FCS Software Plan

Basic certification documents applicable:

- DO178C – DAL B/C (light-weight UAV without any human onboard)
- DO331 (for MBD)
- DO-330 (for Simulink code generation)
- ARP4754A

Development

A Requirements-Based Approach is mandated according to DO178C guidelines.

High Level Requirements

The software should have High Level Requirements (HLR) refining System Requirements. Each HLR should be assigned to a software *"component"*. The FCS architecture can be broken down into smaller software components based on functional and logical properties. For example,

- IO
- Redundancy_Manager
- FDIR_Manager
- FDIR_Notification
- Autopilot_Manager
- Mission_Manager
- Attitude_Controller
- Trajectory_Manager
- Guidance
- State_Estimator
- Algo_Lib
- Math_Lib

The HLR should clearly specify "what" the component is supposed to do to refine system requirements and it should be verifiable.

Low Level Requirements

Low Level Requirements (LLR) break down HLR into design artefacts which tell "how" the HLR is implemented.

Traceability

Bidirectional traceability should be maintained between:

- System requirements and HLR
- HLR and LLR
- LLR and code

- LLR and Unit Test
- HLR and High Level Test

Verification & Validation

A Requirements Based Testing should be performed to comply with DO178C guidelines.

For DAL-B level, decision and statement coverage is required. For DAL-C level, statement coverage is required.

Unit Testing

Unit testing (or Low Level Test) to be performed LLR which are to be verified at Requirement Level.

For Low Level Testing, the specific block implementing the design can be stubbed (using a test harness) to isolate it from external connections. Inputs can be triggered using MCDC input cases, and outputs should be verified against the LLR.

MBD testing strategy: MIL and SIL

High Level Test

High Level Test (HLT) are performed at component level of the software. It can be considered as black box test as you cannot break the connections between the blocks inside the component.

MBD testing strategy: MIL and SIL

Cross Target Testing

The generated C code from SIMULINK auto-code generator needs to be tested for the following criteria on the target platform:

- WCET (Worst Case Execution Time)
- Memory Profiling (RAM usage)
- Deadline Monitor
- Pre-emptive scheduling

MBD testing strategy: PIL

Integration Testing

System-level validation of the fully integrated Flight Control Computer (FCC) with all the electrical sensor, actuator and telemetry interfaces. This test should be performed only after reaching a matured baseline to avoid any major failures.

MBD testing strategy: HIL

Simulated External World

A simulated external world is required comprising UAV flight dynamics, sensor data, and telemetry to be able to perform MIL, SIL, and PIL HLT.

Flight Test Readiness Criteria

- All tests are passed
    - MIL SIL equivalence
    - PIL
- Ground Integration test passed
    - HIL
    - Simulated flight runs on cross platform
    - Fault Injection test passed
    - Telemetry tests
- Sensor calibration
- Aircraft characteristics calibrated

Configuration Management

Each artefact of the software development and V&V (HLR, LLR, code, test files, test reports, executables) can be categorized as CC1 or CC2 (based on DO178C guidelines). For CC1 category, each change in the artefact has to be tracked in baseline evolution (e.g. design, code, test files etc.). For CC2, only milestone changes are needed to be tracked (e.g. test reports, executable).

Git can be used for complete configuration management.

Each update on baseline should ensure:

- Software non-regression
- Review from independent author
- Real-Time performance non-regression