JSBSim Flap Control System – Requirements and Design Specification

Prepared for: CSE Project – JSBSim Requirements Phase
Prepared by:

Saru Karki, ……………………………………

Date: October 19, 2025

CSE 3310

1. Purpose & Scope

The purpose of this document is to define the requirements and high-level design of the **Flap Control System (FCS)** modeled in **JSBSim**.

This system simulates flap control behavior for an aircraft, including **switch**, **gain**, and **kinematic blocks**, as defined by the JSBSim XML schema.

It outlines planned implementation classes, methods, and testing strategy for software verification.

2. Functional Requirements

The detailed system and software requirements for the **Flap Control System (FCS)** are documented in the accompanying Excel file
`JSBSim_Flap_Requirements.xlsx`.
This spreadsheet defines each requirement (IDs FLA-001 through FLA-008.3) following the project's standard format:

- **Requirement ID** — A unique identifier (e.g., FLA-001 to FLA-008.3)
- **Urgency** — Level of implementation priority (High, Medium, Low)
- **Description** — Defines expected behavior of each block (switch, gain, kinematic)
- **Implementation (package.class.method)** — Maps each requirement to the specific Java package, class, and method planned for development
- **Verification** — Describes how each requirement will be tested and validated
- **Priority** — "Must", "Should", or "Nice-to-have" ranking based on system impact
- **Notes** — References to XML mappings or related requirements

3. Static Model and Design Overview

3.1 Package Summary

- **jsbsim.fcs** — Flight control subsystem (FlapController, PureGain, KinematicBlock)
- **jsbsim.model** — Data models such as KinematicSetting
- **jsbsim.xml** — XML parsing and schema mapping utilities
- **jsbsim.sim** — Environment or simulation state providers

3.2 Key Classes

FlapController

- positionRad: double

+ setDefaultPosition()

+ updatePositionForVelocity(velocityKnots, mach)

+ getPositionRad()

PureGain

- gain: double

+ setGain(double)

+ getGain()

+ apply(double)

KinematicBlock

- settings: List<KinematicSetting>

  + configureSettings(List<KinematicSetting>)

  + update(double inputNorm, double dt)

  + getCurrentPosition()

KinematicSetting

  - position: double

  - time: double

4. Java Skeleton Summary

```
// File: jsbsim/fcs/FlapController.java

package jsbsim.fcs;


public class FlapController {

    private double positionRad = 0.0;


    public FlapController() { setDefaultPosition(); }


    public void setDefaultPosition() {
        this.positionRad = 0.0;

    }


    public void updatePositionForVelocity(double velocityKnots, double mach) {

        // TODO: Implement switch logic:

        // if velocity < 250 → 0.349

        // else if mach > 0.9 → -0.349

        // else → default (0.0)

    }


    public double getPositionRad() { return positionRad; }

}
```

5. GitHub Project Board Plan

Issue	Summary	Type	Priority
REQ-FLA-001	Implement FlapController skeleton + unit tests	Task	Must

REQ-FLA-002–003	Implement velocity/mach switch logic	Feature	Must
REQ-FLA-004–006	Implement PureGain block + tests	Task	Should
REQ-FLA-007–008	Implement KinematicBlock and XML mapping	Feature	Must
XML-MAPPING	Parse and map XML elements to objects	Task	Must
INTEGRATION	Integrate FCS components end-to-end	Integration	Must

6. Verification & Testing Plan

Testing will be implemented with **JUnit 5**.
Proposed tests include:

- **FlapControllerTest** – verifies default position, velocity < 250, Mach > 0.9
- **PureGainTest** – verifies gain=2.864789 yields correct normalized output
- **KinematicBlockTest** – verifies configuration and internal settings
- **IntegrationTest** – wires all modules and checks end-to-end flap motion output