Design Rules and Aircraft Hydraulic System

Pedro Alonso Condessa and Murillo Stein

November 27, 2023

1 Design Rules Overview

Design rules, in the context of engineering and system design, refer to a set of guidelines or principles that help ensure the functionality, safety, and performance of a system. These rules serve as a foundation for making informed design decisions and are essential in the development of complex systems.

Design rules can cover various aspects, including material selection, component placement, system architecture, and safety measures. They are a valuable tool for achieving reliable and efficient designs in fields such as aerospace engineering, where the integrity of systems is critical.

2 Aircraft Hydraulic System

The hydraulic system in aircraft plays a vital role in controlling various functions and mechanisms. This system utilizes hydraulic fluid, pumps, valves, actuators, and reservoirs to transmit power and control aircraft components. Some key components of the aircraft hydraulic system include:

- **Hydraulic Pumps:** These devices provide the necessary hydraulic pressure by pressurizing the fluid, allowing it to perform work.
- **Proportional Valves:** These valves regulate the flow of hydraulic fluid to control specific aircraft functions.
- **Hydraulic Reservoirs:** Reservoirs store hydraulic fluid and ensure a consistent supply for the system.
- **Cylinder Hydraulic Actuators:** These actuators convert hydraulic pressure into mechanical work, controlling aircraft components.
- **Hydraulic Filters:** Filters remove contaminants and maintain fluid purity.
- **Pressure Relief Valves:** These valves prevent excessive pressure buildup within the hydraulic system.
- **Aircraft Components:** Aircraft functions such as ailerons and rudders are controlled by the hydraulic system to achieve flight control and stability.

3 Python Code for UML Diagrams

Below is Python code that generates a UML component diagram for an aircraft hydraulic system with a customizable number of functions and circuits. This code utilizes the ‘pyplantuml‘ library to create PlantUML code and render the diagram.

```
import pyplantuml as plantuml

def generate_hydraulic_diagram(functions, circuits):
    plantuml_text = "@startuml\n"
    plantuml_text += "!define HydraulicCircuit class : Hydraulic Circuit\n"
    plantuml_text += "!define HydraulicPump class : Hydraulic Pump\n"
    plantuml_text += "!define ProportionalValve class : Proportional Valve\n"
    plantuml_text += "!define HydraulicReservoir class : Hydraulic Reservoir\n"
    plantuml_text += "!define CylinderHydraulic class : CylinderHydraulic\n"
    plantuml_text += "!define HydraulicFilter class : Hydraulic Filter\n"
    plantuml_text += "!define PressureReliefValve class : Pressure Relief Valve\n"
    plantuml_text += "!define Aircraft class : Aircraft\n"

    for i, circuit in enumerate(circuits, start=1):
        plantuml_text += f"package \"Circuit {i}\" {{\n"
        plantuml_text += f"  [<<component>>HydraulicCircuit] as circuit{i}\n"
        plantuml_text += f"  [<<component>>HydraulicReservoir] as reservoir{i}\n"
        plantuml_text += f"  [<<component>>HydraulicPump] as pump{i}\n"
        for j, function in enumerate(functions, start=1):
            plantuml_text += f"  [<<component>>ProportionalValve] as valve{i}_{j}\n"
            plantuml_text += f"  [<<component>>CylinderHydraulic] as actuator{i}_{j}\n"
            plantuml_text += f"  [<<component>>HydraulicFilter] as filter{i}_{j}\n"
            plantuml_text += f"  [<<component>>PressureReliefValve] as relief{i}_{j}\n"
        plantuml_text += "  [<<component>>Aircraft] as aircraft\n"
        plantuml_text += "}\n"

    for i, circuit in enumerate(circuits, start=1):
        plantuml_text += f"circuit{i} --> reservoir{i}\n"
        for j, function in enumerate(functions, start=1):
            plantuml_text += f"reservoir{i} --> pump{i}\n"
            plantuml_text += f"pump{i} --> valve{i}_{j}\n"
            plantuml_text += f"valve{i}_{j} --> filter{i}_{j}\n"
            plantuml_text += f"valve{i}_{j} --> relief{i}_{j}\n"
            plantuml_text += f"valve{i}_{j} --> actuator{i}_{j}\n"
        plantuml_text += f"actuator{i}_{j} --> aircraft\n"

    plantuml_text += "@enduml"

    return plantuml_text

def main():
    functions = ["Function1", "Function2"]  # Change these to your function names
    circuits = range(1, 3)  # Change this to the number of circuits you need

    diagram_text = generate_hydraulic_diagram(functions, circuits)

    with plantuml.api.PlantUml() as plantuml_api:
        plantuml_api.processes_text(diagram_text)
        diagram_url = plantuml_api.get_url()
        print("Generated UML diagram URL:", diagram_url)

if __name__ == "__main__":
    main()
```

4 UML Component Diagram for Aircraft Hydraulic System

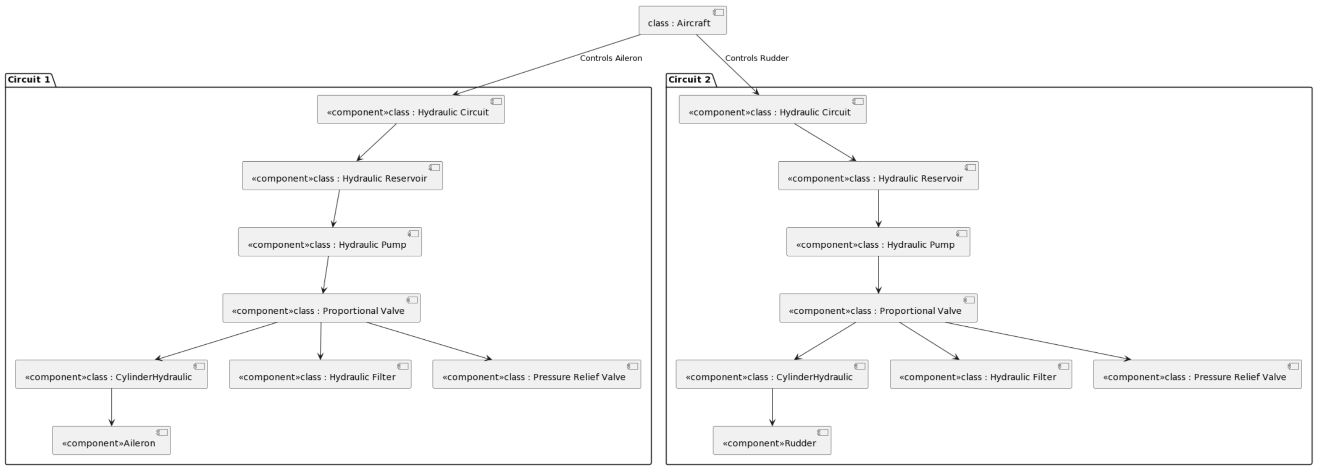
class : Aircraft
Controls Aileron
Controls Rudder
Circuit 1
«component»class : Hydraulic Circuit
«component»class : Hydraulic Reservoir
«component»class : Hydraulic Pump
«component»class : Proportional Valve
«component»class : CylinderHydraulic
«component»class : Hydraulic Filter
«component»class : Pressure Relief Valve
«component»Aileron
Circuit 2
«component»class : Hydraulic Circuit
«component»class : Hydraulic Reservoir
«component»class : Hydraulic Pump
«component»class : Proportional Valve
«component»class : CylinderHydraulic
«component»class : Hydraulic Filter
«component»class : Pressure Relief Valve
«component»Rudder

Figure 1: UML Component Diagram for the Aircraft Hydraulic System