SWEN326 Requirements:

Notes:

Classify into functional and non-functional. Flight simulator and test software for the flight simulator.

https://nuku.wgtn.ac.nz/courses/19441/assignments/67810

Requirements Examples:

https://www.researchgate.net/figure/Samples-of-high-and-low-level-requirements_fig1_283315841

Overview:

Implement an aircraft simulation. Test it using a graphical user interface with a 2D map.

Requirements Plan:

Every time a change is made to the code, it must be extensively tested, and the requirements should be changed.

Flight simulator:

Pilot User Interface:

- Controls must be intuitive and easy to use.
- Input fields for entering data. Input fields must be validated for expected boundary and invalid
    - Latitude
    - Longitude
    - Altitude
    - Speed Restrictions
    - Expected times of arrival at each waypoint.

Control Signal Specifications:

- Autopilot control Frequency

- Execution Check Parameters
- Must update every second at minimum

Testing software:

Sensor Data Simulation and frequency:

- Airspeed Sensor, Altitude Sensor, Attitude Sensor, Engine Parameters

High-level Requirements

https://nuku.wgtn.ac.nz/courses/19441/files/2527695?wrap=1

https://nuku.wgtn.ac.nz/courses/19441/files/2633010?wrap=1

Note to reader: I have classified the types of requirements under letters so that when we want to refer to the first functional requirement, for example, we can simply say Requirement 1A.

Requirements numbering:

Requirements are numbered R1, R2, R3, R4. DO NOT change the number to the requirements. For example, if you get rid of the requirement R2, **do not** change R3 to R2 and R4 to R3. Keep them the same. If we change the numbers, when we refer to them in the low-level requirements we will be pointing to the wrong high-level requirement.

If the requirement is blue, e.g. R1, that means it's been linked to a low-level requirement at the bottom of this document.

Functional Requirements "A"

"What should a system do?"

1. R1: After a control signal is sent, execution must be verified by reading relevant data from the sensors within 200ms of command issuance.
2. R2: After a control signal is sent, the expected change of the sensor data one second after command issuance should be calculated.
3. R3: For control surfaces, if the calculated change due to a control signal is not within a 2% margin of error of the expected change, or within 5% for an engine parameter, the command should be re-sent.
4. R4: If a control signal command has not been responded to correctly three times in a row, a warning should be issued to the pilot via the user interface.
5. R5: Every warning issued to the pilot should be classified under one of three categories, “Critical”, “Major” or “Minor”.
6. R6: Autopilot control should have buttons to engage and disengage autopilot and manual override controls such as altitude, pitch and speed. Indicator lights should clearly indicate the status of the autopilot.
7. R7: All other factors remaining equal, the thrust should be linearly related to the airspeed.
8. R8: All simulated metrics should respond dynamically to changes in thrust, pitch, roll or yaw.
9. R9: All other factors remaining equal, an increase in thrust should increase the pitch and cause the aircraft to climb. Likewise, a decrease in thrust should cause a decrease in pitch.
10. R10: The impact of the ailerons and rudders on roll and yaw should depend on the thrust asymmetry of the aircraft. Thrust asymmetry causes roll and yaw in the direction of least thrust.
11. R11: Maximum and minimum thrust levels, and the cruise level, should be as specified in the specifications of the aircraft model being flown
12. R12: Rate of turn (measured in degrees) should be decreased at higher airspeed, and increased at lower airspeed, assuming equal roll and yaw.

13. R13: When the system receives a turn request from the pilot through the UI, the wing pointing in the direction of the turn should be lowered, resulting in a change in roll.
14. R14: The cruise level of thrust, for our simulation, must be within the idle range for a given aircraft. At cruise level, the aircraft should remain at the same altitude, assuming all other factors (e.g. air density, drag etc) remain the same.
15. R15: If the aircraft drops below or rises above the desired altitude, thrust and the elevators should be adjusted in order to return to the desired altitude.
16. R16: For any exceedance of the normal operational range for typical metrics (i.e. roll, pitch, yaw, thrust), a corrective action should be explicitly defined
17. R17: The entry into the flight plan of an altitude above 42,000 ft or below 0 ft should be ignored, and an explicit error message should be shown.
18. R18: The reading of an altitude above 40,000 ft by the sensors should result in a corrective action occurring.

Assumptions:

1. R17: For the purposes of this project, we will assume that all factors, except thrust, rudder, elevator, altitude and aileron movement, are independent and can be collectively simulated by random, uniformly distributed errors for all metrics i.e. airspeed, pitch, roll and yaw.
2. R18: As much of our specification assumes the plane is an Airbus A350, we will assume that the maximum level of thrust is 400kN, within the range for the Airbus A350.

Flight Plan Management:

1. R19: The system must allow the input of waypoints including latitude, longitude and altitude.
2. R20: It should enable users to input speed restrictions and expected times of arrival at each waypoint.

3. R21: A submit button must be available to load and activate the flight plan.
4. R22: A visual display (map) should show the current position, planned route and waypoints.

Autopilot Control panel:

1. R23: Provide buttons to engage/disengage the autopilot.
2. R24: Offer controls for manual override including altitude adjustment, speed, and heading.
3. R25: Indicator lights must indicate autopilot status (engaged, disengaged, fault condition).

Sensor Data Display:

1. R26: Display digital readouts for airspeed, altitude, pitch, roll, yaw, and engine parameters.
2. R27: Visual indicators should signify data update frequency.
3. R28: Program should respond if sensors give extreme values:
   a. This must trigger an alert mechanism and initiate actions if predefined thresholds exceeds.

Hazard Alerts:

1. R29: A dedicated section should exist for hazard warnings and mitigation actions.
2. R30: Audible and visual alerts must be provided for immediate hazards.
3. R30 Emergency procedures checklist or action plan should be available.

User Interface:

The user interface should contain:

1. R31 Altering the flight plan of the aircraft.
2. R32 Engaging or disengaging autopilot.

Thrust Levels:

1. R44 Maximum thrust level is used for takeoff and some climbing operations.
2. R45 Minimum thrust is used during idle on the ground or during descent.
3. R46 Cruise thrust is approximately 15-30% of maximum thrust, depending on cruise altitude. Note: In this project, we don’t need to consider aircraft load.

Non-Functional Requirements “B”

“What overall qualities should the system have?”

1. R33 A graphical user interface should be programmed. The user interface should be intuitive and easy to use. All buttons and input fields should be tested for boundary, expected and invalid input.
2. R34 Must adhere to safety critical software development principals, and ensure traceability from requirements to code
3. R35 System should be scalable and able to handle different types and amounts of sensor data
4. R36 The system should have thorough documentation of the code design and testing procedures and should be clearly accessible from the code
5. R56. The altitude, airspeed and attitude sensors should all have a 2 out of 3, redundant architecture.

Sensors:

1. R37 Fault detection, tolerance and fail-safe mechanisms should be implemented for abnormal sensor values.
    a. A control signal is considered successful if the sensor data’s expected change is within a margin of error of ±2% for control surfaces and ±5% for engine parameters, and within 1 second of command issuance.
    b. System must handle retry logic up to 3 times before failure notification.

2. R38 Airspeed Sensor: Program should simulate data to represent the aircraft’s speed between 50 to 500 knots.
3. R39 Altitude Sensor: Simulate barometric and GPS altitude data from –1,000 to 50,000 feet AMSL.
    a. Thresholds for sudden changes should be set to indicate potential sensors faults.

Reliability:

1. R40 The system must operate reliably under normal and abnormal conditions.
2. R41 It should ensure minimal downtime and robustness against failures.

Safety:

1. R42 The system must priorities safety, providing accurate data and warnings to prevent accidents.
2. R43 It should have fail-safe mechanisms to handle sensor faults and mitigate potential hazards.

Usability:

1. R47 The user interface should be intuitive and easy to use, considering the stressful nature of flying.
2. R48 It should provide clear and concise information to assist pilots in decision making.

Performance:

1. R49 Sensor data updates should be timely and responsive, meeting specified frequencies.
2. R50 Control signals should be executed promptly with minimal delay.

Maintainability:

1. R51 The system should be easy to maintain and update, with clear documentation and modular design.
2. R52 It should support diagnostic tools for troubleshooting and repair.

User Interface:

1. R53 Sensor data should be displayed on the screen.

Test Aircraft:

R54 Test simulations of real aircraft that are used on international routes in 2024.

Safety Requirements “C”

1. R55 The system should send control signals to the aircraft’s control surfaces at most 500ms after the previous control signal.

SWEN326 Low-Level Requirements:

Remember: All low-level requirements must be linked to a high-level requirement!

Specifications to write the code from them.

Plan:

Notes:

TODO: Convert this into a markdown file

1. API to communicate between sensors. Asynchronous: order of execution is not determined by the lines of code.

Requirements:

1. An asynchronous API will be used for sensors to communicate with the main application – Parent requirement: R1
2. Parent requirement: R2 Display the sensor values on the screen for the pilot to view them. They should be displayed with the units and should be easy to read and understand.
3. After issuing a control signal, the expected change of the data one second after command issuance should be calculated and another

control signal should be sent one second after command issuance. If the observed change is not within the acceptable range (+-2% for control surfaces or +-5% for engine parameters) three times in a row, the pilot should receive a critical warning regarding sensor value – Parent requirements: R2, R3, R4, R16

4. The pilot should be able to easily disengage autopilot. After disengaging autopilot, the plane should continue to self-manage until a control is overridden. Manual-overriding should be impossible in autopilot mode, and the current mode should be clearly displayed to the pilot - Parent requirement: R4
    a. Buttons should be provided on the interface to engage or disengage the autopilot. They should be intuitive and easy to use.
5. Implement a classification system within the software range and which display as “Critical”, “Major” and “Minor” i.e. we can use a decision tree to display and Visual cues i.e. color based like red, green and yellow. Parent requirement: R5
6. An algorithm or function that will ensure that the thrust adjustment is linearly affecting the airspeed. Implement event listeners whenever there is change R7
7. Every calculated sensor value should be subject to random errors following a uniform distribution, to simulate changes in air pressure. Parent requirement: ??
8. Parent requirement: R9 An increase in thrust should cause the simulated aircraft to increase the thrust and rise
9. Cruise thrust will be between 15% - 30% of the maximum thrust, depending on the cruise altitude. Cruise thrust will increase with altitude – Parent requirement: R46
10. When a turn request is received, control signals should be sent to the airelons and rudders to adjust the roll and yaw accordingly. – Parent requirement: R13
11. The size of the turn (in number of degrees) should depend on the airspeed. A higher airspeed results in a smaller turn – Parent requirement: R12

12. If a command to turn left is received, the left airelon should rise and the right aileron should be lowered. This will lower left wing and change the roll, resulting in a turning motion – Parent requirement: R13
13. Each sensor (altitude, attitude and airspeed) should be composed of three redundant components which cooperate to return a single value – Parent requirement: R56
14. The value returned by any sensor (altitude, attitude or airspeed) should be agreed upon by at least 2 out of 3 redundant components. If all components disagree, an explicit error message should be returned to the caller, and the UI should show a critical warning to the pilot - Parent requirement: R56
15. Implement a single, Intuitive form on the User interface, it will include Latitude, Longitude and altitude of waypoints, Speed restrictions and expected time of the arrival at each waypoint. Plus, a submit button Parent requirement: R18,19,20
16. Integrate Dynamic maps that will enable the user to add, remove and modify waypoints directly on the map interface, with change reflected in real time. Alos using visual cues (like color coded routes). Use of Gmapsfx which will be bridge between Google maps with JavaFX. Parent requirement: R19,20,21.
17. Use of JavaFx scene Builder to design to bind the UI components directly to the backend autopilot control mode, this will also include JavaFx LED components to represent the status of the autopilot and Buttons to engage/disengage the Autopilot. Parent requirement: R22,23,24.
18. Visual display and alerts using JavaFX dialogs will help in displaying digital readouts, also will enhance the notifications to inform pilots when sensor reading exceeds. Parent requirement: R25,26,27.
19. Sensors should return an explicit confirmation message after receiving a control signal, within 200ms. To ensure that the correct control signal has been received, the sensor should return an identifier of the control signal, such as a sequence number or the time when the control signal was received – Parent requirement: R1

20. At any given time, the acceptable range of yaw should be +-5 degrees from the heading, where North is defined as yaw = 0, West is defined as yaw = -90, and South is yaw = +-180. This requirement is necessary as we must define when a corrective action must happen – Parent requirement: R57
21. As per the standards of the Airbus, pitch must be between 30 and –15, where 90 is defined as straight up, and negative 90 is defined as straight down. If the pitch exceeds these limits, a corrective action must occur, and the pitch should be returned to 0 – Parent requirement: R57
22. Parent Requirement: R11 The simulation program will have several aircraft models to test on. These models should contain:
    a. Engine Type
    b. Maximum Thrust
    c. Minimum Thrust

    The program should represent these aircraft as objects.
23. The cruise level for any aircraft model will be assumed to be 30% of the maximum level of thrust at the maximum flight altitude (42,000 ft) and 15% at ground level. Between these bounds, cruise thrust is assumed to be linearly related to altitude. If the thrust is 2.5% below the cruise thrust, or 5% above the cruise thrust, and the plane is in cruise mode, a corrective action must occur, and the thrust must return to cruise level – Parent requirement: R57
24. The maximum roll will be assumed to be 25 degrees in either direction. If this limit is exceeded, a corrective action must occur and the plane must return to 0 roll, in other words horizontal – Parent requirement: R57
25. If a corrective action is undertaken, and the control signal fails, a critical warning with a relevant error message should be displayed to the pilot via the UI – Parent requirement: R27
26. If a command is given to increase the altitude of the plane, the system should adjust the plane’s thrust and elevators enough to lift the nose to a pitch of 20. The plane should stay at this pitch until the desired altitude is reached – Parent requirements: R9,15

27. If a command is given to decrease the altitude of the plane, the system should adjust the plane’s thrust and elevators enough to lower the nose to a pitch of –10. This state should persist until the desired altitude is reached – Parent requirements: R9,15
28. A turn command will result in an adjustment of the ailerons sufficient to induce roll of 20 degrees in the direction of the turn. This state will persist until the desired heading is achieved – Parent requirement: R8
29. Asymmetric thrust should inadvertently occur to test the system’s robustness and ability to correct itself. – Parent requirement: R10
30. Asymmetrical thrust (i.e. jet engines on either side producing different amounts of thrust) should result in a yaw in the direction of least thrust and roll towards the least powerful engine. These effects should be proportional to the size of the disparity between the engines. - Parent requirement: R10
31. When the plane is headed straight, the roll should be kept at 0. If the roll exceeds 5 degrees in either direction, a corrective action must occur.
32. Waypoints should be adjustable through the user interface, with fields for longitude, latitude and altitude – Parent requirement: R18
33. The pilot must be able to create waypoints by selecting points on the map – Parent requirement: R18
34. Waypoints should be easily removable through the user interface, preferably by selecting waypoints on the map, and the autopilot must receive the new path and adjust the flight course accordingly – Parent requirement: R18
35. Environment values (i.e. speed, pitch, roll, yaw etc) must be updated every second. The new values must be calculated using the most recent measurements – Parent requirement: R2
36. After a control signal is sent, the controller should receive the environment values exactly one second afterwards, and these should be compared to expected values to ensure the command has been executed correctly – Parent requirement: R2

Flight Plan Management:

1. Parent Requirement: R21 A 2D map should show the location of various places. The aircraft’s current position should be displayed, as well as the aircraft’s coordinates in the world. A coordinate system will need to be designed.

Derived Requirements:

1. If the heading of the plane is not within 2.5 degrees of the direction to the next waypoint (or destination), a corrective action must occur.