs (BNO080 and ICM20602) for realistic testing.
  - Integration of state-space control design and Kalman filter performance.
- Utilized MATLAB/Simulink and custom tools for control strategy validation.

Kalman Filter Implementation Details

- **Purpose:** Enhance sensor fusion and reduce measurement noise.
- **Process:**
  - **Prediction:** Estimate the next state based on prior data.
  - **Correction:** Update estimates using incoming sensor measurements.
  - **Iteration:** Continuous updating for real-time performance.
- Resulted in improved control loop performance and flight stability.

Mathematical Modeling: Battery, Motor, and Propeller Performance

- **Battery:**

$$V = 3.8 \times 3 = 11.4\,\text{V}, \quad I_{\max} = 3.83 \times 20 = 77\,\text{A}.$$

- **Thrust Calculation:**

$$\text{Thrust} = C_T\,\rho\,n^2\,D^4,$$

where

$$C_T = 2 \times 10^{-15}\,\text{RPM}^3 - 4 \times 10^{-11}\,\text{RPM}^2 + 3 \times 10^{-7}\,\text{RPM} + 0.101307.$$

- **Torque Calculation:**

$$\text{Torque} = 4 \times 10^{-14}\,\text{RPM}^3 + 8 \times 10^{-12}\,\text{RPM}^2 + 3 \times 10^{-6}\,\text{RPM},$$

and using the motor constant,

$$K_T = \frac{1}{K_V} = \frac{1}{1400} \approx 0.007, \quad I = 1400 \times \text{Torque}.$$

Mathematical Modeling: Drone Dynamics and Control Forces

- **Translational Dynamics (Newton's 2nd Law):**

$$F_x = ma_x, \quad F_y = ma_y, \quad F_z = ma_z.$$

- **Rotational Dynamics:**

$$M_x = I_x\ddot{\theta}_x, \quad M_y = I_y\ddot{\theta}_y, \quad M_z = I_z\ddot{\theta}_z.$$

- **Moments of Inertia:** For a rod of length L,

$$I = \frac{1}{3}mL^2.$$

Approximations for our drone:

$$I_z \approx 0.007\,\mathrm{kg}\cdot\mathrm{m}^2, \quad I_{x,y} \approx 0.003\,\mathrm{kg}\cdot\mathrm{m}^2.$$

- **Control Moments:**

$$M_x = (F_3+F_4-F_1-F_2)\,d, \quad M_y = (F_3+F_2-F_4-F_1)\,d, \quad M_z = T_4-T_1+$$

Mathematical Modeling: Drag and Overall Force Balance

- **Drag Force:**

$$D = \frac{1}{2}\rho V_{\text{wind}}^2 A C_d,$$

where A is the effective area.

- **Overall Force Balance:**

$$F_x = F_{\text{prop}_x} \pm D_x, \quad F_y = F_{\text{prop}_y} \pm D_y, \quad F_z = F_{\text{prop}_z} - mg \pm D_z.$$

- **Yaw Control:**

$$M_z = T_4 - T_1 + T_2 - T_3.$$

Dual-Rate GPC MATLAB Simulation Implementation

- **System Setup:**
  - Defined continuous-time dynamics with inertia (J) and damping (b) parameters.
  - Constructed the state-space model (A_c, B_c, C_c, D_c) and discretized it using a fast sampling period ($T_s = \frac{1}{1200}$).
- **Lifting and GPC Matrices:**
  - Used a lifting technique to build the lifted model over a horizon N via `buildLiftedModel`.
  - Constructed the prediction (Φ) and control (Γ) matrices with `buildGPCmatrices` for a prediction horizon P.
- **Control and Estimation:**
  - Designed a dual-rate GPC by minimizing a quadratic cost function.
  - Computed the Kalman filter gain using `dlqe` for robust state estimation.

Simulation Results

- Plots for pitch, roll, and yaw axes display:
  - The output angle (in degrees) versus the setpoint.
  - The corresponding control input over time.
- Results illustrate:
  - Fast response with minimal overshoot in the fast single-rate GPC.
  - The dual-rate approach achieving performance between fast and slow single-rate cases.

(See MATLAB figures for Pitch, Roll, and Yaw responses.)

Conclusion and Future Directions

- **Achievements:**
  - Developed a robust digital controller using state-space techniques validated via simulation.
  - Integrated sensor processing with the ICM20602 and BNO080 and enhanced performance using a Kalman filter.
  - Implemented dual-rate GPC and comparing with PID with a lifted model approach in MATLAB, enabling improved control performance.
- **Future Work:**
  - Further optimization of control algorithms and simulation parameters.
  - Extension to more complex dynamical systems and adaptive control strategies.
  - Deep-dive into stability robustness analysis under model-plant mismatch.

References

- Jie Sheng Tongwen Chen, "On Stability Robustness of Dual-Rate Generalized Predictive Control Systems," *Proceedings of the American Control Conference*, Arlington, VA, June 25-27, 2001.

PNC *Digital Control System Analysis and Design*, 4th ed., by Charles Phillips, H. Nagle, and Aranya Chakrabortty, 2015.

KP *Practical Design and Application of Model Predictive Control*, by Nassim Khaled and Bibin Pattel, available at `https://www.elsevier.com/books/practical-design-and-application-of-model-predictive-cont khaled/978-0-12-813918-9`.

WANG *Model Predictive Control System Design and Implementation Using MATLAB*, by Liuping Wang, Springer, 2009, ISBN: 978-1848823303.

- MathWorks PDF documentation for the Model Predictive Control Toolbox.