Knowledge-Based Engineering Application For Fuselage Integration And Cabin Design

Master Thesis Report

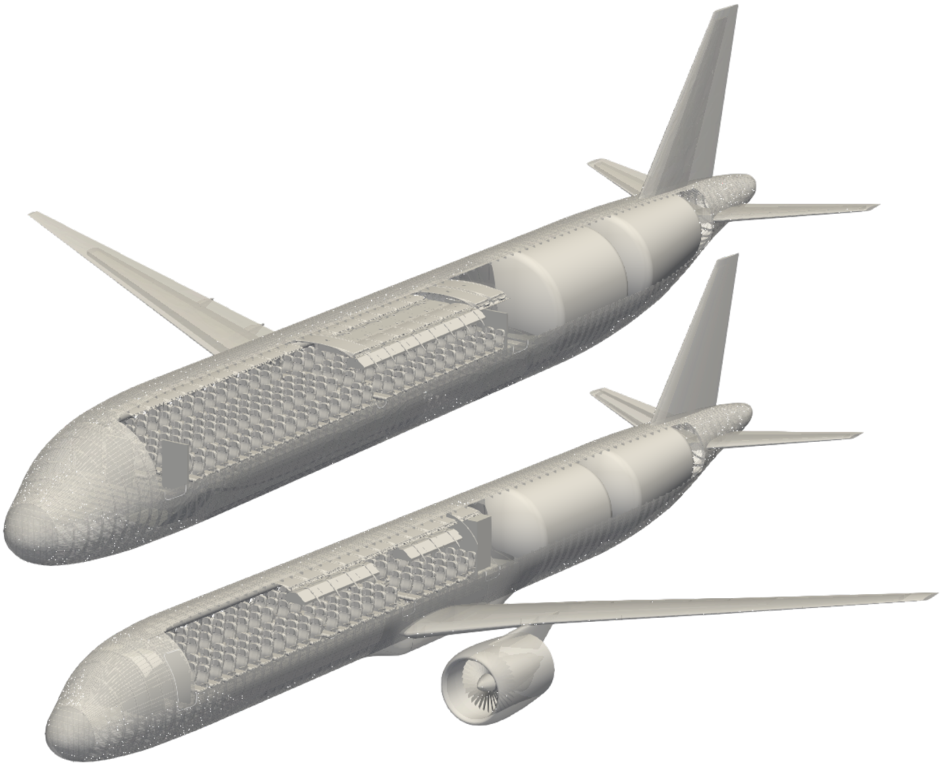
Nikhil Bhargav

Vasanth Elangovan

University (LiU) Supervisor: Raghu Chaitanya Munjulury

University (LiU) Examiner: Christopher Jouannet

Organization (DLR) Supervisor: Christian Hesse



DLR Deutsches Zentrum für Luft- und Raumfahrt
German Aerospace Center

LINKÖPINGS UNIVERSITET

Linköpings universitet
Institutionen för ekonomisk och industriell utveckling
Examensarbete 2023|LIU-IEI-TEK-A–23/04739–SE

Linköpings universitet
Institutionen för ekonomisk och industriell utveckling
Ämnesområdet Fluida och Mekatroniska System
Examensarbete 2023|LIU-IEI-TEK-A–23/04739–SE

Knowledge-Based Engineering Application For Fuselage Integration And Cabin Design

Master Thesis Report

Nikhil Bhargav

Vasanth Elangovan

University (LiU) Supervisor: Raghu Chaitanya Munjulury

University (LiU) Examiner: Christopher Jouannet

Organization (DLR) Supervisor: Christian Hesse

DLR Deutsches Zentrum für Luft- und Raumfahrt
German Aerospace Center

LINKÖPINGS UNIVERSITET

Linköping universitet
SE-581 83 Linköping, Sverige
013-28 10 00, www.liu.se

Abstract

The pace of development in aviation technology is increasing, and there is a constant need for new concepts to keep up. An innovative concept is desired to reach the net-zero emission and sustainability target visualized in Flight path 2050 [1]. Introducing digital models and virtualization into aviation fields reduces time consumption on manual modelling and increases design accuracy. Digital mock-up models also help in minimizing costs due to errors in the later stage of development or manufacturing. The Institute of Systems Architecture in Aeronautics at German Aerospace Center (DLR) works in digitizing cabin design environments with extensive implementation of the Knowledge-Based Engineering (KBE) approach. The virtual cabin design system tool also known as Fuselage Geometry Assembler (FUGA) provides a digital model of the cabin of both single and twin aisle configurations of commercial aircraft. The information of aircraft characteristics is provided to FUGA using Common Parametric Aircraft Configuration Schema (CPACS). CPACS coupled with FUGA provides the user with a consistent model of aircraft and cabin design, when viewed through a virtual platform provides an immersive experience to be inside an aircraft cabin before physical production. The multidisciplinary capability of FUGA provides experts from different disciplines to perform analysis such as vibration analysis on the cabin environment. For ease of usage and better visualization of information from FUGA, a web-based application through Flask is hosted for FUGA. This enables the user to access the FUGA tool without the need of installing the tool on their devices. With the world now moving towards a greener approach, an alternative propulsion system may require a different fuel tank configuration. Retro-fit of liquid hydrogen fuel tank into an existing aircraft's fuselage is done using FUGA tool and aircraft performance analysis is conducted and the outcomes are studied. The enhanced and advanced model of twin-aisle configuration, now on par with single-aisle configuration is used for hydrogen tank sensitivity analysis. The comparative study of different aisle configurations retro-fitted with liquid hydrogen fuel tank is further conducted for arriving at an optimal design point for a balance in range and passenger capacity.

Acknowledgements

We would like to extend our gratitude and thanks to Dr. Christian Hesse, Dr. Raghu Chaitanya Munjulury, and Mr. Philip Satwan for providing unparalleled support both in academic and technical fields throughout the master thesis.

We would like to extend our utmost gratitude to Dr. Bjorn Nagel for providing us the opportunity to do our master's thesis in DLR and conducting us throughout the thesis.

We would like to thank DLR for providing us with the platform and the environment for proceeding with our master's thesis and enabling us to achieve great results for the thesis.

We would like to thank Dr. Christopher Jouannet and Dr. Ingo Staack for being our examiners and helping us keep track of our master's thesis.

We would like to thank Linkoping University for providing an extraordinary platform in the academic and professional field to enhance and develop our academic knowledge in the specific field which enabled us to proceed ahead with our master's thesis.

We would like to thank our friends and family for being supportive and maintaining our psychological well-being throughout the focus time of our master thesis.

We would like to thank our thesis opposition Fredrik Prokic and Clas Eriksson for pushing us forward in the opposing side to point out the flaws that were present and improving our overall results of the master thesis.

Nikhil Bhargav
November 20, 2023

Vasanth Elangovan
November 20, 2023

Nomenclature

Abbreviations and Acronyms

Abbreviation	Meaning
2D	2-Dimensional
3D	3-Dimensional
CAD	Computer-Aided Design
CFD	Computational Fluid Dynamics
CPACS	Common Parametric Aircraft Configuration Schema
CPU	Central Processing Unit
CSS	Cascading Style Sheets
C.O.G	Center Of Gravity
CGmin	Minimum Center of Gravity
CGmax	Maximum Center of Gravity
DAG	Directed Acyclic Graph
DLR	Deutsches Zentrum für Luft- und Raumfahrtr
DEE	Design and Engineering Engine
ETOPS	Extended Twin Engine Operations
FPG	Full Problem Graph
FUGA	Fuselage Geometry Assembler
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
IPDE	Internet based Product Data Environment
IRPD	Internet based Rapid Product Development
JSON	JavaScript Object Notation
KBE	Knowledge Based Engineering
KOMPRESSA	Knowledge-Oriented Methodology for the Planning and Rapid Engineering of Small-Scale Application
KNOMAD	Knowledge Nurture for Optimal Multidisciplinary Analysis and Design
LiU	Linköping University
LH2	Liquid Hydrogen
L/D	Lift/Drag
MAC	Mean Aerodynamic Chord
MCG	Maximal Connectivity Graph
MDO	Multi-Disciplinary Design Optimization
MMG	Multi-Model Generator
MOKA	Methodology and software tool Oriented to Knowledge-Based Engineering Application
mTOW	Maximum Take-Off Weight
OCC	Open CASCADE
OCCT	Open CASCADE Technology
OKP	One of a Kind Production
OML	Outer Mold Line

Abbreviation	**Meaning**
OEM	Operational Empty Weight
Pax	Passenger
PLC	Programmable Logic Control
PSG	Problem Solution Graph
RPD	Rapid Product Development
SAF	Sustainable Aviation fuel
SME	Small to Medium Scale Enterprises
SM	Static Margin
STL	Standard Tessellation Language
TSFC	Thrust Specific Fuel Consumption
uID	unique ID
URI	Uniform Resource Identifier
URL	Uniform Resource locator
VTK	Visualization Toolkit
VTM	VTK Multiblock set
VTP	Visualization Tool Protocol
WBA	Web-Based Application
WP1	Work-Package 1
WP2	Work-Package 2
WP3	Work-Package 3
WWW	World Wide Web
XML	Extendable Markup Language

List of Figures

List of Tables

Contents

1 Introduction

In the 21^{st} century, the development of aviation technology is increasing at a rapid pace [8]. With the introduction of new concepts being quicker than in the previous century, there is a visible shift of conceptual designing from pen and paper to computer and cloud. This transition, thus increases the accuracy of designs and reduces the lead time, environmental impact, and expenses (time and money invested in designing models and prototypes). The budget and manpower required to design an aircraft, or the system that the aircraft depends on, without the help of computers and simulations, is high and the process is time-consuming. The path of aviation trends towards flight-path 2050 [1] focuses on moving towards a greener aviation industry and contribution put by different platforms and companies for research and development in reducing emissions and increasing propulsion efficiency. Net-zero emission motivates the need for digital technology in the development phase of aircraft conceptual design.

The technological advancements achieved in the 21^{st} century, enable us to develop an aircraft, or its system in a virtual environment and conduct the necessary tests. The digital model or system can also be modified as desired, without the need to spend excess time and money to reconstruct a flawed design from scratch. With the collaboration of engineering and years of experience in the field of aviation, a digital application can be configured to design an aircraft or its system with the knowledge acquired by expert engineers. This knowledge-based engineering (KBE) [9] approach plays a vital role in reducing the computational time for designing and providing the designer more time for creative tasks.

The Institute of System Architecture in Aeronautics inside the German Aerospace Center (DLR) focuses on digitizing aircraft system architecture to increase the interdisciplinary compatibility of one design architecture to another. This enables the designer of one field to access information and data from a different architecture. One core project that unifies the characteristics and parameters of a whole aircraft is CPACS (Common Parametric Aircraft Configuration Schema). It is a data definition for the air transportation system. CPACS enables engineers of multiple fields/disciplines to exchange information about the same aircraft between their tools. It is, therefore, a driver for multi-disciplinary and multi-fidelity design in distributed environments [10].

With the external details of the aircraft concentrated and explained until now, the internal cabin design and system configuration are of high importance for both passenger comfort and financial profit. Hence, a cabin design tool is desirable to have a digital mock-up of an actual cabin layout. A good and accurate visualization of a cabin could be possible, without the need to manufacture and produce it physically. FUGA (Fuselage Geometry Assembler) is developed internally in DLR and requires the input of CPACS to design the exterior of the aircraft and certain interior parameters for the cabin layout. The overall high-fidelity layout of the cabin interior design system provides the user control over where the cabin components could be positioned and how they would look at a certain orientation. The process of developing such high-fidelity models and it's supporting applications requires sufficient knowledge in the field of programming and coding. To ease the operation of FUGA and provide user-friendly features, an introduction of a Web-based application yields to be a great start. The Web-based application lets the calculations and programs run in the back-end while the user only sees the information that the program requires as inputs (CPACS files, .json files, etc) and the information that the user wants to see as outputs (2D Cabin layout, 3D Cabin layout, etc). This would also reduce the time required for the user to study and understand the working of the application or tool.

With a greener approach towards aviation being a motive for the upcoming years, liquid hydrogen has been a major game-changer for the aviation industry [11]. The integration of a liquid hydrogen tank into existing fuselages of single and twin-aisle configurations would aid in the transition from conventional power to hydrogen power. The completed twin-aisle configuration and single-aisle configuration in FUGA are modified to introduce liquid hydrogen tanks into the fuselage and studies related to the performance of the configuration, payload capacity, range, and center of gravity, and additional studies are conducted for each iterative design for single and twin aisles. This analysis provides further understanding of including liquid hydrogen tanks in existing aircraft models and also trade-offs between different variants are analyzed in the study.

1.1 Aim and Deliverables

In the scope of the Master's Thesis, a knowledge-based engineering approach is utilized in aircraft fuselage integration and cabin design. The aisle configurations (Single and twin-aisle) in FUGA are to be unified with additional refinement of cabin components with the help of knowledge-based engineering approaches. This yields more adapted alignment and accurate positioning of cabin systems and components in a multi-disciplinary design environment.

The topics for the master thesis are categorized into the development of Web Based Application for FUGA, Integration of single aisle and twin aisle configuration into a single package, and model generation & refinement. The work package inside these categories is explained in further sections.

The following tasks are planned to be completed at the end of the thesis:
(Note: Work package refers to the task to be completed)

1.1.1 Web Based Application

- Work-package 1: Investigation and Implementation of Python frameworks for the integration of web-based applications.

1.1.2 Model Generation and Refinement

- Work-package 2:
  - Advancement of twin-aisle configuration with the capabilities of knowledge-based engineering approach.

1.1.3 Study and Analysis

- Work-package 3: Study of Liquid hydrogen tank in single and twin-aisle configuration. Passenger and range sensitivity analysis with variation in tank quantity and size.

1.2 Research Question

1. How can a web-based application effectively represent knowledge-based engineering, and to what extent can it be universally applied to KBE projects?
2. How knowledge-based engineering approach is utilized in parametric aircraft designing and cabin designing?
3. What are the benefits of a liquid hydrogen tank being retrofitted into an existing fuselage? What are the trade-offs?
4. How does the performance of aircraft vary from single and twin-aisle with liquid hydrogen tanks?

1.3 Delimitation

The scope of the project is limited to the following delimitation. The scope is determined based on provided time for project completion and the current level of complexity.

- The time consumed in the development of the web-based application is defined by the complexity required. And since a specific program is used, it is predominantly dependent on it.
- The accuracy and current effectiveness of the twin-aisle configuration are limited to the capability of the design coding language and version.
- The current knowledge of liquid hydrogen tanks for the purpose of integration into aircraft fuselage and usage is limited.

1.4 Work-package Descriptions

The thesis is divided into 3 different work-package to keep track of and establish uniform progress. A detailed description of each work package is written in this section. The scope and fields covered by the work package provide an in-depth view and knowledge about the direction of individual work packages.

1.4.1 Work-package 1: Investigation and Implementation of Python frameworks for the integration of web-based applications.

Investigation and Implementation of web-based applications will make FUGA easily accessible within the institute and other institutes within the same organization - such as other research teams for their own projects.

In addition, external partners (such as companies or organizations) can also benefit from FUGA's capabilities. This could facilitate collaborations between the institute and external partners.

In the Web-based application, the user has to input the necessary files and also other necessary inputs to get the desired outputs on the web application.

The outputs can be the 2D draft of the cabin front section or the cabin layout plan or 3D visualization platform or have some validation or comparison results (example: comparison of single-aisle and twin-aisle).

The depiction of the web-based application on the FUGA is represented in figure 1.

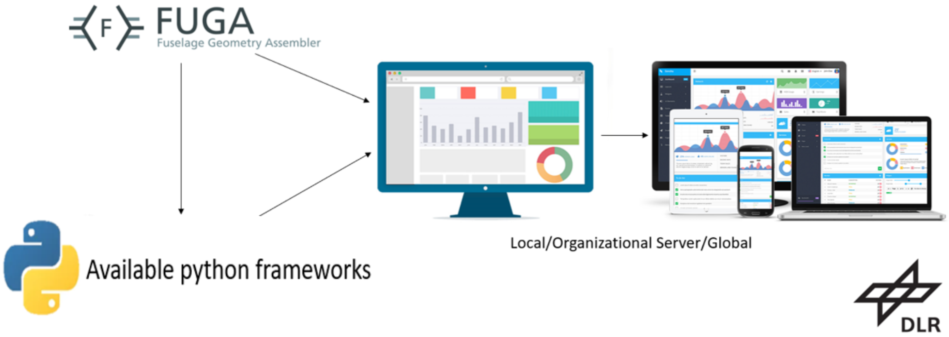

FUGA
Fuselage Geometry Assembler
Available python frameworks
Local/Organizational Server/Global
DLR

Figure 1: Depiction of Web-Based Application

1.4.2 Work-package 2: Advancement of twin-aisle configuration capability in Fuselage Geometry Assembler

FUGA currently has 2 different methods of computing and displaying single-aisle configurations and twin-aisle configurations. The twin-aisle configuration present in FUGA is currently not capable of providing scientific information and analytical possibilities as can be done with a single aisle configuration. Further refinement has been done for the twin-aisle configuration. The following inputs are required to completely utilize the FUGA tool:

- CPACS File (.xml) is required by the FUGA tool to set the cabin boundaries and model the external model of the aircraft
- Aircraft structural file (.xml) is an augmented .xml file that contains critical data for the cabin design such as bulkhead, cutouts, stringers, floors, etc. These data are crucial to set the cabin system boundaries and constraints within the aircraft fuselage.
- Cabin design file (.json) Contains the seating configuration, number of aisles, class specifications, lavatories, and position, galleys, and positions, exits, and types, etc. These inputs are the vital parameters for FUGA as it is designed around the cabin parameters.
- Twin aisle cabin design file (.json) is another file that is additionally required apart from the existing cabin design file. This file is needed only for twin-aisle configuration and not for single-aisle configuration. This overwrites the directories of cabin components needed for the twin-aisle as well as seating configurations, number of classes, class dividers, and so on. This additional information provides the cabin layout for twin-aisle configuration, but certain essential information such as exit layout and types are still retained from the previous cabin .json file (single aisle configuration file).

The work that would be undertaken in this task is enlisted below:

- Refinement of rule sets currently implemented in twin-aisle configuration and further increase of compatibility of the twin-aisle configuration with the existing single-aisle configuration.
- Refinement of the luggage compartment and ceiling panels of Twin aisle configuration.
- Introduction of Cargo liners for both aisle configurations.
- The final result of this work would be integrating with the work package 1.

1.4.3 Work-package 3: Study and Comparison of Aisle Configurations using Knowledge-Based Engineering approach with Sustainability into consideration

Liquid Hydrogen tank is a future scope of fuel for commercial aviation and DLR aims to highlight the focus on hydrogen utilization. In Work-package 3, a trade-off study and comparison between different parameters of single and twin-aisle configuration with the hydrogen tank taken into account is performed. The hydrogen tank is initially decided to be placed at the aft region of the fuselage and it may result in various changes from an existing commercial aircraft cabin model. The following points are the result of introducing a hydrogen tank into the digital aircraft concept against an existing model:

The current simulation model requires some higher fidelity model updates. This would contain the following changes:

- Comparison of Cabin layout with and without the hydrogen tank.
- Performance analysis and difference with the introduction of the hydrogen tank.
- Comparison of tank capacity and trade-off of the passenger capacity of the same cabin model.
- Shift and difference of aircraft C.G with the introduction of the tank.

2 Literature Review

The information and knowledge that is studied and gathered for the thesis are written in detail in this section. The literature data that helps in better understanding the thesis and parts where attention is required is further briefed in detail in this section.

2.1 Harnessing the Knowledge-Based Technique in Web-Based Application

In the field of KBE where the automation takes place to get the desired product and results, a choice of appropriate platform for interaction and display of the results for the user plays a vital role. With increasing demand in collaboration and reachability for the Tool or product, a web application seems to be the solution.

A knowledge-based engineering approach plays an important role in the initial stages of aircraft design (conceptual design phase), the work carried out by Munjulury, R.C. [12] on the knowledge-based aircraft conceptual design applications using a data-centric approach by using XML for parametric data definition, allowing 3D CAD integration. There were 3 modules integrated with the XML as the centre database which are: Sizing tool (BeX) with CAD module(RAPID and CATIA), an estimation and analysis module for aerodynamics and other estimation and integration of the system architecture for analysis of the system to system interaction. The work also describes the use of a multidisciplinary design optimization platform, modeFRONTIER for evaluating different designs. And for the interaction between different applications, a multifaceted user interface was developed.

Reddy et al. [13], imply the role of a Web application for a KBE system which enhances the interaction with the user where the customer's needs are effectively taken into consideration than the traditional design. Product visualization in [13] includes 2D drafting and 3D models with an interactive window for the user to pick the requirement for better understanding, also with an enriched experience with audio, video, text, and animation using HTML technology.

Effective inter and intra-communication between the companies is dealt by Tu Y. and Xie S. in [14], which makes the tool more effective, that is by having a WWW platform (a holistic point of view). This helps to have quicker response to the customer requirement with saved production costs and better globalization.

The work carried out by Xie S. and Tu Y. in [2], which deals with One Kind Production (OKP) with Rapid Product Development (RPD) using an Internet-based approach which gives a global customer interface as shown in figure 2. figure 2 depicts the Internet-based RPD which provides an Internet-based product development (PD), with virtual-based process planning and cost optimizing model. Thereby bridging the gap between the customers plays a vital role which is done through

Graphical User Interfaces (GUI's) [2].

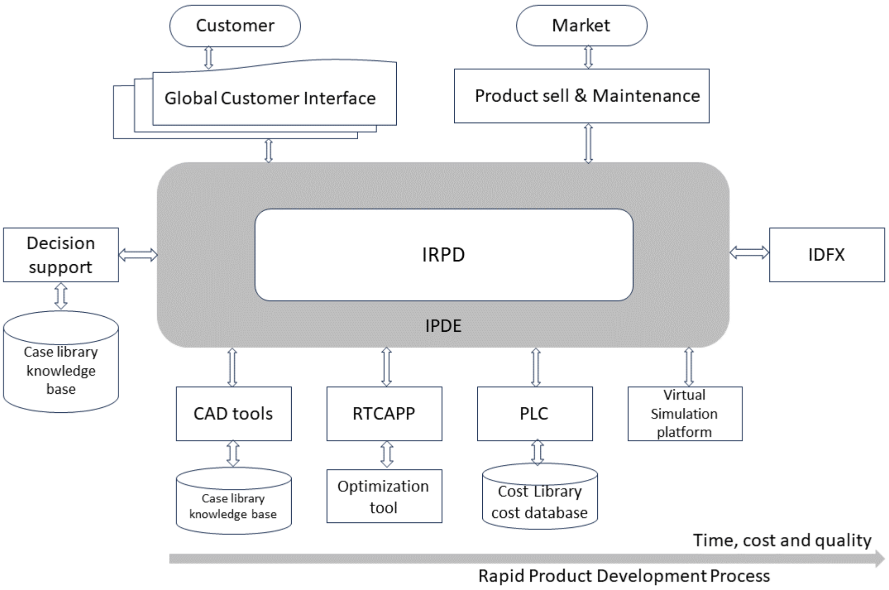

Customer
Market
Global Customer Interface
Product sell & Maintenance
Decision support
IRPD
IPDE
IDFX
Case library knowledge base
CAD tools
RTCAPP
PLC
Virtual Simulation platform
Case library knowledge base
Optimization tool
Cost Library cost database
Time, cost and quality
Rapid Product Development Process

Figure 2: Structure of the system, adapted from Xie S. and Tu Y. [2]

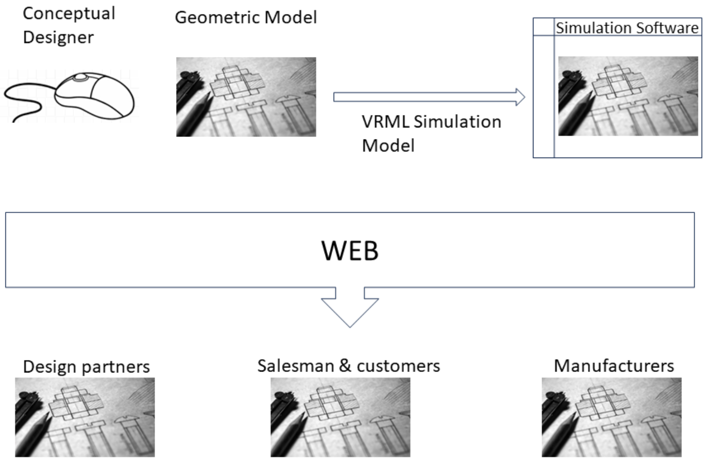


Figure 3: Adapted Example showcased by Qin SF. [3] for a potential application scenario.

Also, another research from Xie S. [15], describes the information reach and information range done through web or internet technologies to share the information between companies by introducing a virtual manufacturing concept to meet the changing customer needs, in which customers have the constant connection between product and manufacturing information, which basically covers the entire product life cycle.

A Web app was developed by Qin SF. group [3] as seen in figure 3, which connects the design partners, sales, customers, and manufacturers. Also in which the design through the internet is quickly simulated for the product behavior and results are displayed.

2.2 Study of wide-body aircraft

The Demand for wide-body aircraft in the market steadily rose with the increase in air traffic and passenger population over the years[16]. This raise for wide-body aircraft may be due to the need for transporting a large number of passengers in a single flight and increased comfort as well since wide-body cabins are large and spacious. Aircraft with more than a single aisle are called wide-body aircraft. These aircraft are capable of transporting more passengers for the equivalent length and also increase the cabin space. These aircraft are designed to cover medium to long-haul flights. Since wide-body aircraft are designed to cover long distances and are relatively large in size, they require larger gates in airports and consume more fuel for flights.

In the early times of aviation, aircraft with wide-body configurations normally had more than 2 engines for redundancy and safety to fly over large water bodies. With the advancement in the aeronautical sector over the period of time, along with it, the development of turbine engines. Now the ETOPS [17] rating allows twin engines aircraft to fly over large water bodies. With the hub and spoke flight path being converted to a point for passengers to reach their destination with low layovers and connections, wide-body aircraft with 2-engine configurations (eg: Boeing 777, 787, A330, A350, etc) are chosen over traditional 4-engine wide-body aircraft(eg: Boeing 747, A380, etc).

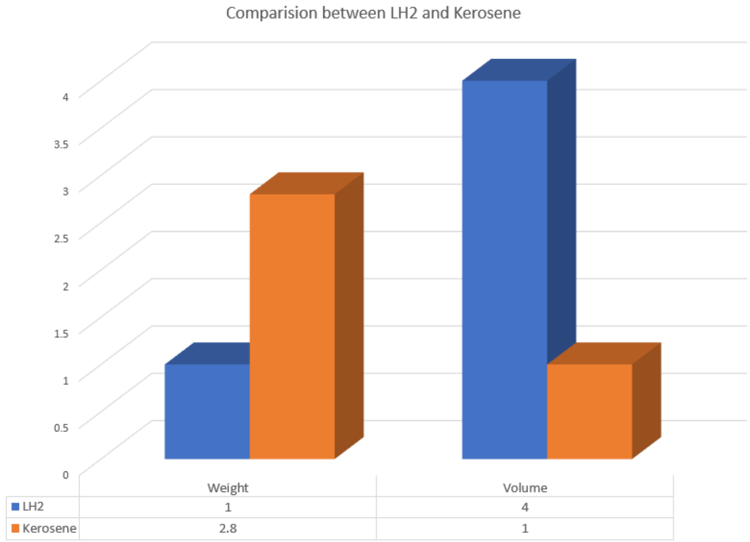
2.3 Liquid Hydrogen Tank in Aviation Industry

The aviation industry, as previously explained in Introduction section 1, is a fast-growing industry that is predicted to have an increase in air transport demand by 4.3% in the next 20 years [1]. With the increase in air transport demand, and the current aviation industry consuming 2% of the global greenhouse gas mission, It also contributes to 8% of total world economic activities in GDP. Aviation industries are hard at work transitioning from using fossil fuel to power gas turbine engines to using Sustainable Aviation fuel (SAF) and hydrogen fuel due to the raising CO2 global emission projected at 61 % every 5 years[18]. The transition requires intensive research, development, testing, certification, and multiple processes within the industrial standard to bring to the public and commercial flight. To aim at utilizing green emissions by around the year 2050, it is important to progress and start research and development towards green energy in the current aviation industry [18].

The conversion from traditional aviation fuel to hydrogen fuel requires significant changes in the storage of fuel for the aircraft. The traditional way of storing fuel inside the wings is not feasible as certain design requirements are to be met for proper and safe functioning and usage of liquid hydrogen tanks [19]. Hydrogen has 3 times more heat content for each mass than that of kerosene. This will therefore affect adversely the range of the aircraft. And since positioning it in the wings would mean that the tank would experience sharp edges, this would not be ideal for a pressurized structure (Round smooth curved edges are desired). The hydrogen tank comes in 2 different configurations. Integral and Non Integral tanks. Integral tanks come equipped with the aircraft fuselage and hence, a larger volume is available for the occupied area of the tank. Since this configuration is built along with the Fuselage, retrofit of such tanks is not necessary as it comes built with the fuselage, this requires extensive modification to the existing aircraft model, but in turn, a lower fuselage size and less drag are achieved. On the other hand, non-integral tanks carry volumes of fuel inside an insulated tank which can be placed inside an existing aircraft's fuselage. This does not require extensive remodeling of the external geometry of the aircraft, but larger fuselage weight and more drag is a drawback.

The study focuses on introducing a hydrogen fuel tank into an existing aircraft model so that not much time, effort, and extensive redesigning is necessary for the swift introduction of liquid hydrogen into the aircraft. This means the suitable configuration of the non-integral tank would be considered for this study. Non-integral tanks require insulation to either the interior or exterior of the tank. Since the tank should be able to sustain and maintain very low temperatures throughout the life cycle [19], the insulating material needs to handle extreme conditions, additionally, the tank must be accessible internally for repairs and maintenance.

W Xu, Q Li, and M Huang [20], discuss the practical implementation and design of cryogenic hydrogen storage tanks for unmanned aircraft. When you consider kerosene and hydrogen of the same weight, the hydrogen has 2.8 times more energy which can be used for longer cruising time. However, when you take into consideration of the volume of liquid hydrogen it takes 4 times more than kerosene or any conventional aviation fuel, the illustration of weight and volume is shown in figure 4, which requires a durable and insulated tank which plays a major role. They take into consideration of spherical tank instead of a cylindrical tank due to the effect of sloshing which is maximum in cylinder tanks and the introduction of slosh plates will lead to a higher weight penalty. Additionally, the boil rate in the liquid hydrogen tank must be taken into factor as it shows that the point contact support structure can reduce heat leakages by 85% but mechanical properties must also be accounted for, as the stress concentration and structural integrity.



	Weight	Volume
🟦 LH2	1	4
🟧 Kerosene	2.8	1

Figure 4: Ratio of weights and volumes for LH2 and Kerosene

C Winnefeld et al. [21], describe the design and modeling of cryogenic tanks for liquid hydrogen for future aircraft applications, initially studies on mission parameters have to be done before designing the tank which can include the mission of the aircraft, the altitude at which it is flying, which in turn decides the heat input to the tank, insulation of the tank and according to this the boil-off rate inside the tank is determined leading to the venting situation and maximum allowable pressure. The next step is deciding the tank structure, the available options are: integral and non-integral, and both have their own advantages and disadvantages. Integral type is the ones that are structurally integrated with the airframe and are subjected to the same loading as the airframes and the non-integrals, as the name is self-explanatory, which does not merge with the structure, can be either inside the fuselage or external like a non-integral pod tank. Thermal insulation of the tanks plays a vital role as they balance the pressure inside the tank keeping the boil-off rate at the nominal rate, this is decided by the structural design, insulation material, and thermal modeling, thereby the design of the tank is an iterative process between geometric, mechanical and thermal design.

D Verstraete [22], describes the potential of hydrogen usage for transport aircraft making it a long way for sustainability in aviation. The semi-retrofit design is done on a narrow-body and wide-body configuration instead of extending the aircraft fuselage completely based on the tank volume needed the cabin was modified (reduced) and decreasing the fuselage length. Integral tank configuration was taken into consideration were 2 designs opted, one being 2 integral tanks one at the front and one at the back, this leading to the separation of the cockpit and cabin, there by a second design was considered were so-called top tank where the tank was on the top and in the aft region allowing the cockpit and cabin to be integrated, but this also led to a drastic increase in the tank weight. Using the top tank led to 28.1% heavier than the integral tanks for short range and a 50% increase in medium-range aircraft. The observation on the energy efficiency on the long-range mission is around 12%, on the other hand for short-range mission it seemed to be an energy penalty of 18%

which is due to the weight of the tank.

The FUGA tool which will be used extensively in the thesis study also incorporates the ability to perform analysis and study with a liquid hydrogen tank being a part of the cabin design system. [4]. Though the ability of FUGA limits the position of the tank to the rear region of the fuselage, this will help in studying the performance change in the aircraft's payload, range, and center of gravity variation with different tank configurations and types of aircraft. Preliminary design variations such as different tank configurations and their findings are presented [4], an extensive study of further improvement of such tank configurations for aircraft of different aisle configurations and payload criteria can be carried out with the existing tool thanks to the parametric ability and knowledge-based engineering approach for arriving at feasible result based on provided data.

3 Framework

After having a brief overview of topics and sections about the thesis body, the framework upon which the thesis is worked is explained in brief in this section. The crucial information on which the thesis is based is explained in this section and the framework is the environment in which the studies are conducted.

3.1 Fuselage Geometry Assembler

Fuselage Geometry Assembler (FUGA) is a tool that provides preliminary cabin design, structural design, cargo hold designs, and aircraft design closely interlinked into a single frontier. The main design characteristics of the aircraft such as fuselage profile, wing box location, wing properties, etc are derived from CPACS.

The cabin characters and criteria such as floor height, seating arrangement, seat pitch, and cabin component positioning are inserted into FUGA using a .json document and the boundaries that maintain the consistency between the outer fuselage and the inner cabin design system are provided by the .xml augmentation file. By using the knowledge-based engineering approach, various rulesets are embedded into FUGA that maintain the consistency of the aircraft model and geometric model generations.

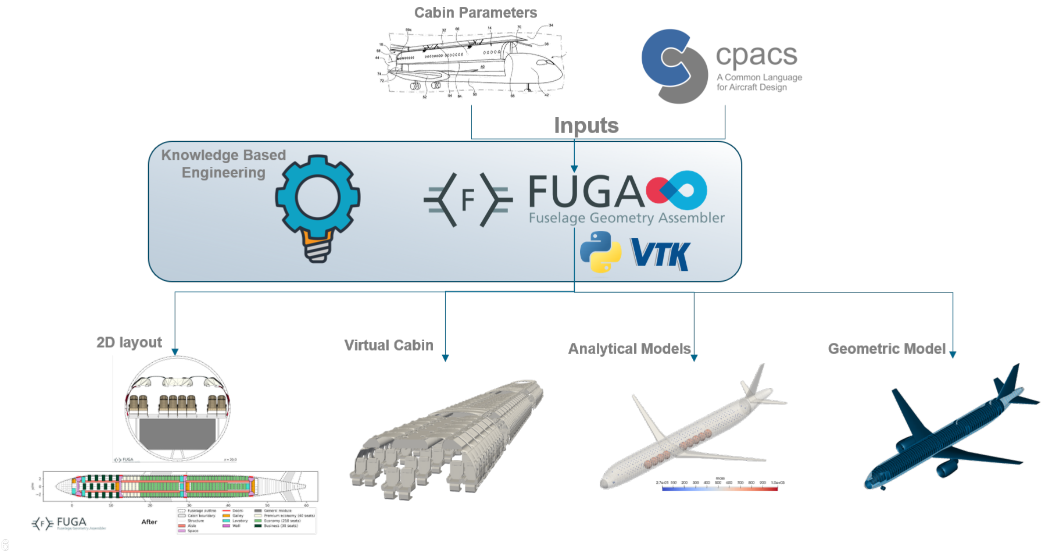


Figure 5: Overview of FUGA

The fundamental software architecture of FUGA's design framework is coded in Python and consists of 3 principal components that are taken from knowledge-based engineering methodology. The structural part of FUGA is modelled using Open Cascade Technology (OCCT) [23]. The structure and the functionality of FUGA are illustrated in figure 5.

3.1.1 Fuselage Geometry Assembler - Rulesets

The position of components that are present inside the cabin (eg: floors, seat rails, seats, side walls, etc.) needs to be co-related to one another and should not overlap. This consistency is maintained through various numbers of rulesets that are implemented in FUGA using a knowledge-based engineering approach. As illustrated in figure 6 from Walther JN, et al. [4].

The maximal connectivity graph (MCG) 6 represents the relationship between different components of the design system. At the initialization of the FUGA tool, the necessary data required for the optimum functioning of the tool is imported from the data repository, which is CPACS for FUGA. The nodes which do not have any incoming connections are the root nodes that contain the data retrieved from the data repository and the leaf nodes are the nodes that contain information/data that are required/requested.

The data requested are determined by the full problem graph (FPG) and then the KBE solution is formulated from the available and requested data. Based on the FPG, the corresponding and related rules are analysed and executed which is known as problem solution graph (PSG). This KBE approach works efficiently regardless of the active rules and as long as the processing connections can be traced back to the requested nodes.

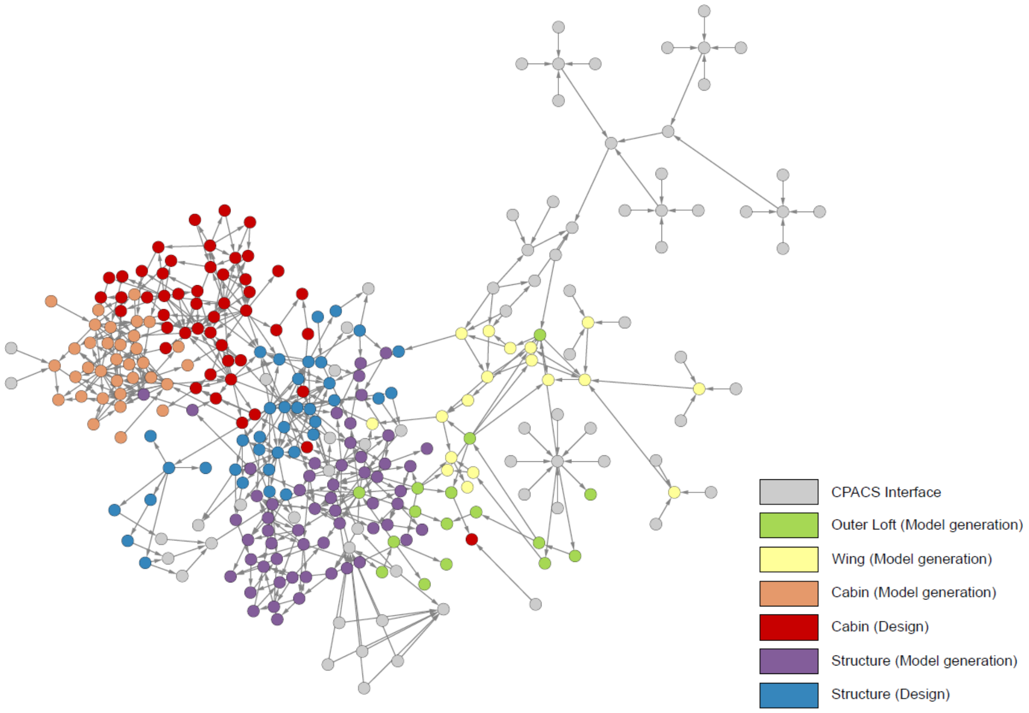


Figure 6: Maximal connectivity graph of rule sets.[4]

The rulesets that are visualized above are coded in python using the NetworkX python package as the basis. [24]. This nodal-based approach and graphical visualization help FUGA in easing the implementation of the KBE approach and providing a sound relationship between multiple design components within the design system.

The design rule sets that are used in FUGA are stored as classes in Python. Each of these classes inherits from a common base class. The functional paradigm states that each rule can only provide the value of a single entry of the data repository, which can be accessed through its more unique resource identifier (URI). The following attributes are primarily utilized in the FUGA class hierarchy:

- **provides**: URI are stored in here.
- **requires**: the corresponding URI that is requested from the data repository is displayed here. The specific class does not function if all the requirements are not present.
- **compute**: It takes in the inputs as python-dictionary which is specified in requires attribute and returns the formulated value from the specified provided attribute.

The information provided by the user's definition of the cabin configuration is stored in the data repository. The knowledge repository contains rules and rulesets that maintain the relations between multiple components such as the seat rails, hatrack position concerning flooring, etc. The rules are defined from CPACS and other CAD geometries(eg: .stl files provided for Cabin components).

The inference engine utilizes the corresponding rulesets and data provided by the user and executes the existing data to yield a model which represents the model that the user wants and checks if it is within the consistency of the limitations for an aircraft and cabin design, if not, the engine fails to execute. The reason for multiple rulesets is due to the constrain of one return value per function, resulting in multiple rulesets to provide a network of data for the provided data repository.

One of many factors that makes FUGA unique is the outside-in cabin design structure. This method fits a cabin design system based on the outer fuselage design dimensions. figure 7 provides a brief overview of the relationship between multiple components that are inter-linked in FUGA tool infrastructure. The N^2 Chart provides the dependencies between different rule sets used for different structures of the system. The diagonal highlighted gray boxes indicate the hierarchical dependency as it goes diagonally down.

The boxes that are present above the highlighted diagonal boxes indicate the dependency from the system horizontal to the box to the system to which the highlighted diagonal box lies horizontally. This indicates that one or more system is dependent on one or more other system to function at optimal level. The boxes that are below the diagonal boxes indicates the inter-dependency of one or more system downstream of the system hierarchy to one or more system upstream of the system hierarchy. This inter-dependency structure further goes into a loop to arrive at the closest accurate result possible without extensive usage of time or resources. It is

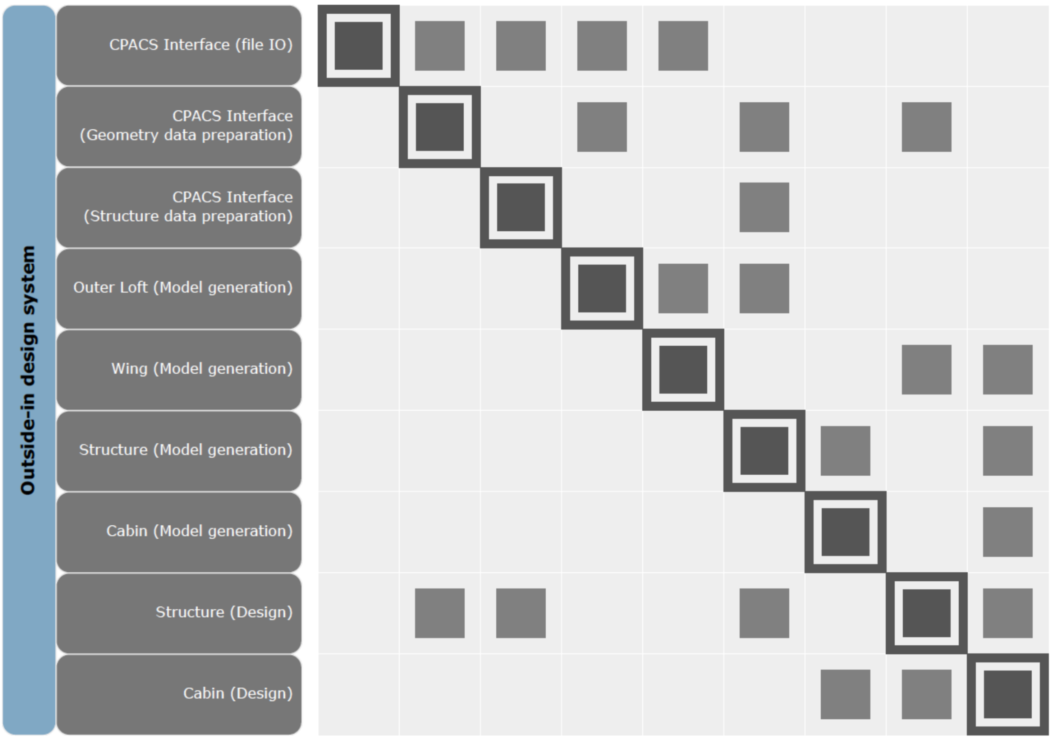

Outside-in design system
CPACS Interface (file IO)
CPACS Interface (Geometry data preparation)
CPACS Interface (Structure data preparation)
Outer Loft (Model generation)
Wing (Model generation)
Structure (Model generation)
Cabin (Model generation)
Structure (Design)
Cabin (Design)

Figure 7: N^2 chart showing rules sets for basic outside-in design [4]

also important to know that different design rules or systems can be mutually dependent as well.

3.2 Open Cascade Technology

The Open Cascade Technology (OCCT) is used as the basis for CAD generation of the cabin design system and external geometry from the FUGA tool. It is used to model the shells of stringers, frames, fuselage, wings, and lots of additional geometries that are specified in CPACS. The input for OCCT is mostly supported by an addon occhelpers in FUGA. the helpers simplify the utilization of the code-to-CAD method.

The method of doing code-to-CAD is tedious, time-consuming, and effort from the coder to bring points and lines into visualization through code. Then taking these simple shapes into an inter-relation-able CAD model requires hundreds of lines of code. To ease the efforts put in by the coder.

This method though can be tedious and time-consuming, is highly parametric and provides the user the flexibility of adapting the CAD model to rules and constraints set by the KBE approach. The conceptual method, that different analyses require respective abstractions of the geometry is known as Multi-fidelity. FUGA has the capability of Multi-Model Generators (MMG) that satisfies the user requirements.

The geometrical requirement for MMG is made possible by a feature tree derived from CAD application apart from the OCCT library. This tree links the geometrical shapes and operations of different components through boolean operations, extrude, etc. This intersection or interaction of stringers and frames creates a new geometric component as a result of its parent shapes.

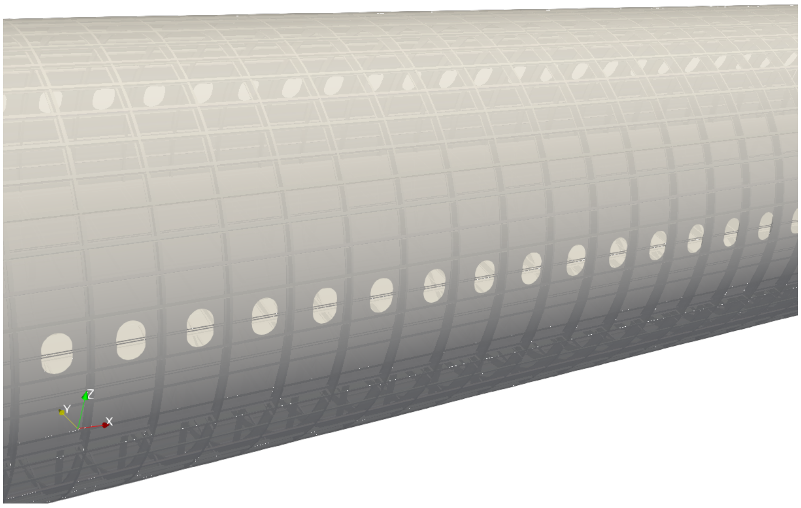


Figure 8: Fuselage structure of CPACS aircraft designed in OCCT

Visualization of how CAD is done in OCCT which is implemented in FUGA is illustrated in fig 8. The fuselage structure is stacked up on a tree configuration in CPACS along with the interaction of the extrusion plane, line, and profile. They are coupled through the ruleset used in FUGA that creates a harmonic relation between individual CAD operations.

3.3 Visualization ToolKit

Visualization ToolKit (VTK), is an open-source software package designed for the manipulation and visualization of scientific data being a versatile toolkit it offers advanced 3D Rendering tools and also capabilities in 2D plotting [25], with the power of VTK, researchers and engineers can process and showcase their data in a visually compelling manner.

One of the advantages of visualizing the generated model through VTK is the flexible data formats. VTK supports multiple data formats which include VTP (VTK Polydata) and VTM (VTK Multiblock). Multiple VTP blocks can be integrated into one VTM model, which is suitable for exchanging data, and Paraview, which is an open-source post-processing visualization engine [26] can be used to visualize the generated VTM and other geometric model definition.

3.4 Knowledge Based Engineering

The competitive world and demanding markets require means of increasing the rate of product development while reducing the time and costs that incur in the process. To achieve the demand, especially in a design domain, the technology that supports rapid and modular design is Knowledge-Based Engineering (KBE) [9].

Knowledge-based engineering is utilized to automate repetitive design procedures and arrive at the optimum design, saving time and enabling the designer to explore vast design points [27], hence, Knowledge-based engineering has been a major advantage for aerospace and automotive industries.

Knowledge-Based Engineering, defined by Ammar et al [28] ”Knowledge-based engineering has been defined as being an engineering methodology in which knowledge about the product, e.g., the techniques used to design, analyze, and manufacture a product, is stored in a special product model. The product model represents the engineering intent behind the geometric design. It can store information, attributes of the physical product such as geometry, material type, functional constraints, etc. as well as process information, the processes by which the product is analyzed, manufactured, and tested.”

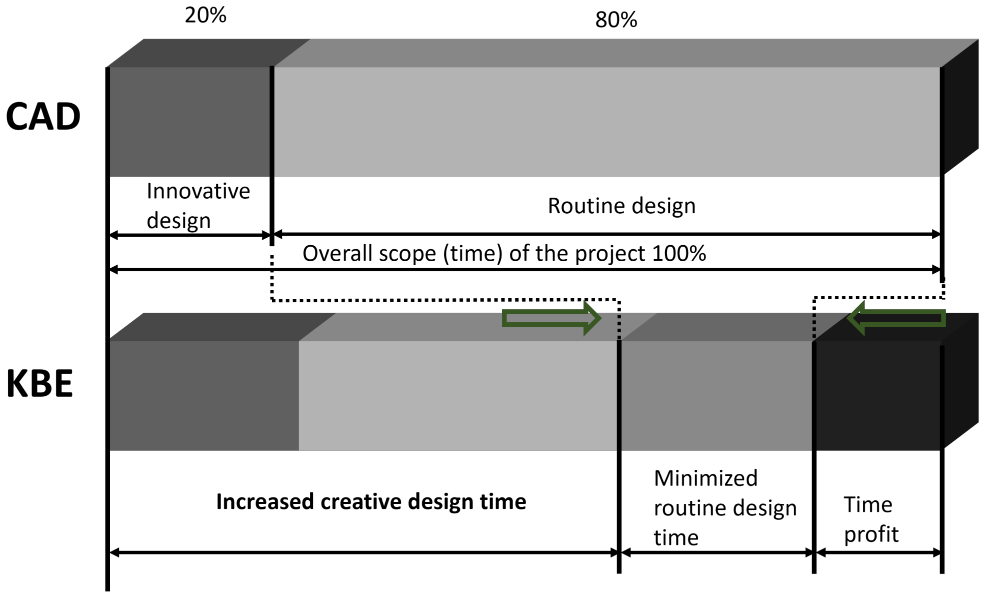

20%
80%
CAD
Innovative design
Routine design
Overall scope (time) of the project 100%
KBE
Increased creative design time
Minimized routine design time
Time profit

Figure 9: Time Profit through KBE approach diagram adapted from Skarka W [5]

Skarka W. [5] states that there is no unambiguous definition of KBE. But the noticeable advantage of KBE is illustrated in figure 9 which shows the time profit achieved by using KBE. The KBE approach is used to automate repetitive design tasks which allow the designer to focus on creative tasks.

There are many methodologies available for the development and support of KBE applications. Methodology and software tools Oriented to Knowledge-Based Engineering Application (MOKA) are one of the methodologies. It is based on the KBE life-cycle steps and is expressed in accompanying case-specific models designed to take a project towards industrialization [29].

Another methodology available for KBE is Knowledge-Oriented Methodology for the Planning and Rapid Engineering of Small-Scale Applications (KOMPRESSA). This methodology focuses on the usage of the KBE approach at small to medium-scale enterprises (SMEs) and shares similar traits with MOKA and had been developed parallel with it. The unique feature that distinguishes KOMPRESSA is an increased emphasis on risk analysis and management [30].

Existing knowledge-based engineering methodologies are at their early stage and provide room for improvement. The multi-disciplinary character of knowledge-based engineering is not well established and the current methodologies are not optimal for multi-disciplinary application. the Knowledge Nurture for Optimal Multidisciplinary Analysis and Design (KNOMAD) [31].

Knowledge-based engineering method has been used in cabin automation and aircraft parametric designing due to its flexible capabilities and potential to adapt to conditions. One such implementation of KBE is Robust Aircraft Parametric Interactive Design (RAPID). The aircraft's geometrical integrity as well as cabin dependency is strongly governed by knowledge-based engineering which yields a flexible aircraft model that is capable of generating sound designs while providing the user plethora of features to modify the model [12].

This methodology could be potentially used for the development of the FUGA tool, it closely resembles for knowledge management system approach, which is not desired for FUGA. The KBE methodology that closely incorporates KBE within an overall multidisciplinary design optimization (MDO) approach is the DEE or Design and Engineering Engine approach by Rocca et al.[32]. This methodology basis on a Multi-Model Generator (MMG) to describe a product family that is developed from common rule sets. MMG can automatically instantiate geometry and provide multiple perspectives of the inputted model. The overall analytical result is the input for the overall MDO process.

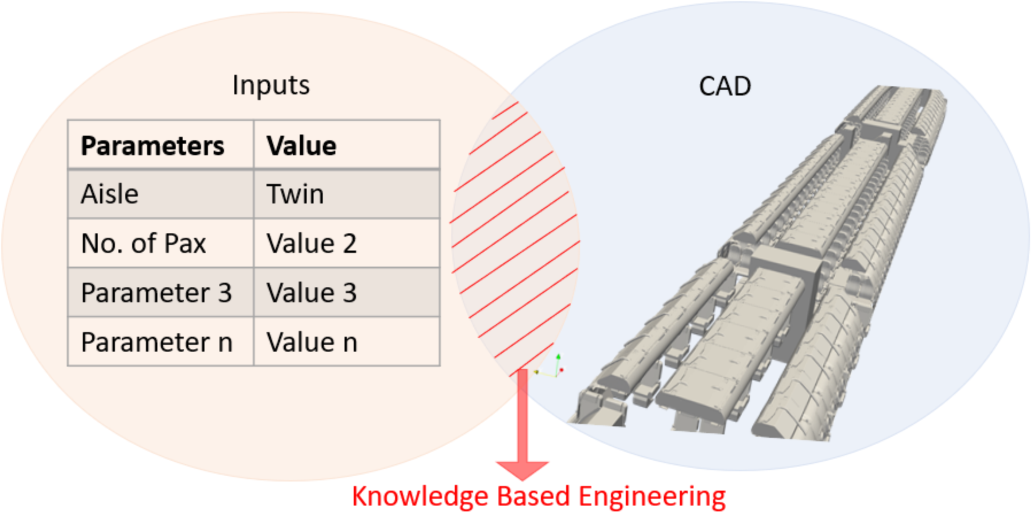
Through the merits and advantages of the Knowledge-Based Engineering approach has room for development and is a promising method, the approach also has some limitations noted by Verhagen et al.[9] and some of which are tabulated below:

- Case-based Development of KBE application:
  - The development of KBE approaches is specific for a specific case problem and it is not developed as a structural framework or methodology of application.
- A tendency towards development of 'black-box' applications:

- The current development of KBE applications is oriented toward black-block applications, representing captured knowledge as mere data and formulas without context. there is no explication of formulas and the actual meaning and context of the captured knowledge, let alone provisions for capturing design intent.

- Lack of knowledge re-use
  - As stated above, the black-box application has made it difficult for inviting knowledge to re-use [33]. High-level knowledge such as project constraint reasoning, problem resolution methods, etc. is not captured accordingly.

The parametric aspect of the FUGA tool is a great advantage for virtual visualization and analysis of cabin design for different aircraft configurations which is explained in brief in section 3.1.1. But the parametric aspect also comes with the disadvantage of lack of consistency to adapt to the changes in the architectural requirements and conditions over the period. Hence, "a knowledge-based approach is introduced to automate data initialization task and assert consistency with previously generated product data", quoted from [4].



Parameters	Value
Aisle	Twin
No. of Pax	Value 2
Parameter 3	Value 3
Parameter n	Value n

Figure 10: Depiction General KBE system

The KBE methodology of the Design and Engineering Engine (DEE) is a more comparable approach to FUGA tool, and with some modification from the existing methodology to adapt to the working purpose of the tool. It is possible to provide multiple parametric models of the aircraft and its cabin, which is not only inclined to the user criteria but is also method-oriented toward MMG. The KBE approach in FUGA consists of 3 fundamental elements of the KBE approach as illustrated by La Rocca [34], and also mentioned as "La Rocca's Pillars", as mentioned in [4]:

- A workspace or data repository of product data.

- A knowledge repository, containing design rulesets.
- An inference (or reasoning) engine, which selects the rules to generate additional data.

A rule-based KBE approach is crucial in the construction of the FUGA tool. It is best suited for a structural approach and based on product development in cases where exercises and repetition can be tolerated. With procedures that required extensive usage of sketch and blurred attributes, Knowledge-based engineering might not be a proper fit. With aircraft designing and the concrete information and data that is used in CPACS, the structural approach and flexible requirements for a cabin design system make knowledge-based engineering a very feasible approach for FUGA.

To adapt to the assumptions and rough estimations done in multiple calculations and studies, the effect of the "Rule of Thumb" is used in the knowledge-based engineering approach. The variation in geometric modeling and handling of problems for solving multiple criteria and conflicts is generally more suitable for knowledge-based engineering. An iterative process that requires multiple design evaluations and corrections is suitable for KBE to be utilized. KBE can design and analyse multiple iterations and variations of CAD or formulate problems to produce and store various optimum solutions.

3.5 Common Parametric Aircraft Configuration Schema

Common Parametric Aircraft Configuration Schema (CPACS) is a crucial integral part of the FUGA tool as most of the components and parts that are required and used in FUGA are derived from CPACS. CPACS is a widely used central data exchange format that is used to model flexible aircraft models, making it a source for creating multi-fidelity models, and has been used in the application of many projects in the German Aerospace Center [35]. It also acts as a common platform in which experts from multiple disciplines can share their opinion on a single aircraft model.

CPACS was first introduced by Liersch and Hepperle [36], in which the outer mold line (OML) of the aircraft structure is discussed in brief. CPACS is designed in XML based file format because of the hierarchical data structure. This hierarchical structure yields advantages for parametric abstracts and mapping of aircraft geometry [36]. A more detailed structure of CPACS is seen in figure 11

The aircraft structural properties such as the fuselage profiles, sections, wing profiles, engine configuration and sections, etc. are derived from CPACS for FUGA to model the outer geometry of the aircraft.

The primary purpose of CPACS was initially discussed to bridge the gap between different institutes and fields and to establish a co-relation between the professionals

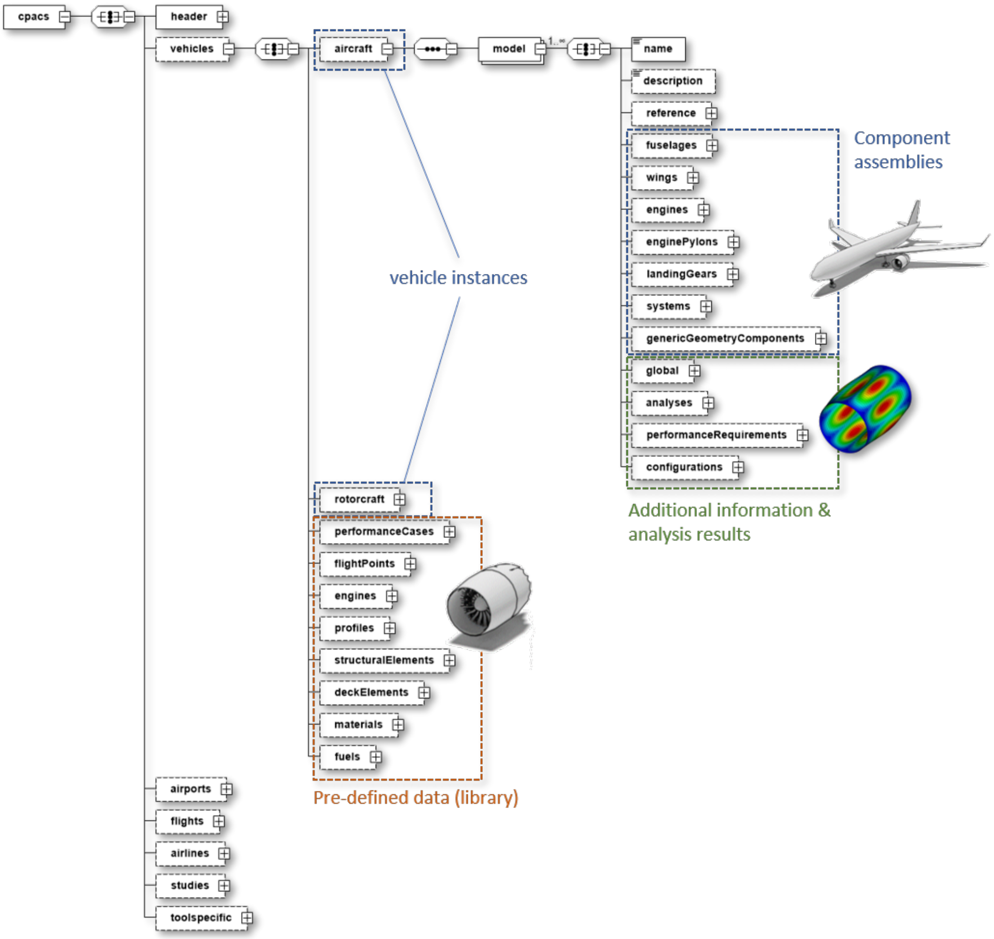


Figure 11: CPACS Data structure tree illustrated in [6]

in the respective fields [37]. This would in turn provide a whole infrastructure of an aircraft, which not only defines a structural understanding of the aircraft, but also the performance, system details, the flight path taken by the particular aircraft, and much more.

CPACS allows the user to link a single analysis tool into multiple process chains of analysis within the tool or other tools as well. CPACS also stores the data of the results from the analysis into it which can be transferred to different multidisciplinary tools. CPACS acts as a common ground to which whatever analytical tools can be coupled and provide results that aid in the development of the aircraft concept or understanding of a complicated design.

CPACS is designed in a XML schema definition (.xsd) document for its advantage of the hierarchical structure, this is also highly defined for the development phases of an aircraft. For example, the wing can be broken down into ribs, spares, bolts, and more. With this top-down and bottom-up methods are more easily understood and traced in a hierarchical structure.

Additionally, XML-based files have the option to add attributes with unique IDs (uID) or symmetry flags. This is also sometimes called "metadata". Each component of the aircraft has a designated uID from which that particular part can be called for external purposes such as analysis, parameterizing, etc.

3.5.1 CPACS influence in FUGA

The outer fuselage geometry structure, engine and landing gear configuration and sizing, and many more characteristics of the aircraft are derived from CPACS. Though CPACS contains information such as aircraft performance mapping, compatible airports, flight paths, etc. Those parameters are not utilized in FUGA since it has little to no importance or dependency on the functionality of FUGA.

The FUGA tool is based on the outside-in approach, which means it requires a well-defined fuselage structure of the aircraft to create the corresponding relation between the internal cabin component parameters to the fuselage structure parameter.

CPACS is the most optimal solution for the fuselage structure primarily thanks to its parametric design structure, utilizing the hierarchical advantage of XML data sets, which is further explained in [7]. The fuselage structure definition and its details in CPACS are thoroughly explained in Scherer [10]

Various internal structure components of the aircraft's fuselage can be described in detail using CPACS such as stringers, long beams, cross beams, seat rails, etc. The cargo door and passenger door cutouts are also defined in CPACS along with window cutouts. The aircraft structure description of CPACS is divided into two parts: profile-based components and sheet-based components.

The Profile-based elements are defined in an extrusion along a curve based on the normal structure obtained from a profile defined as a 2-dimensional cross-sectional profile. This method is used in defining the stringers, struts, seat rails, and extruding components that protrude in longitudinal and rotational components along the x-axis. This method not only provides a detailed view of the structural element of the aircraft but is also parametric taking the outer curvature and sheet of the aircraft fuselage or the dependent surface. The profile-based elements are stored as nodes for structural elements and have their unique uID's which can be utilized as a reference for future components.

In the current CPACS release of 3.4 (at the time of writing), the node "decks" provides a simplified description of the cabin layout and boundaries by a grid point sampling method using contour lines upon which, seats, galleys, lavatory, etc. could be positioned and oriented. The current discrepancy that is faced is the misalignment of cabin definition with the fuselage parameter in the event of changing the fuselage or cabin structure. This would create an overlap of components and be invalid.

Sheet-based elements provide thickness to pre-defined surfaces or profiles that are created initially. This gives a skin panel that can be a reference for stringers, struts, frames, etc. Both profile and sheet-based elements provide material information for various analysis purposes.

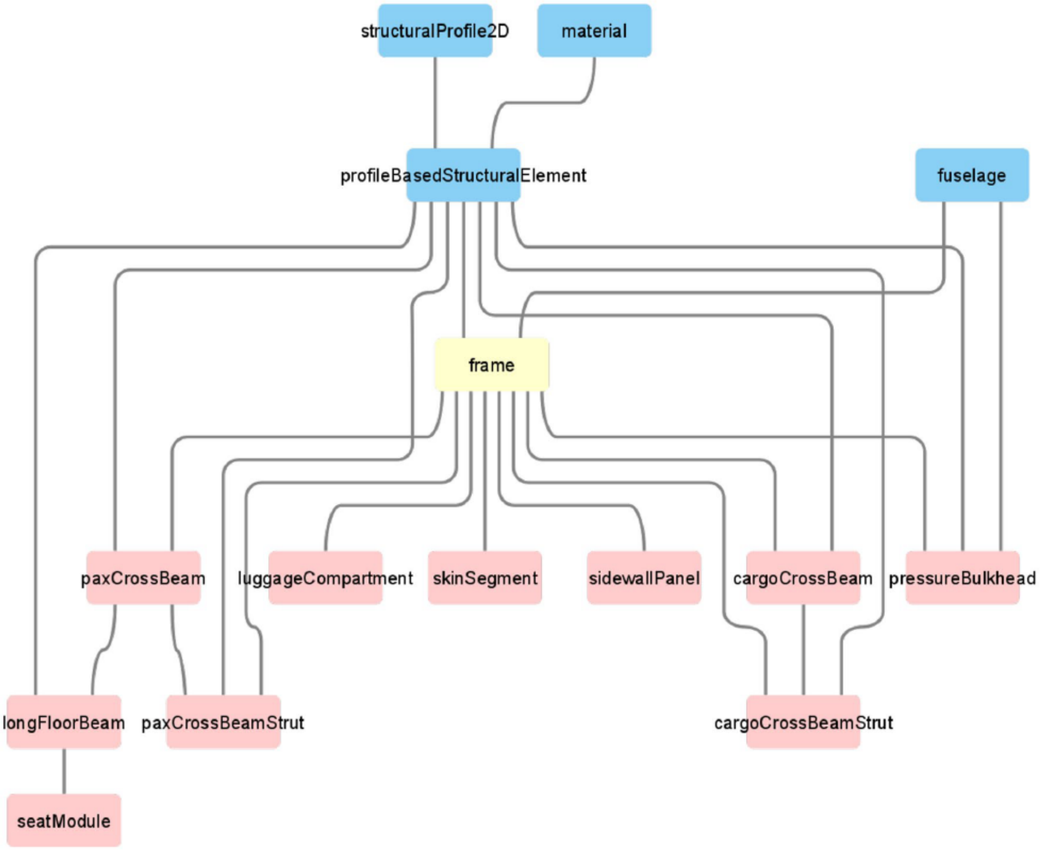


Figure 12: Dependency map for frame node in CPACS [7]

figure 12 from Walther et al [7], provides a structured representation of the relationship between nodes. In this example, it takes into account the data transition between nodes in frames. In the frame node, it is evident that it is crucial for scaling and positioning of the luggage compartment, skin segment, sidewall panels, etc. Hence frame is the parent of the downstream nodes, while the frame is the child of profileBasedStructuralElement and fuselage node. The directed acyclic graph (DAG) dependency construction is illustrated above which further provides the means to query the parents and children of any given nodes.

3.6 Web Based Application

Web Based Application (WBA) is also known as a web application or web app, which is a software application that can be easily accessed by any operating system using the internet.

It is unlike installing a desktop application or as an app on the mobile, instead it is hosted on the server by which it can be accessed either remotely or through the network.

The advantages of having a web app for FUGA makes it easily accessible within the institute, other institutes within the same organization, or external partners.
Web-based application is extensively used nowadays due to its many advantages like accessibility, Cross-platform compatibility, Seamless collaboration with different projects, scalability, data backup and security, and more.

With the requirement for interactive visualization based on the desired description produced by the FUGA tool, a web application emerges as the ideal solution, as described by Lukasczyk et al. [38], a web app developed for interactive frameworks also provides the user with visualization of the data set, for this thesis a similar perspective will be employed to show the results produced using the FUGA tool.

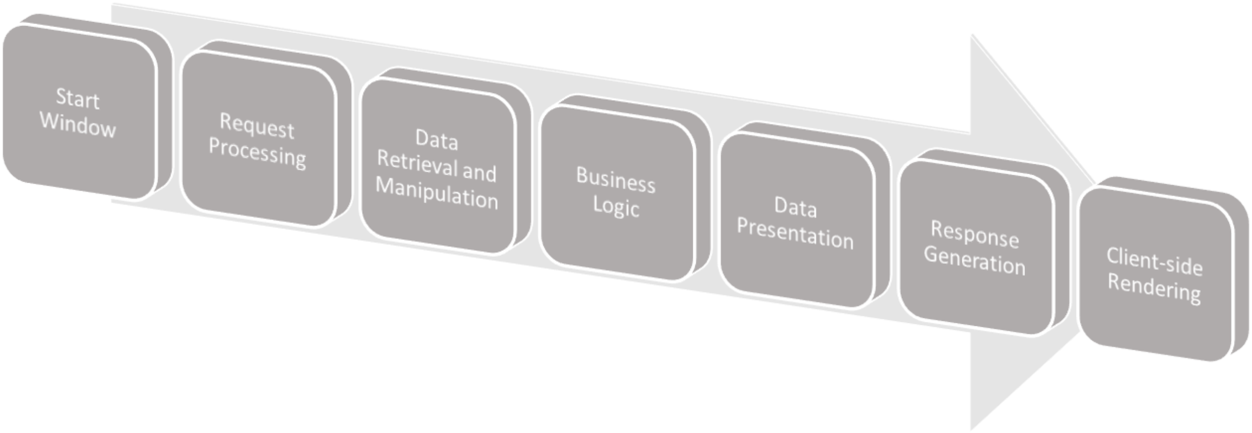


Figure 13: General Working of Web application

figure 13 gives a general flow of the working of a web application, it can be broken down into 7 steps, which are briefly explained below,

- Start Window: It is the main initial window where the request from the client (user) is taken and sent to the processing.
- Request Processing: Depending on the request, it is examined and sent to or routed to the appropriate module or component in the framework
- Data Retrieval and Manipulation: Consider some inputs and files are submitted from the user, it is then taken into the database and necessary operation on the data is performed.
- Business Logic: Based on the inputs and data received the application is processed and manipulated based on the rule set defined, this can be also said for the main background engine.

- Data Presentation: once the data is processed, it is then made ready for the presentation stage, where the data is formatted and prepared for rendering.
- Response Generation: Once the data is ready to be presented, it is then processed with the appropriate template and ready to be rendered.
- Client-side Rendering: Based on the request received, the processed data with the appropriate template is rendered and displayed to the client.

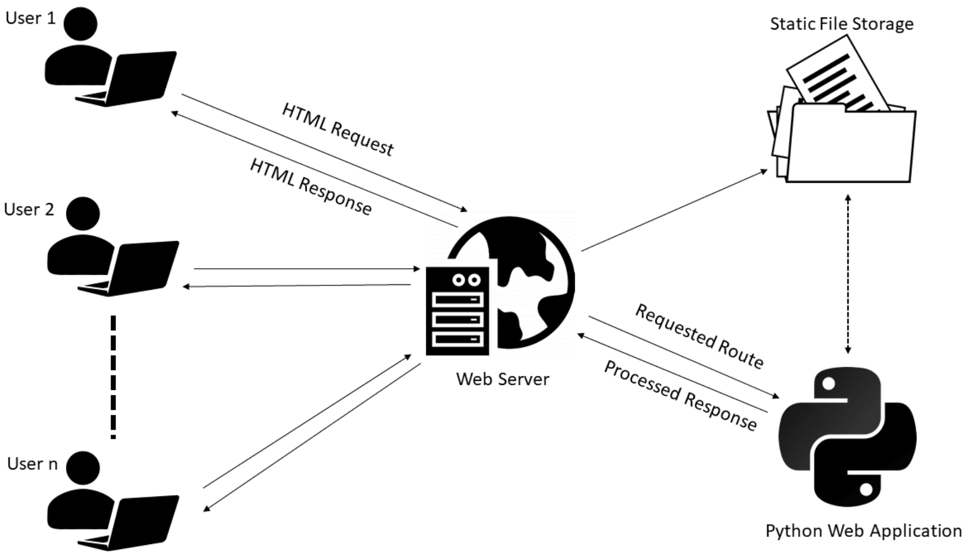


Figure 14: Flow of Request Processing and Response with Python web application

figure 14 depicts the working of a web application on a global level where n-users can access the website through a web server with HTML Request connected to it, which works on a Web application framework accepting a particular route or component to be executed with the Static folder available for storage. Once a particular component is complete a processed response is sent to the web server which is shown to the user with a help of an HTML response.

4 Methodology

The steps taken to perform each task within individual work packages are explained in detail in this section. A brief explanation of the reason for each step is also provided to give the readers a knowledgeable walk-through of individual tasks.

4.1 Implementation of Web Application for FUGA

Python has gained significance over the years and it is the most popular language in the world [39], several Python frameworks can be used to develop web applications, When it comes to their advantages and applicability, it is discussed in this section.

Additionally since the FUGA tool is coded in Python, it would be easier to implement the tool in a Python-based web application.
To implement a Python-based Web application, the following are the available frameworks as shown in table 2.

The different web-application is analyzed with the evaluation factors like Lightweight and flexible, Easy to learn, Modularity, Scalability, and cross-platform compatibility, Here is an explanation of each evaluation factor: (The Information was taken from the official documentation, Flask [40], Django [41], CherryPy [42], Pyramid [43], Tornado [44], Sanic [45], and Dash [46])

- Lightweight and flexible: This is the minimalist and flexible approach for developing a web application and integrating different application structures and components into the framework.
- Easy to Learning: When dealing with using a particular application, it is very important to know how easily the user can grasp the concept and be implemented for the project, it depends on the available documentation, tutorials, straightforward syntax, and supportive community.
- Modularity: Depends on how modular a framework can be, which makes the modules more reusable and reduces the line of code drastically
- Scalability: The performance of the framework can be seen with handling performance of user traffic, data volume, and handling parallel requests.
- Cross-platform compatibility: This denotes the web app running on different operating systems and browsers with the same effectiveness, by this, it receives a larger audience.
- Score: ”Yes” and ”No” would be the outcomes of evaluating each web framework against each evaluation factor, and ”Yes” would get a point, and ”No” will be null.

By looking at the results from table 2, Flask has the highest score, and the most key factor for the thesis would be Modularity and Ease of learning, which Flask has

Table 2: Comparison of Web Frameworks

Framework	Lightweight and Flexible	Ease of Learning	Modularity	Scalability	Cross-Platform Compatibility	Score
CherryPy	No	Yes	Yes	Yes	Yes	4/5
Dash	No	No	Yes	Yes	Yes	3/5
Django	No	No	Yes	Yes	Yes	3/5
Flask	Yes	Yes	Yes	Yes	Yes	5/5
Pyramid	No	Yes	Yes	Yes	Yes	4/5
Sanic	No	No	Yes	Yes	Yes	3/5
Tornado	No	No	Yes	Yes	Yes	3/5

an easy learning curve due to the tutorials and documentation and for a beginner it would be approachable. Thereby Flask would be preferred as a web framework for this thesis.

4.1.1 Flask Basics and Implementation

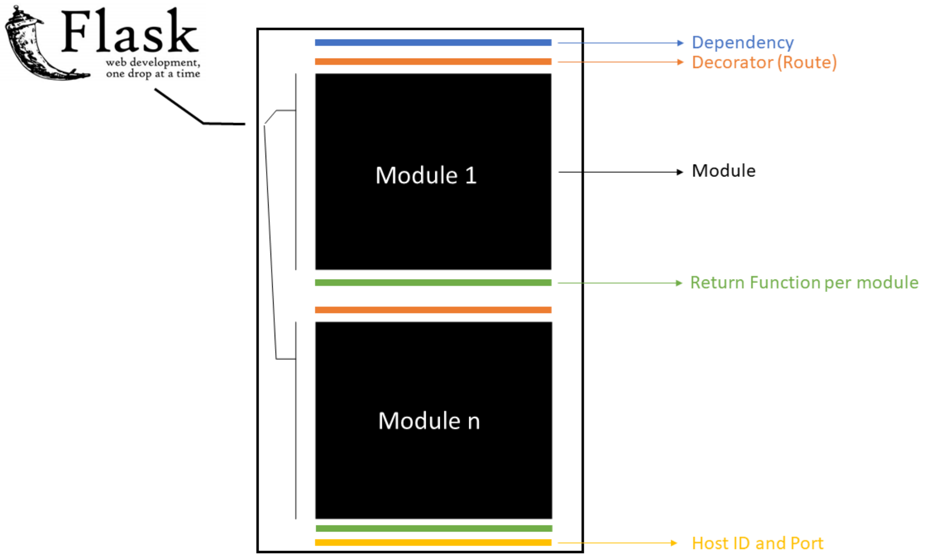


Figure 15: Components of FLASK (Python Web-based application)

The general architecture of flask can be seen in figure 15, the key aspects of the architecture are discussed below:

- Dependency (Libraries): This comes at the beginning of the code which is used to import required libraries or external modules which will be required for the application to run. An example of the same would be:

```
import numpy as np
import pandas as pd
```

Here we import the numpy and pandas, which are a library for Python programming language that is used for scientific computing and data analysis and manipulation.

- Decorator (Route): Decorator which is also called route, contains the URL paths that the application responds and by accessing the particular URL you access the particular module to be executed and is defined as,

  ```
  @app.route("/Path")
  ```

  for example, consider you have a host ID ”127.0.0.1” and want to access the result module of the flask then the URL would be ”https://127.0.0.1/Result”, this would access the module defined as

  ```
  @app.route("/Result")
  ```

- Module: This is the main section of the flask which contains the core code to obtain the result. The modules are executed when their particular Decorator is called as explained above. Modules help organize and structure related groups together, an example for the same would be to consider having an option for the user to input a file that has to be accessed and then input to be taken, so there can be two modules defined one for taking the file and storing it in the database and the other would be to take the input variables.

- Return Function: It can be called the view function, where every module should end with a return function, which is connected to an HTML file that renders the template. It is usually coded with HTML, JavaScript, and CSS script, there is an option also to pass variables/arguments called the query strings to or from a template or to a flask module itself.

- The network interface on which the Flask application should run is defined in Host ID, an example of a Host Id would be ” 127.0.0.1”, this will be the base ID and any further modules that need to be accessed needs to be addressed post to this address tag as mentioned above. An example of defining the host ID is as follows:

  ```
  app.run(host="127.0.0.1", port=8080, debug=True)
  ```

4.1.2 FUGA in Flask

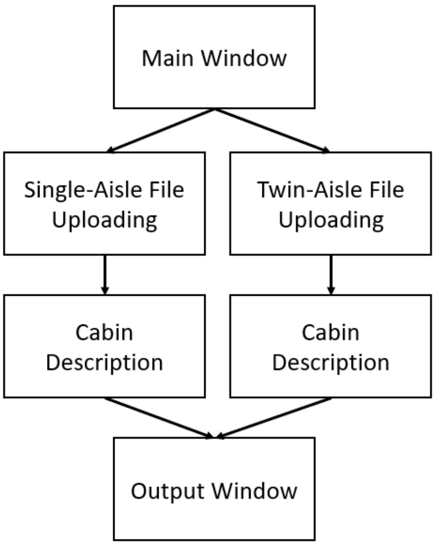


Figure 16: General flow in the web for FUGA application using Flask.

The general overview of the flow in the web page and general architecture implemented for FUGA is represented in figure 17 and 16 respectively, where the user is given a main window which has an option for either single or twin-aisle configuration followed by cabin description and output window, here the file uploading is the relevant CPACS file and cabin description json file with an optional cabin description window. The outputs will be 2D layouts and 3D models.

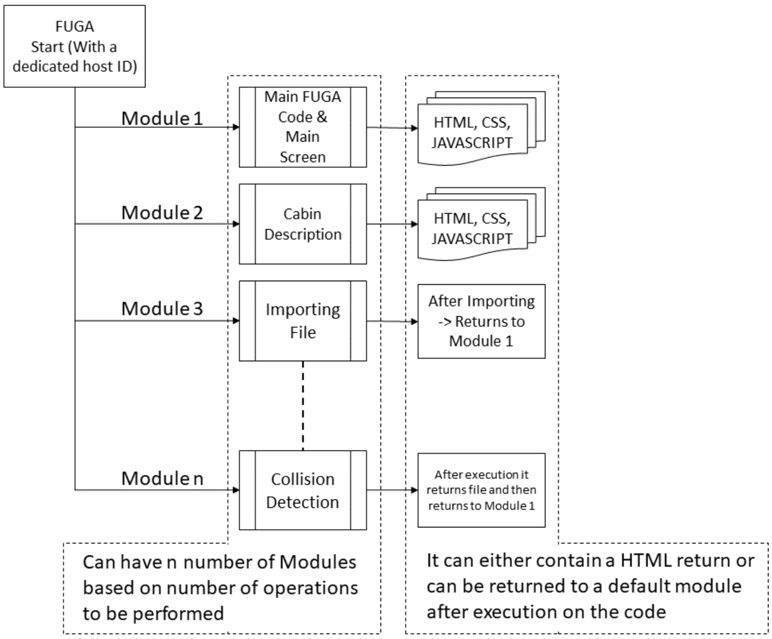

FUGA Start (With a dedicated host ID)
Module 1
Module 2
Module 3
Module n
Main FUGA Code & Main Screen
Cabin Description
Importing File
Collision Detection
HTML, CSS, JAVASCRIPT
HTML, CSS, JAVASCRIPT
After Importing -> Returns to Module 1
After execution it returns file and then returns to Module 1
Can have n number of Modules based on number of operations to be performed
It can either contain a HTML return or can be returned to a default module after execution on the code

Figure 17: General Flask architecture for FUGA application

The detailed Flask architecture implemented to FUGA is depicted in figure 18.

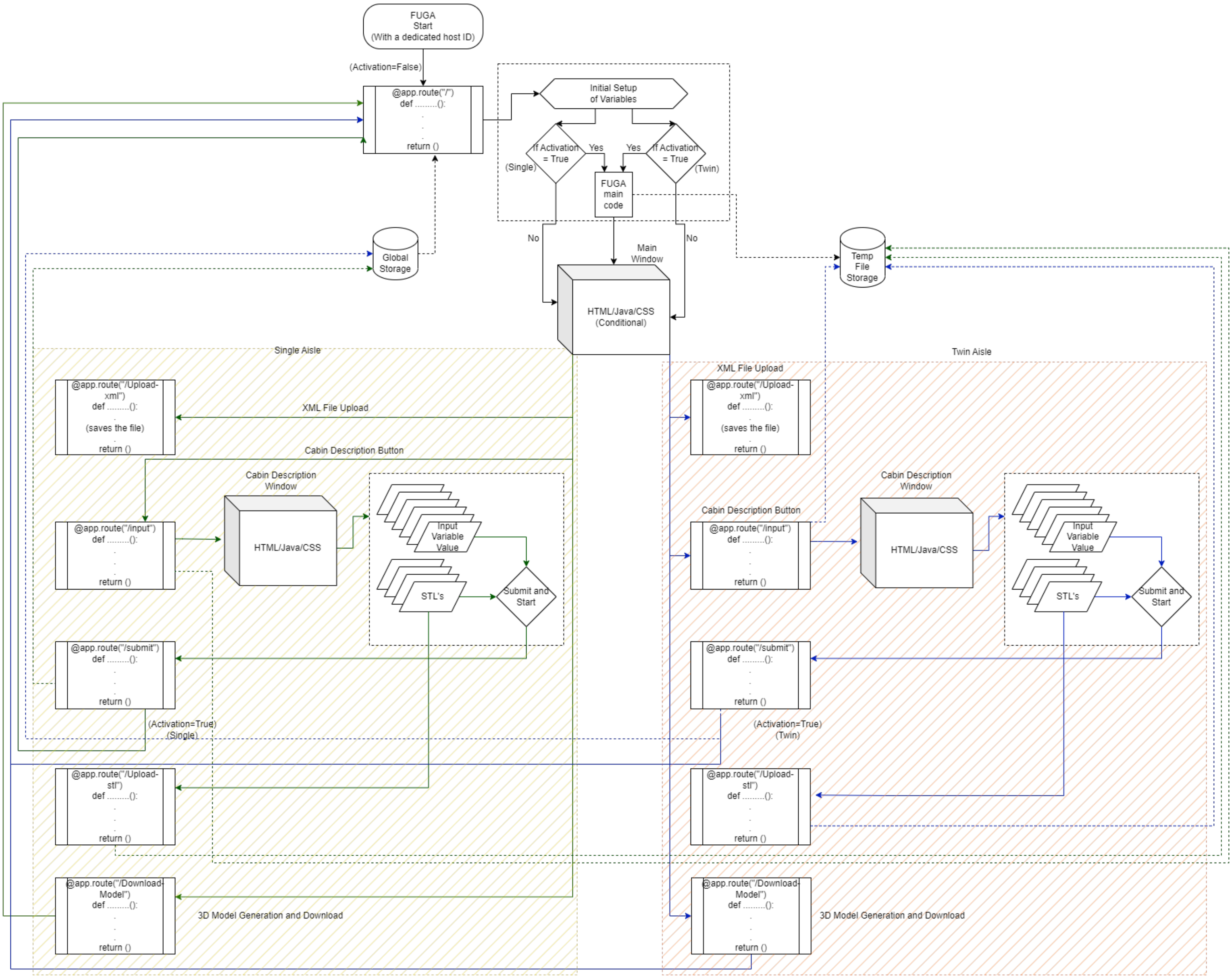


Figure 18: Detailed Flask architecture for FUGA application

The detailed explanation of the architecture in figure 18 is explained below:

- CPACS File Upload:
  The user is given the option to upload the CPACS file for both single and twin-aisle configurations and this stores the file in a temporary-static storage file which will be accessed later when the main FUGA code runs.
  To have fail-proof architecture, there is a default file available, if the user does not upload a file the default file is taken.

- Cabin Description: The Cabin description is taken from a JSON file, which is a lightweight format for storing and transporting the data. The parameters taken for or given to the user are indicated in figure 19

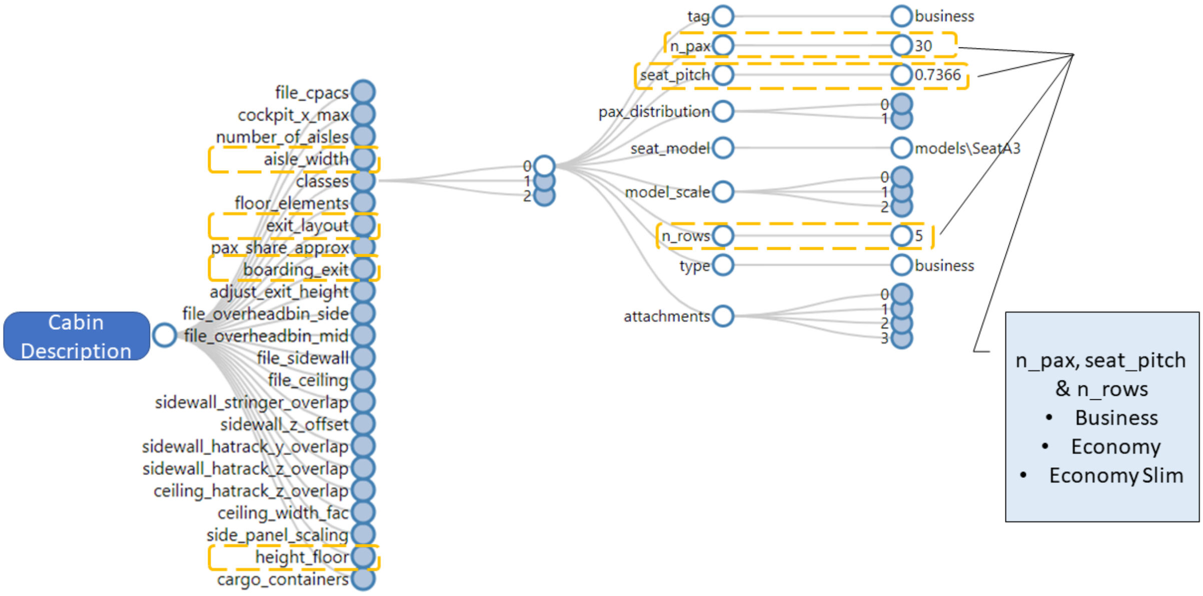


Figure 19: Cabin Description inside JSON ile

  - JSON Update: Once the inputs are taken in the 'Cabin Description Window' can be refereed in figure 18, the JSON file as to updated based on the inputs. Thereby once the 'Submit and Start' is pressed, it activates a flask module which reads the JSON file, then updates the user values, and then redirects the flask module to the main window module where the FUGA main code is activated based on the activation command received (either single or twin).
    In addition to this, the user gets a default variable displayed and assigned to the parameters if the parameter value is not assigned making the system more robust.

- Collision Detection:

  - The working is as above the input file must be submitted following which the desired stl seat module-file will be taken or uploaded, when the run/compile is executed it runs a program with the help of VTK collision detection filter to give a model of contact cell VTM model, the detailed explanation can be found in the section 4.1.3.

- Output:

  - 2D: The generated result is stored in temporary-static file storage from which the result can be viewed by the user with a specific button assigned

to each.
There will be a temporary image assigned which will be displayed if the button is accessed before generating or running the model.

- 3D: For this, the user can select the desired model from the list and then generate and download the file (The user will be provided with VTM and its corresponding VTP files, which can be visualized using software like Paraview) (Note that the files provided will be in a zipped manner)

4.1.3 Collision Detection

A collision detection between the seat and the inner fuselage is going to be performed using the help of a VTK library called 'vtkCollisionDetectionFilter', this filter helps to find the collision cells between two triangulated objects, the working principle is shown in figure 20, where 'A' and 'B' are two triangulated objects between which you have to find the contact cell of 'B' on 'A', the hatched section in the rightmost image in figure 20 indicates the contact cell on 'A'.This principle will be applied to the inner fuselage and seat modules as shown in figure 21.

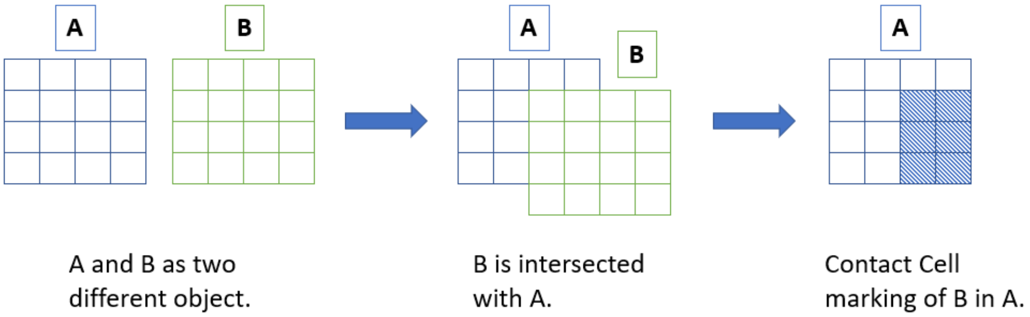

A
B
A
B
A
A and B as two different object.
B is intersected with A.
Contact Cell marking of B in A.

Figure 20: Representation of the contact cell on object A by object B using VTK collision detection filter

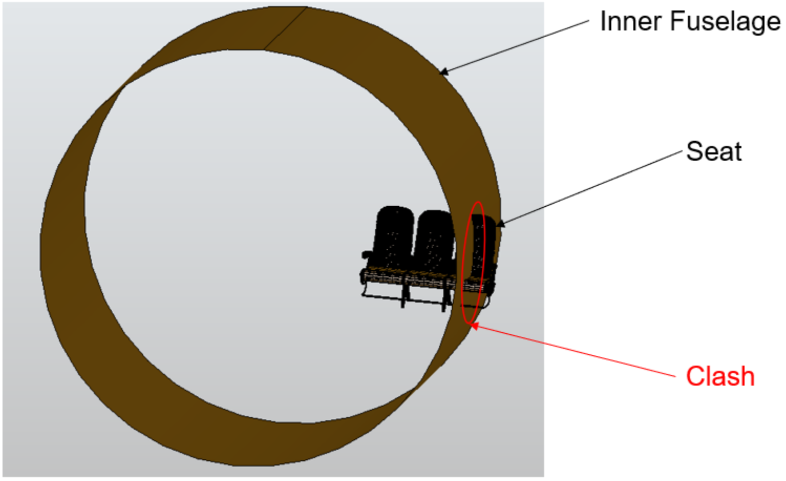


Figure 21: Basic representation of clash detection between the inner fuselage and seat

4.2 Advancement of twin-aisle configuration capability in Fuselage Geometry Assembler

The internal cabin layout and CAD model for a single aisle configuration (popular in narrow-body aircraft) and twin-aisle configuration (popular in wide-body aircraft) can be accomplished in the FUGA tool. The tool however utilizes different approaches in bringing forth the configuration.

The following input files as seen table 3 are required by FUGA to generate the layout configuration and CAD model of the respective aisle configuration layouts:

Table 3: Aisle Configuration Input files

Single aisle configuration	Twin aisle configuration
Cabin Parameter file (.json)	Cabin Parameter file (twin-aisle)(.json)
Fuselage Augmentation file (.xml)	Fuselage Augmentation file (.xml)
CPACS file (.xml)	CPACS file (.xml)
-	Cabin Parameter file (single-aisle) (.json)

The twin-aisle configuration utilizes the same structure that was used to design the single-aisle configuration of FUGA. The base class of twin-aisle changes from that of the single-aisle configuration to incorporate the introduction of new components. (eg: 2 aisles, middle seat, middle hatracks, etc.)

4.2.1 Twin Aisle Configuration - Ceiling Panels

The ceiling panels of the twin-aisle configuration initially inherited the characteristics and parameters of the single-aisle configuration. This created a floating ceiling panel that was not connected to the side hatracks nor has been scaled accordingly to cover the overhead ceiling completely.

The class that calls the ceiling into FUGA namely ”CeilingPanels” utilizes the requires and computes attributes as mentioned in section 3.1.1, these attributes help the ruleset in confirming the availability of the required uIDs and components that have been formulated and stored in the data repository during the initialization of FUGA tool.

The Ceiling Panel initially retrieves the variables and data necessary for the instantiation of ceilings. Upon retrieving, it processes the data into computing the 2d positions, CAD model uID, scaling of the panel, the total number of panels needed for the frames generated, transition and positioning of panels, etc. This processing method utilized for the ceiling panel was taken from a single aisle configuration initially.

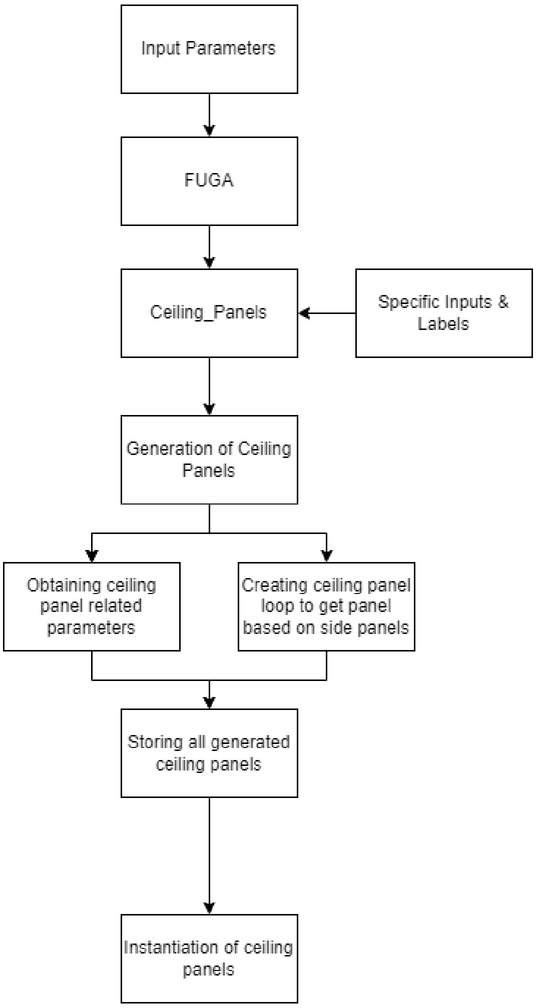

Input Parameters
FUGA
Ceiling_Panels
Specific Inputs & Labels
Generation of Ceiling Panels
Obtaining ceiling panel related parameters
Creating ceiling panel loop to get panel based on side panels
Storing all generated ceiling panels
Instantiation of ceiling panels

Figure 22: CeilingPanel class flowchart

Single aisle configuration did not take into account the shift of ceiling panels to the slots between the middle hatrack and the side hatracks. Hence, with the shift of the panel to the side, it needs to be symmetrical to instantiate on the other side of the cabin ceiling. Furthermore, It did not take into account the change of panel positioning and scaling at each intersection of galleys, lavatories, and entries.

The following method was carried out to the ceiling panel for the twin-aisle configuration:

- Step 1: The panel positioning frames have been separated into 2 containers A and B. Container A Contains the total number of panel positioning frames that has sidewall panels involved. Container B contains the total number of panel positioning frames that contain no sidewall panels.
- Step 2: For every panel frame in container A and B, the scaling, translation, and z rotation of the ceiling is appended into a separate variable respectively.
- Step 3: The existing global translation definition is used to return the translation of the CAD model of the ceiling to the respective position depending on

the type of panel the ceiling is above.

- Step 4: The transformation data frame is created individually for Containers A and B.

- Step 5: The data frames are merged before returning the class.

An illustration of the Python code is seen in figure 22. The simple flowchart explains the basic functionality on how the ceiling panels are getting instantiated inside FUGA, a more in-detail working of how the ceiling panels are getting instantiated is seen in the appendix A.1.1 71

4.2.2 Twin Aisle Configuration - Mid Hatracks

The middle hatrack, which is also called the middle luggage compartment is used specifically in twin-aisle configuration aircraft or wide-body aircraft. The middle hatrack was integrated into the FUGA tool by inserting additional codes into the "luggage compartment" class in the twin-aisle configuration base.

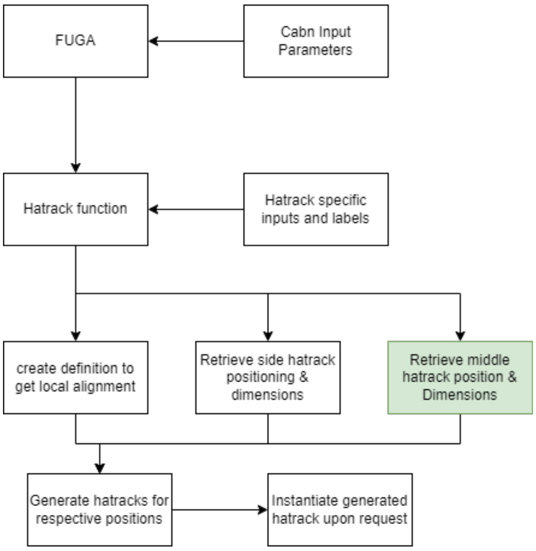

FUGA
Cabn Input Parameters
Hatrack function
Hatrack specific inputs and labels
create definition to get local alignment
Retrieve side hatrack positioning & dimensions
Retrieve middle hatrack position & Dimensions
Generate hatracks for respective positions
Instantiate generated hatrack upon request

Figure 23: Hatracks instantiation flowchart

The Luggage Compartment class initializes a definition "get_local_alignment" which gets the translation of the hatrack in y direction at the given position panel frame. The compute definition, as earlier familiarized in CeilingPanel, runs the class and provides the desired output from the class. The *requires* attribute as mentioned in section 3.1.1, only executes the class if all the requirements are satisfied. The following methods were carefully followed and performed to insert the mid-hatrack into the FUGA twin-aisle configuration.

- Step 1: Additional variables that require middle positioning coordinates in the x and y directions are introduced.

- Step 2: The dimensional boundaries of the mid hatrack (eg: z position, scaling, panel position frame specific translation, etc.) are calculated for the hatrack for its specific position.

- Step 3: The translation of all the hatracks within the panel frames is done globally.

- Step 4: The newly created hatracks that are created are stored in a separate Transformation data frame.

- Step 5: The data frame columns created for the new mid hatracks are appended to the existing list of side hatracks

In figure 23, a simple flowchart representing the function of the hatrack class is displayed. The process of commands is seen and the additional function for the existing class is marked in green. Labels are used to call in a set of information or models that have already been created in FUGA, or created during the operation of FUGA. A more detailed view of the flowchart is presented in the appendix A.1.1 72 in which, the added change to the existing system is illustrated as a yellow box. These additional lines of code inserted into the Python class would include the newly created mid-hatrack into the data frame. But since the number of panel positioning frames and the total number of hatrack does not match (Total number of hatracks pp_frames), additional lines of codes in equation 24 are interpreted into the 2d_plot module.

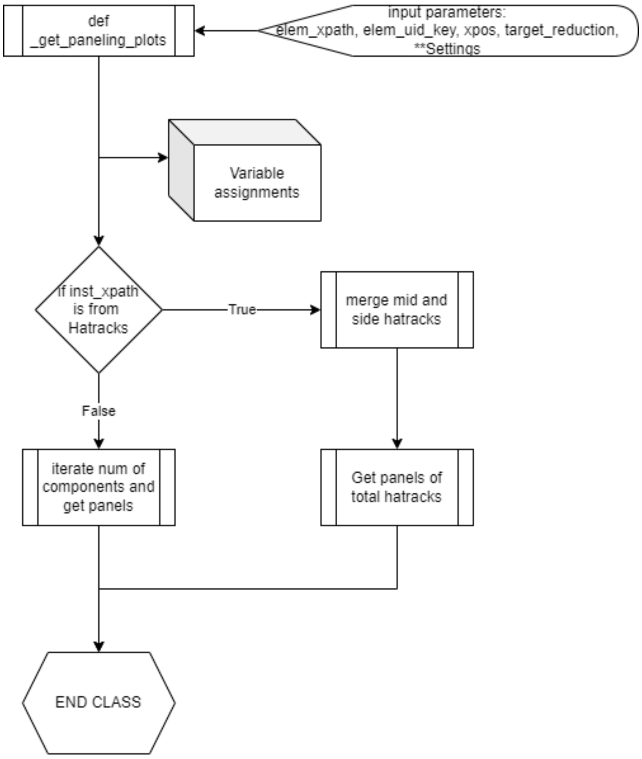

def
_get_paneling_plots
input parameters:
elem_xpath, elem_uid_key, xpos, target_reduction,
**Settings
Variable assignments
if inst_xpath is from Hatracks
True
merge mid and side hatracks
False
iterate num of components and get panels
Get panels of total hatracks
END CLASS

Figure 24: plot_2d hatrack detection flowchart

In the 2d_plot module, there is a common definition for all the cabin design components named "_get_paneling_plots". This definition iterates the number of panels utilized in the specified instantiating source path and returns the total number of iterations that the plot must repeat to instantiate all the components. A condition was inserted into the definition which turns true only upon detecting the source instantiating path to be "hatracks" as illustrated in flowchart 24. Upon detection, the total number of pp_frames are formulated as eq 1

$$tot\_pp\_(frames) = pp\_(frames) + \left(\frac{pp\_(frames)}{2}\right) \tag{1}$$

4.2.3 Twin Aisle Configuration - Lavatory Alignment

The .stl model of the lavatory when instantiated into a 3D cad model gets positioned in the same oriented direction of the default lavatory model when mirrored to the other side. This could be overcome by introducing a small conditional code into the _iterate_element_mesh definition. The iteration of the element mesh instantiates the number of elements required for the CAD component, this requires the total number of elements present in the data frame. Once iterated, the components are visible in the 3D cabin design system.

The conditional statement that is introduced into this definition is illustrated as a flow chart in fig 25. Multiple if conditions are introduced to verify and validate the presence of the lavatory cad model before initializing the conditional statement. Once the lavatory types and presence is validated, the y position of the lavatory which is the lateral direction in the aircraft fuselage is verified to be in the opposite direction, the condition transforms the mesh and rotates it 180 degrees. If the CAD model is not in the opposite direction, the CAD model is just transformed and not rotated.

4.2.4 Cargo Liners

The cargo liners being an integral part of the cargo compartment in aircraft are taken from the curves of the existing frames, beams, and beam struts of the aircraft fuselage. The cargo liner is created for both single and twin-aisle configuration and hence, it's a multi-functional model. The curves that are referenced from the beams and beam struts that are taken from the specific frame as seen in figure 26. The cargo liners serve as a heat shield in the event of a fire breakout and help in containing fire-induced damages. It also keeps the cargo compartment reinforced and helps keep containers in order. Such liners are quite heavy and it is beneficial to model and exhibit the liners in CAD modelling and meshing the liners aid in analytical studies such as individual weight extraction and contribution to the aircraft's overall weight distribution.

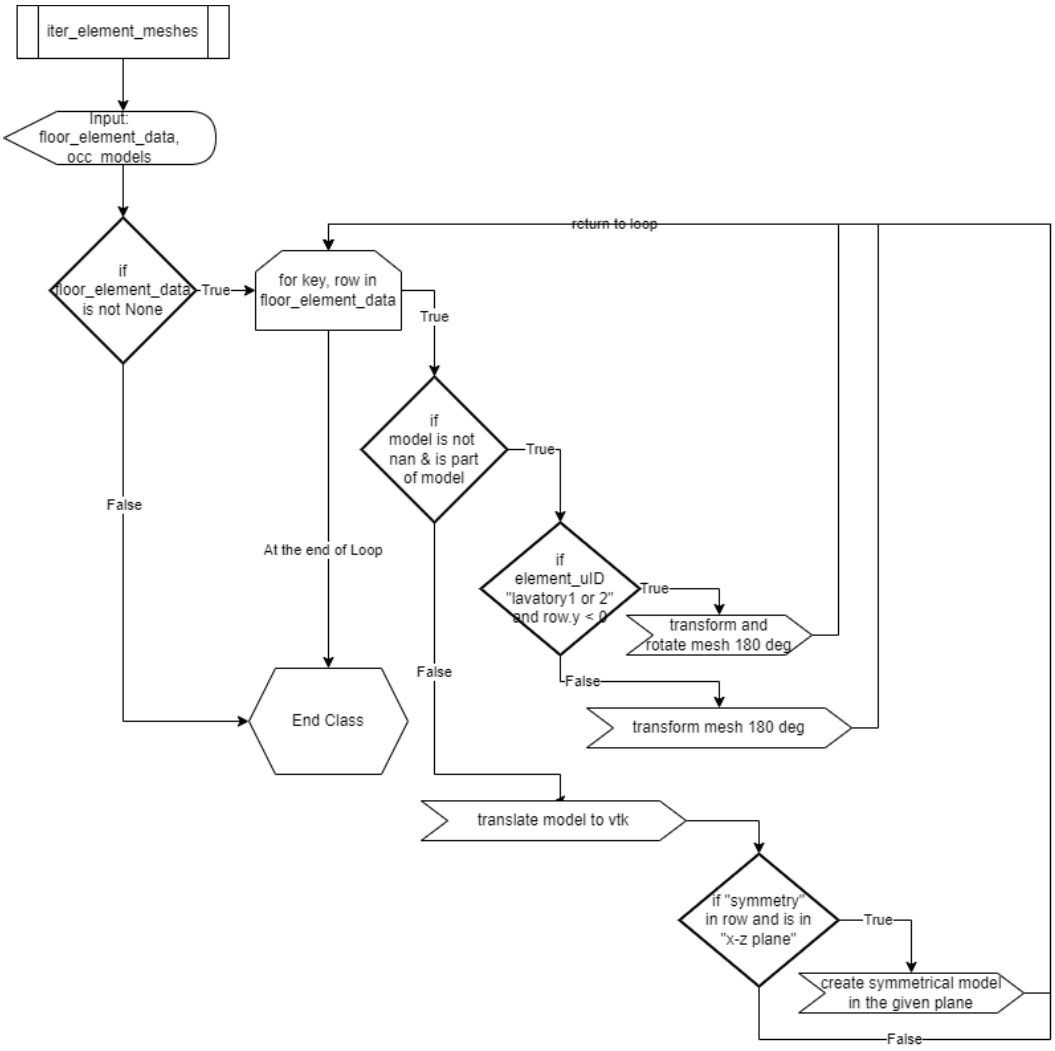

iter_element_meshes
Input: floor_element_data, occ_models
if floor_element_data is not None
True
for key, row in floor_element_data
return to loop
True
if model is not nan & is part of model
True
if element_uID "lavatory1 or 2" and row.y < 0
True
transform and rotate mesh 180 deg
False
transform mesh 180 deg
False
At the end of Loop
False
End Class
translate model to vtk
if "symmetry" in row and is in "x-z plane"
True
create symmetrical model in the given plane
False

Figure 25: Lavatory rotation condition flowchart

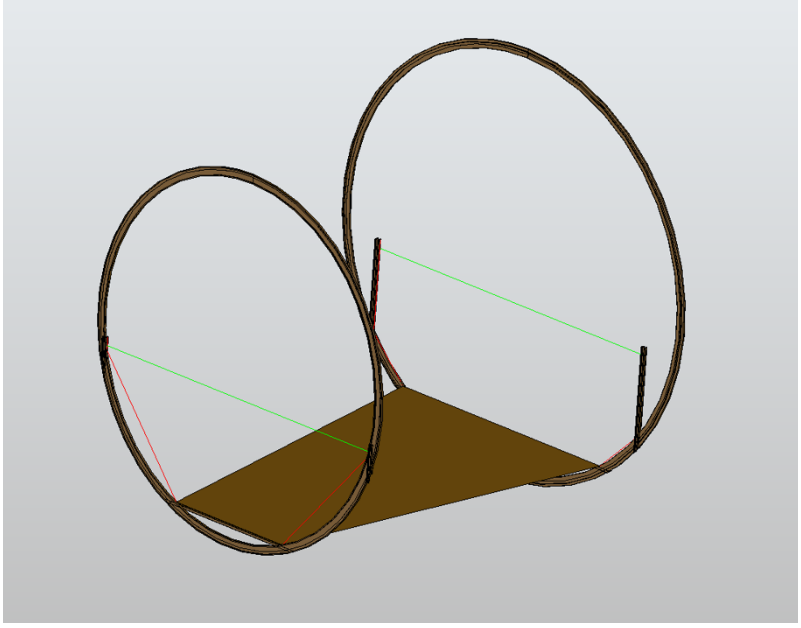


Figure 26: Cargo Liner Reference curves

The cargo liners of one segment are defined as the cargo line coverage from one frame to another in the longitudinal direction. The cargo liner of one segment consists of 4 individual cargo liner panel sides. The cargo liners. The horizontal green line visible in figure 26 shows the passenger floor deck and the red vertical and horizontal lines are the beam struts and beam respectively. These are the reference curves that are used to loft the cargo liners.

The parametric ability of the cargo liner code enables flexible operation and necessary boundaries being implemented to restrict the liner's directional error. The total length of the cargo deck is the limitation of the liner as well as the wing box region taken into consideration. The liners won't be instantiated in that region. An interval feature is also included to reduce the total number of panels being created, reducing the number of panels also reduces the time required to run the class. A more in-depth flow of how the cargo liners are being implemented into FUGA is illustrated in figure 27.

Initially, the cargo liner requires multiple labels that call in the reference curves, aircraft cargo deck X limitation taken from bulkhead X limit, and the wing box boundary in X directions. Once the labels are present, the cargo liner code initiates, If any one of the labels is not present, the cargo liner code will not execute. Once the class is executed, the class first unpacks all the labels and retrieves the necessary parameters into the specific variables it is assigned to store, and returns the variable that has the set of information from the labels.

The class has 3 functions apart from unpacking. The first 2 functions reduce the total repetition of large segments of lines that are required to retrieve the reference curves. The compute definition runs the main body of the class. This first take in all the variables that are returned from unpack and runs a "for" loop that starts from the origin position of the cargo floor of the aircraft till the start of the wing box boundary and takes the x position of each iteration with the interval as provided. Once the positions are mapped and stored, the definitions used to retrieve the reference curves are executed.

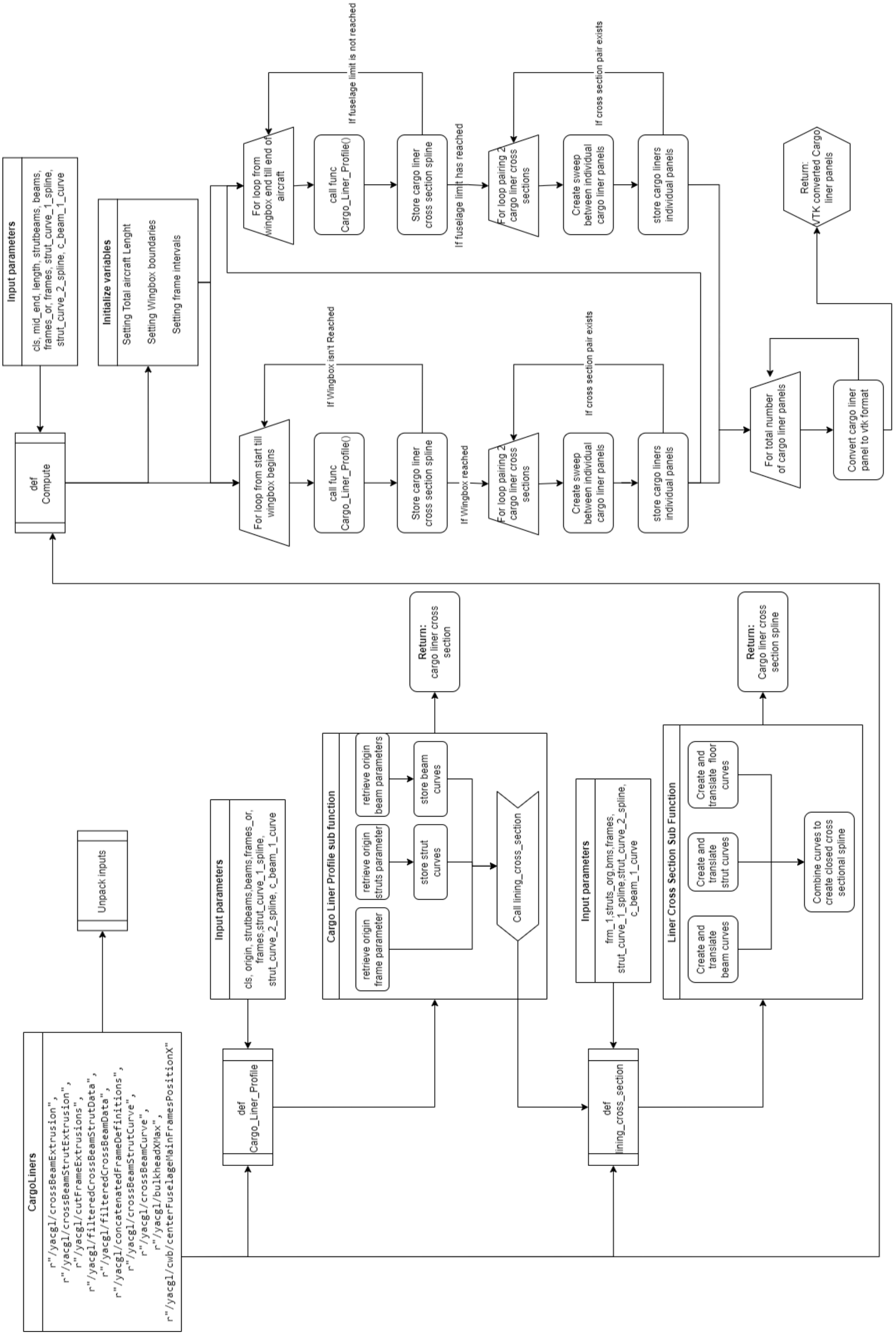

Input parameters
cls, mid_end, length, strutbeams, beams, frames_or, frames, strut_curve_1_spline, strut_curve_2_spline, c_beam_1_curve
Initialize variables
Setting Total aircraft Lenght
Setting Wingbox boundaries
Setting frame intervals
def Compute
For loop from start till wingbox begins
call func Cargo_Liner_Profile()
Store cargo liner cross section spline
If Wingbox isn't Reached
If Wingbox reached
For loop pairing 2 cargo liner cross sections
Create sweep between individual cargo liner panels
store cargo liners individual panels
If cross section pair exists
For loop from wingbox end till end of aircraft
call func Cargo_Liner_Profile()
Store cargo liner cross section spline
If fuselage limit is not reached
If fuselage limit has reached
For loop pairing 2 cargo liner cross sections
Create sweep between individual cargo liner panels
store cargo liners individual panels
If cross section pair exists
For total number of cargo liner panels
Convert cargo liner panel to vtk format
Return: VTK converted Cargo liner panels
Unpack inputs
CargoLiners
r"/yacgl/crossBeamExtrusion",
r"/yacgl/crossBeamStrutExtrusion",
r"/yacgl/cutFrameExtrusions",
r"/yacgl/filteredCrossBeamStrutData",
r"/yacgl/filteredCrossBeamData",
r"/yacgl/concatenatedFrameDefinitions",
r"/yacgl/crossBeamStrutCurve",
r"/yacgl/crossBeamCurve",
r"/yacgl/bulkheadXMax",
r"/yacgl/cwb/centerFuselageMainFramesPositionX"
Input parameters
cls, origin, strutbeams,beams,frames_or, frames,strut_curve_1_spline, strut_curve_2_spline, c_beam_1_curve
def Cargo_Liner_Profile
Cargo Liner Profile sub function
retrieve origin beam parameters
store beam curves
retrieve origin struts parameter
store strut curves
retrieve origin frame parameter
Call lining_cross_section
Return: cargo liner cross section
Input parameters
frm_1,struts_org,bms,frames, strut_curve_1_spline,strut_curve_2_spline, c_beam_1_curve
def lining_cross_section
Liner Cross Section Sub Function
Create and translate floor curves
Create and translate strut curves
Create and translate beam curves
Combine curves to create closed cross sectional spline
Return: Cargo liner cross section spline

Figure 27: Cargo liner code flowchart

The repetitive definitions first take the filtered definitions of frames, beams, and strut beams' total data frame and start to search and locate the X position of the frames. For each loop, the X position increments at the range of interval specified. The location and uID of beams, struts, and frames are identified and stored in a variable to be used for taking reference of their spline curves. This now calls in the next class definition that would then, create a copy of the spline curves used for creating the structure, and translates it into a position where it does not intersect the profile of the structures. Once the spline curves are duplicated and translated, they are merged as one fully closed circle of reference lines, which now creates a closed loop and also, into a cross-sectional line.

Once the cross-section line of all the intervals of frames is appended and stored in a variable, it is then taken into a "for" loop which will now create the links between follow-up cross-sections to create the individual cargo liner panels. The linked lines then form a closed longitudinal polygon. This closed polygon then is extruded as a shell to create the panels for the cargo liners. Each panel is then converted into vtk format for analytical purposes.

4.3 Hydrogen Tank implementation (WP3)

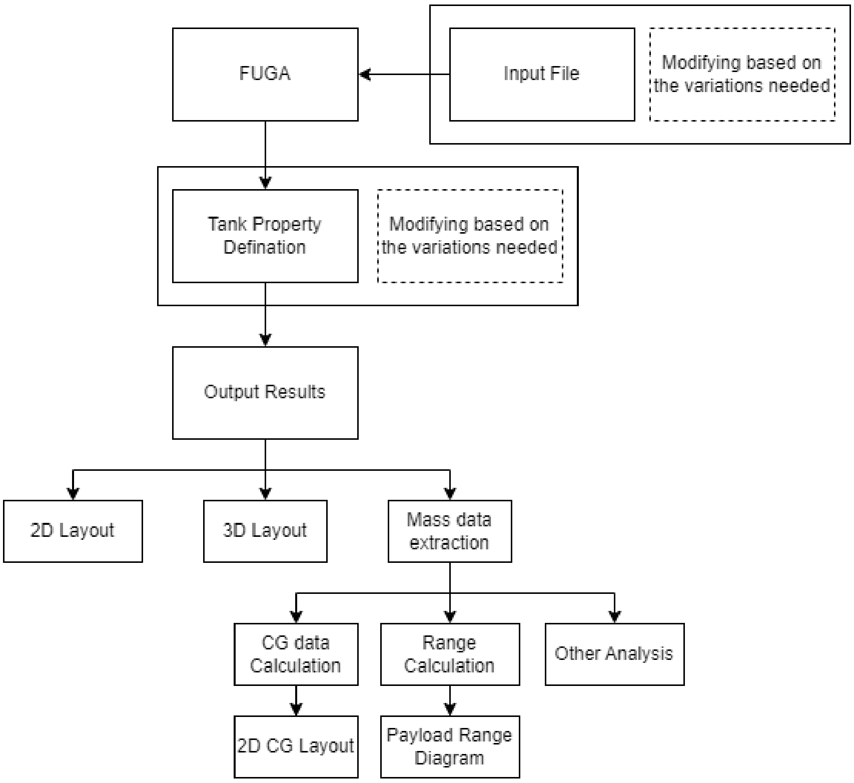


Figure 28: Process of generating results for different variations of pax capacity with tank configuration

The geometric model used for illustration and analytical purposes of FUGA is parametric in nature and is generated by the loft of 2 cross-sectional areas of the fuselage. The hydrogen tank configuration has options to vary in offsets, number of tanks, shape alterations, the distance between tanks (If multiple tanks), the distance between start and end points, and the ratio of tank sizing. The flow process of generating the data can be seen in figure 28. This flexible implementation of the tank enables the user to design the ideal tank configuration for their cabin setup and provide information as detailed as required. Furthermore, once the configuration is executed, various output data are provided. The tank volume, fuel weight, and total tank mass including miscellaneous weights are also calculated and provided for individual tanks if multiple tanks are provided. This includes the additional support weight used to sustain the tank. This enables the analytical study and understanding of including a liquid hydrogen tank.

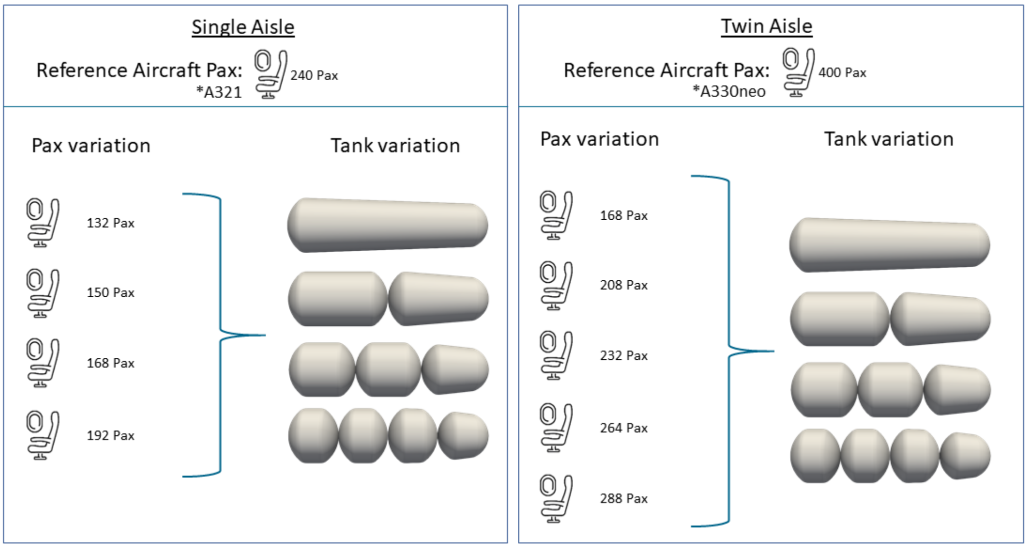
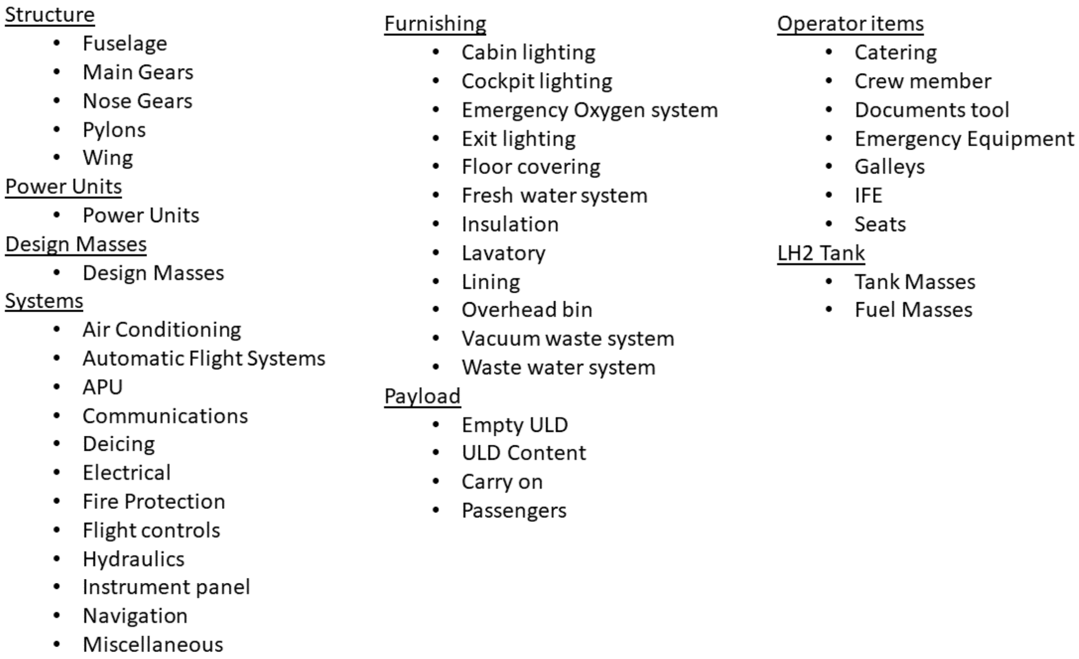


Figure 29: Number of variants evaluated in single and twin-aisle configuration

The variations of configuration with respect to tank quantities and passenger capacity is to be evaluated and taken into analysis for obtaining the optimum design for single and twin-aisle configuration. It is a common practice to analyze a wide range of variations that reaches all the points of a design graph. Specifically, the single-aisle configuration contains 12 variants and the twin-aisle configuration contains 20 variants with multiple combinations of different passenger capacities and hydrogen tank quantities as seen in figure 29.

4.3.1 Weight and CoG



Structure
- Fuselage
- Main Gears
- Nose Gears
- Pylons
- Wing

Power Units
- Power Units

Design Masses
- Design Masses

Systems
- Air Conditioning
- Automatic Flight Systems
- APU
- Communications
- Deicing
- Electrical
- Fire Protection
- Flight controls
- Hydraulics
- Instrument panel
- Navigation
- Miscellaneous

Furnishing
- Cabin lighting
- Cockpit lighting
- Emergency Oxygen system
- Exit lighting
- Floor covering
- Fresh water system
- Insulation
- Lavatory
- Lining
- Overhead bin
- Vacuum waste system
- Waste water system

Payload
- Empty ULD
- ULD Content
- Carry on
- Passengers

Operator items
- Catering
- Crew member
- Documents tool
- Emergency Equipment
- Galleys
- IFE
- Seats

LH2 Tank
- Tank Masses
- Fuel Masses

Figure 30: List of Weights obtained by FUGA tool

For the calculation of the Center of Gravity (CoG) and the Static Margin (SM), the weights of the components were extracted from the Mass model which was developed in the FUGA tool as also shown in figure 28, this included the x, Y and z locations with mass. The masses which were considered are shown in figure 30 The Static Margin which was calculated as a percentage of Mean Aerodynamic Chord (MAC) was obtained by the equation 2.

$$StaticMargin(SM) = \frac{NeutralPoint(NP)location[m] - Mass(x/y/z)Location_{min/max}[m]}{MeanAerodynamicChord[m]} * 100 \tag{2}$$

4.3.2 Weight and Range

The range of the aircraft is calculated using Berguet's range equation [47]. This equation 3 takes into account aircraft flying conditions, engine performance, weight differentiation, and flying performance.

$$R = \frac{v}{g * TSFC} \frac{L}{D} ln\left(\frac{W_{initial}}{W_{final}}\right) \tag{3}$$

The velocity(v) is provided in m/s, gravity (g) is a constant of 9.81 m/s, and the thrust-specific fuel consumption is calculated in $kg/kN.s$. The Thrust-specific fuel consumption differs from single and twin-aisle as the reference aircraft are also different, it is assumed that the hydrogen-powered gas turbine has 10 % more efficiency and hence, the reference aircraft's $TSFC * 0.9$. The L/D ratio is a constant for all the variants and configuration and the crucial calculations and variation in results that takes place in the range is in the initial weight and the final weight of the aircraft.

4.3.3 Liquid Hydrogen Tank In Single Aisle Configuration

In the LH2 tank in single-aisle configuration, a reference aircraft of Airbus A321[48] is considered, for comparison, and a similar geometry is provided by CPACS for liquid hydrogen tank integration. The table 4 shows all data for all the factors taken into account for all the variant considerations and the reference aircraft. The LH2 aircraft is assumed to fly at 35,000 $feet$ at the velocity of 220 m/s. The aircraft has the same L/D ratio as the reference aircraft of 17 and has 10% more efficient than the reference aircraft Thrust Specific Fuel Consumption (TSFC) because the engine Hydrogen powered turbine engine is assumed to be 10% more efficient than Pratt & Whitney PW1000G turbofan engine.

Table 4: Single-aisle configuration variant parameters

	Variant 1	Variant 2	Variant 3	Variant 4	Reference A321
Pax Capacity	132	150	168	192	240
Number of Rows	22	25	28	32	40
TSFC (kg/kNs)	1.603E-5	1.603E-5	1.603E-5	1.603E-5	2.004E-5
Cruise Velocity (m/s)	220	220	220	220	220
Cruise Altitude (ft)	35000	35000	35000	35000	35000
L/D Ratio	17	17	17	17	17

4.3.4 Liquid Hydrogen Tank In Twin Aisle Configuration

The LH2 tank in twin-aisle configuration has a slight advantage over the single-aisle configuration thanks to the widened diameter of the fuselage, this benefits with larger volume, but as a trade-off, a larger tank and additional weights are required, which can affect the configuration adversely. For comparison, a reference aircraft of Airbus A330-NEO[49] is considered and a similar geometry is provided by CPACS for liquid hydrogen tank integration. The following comparison data Table 5 is taken for liquid hydrogen tank integration for twin-aisle configuration. The LH2 aircraft is assumed to fly at 35,000 $feet$ at the velocity of 220 m/s. The aircraft has the same L/D ratio as the reference aircraft of 17 and has 10 % more efficient than the reference aircraft Thrust Specific Fuel Consumption (TSFC) because the engine of Hydrogen powered turbine engine is assumed to be 10 % more efficient than Trent 7000 turbine engine.

4.3.5 Comparison Between Aisle Configuration

When the most optimum tank configuration for individual aisle configuration was studied. The comparison between aisle configuration with the same tank quantities and passenger capacity would provide the viewer with a more in-depth view and understanding of the trade-offs and merits that each aisle configurations hold against the liquid hydrogen tanks. Taking into account the change in the tank quantities. The discussions on how will the fuel weight vary with different aisle configurations and also how that contributes to the total aircraft performance being affected. It is noted that the number of passengers is kept constant and equal in both aisle configurations to have a uniform comparison.

Table 5: Twin aisle configuration variant parameters

	Variant 1	**Variant 2**	**Variant 3**	**Variant 4**	**Variant 5**	**Reference A330-neo**
Pax Capacity	168	208	232	264	288	400
Rows in Middle	20	27	30	34	37	50
Rows in Side	22	25	28	32	35	50
TSFC (kg/kNs)	$1.32*10^{-5}$	$1.32*10^{-5}$	$1.32*10^{-5}$	$1.32*10^{-5}$	$1.32*10^{-5}$	$1.65\text{x}10^{-5}$
Cruise Velocity (m/s)	220	220	220	220	220	220
Cruising Altitude (ft)	35000	35000	35000	35000	35000	35000
L/D Ratio	17	17	17	17	17	17

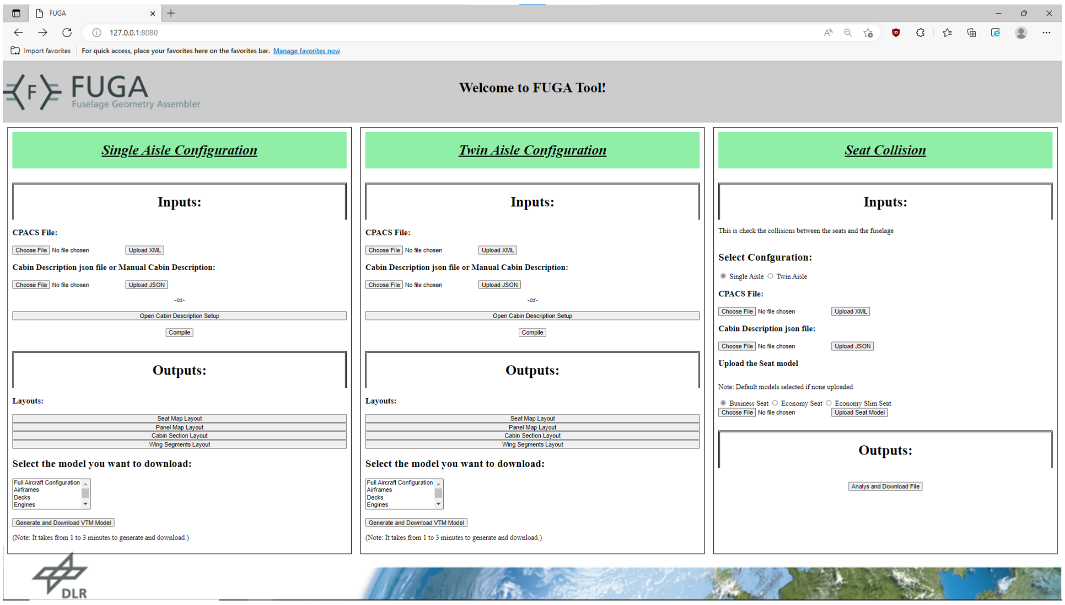
Different variants of pax capacity for twin-aisle configuration provide insight into the configuration performance and trade-offs that are experienced with variation in passenger capacity/payload of the aircraft. This variation parameter is performed for 4 different tank quantities of 1,2,3 and 4. Several tanks influence the weight of the aircraft structure as well as useful fuel for aircraft range. Though theoretically, a single tank would provide more volume for mass, it is not redundant, while a circular shape of each tank can be achieved for more tanks.

5 Results

The outcomes of each task are briefly exhibited in this section. The result section plays a crucial role in the thesis overall since it shows the findings and outcome of the research, method, and application.

5.1 Implementation of Flask in FUGA

The 'Main Window' shown in the architecture in figure 18 is the same main window web browser is shown in figure 31. Refereeing to the same figure 31 there are three dedicated sections, the left one dedicated to the single-aisle configuration, the mid-section for the twin-aisle configuration, and the rightmost for Seat Collision detection, can see that input for CPACS and json files are provided with a compile button for the initial execution of the Code to get the outputs as shown in figure 34.


FUGA
127.0.0.1:8080
Import favorites
For quick access, place your favorites here on the favorites bar. Manage favorites now
FUGA
Fuselage Geometry Assembler
DLR

Welcome to FUGA Tool!

Single Aisle Configuration

Inputs:

CPACS File:

Choose File No file chosen Upload XML

Cabin Description json file or Manual Cabin Description:

Choose File No file chosen Upload JSON

-or-

Open Cabin Description Setup

Compile

Outputs:

Layouts:

Seat Map Layout
Panel Map Layout
Cabin Section Layout
Wing Segments Layout

Select the model you want to download:

Full Aircraft Configuration
Airframes
Decks
Engines

Generate and Download VTM Model

(Note: It takes from 1 to 3 minutes to generate and download.)

Twin Aisle Configuration

Inputs:

CPACS File:

Choose File No file chosen Upload XML

Cabin Description json file or Manual Cabin Description:

Choose File No file chosen Upload JSON

-or-

Open Cabin Description Setup

Compile

Outputs:

Layouts:

Seat Map Layout
Panel Map Layout
Cabin Section Layout
Wing Segments Layout

Select the model you want to download:

Full Aircraft Configuration
Airframes
Decks
Engines

Generate and Download VTM Model

(Note: It takes from 1 to 3 minutes to generate and download.)

Seat Collision

Inputs:

This is check the collisions between the seats and the fuselage

Select Confguration:

● Single Aisle ○ Twin Aisle

CPACS File:

Choose File No file chosen Upload XML

Cabin Description json file:

Choose File No file chosen Upload JSON

Upload the Seat model

Note: Default models selected if none uploaded

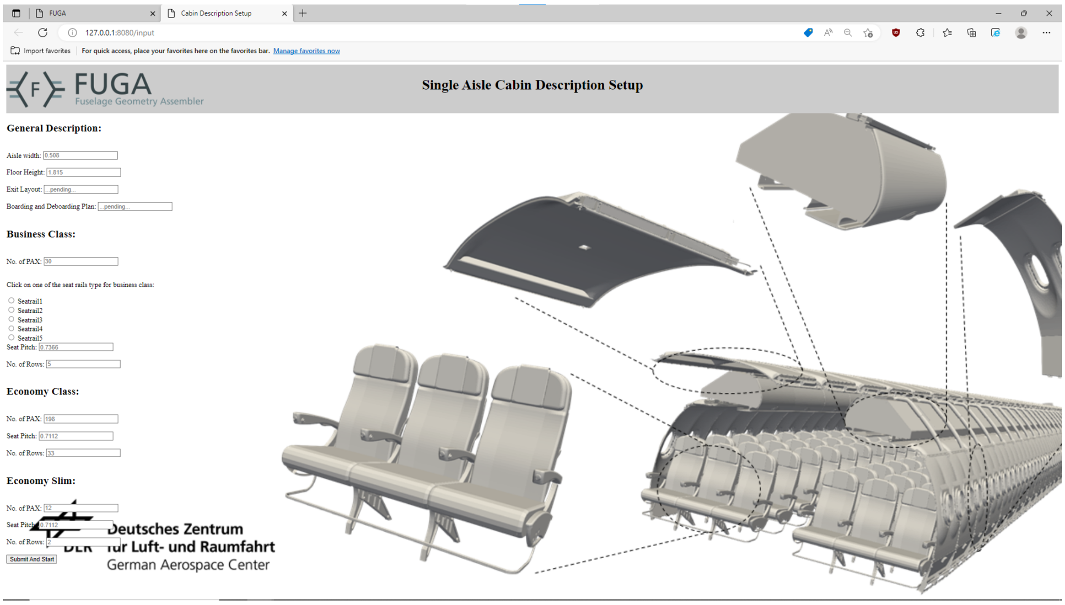
● Business Seat ○ Economy Seat ○ Economy Slim Seat
Choose File No file chosen Upload Seat Model

Outputs:

Analys and Download File

Figure 31: FUGA main web page

figure 32 and figure 33 denote the cabin description for the single-aisle and twin-aisle configuration respectively, the parameters shown in these web screens are the selected parameters seen in figure 19. These screens have the 'Submit and Start' button which sets the activation of the respective aisle configuration to be run and sends it to the main window module of the flask.


FUGA
Cabin Description Setup
127.0.0.1:8080/input
Import favorites
For quick access, place your favorites here on the favorites bar. Manage favorites now
FUGA
Fuselage Geometry Assembler
DLR
Deutsches Zentrum für Luft- und Raumfahrt
German Aerospace Center

Single Aisle Cabin Description Setup

General Description:

Aisle width: 0.508

Floor Height: 1.815

Exit Layout: ...pending...

Boarding and Deboarding Plan: ...pending...

Business Class:

No. of PAX: 30

Click on one of the seat rails type for business class:

○ Seatrail1
○ Seatrail2
○ Seatrail3
○ Seatrail4
○ Seatrail5

Seat Pitch: 0.7366

No. of Rows: 5

Economy Class:

No. of PAX: 198

Seat Pitch: 0.7112

No. of Rows: 33

Economy Slim:

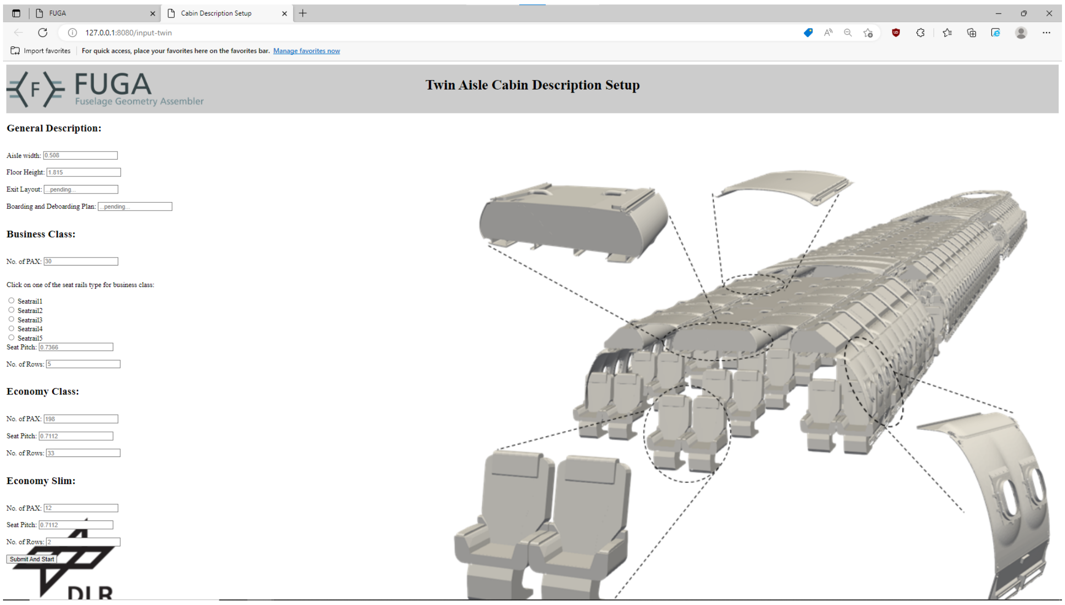
No. of PAX: 12

Seat Pitch: 0.7112

No. of Rows: 2

Submit And Start

Figure 32: Cabin Description window of Single-aisle Configuration


FUGA
Cabin Description Setup
127.0.0.1:8080/input-twin
Import favorites
For quick access, place your favorites here on the favorites bar. Manage favorites now
FUGA
Fuselage Geometry Assembler
DLR

Twin Aisle Cabin Description Setup

General Description:

Aisle width: 0.508

Floor Height: 1.815

Exit Layout: ...pending...

Boarding and Deboarding Plan: ...pending...

Business Class:

No. of PAX: 30

Click on one of the seat rails type for business class:

○ Seatrail1
○ Seatrail2
○ Seatrail3
○ Seatrail4
○ Seatrail5

Seat Pitch: 0.7366

No. of Rows: 5

Economy Class:

No. of PAX: 198

Seat Pitch: 0.7112

No. of Rows: 33

Economy Slim:

No. of PAX: 12

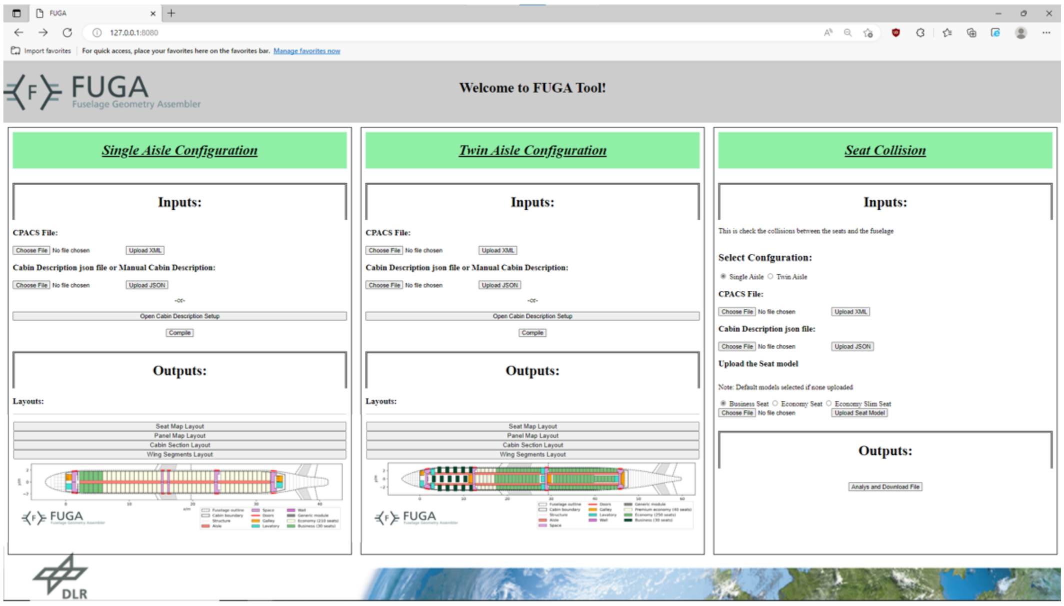
Seat Pitch: 0.7112

No. of Rows: 2

Submit And Start

Figure 33: Cabin Description window of Twin-aisle Configuration

The instant view of the 2D layouts are made available on the main screen once the module run is finished and the result of which can be seen in figure 34.



Welcome to FUGA Tool!

Single Aisle Configuration

Inputs:

CPACS File:

Choose File No file chosen | Upload XML

Cabin Description json file or Manual Cabin Description:

Choose File No file chosen | Upload JSON

-or-

Open Cabin Description Setup

Compile

Outputs:

Layouts:

Seat Map Layout
Panel Map Layout
Cabin Section Layout
Wing Segments Layout

Twin Aisle Configuration

Inputs:

CPACS File:

Choose File No file chosen | Upload XML

Cabin Description json file or Manual Cabin Description:

Choose File No file chosen | Upload JSON

-or-

Open Cabin Description Setup

Compile

Outputs:

Layouts:

Seat Map Layout
Panel Map Layout
Cabin Section Layout
Wing Segments Layout

Seat Collision

Inputs:

This is check the collisions between the seats and the fuselage

Select Confguration:

● Single Aisle ○ Twin Aisle

CPACS File:

Choose File No file chosen | Upload XML

Cabin Description json file:

Choose File No file chosen | Upload JSON

Upload the Seat model

Note: Default models selected if none uploaded

● Business Seat ○ Economy Seat ○ Economy Slim Seat

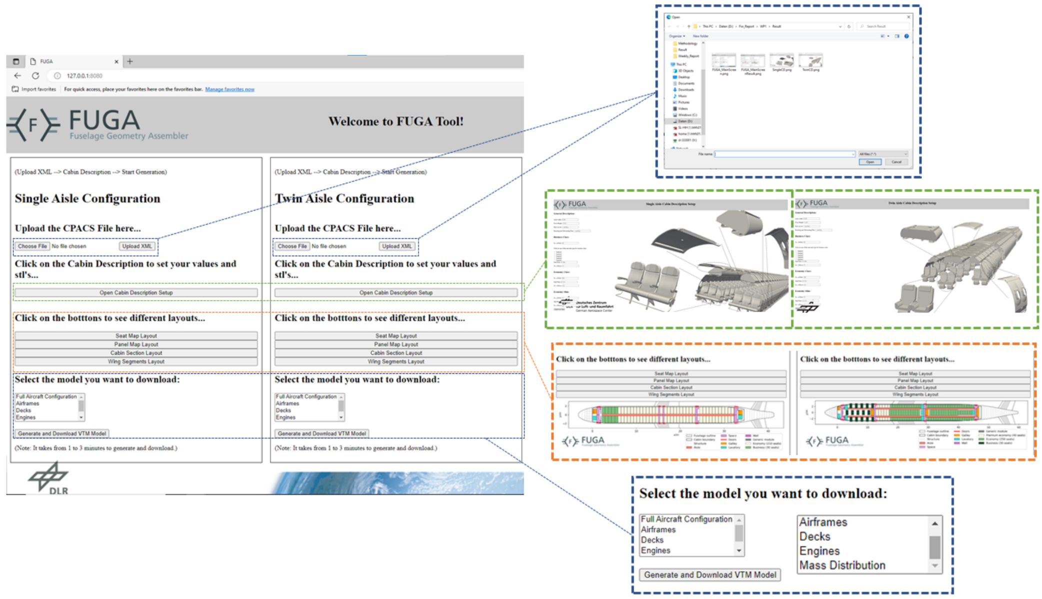
Choose File No file chosen | Upload Seat Model

Outputs:

Analys and Download File

Figure 34: Display of 2D layouts in FUGA main screen

A summarized view of all the options and windows is shown in figure 35, which includes the file uploading, the cabin description, 2D layout output, and 3D model generation and download.



Welcome to FUGA Tool!

(Upload XML --> Cabin Description --> Start Generation)

Single Aisle Configuration

Upload the CPACS File here...

Choose File No file chosen | Upload XML

Click on the Cabin Description to set your values and stl's...

Open Cabin Description Setup

Click on the botttons to see different layouts...

Seat Map Layout
Panel Map Layout
Cabin Section Layout
Wing Segments Layout

Select the model you want to download:

Full Aircraft Configuration
Airframes
Decks
Engines

Generate and Download VTM Model

(Note: It takes from 1 to 3 minutes to generate and download.)

(Upload XML --> Cabin Description --> Start Generation)

Twin Aisle Configuration

Upload the CPACS File here...

Choose File No file chosen | Upload XML

Click on the Cabin Description to set your values and stl's...

Open Cabin Description Setup

Click on the botttons to see different layouts...

Seat Map Layout
Panel Map Layout
Cabin Section Layout
Wing Segments Layout

Select the model you want to download:

Full Aircraft Configuration
Airframes
Decks
Engines

Generate and Download VTM Model

(Note: It takes from 1 to 3 minutes to generate and download.)

Select the model you want to download:

Full Aircraft Configuration
Airframes
Decks
Engines

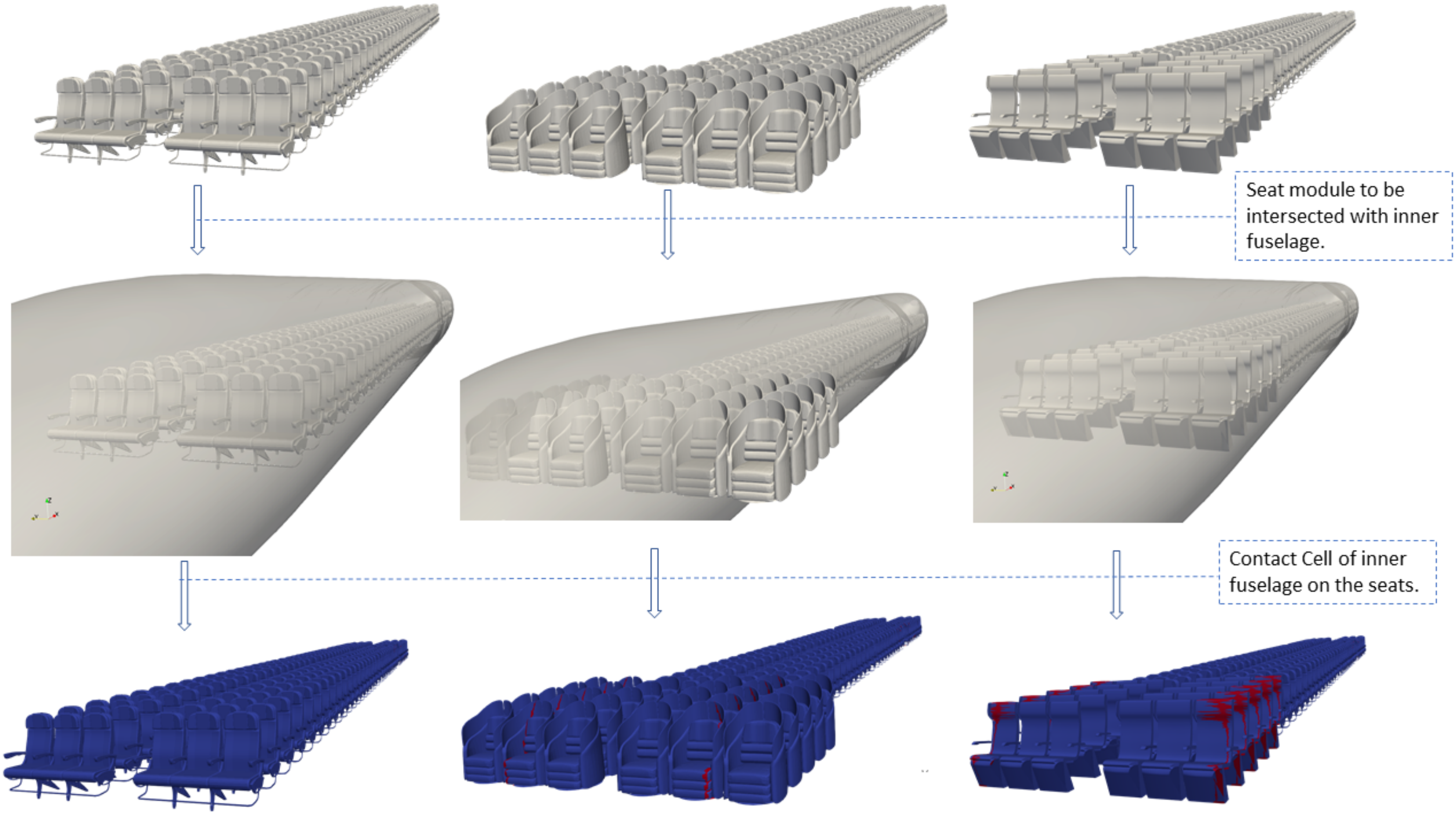
Airframes
Decks
Engines
Mass Distribution

Generate and Download VTM Model

Figure 35: Summarised view of all the options and web screens

5.1.1 Collision Detection

The representation of the clash detection is shown in figure 36, initially the seat modules are taken, followed by the fuselage, the collision of the fuselage on the seat is executed, and contact cells marked in red are shown in the last row in the figure 36.


Seat module to be intersected with inner fuselage.

Contact Cell of inner fuselage on the seats.

Figure 36: Collision detection between the seats and fuselage

5.2 Advancement of twin-aisle configuration capability in Fuselage Geometry Assembler

The results of changes done to the twin-aisle configuration are displayed in this section. The results in both 2D and 3D models of the cabin design system are illustrated with a comparison to its previous model/version.

5.2.1 Twin Aisle Configuration - 2D Cabin layout

The changes done in the ceiling region of the twin-aisle configuration in the 2D plot are largely seen in figure 37. The ceiling panels as well as the integration of the middle hatrack are noticed. When compared to the previous version of the twin-aisle configuration, there are no middle hatrack components visible in 2D as well as 3D models and the ceiling panels were positioned in the middle of the roof. This positioning visualizes the ceiling hanging/floating in the air. This is now corrected and adapted accordingly to the position of the middle hatrack and at the regions where there is no middle hattrack, the ceiling panels sizes appropriately to cover the overhanging voids in places where there was a hatrack.

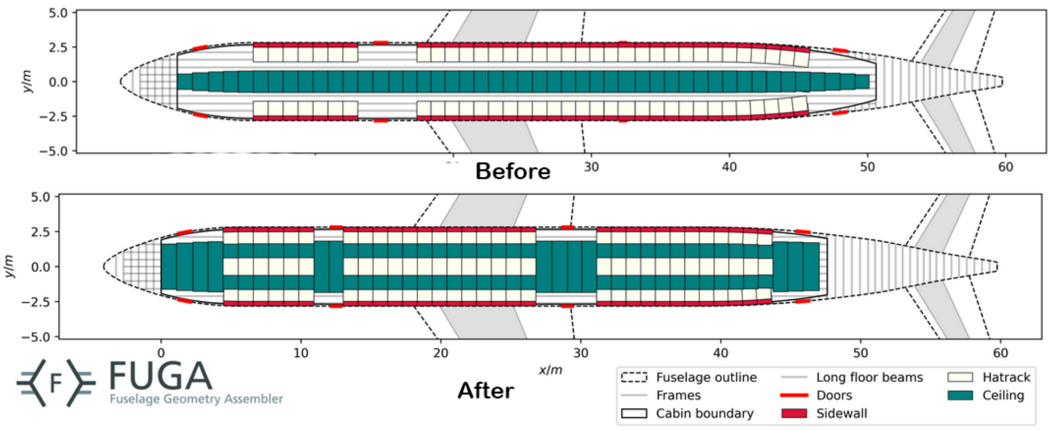


Figure 37: Twin aisle configuration ceiling layout modification

The missing luggage compartment in the middle is more clearly seen in figure 38 along with the floating ceiling panels at the left side figure. The corrected ceiling panels and introduction of middle hatracks are shown in the right side figure.

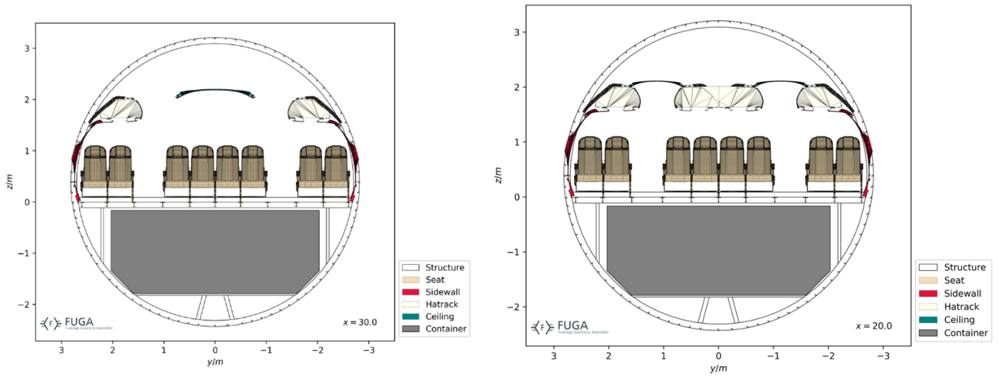


Figure 38: Cross Section - Twin Aisle Original Configuration (LEFT) and Twin aisle Configuration Modified - (RIGHT)

Before arriving at the final version of the modifications done to the twin-aisle configuration. There were some iterations done to the original model, taking the process step by step. The first initial step of implementing the middle hatrack and alignment of galleys and seats is seen in figure 39. In this figure, it is clearly visible that the hatracks have been positioned and placed in the middle region of the ceiling, but the corrected ceiling panels have not yet been introduced. Additionally, due to the extended length of the aft region of the fuselage, there is additional area behind the rear exits and the tool-generated hatracks in these regions.

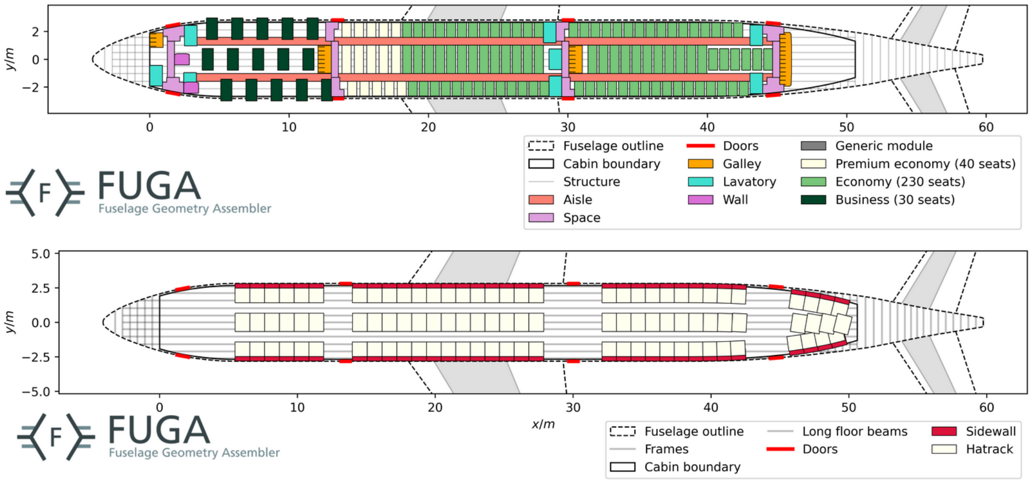


Figure 39: 1st iteration of cabin layout configuration

The second iteration of the twin-aisle configuration introduces common ceiling panel elements that cover the floating hatracks issue as seen in fig 40. Additionally, the cabin layout boundary and placement of bulkheads were modified by the aircraft's pax capacity. The ceiling and hatracks are positioned accurately and are properly sized.

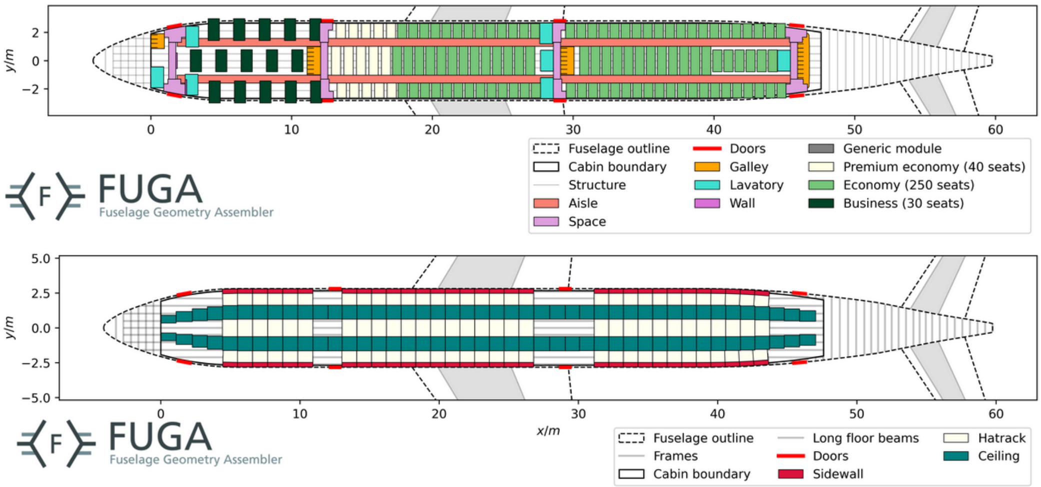


Figure 40: 2nd iteration of cabin layout configuration

The third and final iteration as seen in figure 41 of the twin-aisle configuration introduces a variable ceiling scale upon detection of the absence of sidewall panels. This is parametric and does not require user inputs to vary.

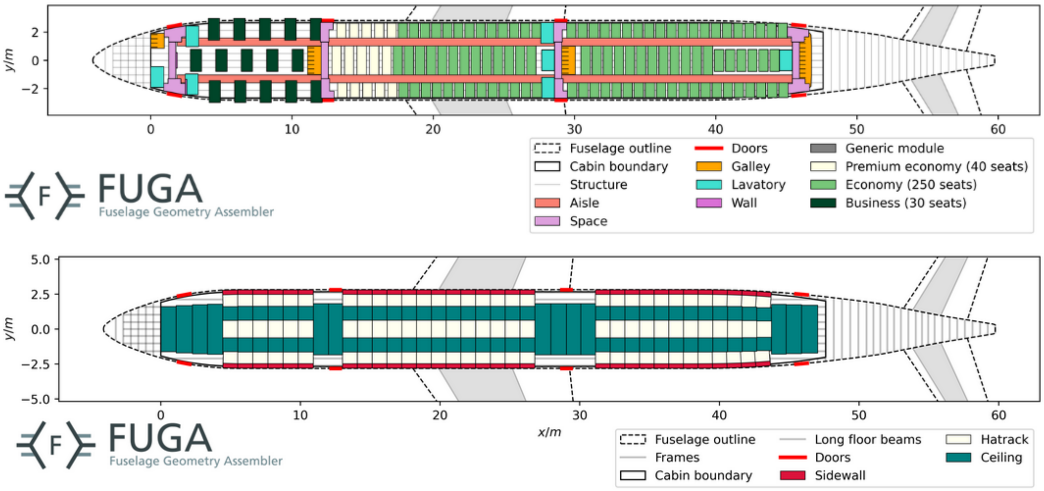


Figure 41: 3rd iteration of cabin layout configuration

5.2.2 Twin aisle configuration - 3D Cabin Layout

The integration of mid hatracks and the ceiling panels are also implemented in a 3D CAD model as seen in figures 42 and 43. The previous model was the original CAD cabin layout and the progressive update showcases the integration of mid-hatracks followed by ceiling panels which were seen in the final iteration 37. In the figures, it is also visible the correction of the lavatory in the 2nd iteration and the complete cabin design in the final iteration. From an external point of view, the positioning and sizing of both hatracks and the ceiling panels do not physically interact and are sized within the fuselage geometry. In the internal view of the cabin, the 3D model of the floating ceiling is vividly visible. The floating ceiling is then replaced with the hatracks in the right position and following the hatracks, the corrected ceiling panels are positioned.

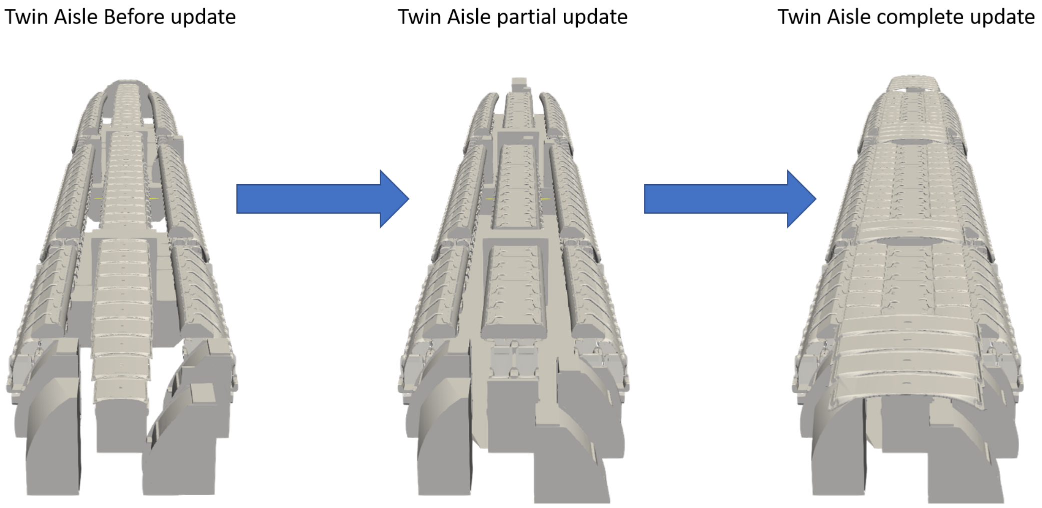


Figure 42: Overall ceiling improvement

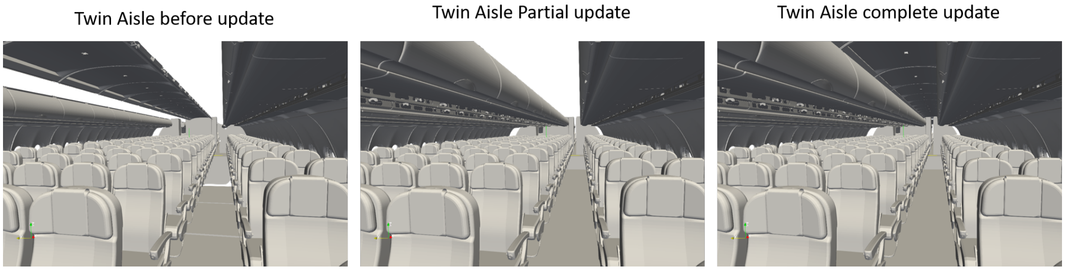


Figure 43: Overall ceiling linear improvement Interior perspective

5.2.3 Fuselage Geometry Assembler - Cargo Liner

The cargo liners which are shown in fig 44 illustrate the twin-aisle configuration with the wing box taken into consideration during the instantiation, it is seen as a space in between. The face walls of the cargo liners are open for visual representation. The structural view provides a skeleton view of the aircraft fuselage along with the placement of the liners. Though the liners are seen in OCCT view, the liners are converted into vtk format for analytical freedom.

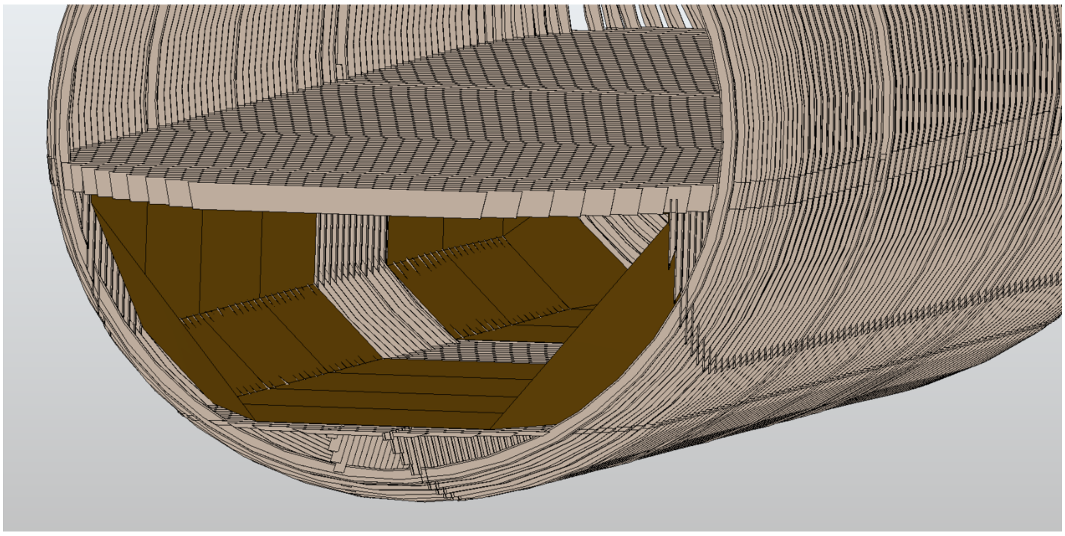


Figure 44: Cargo Liner for twin-aisle configuration - Structural view

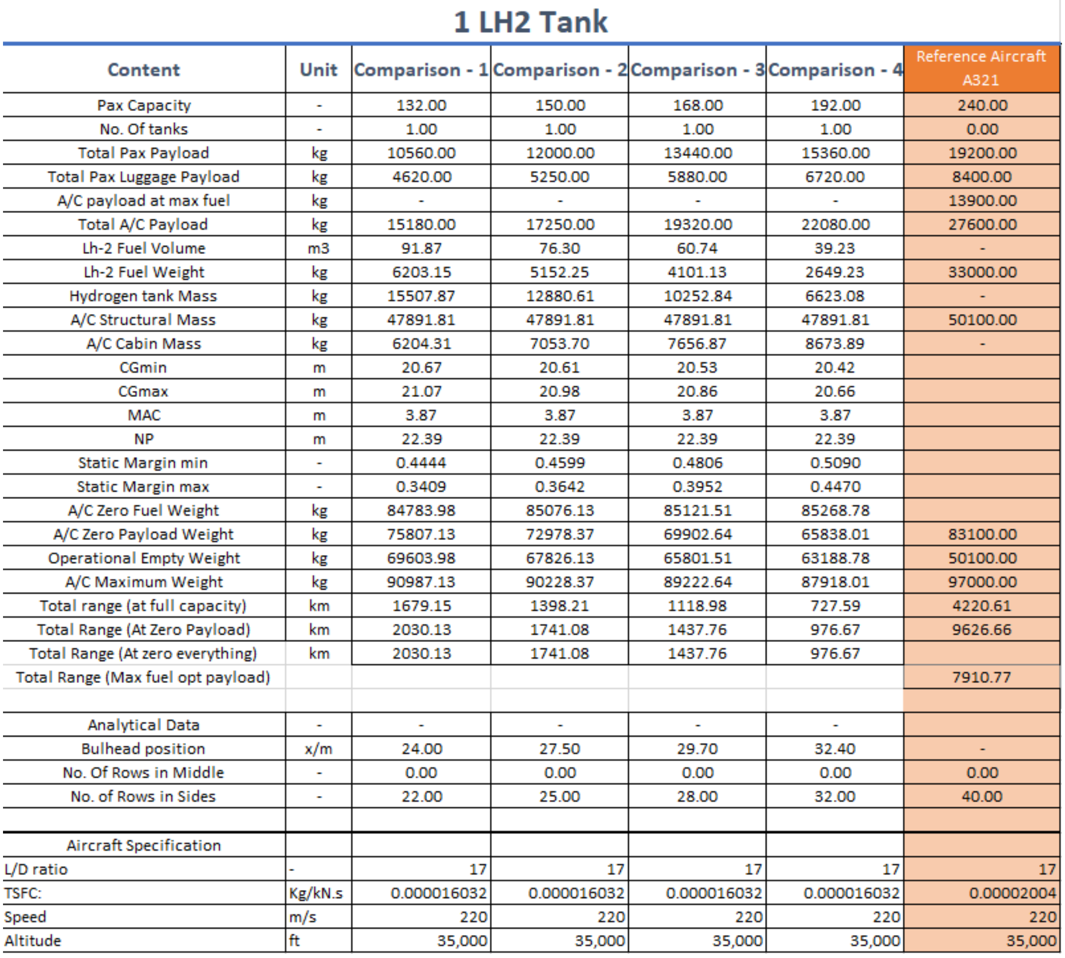
5.3 Study and Sensitivity Analysis of Liquid Hydrogen Tank

The results of the study and calculations for the liquid hydrogen tank integrated into single and twin-aisle configuration is explained in 3 divisions, separating them individually and cross-comparison between 2 different aisle configurations, keeping pax capacity the same.

5.3.1 Single Aisle Configuration

All the data, figures, and analyzed results on single-aisle configuration are illustrated in this section. The variation on the single-aisle configuration was made based on 4 different pax configurations (132, 150, 168, and 192) with variations in number of tanks (1 to 4 tanks) in all the configurations.

Data Sheet: figure 45 depicts the data collected for different pax configurations for a single aisle with fixed 1 tank configuration. Similarly, the data collected for other tank configurations can be found in Appendix section A.1.3 in figures 89, 90 and 91.



1 LH2 Tank

Content	Unit	Comparison - 1	Comparison - 2	Comparison - 3	Comparison - 4	Reference Aircraft A321
Pax Capacity	-	132.00	150.00	168.00	192.00	240.00
No. Of tanks	-	1.00	1.00	1.00	1.00	0.00
Total Pax Payload	kg	10560.00	12000.00	13440.00	15360.00	19200.00
Total Pax Luggage Payload	kg	4620.00	5250.00	5880.00	6720.00	8400.00
A/C payload at max fuel	kg	-	-	-	-	13900.00
Total A/C Payload	kg	15180.00	17250.00	19320.00	22080.00	27600.00
Lh-2 Fuel Volume	m3	91.87	76.30	60.74	39.23	-
Lh-2 Fuel Weight	kg	6203.15	5152.25	4101.13	2649.23	33000.00
Hydrogen tank Mass	kg	15507.87	12880.61	10252.84	6623.08	-
A/C Structural Mass	kg	47891.81	47891.81	47891.81	47891.81	50100.00
A/C Cabin Mass	kg	6204.31	7053.70	7656.87	8673.89	-
CGmin	m	20.67	20.61	20.53	20.42	
CGmax	m	21.07	20.98	20.86	20.66	
MAC	m	3.87	3.87	3.87	3.87	
NP	m	22.39	22.39	22.39	22.39	
Static Margin min	-	0.4444	0.4599	0.4806	0.5090	
Static Margin max	-	0.3409	0.3642	0.3952	0.4470	
A/C Zero Fuel Weight	kg	84783.98	85076.13	85121.51	85268.78	
A/C Zero Payload Weight	kg	75807.13	72978.37	69902.64	65838.01	83100.00
Operational Empty Weight	kg	69603.98	67826.13	65801.51	63188.78	50100.00
A/C Maximum Weight	kg	90987.13	90228.37	89222.64	87918.01	97000.00
Total range (at full capacity)	km	1679.15	1398.21	1118.98	727.59	4220.61
Total Range (At Zero Payload)	km	2030.13	1741.08	1437.76	976.67	9626.66
Total Range (At zero everything)	km	2030.13	1741.08	1437.76	976.67	
Total Range (Max fuel opt payload)						7910.77
Analytical Data	-	-	-	-	-	
Bulhead position	x/m	24.00	27.50	29.70	32.40	-
No. Of Rows in Middle	-	0.00	0.00	0.00	0.00	0.00
No. of Rows in Sides	-	22.00	25.00	28.00	32.00	40.00
Aircraft Specification						
L/D ratio	-	17	17	17	17	17
TSFC:	Kg/kN.s	0.000016032	0.000016032	0.000016032	0.000016032	0.00002004
Speed	m/s	220	220	220	220	220
Altitude	ft	35,000	35,000	35,000	35,000	35,000

Figure 45: Data Sheet of Variation of Pax capacity (132, 150, 168 and 192 Pax) for fixed tank configuration (1 Tank)

3D and 2D layouts of Single Aisle configuration with Hydrogen Tank: figure 46, 47 and figure 48, 49 shows the result obtained in 3D and 2D, In particular, the former set of figures depicts the variation of tank configuration in fixed pax capacity and the later set depicts the variation of pax capacity for fixed tank configuration. The 3D data or model was visualized in ParaView software and for better illustration of pax and tank configuration it was sliced. The other obtained models can be found in appendix section A.1.2 .

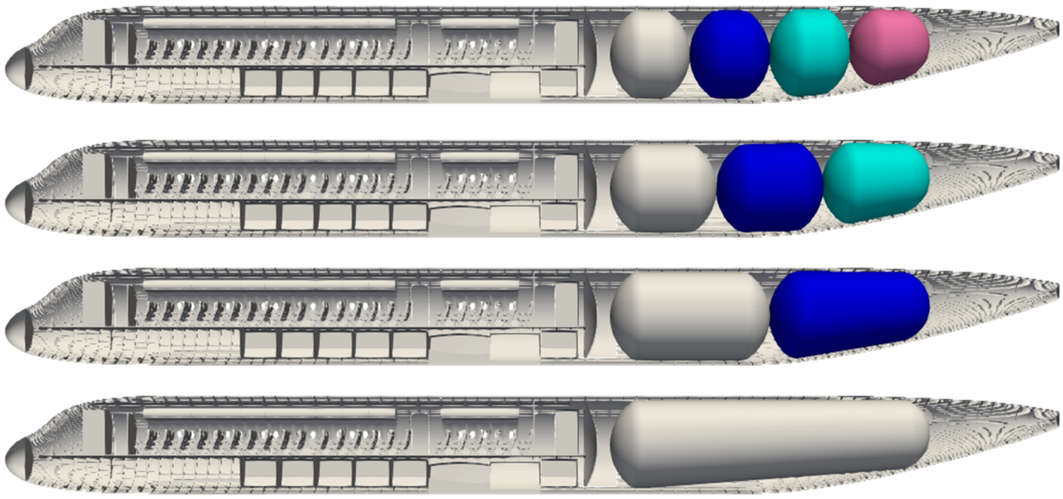


Figure 46: Variation of Tank Configuration for fixed Pax capacity (132 Pax) in 3D

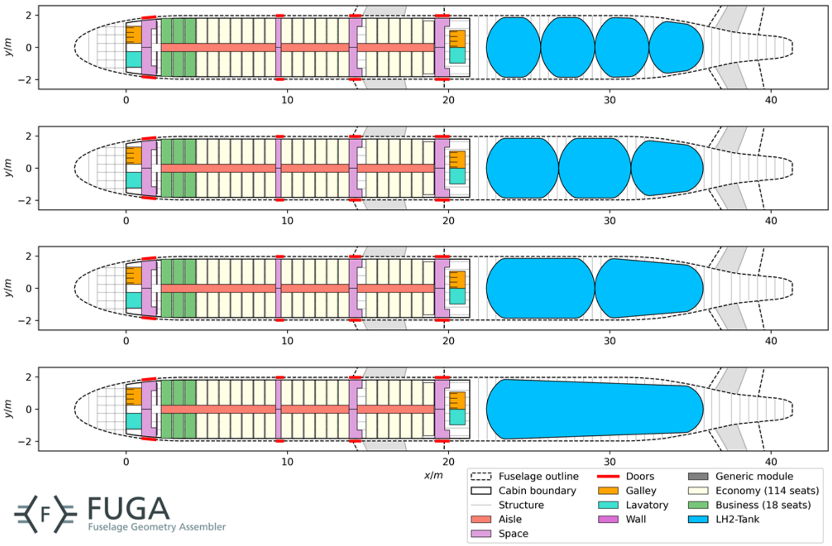


Figure 47: Variation of Tank Configuration for fixed Pax capacity (132 Pax) in 2D

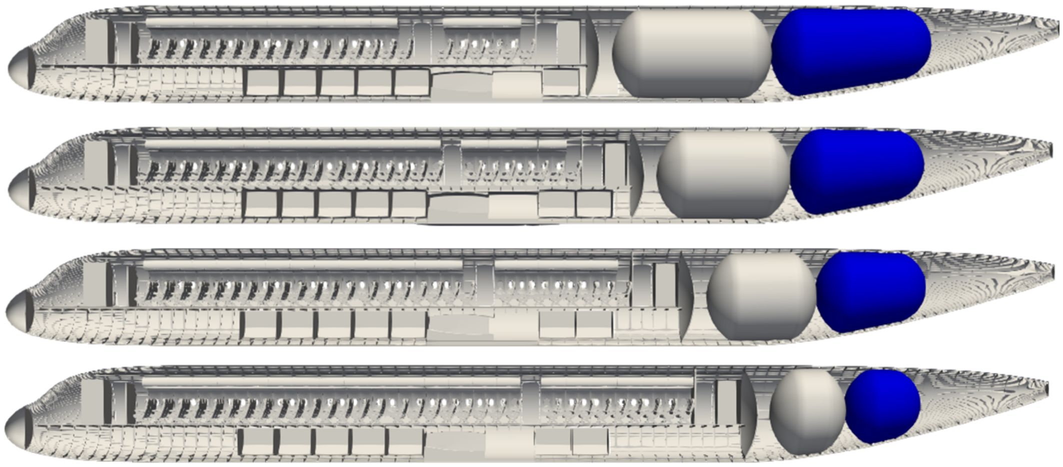


Figure 48: Variation of Pax capacity (Top to Bottom: 132, 150, 168 and 192 Pax) for fixed tank configuration (2 Tanks) in 3D

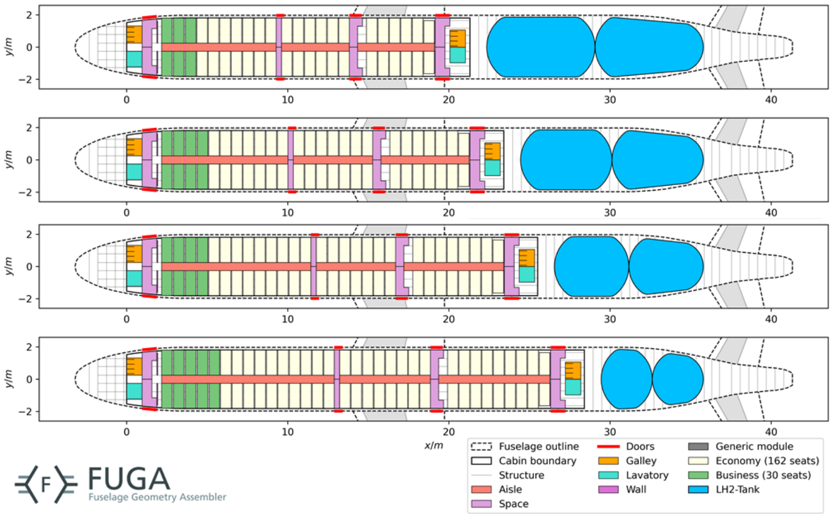
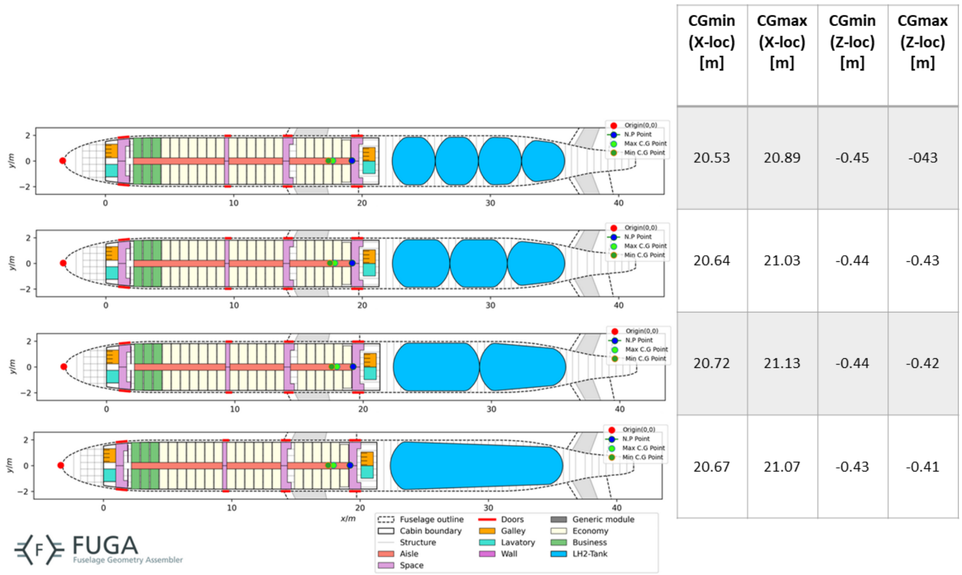


Figure 49: Variation of Pax capacity (Top to Bottom: 132, 150, 168 and 192 Pax) for fixed tank configuration (2 Tanks) in 2D

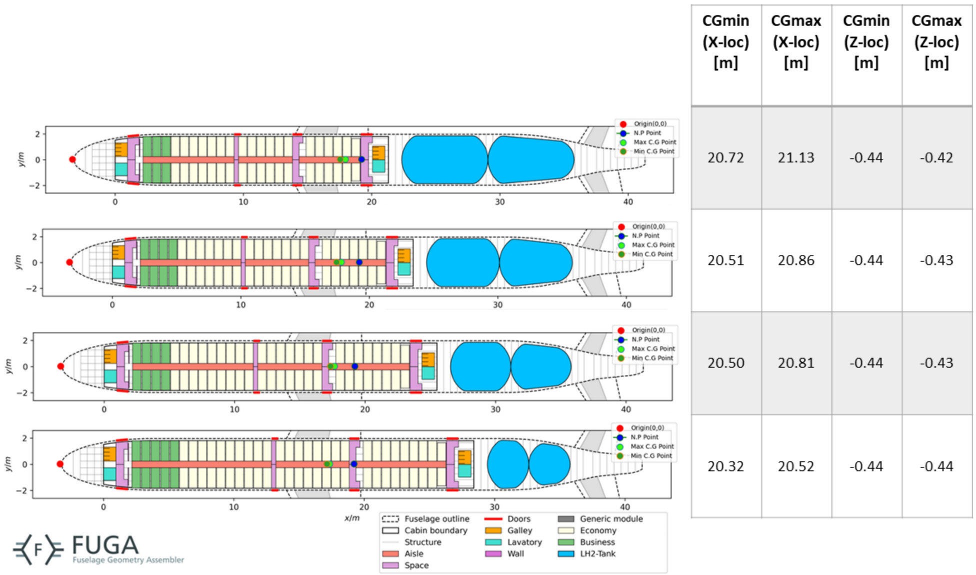
Center of Gravity Variation: When we consider a liquid hydrogen retrofit analysis in the aft region, the C.G variations play a major role in the stability of the aircraft. Two variations of results with CGmin and CGmax are shown in figure 50, which displays the result with constant pax capacity and variation of tank configuration, where it can be seen when the number of tanks is increased the $CG_{min/max}$ tends to decrease in the x-direction and figure 51 demonstrates the variation of same tank configuration with different pax capacity, however in this variation the difference between the $CG_{min/max}$ is comparatively low. The variation from CGmax to

CGmin is shown in Appendix section A.1.2 in figure 88.



CGmin (X-loc) [m]	CGmax (X-loc) [m]	CGmin (Z-loc) [m]	CGmax (Z-loc) [m]
20.53	20.89	-0.45	-043
20.64	21.03	-0.44	-0.43
20.72	21.13	-0.44	-0.42
20.67	21.07	-0.43	-0.41

Figure 50: Variation C.G for 132 Pax capacity in tank configuration (1, 2, 3 and 4 Tanks)



CGmin (X-loc) [m]	CGmax (X-loc) [m]	CGmin (Z-loc) [m]	CGmax (Z-loc) [m]
20.72	21.13	-0.44	-0.42
20.51	20.86	-0.44	-0.43
20.50	20.81	-0.44	-0.43
20.32	20.52	-0.44	-0.44

Figure 51: Variation of C.G for Pax capacity (132, 150, 168 and 192 Pax) in fixed tank configuration (2 Tank)

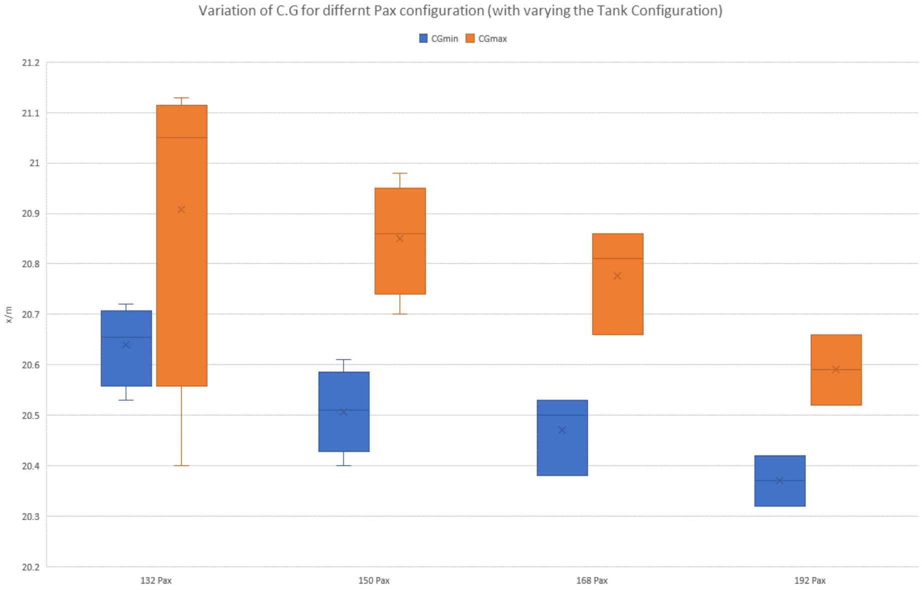


Figure 52: Variation of C.G for Pax capacity in 1,2,3 and 4 tank configurations

Payload Range: The Payload Range diagram for variation in Pax capacity (132, 150, 168 and 192 Pax) in fixed tank configuration (1 Tank) can be seen in figure 53, this figure also contains the reference aircraft of Airbus A320- Neo's payload range diagram. Additionally, figure 54 represents the change in range when tank configuration is changed for 132 pax capacity. The Payload range diagrams of other configurations are accessible in the appendix section A.1.3.

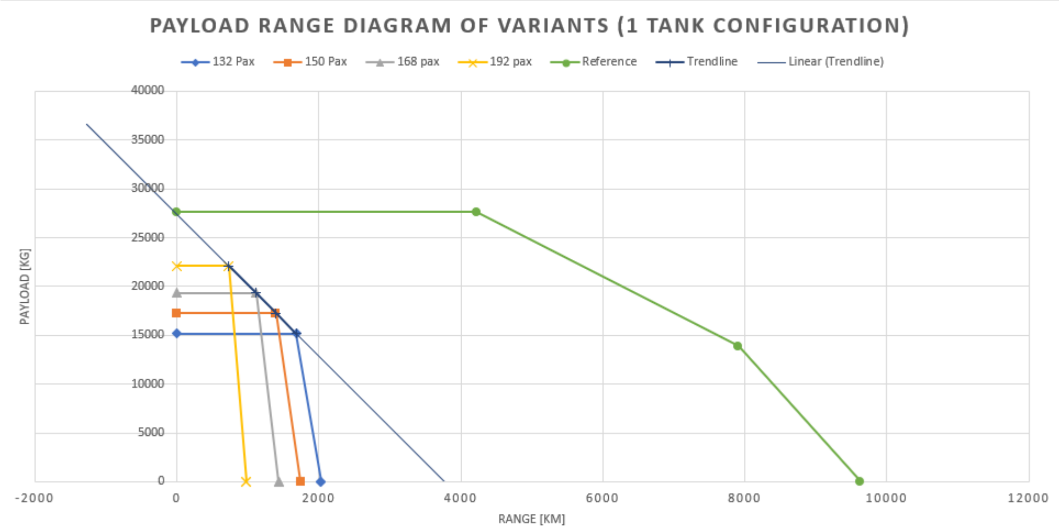


Figure 53: Variation of Payload Range Diagram for Pax capacity (132, 150, 168 and 192 Pax) in fixed tank configuration (1 Tank)

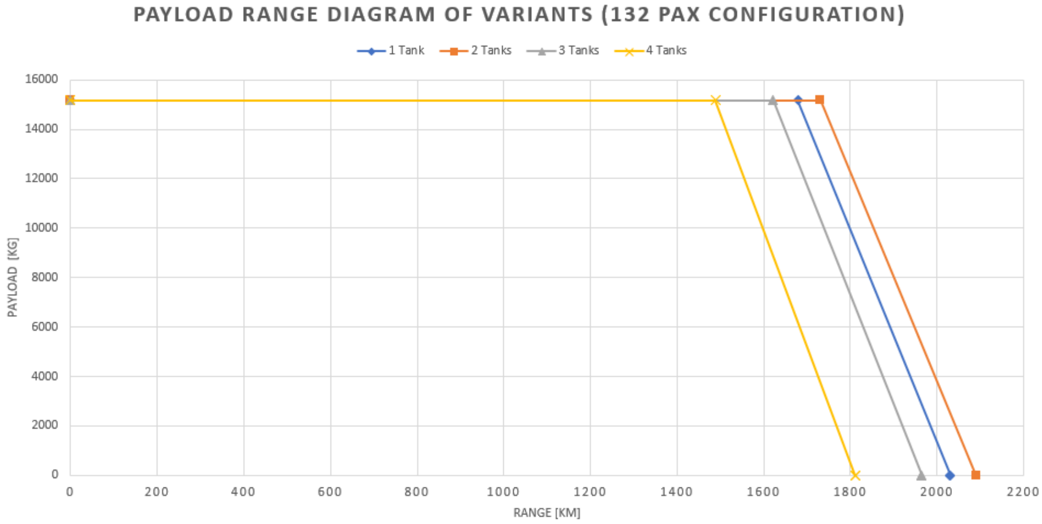


Figure 54: Variation of Payload Range Diagram for 132 Pax capacity in tank configuration (1, 2, 3 and 4 Tanks)

Co-relation Analysis: figure 55 depicts the variation of Total Fuel Weight for all the pax configurations but grouped according to several tank configurations, and on the secondary axis, the range is taken into account. It can be also seen in the reduction in range and fuel capacity or given pax configuration as the number of tanks is increased.

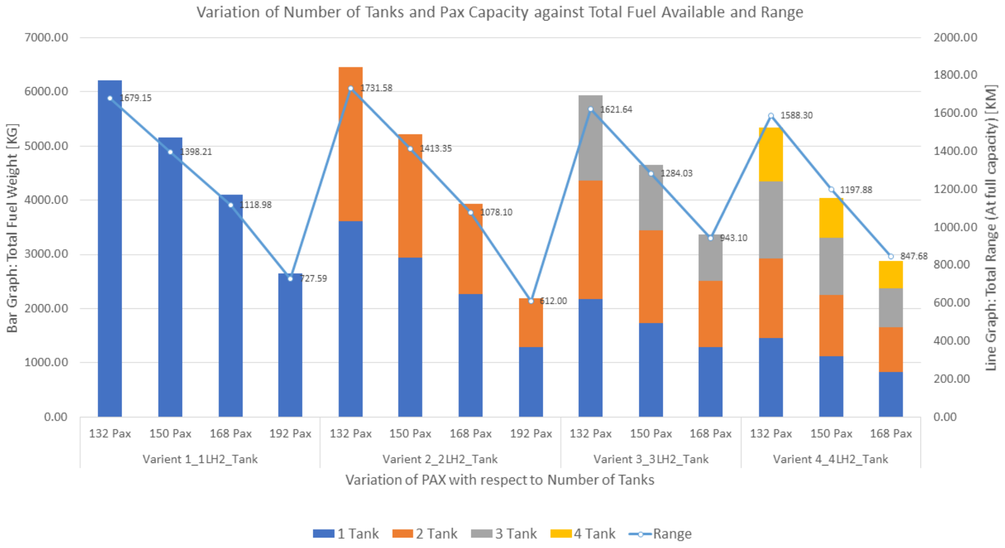


Figure 55: Variation of Number of Tanks and Pax Capacity against Total Fuel Available and Range

Weight distribution study with fuel weight, tank weight, payload, and structural weight as parameters is done on all the configurations, two sets of results, one with constant pax capacity and the other with constant tank configuration can be seen in figure 56 and figure 57 respectively, other comparison charts can be found in Appendix Section A.1.2.

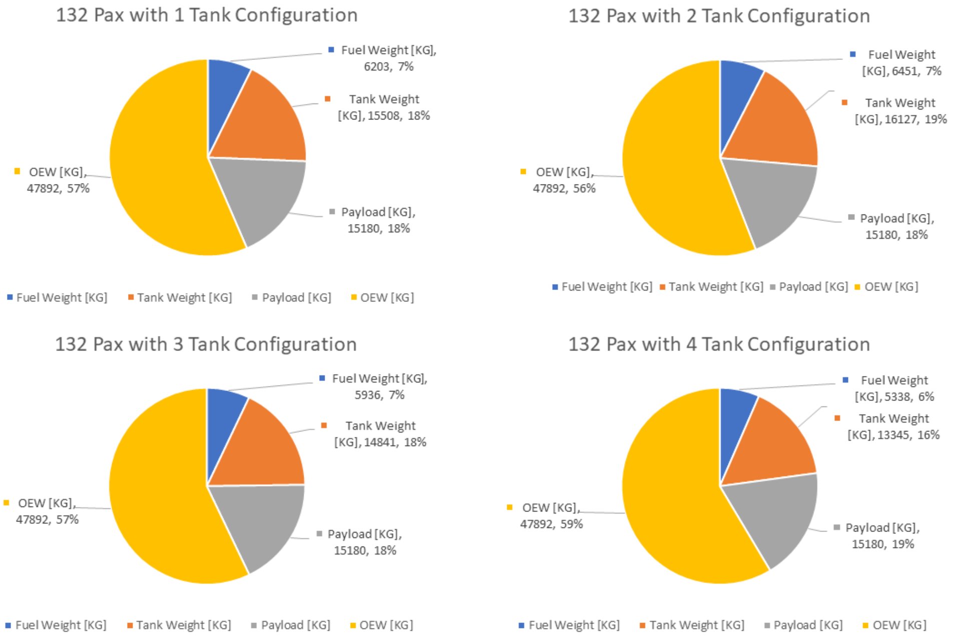


Figure 56: Weight distribution for 132 pax with different tank configurations

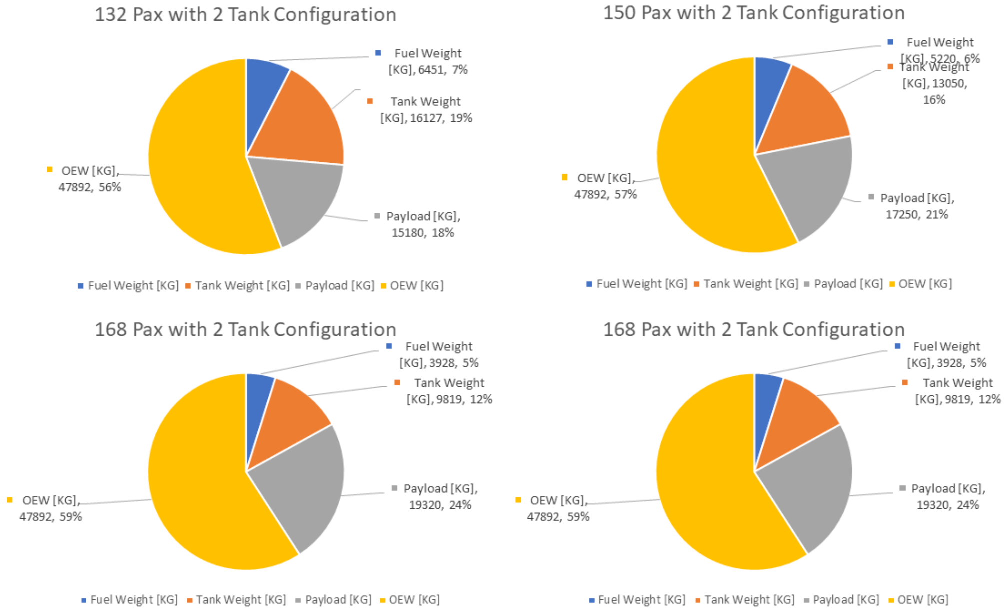
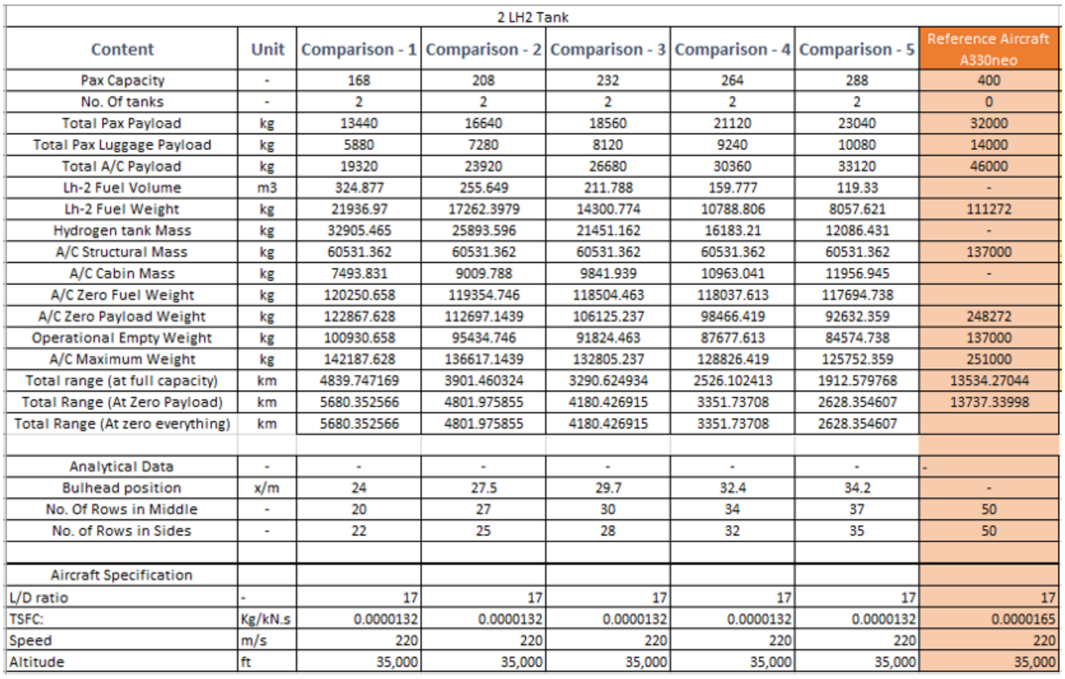


Figure 57: Weight distribution for 2 Tank Configuration with different pax configuration.

5.3.2 Twin Aisle Configuration

The analysis performed and the results obtained are shown in this section for the retrofitting of a liquid hydrogen tank in a twin-aisle configuration.

Tank Configuration Data
The total data collected for all passenger configurations taken into study for 2 tank configuration is displayed in figure 58. The data displayed and assumptions taken into consideration are displayed in the figure. The data collected for different tank configurations are accessible through the appendix A.1.3. in the Tank configuration data figure, it is observed that the total range of the aircraft decreases with an increase in passengers and the total cabin mass increases with an increase in passenger capacity. Additionally, the tank volume decreases with an increase in passenger capacity.

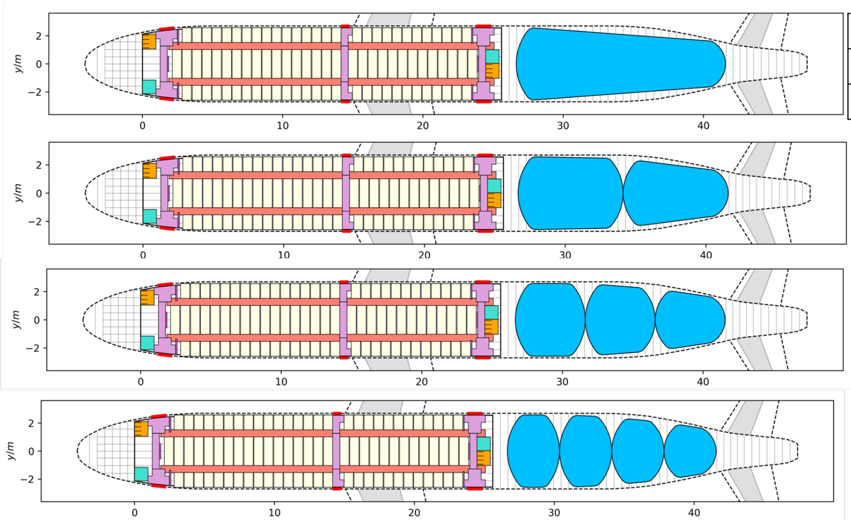


2 LH2 Tank

Content	Unit	Comparison - 1	Comparison - 2	Comparison - 3	Comparison - 4	Comparison - 5	Reference Aircraft A330neo
Pax Capacity	-	168	208	232	264	288	400
No. Of tanks	-	2	2	2	2	2	0
Total Pax Payload	kg	13440	16640	18560	21120	23040	32000
Total Pax Luggage Payload	kg	5880	7280	8120	9240	10080	14000
Total A/C Payload	kg	19320	23920	26680	30360	33120	46000
Lh-2 Fuel Volume	m3	324.877	255.649	211.788	159.777	119.33	-
Lh-2 Fuel Weight	kg	21936.97	17262.3979	14300.774	10788.806	8057.621	111272
Hydrogen tank Mass	kg	32905.465	25893.596	21451.162	16183.21	12086.431	-
A/C Structural Mass	kg	60531.362	60531.362	60531.362	60531.362	60531.362	137000
A/C Cabin Mass	kg	7493.831	9009.788	9841.939	10963.041	11956.945	-
A/C Zero Fuel Weight	kg	120250.658	119354.746	118504.463	118037.613	117694.738	
A/C Zero Payload Weight	kg	122867.628	112697.1439	106125.237	98466.419	92632.359	248272
Operational Empty Weight	kg	100930.658	95434.746	91824.463	87677.613	84574.738	137000
A/C Maximum Weight	kg	142187.628	136617.1439	132805.237	128826.419	125752.359	251000
Total range (at full capacity)	km	4839.747169	3901.460324	3290.624934	2526.102413	1912.579768	13534.27044
Total Range (At Zero Payload)	km	5680.352566	4801.975855	4180.426915	3351.73708	2628.354607	13737.33998
Total Range (At zero everything)	km	5680.352566	4801.975855	4180.426915	3351.73708	2628.354607	
Analytical Data	-	-	-	-	-	-	-
Bulhead position	x/m	24	27.5	29.7	32.4	34.2	-
No. Of Rows in Middle	-	20	27	30	34	37	50
No. of Rows in Sides	-	22	25	28	32	35	50
Aircraft Specification							
L/D ratio	-	17	17	17	17	17	17
TSFC:	Kg/kN.s	0.0000132	0.0000132	0.0000132	0.0000132	0.0000132	0.0000165
Speed	m/s	220	220	220	220	220	220
Altitude	ft	35,000	35,000	35,000	35,000	35,000	35,000

Figure 58: Total Calculation data collected for 2 tank configuration in Twin aisle

Tank Integration
The 2D cabin view of the liquid hydrogen tank integrated into the twin-aisle configuration is seen in figure 59. This 2D view focuses on variants of tank configuration from tank 1 to tank 4 for 232 passenger capacity. Further detailed analysis for different passenger capacities is provided in appendix A.1.3. It is observed that the total tank volume for 2 tank configuration has greater than 1 tank and others, this is due to the linear loft line from 2 ends of the tank since this does not take into consideration the curvature of the fuselage at the aft region, the volume that occupies the curvature of the fuselage is not utilized by the liquid hydrogen tank.



Tank Weight	20106.83 kg
Fuel Volume	198.51 m^3
Fuel Weight	13404.55 kg

Tank Weight	21451.16 kg
Fuel Volume	211.78 m^3
Fuel Weight	14300.77 kg

Tank Weight	20239.56 kg
Fuel Volume	199.82 m^3
Fuel Weight	13493.04 kg

Tank Weight	18710.86 kg
Fuel Volume	184.73 m^3
Fuel Weight	12473.91 kg

Figure 59: Cabin top view with 232 pax capacity and variation in tanks

The 3D view of tank integration into 232 passenger capacity is illustrated in fig 60. The 3D view incorporates external fuselage, internal cabin design, positioning of bulkheads, and generated tank configurations.

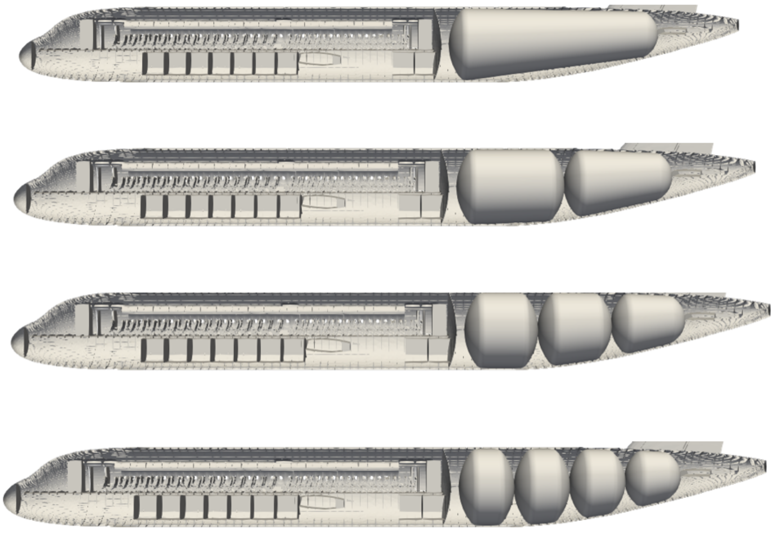
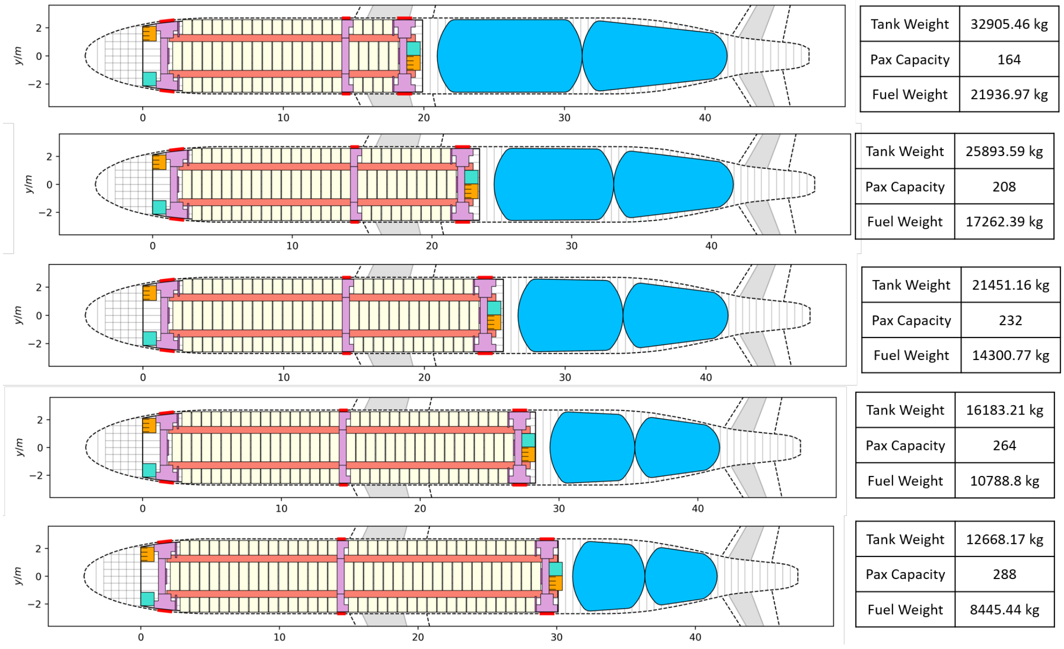


Figure 60: Cabin 3D view with 232 pax capacity and variation in tanks

The 2D view of the transition from 168 passenger capacity to 288 passenger capacity is seen in figure 61. 2 Tank configuration is set constant for the study.



Tank Weight	32905.46 kg
Pax Capacity	164
Fuel Weight	21936.97 kg

Tank Weight	25893.59 kg
Pax Capacity	208
Fuel Weight	17262.39 kg

Tank Weight	21451.16 kg
Pax Capacity	232
Fuel Weight	14300.77 kg

Tank Weight	16183.21 kg
Pax Capacity	264
Fuel Weight	10788.8 kg

Tank Weight	12668.17 kg
Pax Capacity	288
Fuel Weight	8445.44 kg

Figure 61: Cabin top view for different pax configuration with 2 tank configuration

Payload Range Diagram

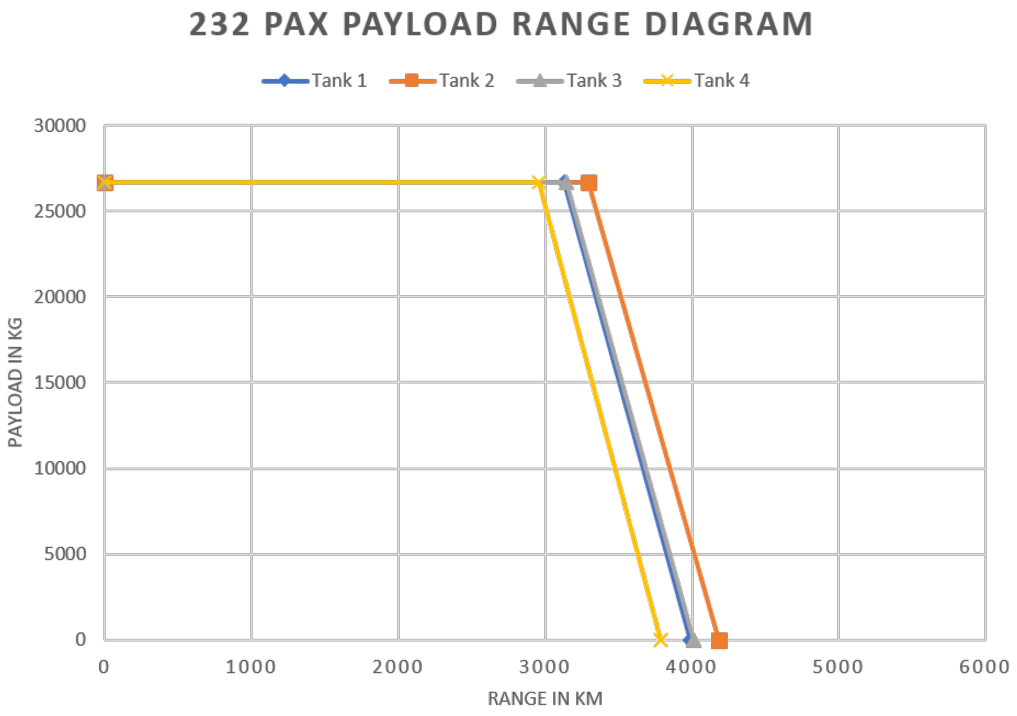


Figure 62: Payload Range diagram of 232 pax capacity

The payload range diagram of the Twin aisle configuration for the 232 passenger variant is depicted in fig 62. The maximum range for the corresponding tank configuration is provided in this figure. The payload range diagrams for other passenger configuration is provided in the appendix A.1.3. The constant tank configuration of 2 tanks is considered and the passenger configuration is varied in figure 63. This figure also contains the reference aircraft of Airbus A330-Neo's payload range diagram. The Payload range diagrams of other configurations are accessible in the appendix A.1.3.

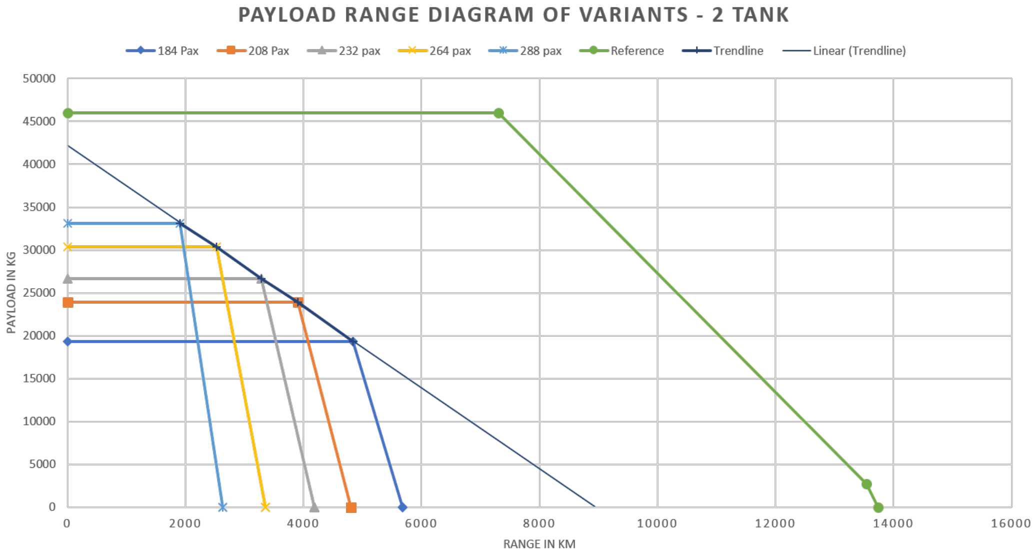


Figure 63: Payload Range diagram of 2 number of tanks

Center Of Gravity

The Center of gravity shifts with different passenger configurations to the tank configuration are depicted in figure 64. The shift of C.o.G for each passenger capacity is displayed inside the figure with 168 passengers to 288 passengers as the line progresses from down to up.

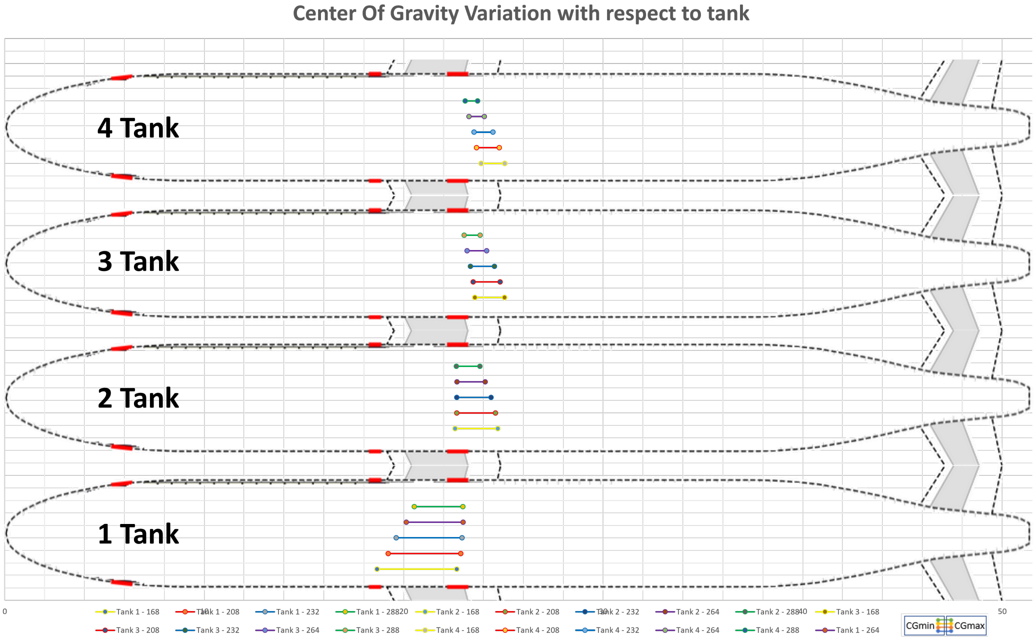


Figure 64: Center of Gravity variation concerning Tank

Fuel weight to Range concerning tank quantities

figure 67 shows the total fuel weight of each tank configuration and the individual weight of each tank. The secondary axis of the range is also displayed in the figure to display the relation between fuel weight and the total range of the aircraft.

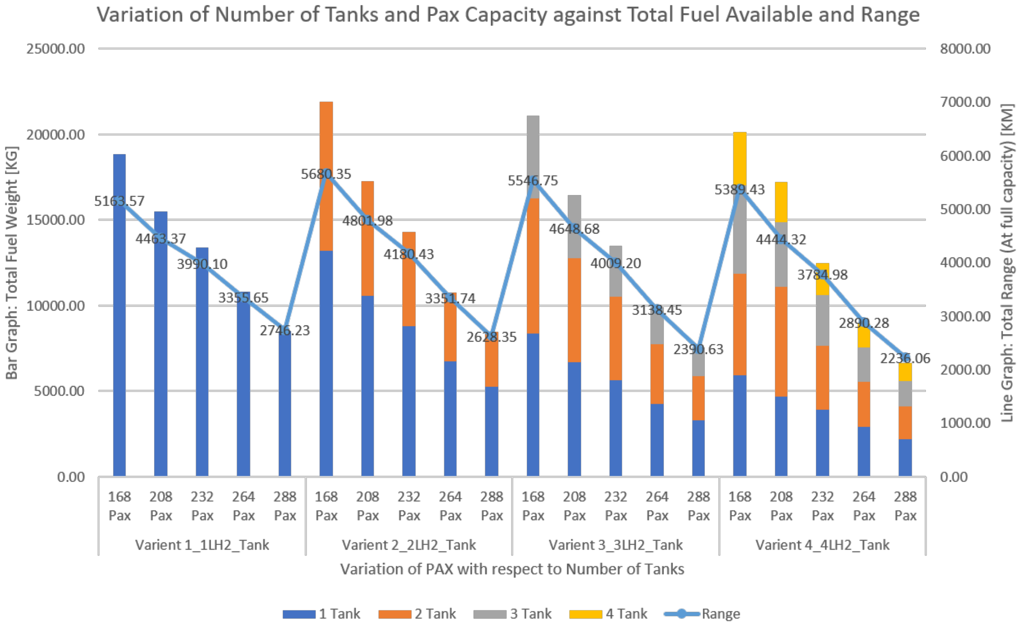


Figure 65: Variation in Range concerning Fuel weight due to tank quantities

Total Weight Distribution of aircraft

The total weight of the aircraft for 232 passenger capacity for different tank configurations is displayed in figure 66. The weight distribution of different passenger capacities is available in the appendix A.1.3.

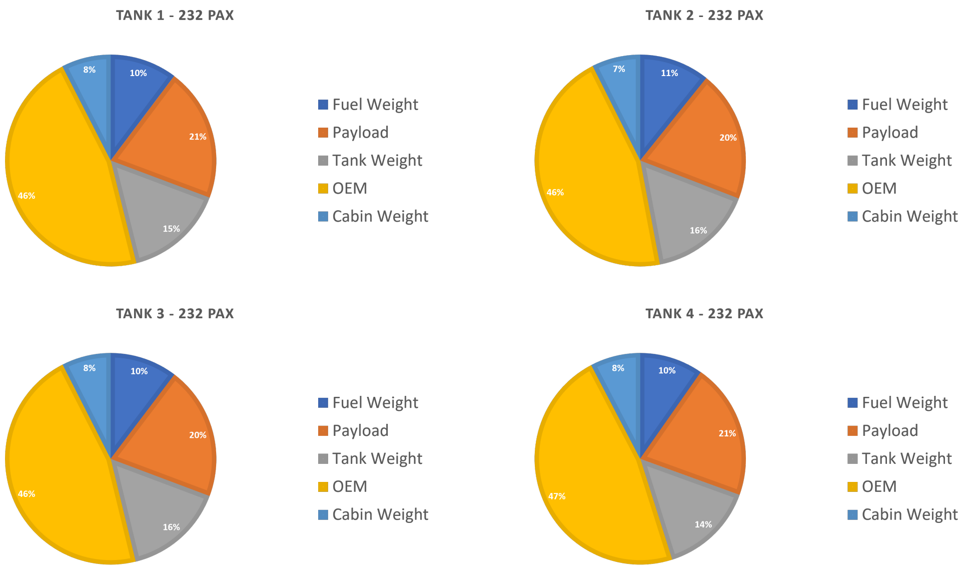


Figure 66: Weight distribution for 232 pax with different tank configurations

The total weight distribution of aircraft for 2 tank configurations and different passenger capacities is displayed in figure 67.

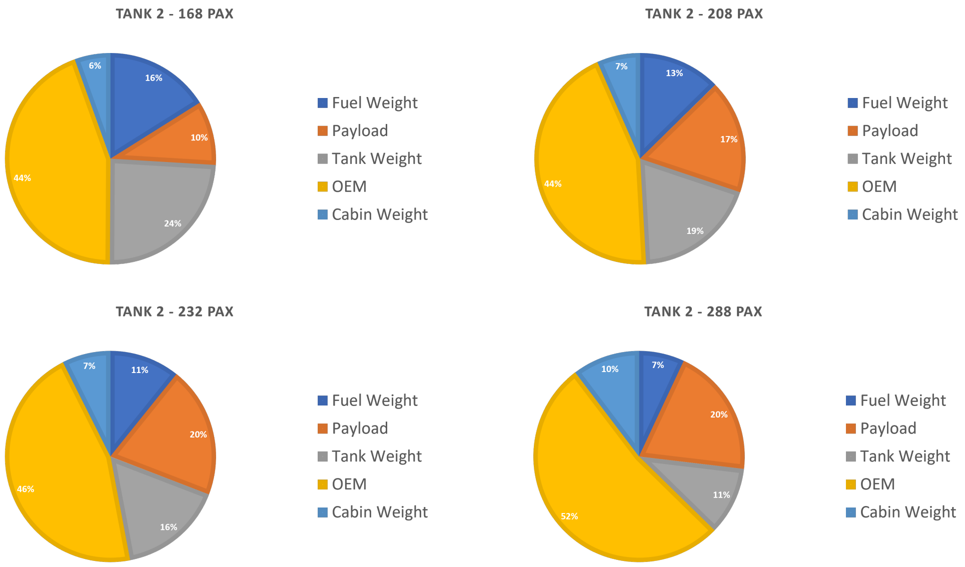


Figure 67: Weight distribution for 2 tank configuration and different pax capacities

5.3.3 Aisle Configuration Comparison

Cross-comparisons between different aisle configuration results are shown in this section. This cross-comparison shows crucial information about the potential difference between single and twin-aisle configurations as well as the benefits of liquid hydrogen tanks for aisle configurations as their own.

Center Of Gravity Comparison between Aisle Configuration

The center of gravity shift comparison requires similar passenger configurations for both single and twin-aisle. Hence a pax capacity of 168 is taken into consideration for the analysis. The change in CG for each tank configuration is shown in figure 68.

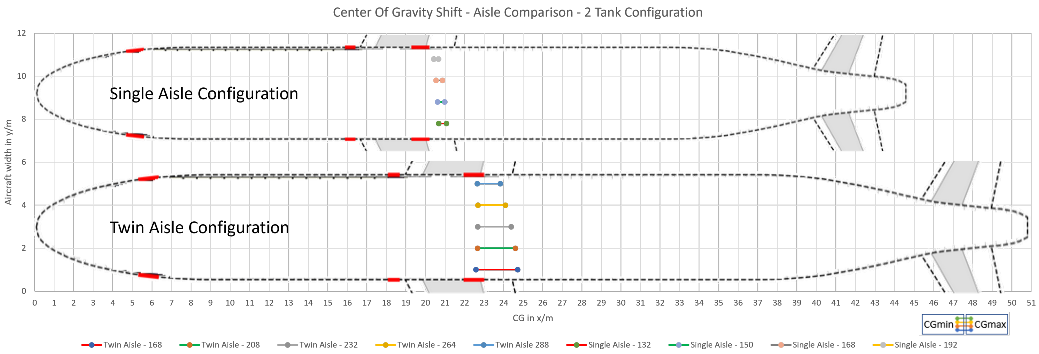


Figure 68: Center Of Gravity shifts between tanks in 168 pax capacity

Payload Range Difference between Aisle Configuration

The payload range diagram for single and twin-aisle configurations for the same passenger capacity of 168 is shown in figure 69. This also takes into consideration the different tank configurations.

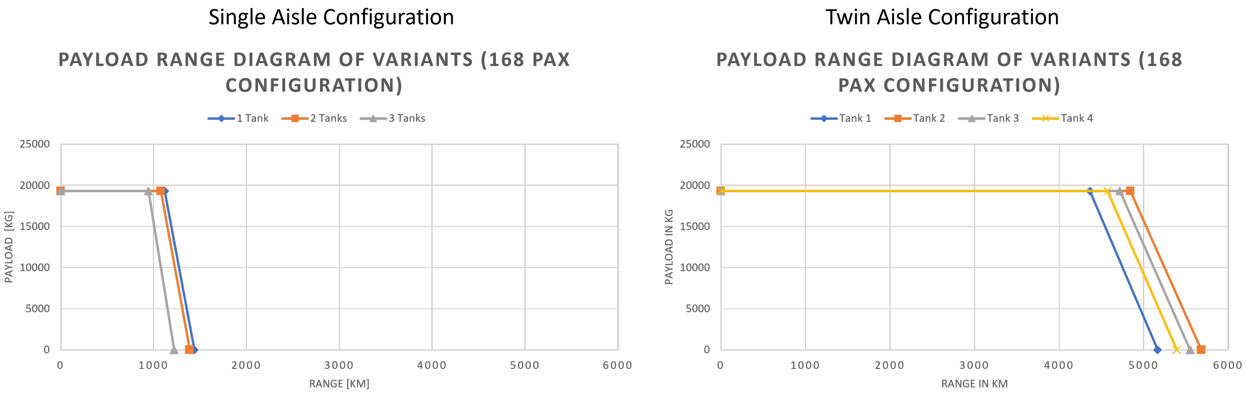


Figure 69: Payload range of aisle configurations for 168 pax capacity

Weight Comparison for aisle configuration

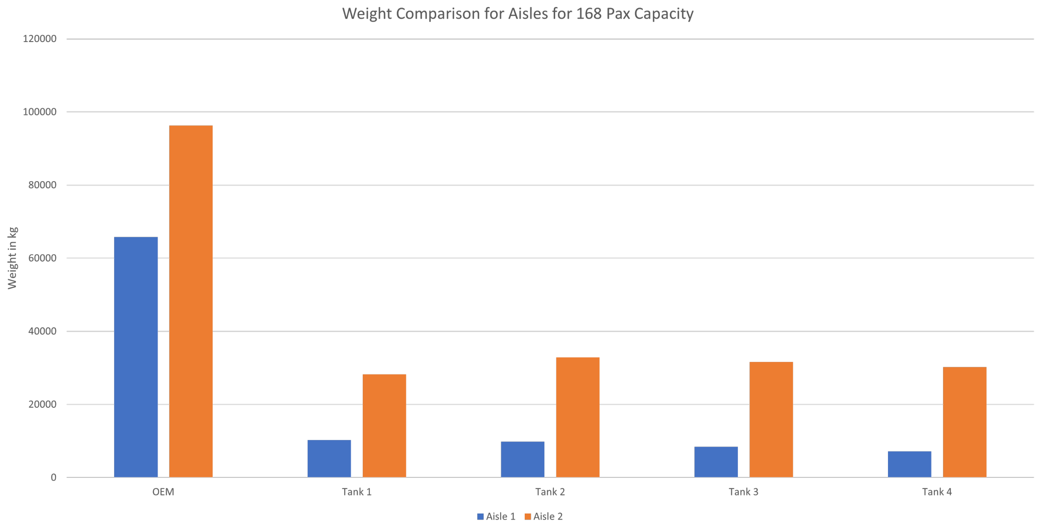


Figure 70: Aircraft Weight and tank quantity comparison for aisle configuration

The difference in total aircraft Operational Empty Weight and individual tank weights are shown in figure 70. The total weight of each tank configuration is compared with its adjacent aisle configuration

6 Discussion

This section covers the various aspects that the results could take shape depending on the parameters that are changed. The discussion of different outcomes provides a broader scope to the tasks that were done.

6.1 Implementation of Web Application for FUGA

After an evaluation of different available Python-based web frameworks, the Flask proved to be the most feasible framework to opt for this work. This was mainly based on the 3 main evaluation factors referring to the table 2,

Modularity: this makes the architecture more dynamic, for example, this architecture has a separate module for receiving a file, JSON file updating (for cabin description), and running the main FUGA code based on the aisle configuration, having such different modules is easier to route it to the required module in Flask and also have its return type, Which makes the code less complex.

Ease of learning: Flask has extensive support and compatibility when implemented in a tool or application. For example, Flask has a straightforward syntax, documentation, and a large user community, which makes it easy to implement Flask.

Lightweight and flexible: When working with the web app, integrating different components into the framework plays a factor as also the communication of the components with Flask. To also deploy the web app in any server with all the effectiveness also plays an important role.

Also, another thing that makes it more dynamic is its passing the values between different modules, which makes it easier to get any values or activation values to different modules, additionally having a temporary-static storage system, makes data storage and data management more efficient.

Making the architecture more feasible by having a default CPACS file and a cabin description, where the user can just run the tool to see the output when the 'Submit and Start' button is accessed as seen in figure 32 and figure 33.

Having such a web-based application for the tool makes it more reachable to the other groups within the organization and also can be used for external collaborations.

Additionally having a Clash detection module integrated will make the model more robust and have a realistic approach, initially, when the FUGA tool is finished with initializing the design, we take just the eat mesh and the inner fuselage mesh, later use the VTK collision detection filter the collision of the fuselage is obtained on the seats as represented in the last row in figure 36. The model is then realized with the help of ParaView software. The effectiveness of seat collision detection depends on the mesh of the seat and fuselage modules, it can be seen the figure 36 in column 2 and column 3 figure, the column 2 figure has a clear definition of the fuselage curve on the seats this is due to the good meshing size of the seat, on the other hand when we see the result in the 3 column we see a spread-out contact cell definition due to the poor meshing of the seat. It must be also noted that if the meshing of both seat and fuselage is of superior quality, the time taken to execute the collision detection is longer.

6.2 Twin Aisle Configuration Capabilities

The introduction of cabin components and enhancement of existing components provides various possibilities for improvement for cabin virtualization in the future. The rule sets and design being parametric provides versatility and options to change position and CAD models.

The modifications in ceiling panels briefed in section 5.2.2 into the 3D cad model opens up various levels of possibilities for the cabin design system of twin-aisle configuration. These changes implemented have produced a closed cabin environment that can be visualized through either, virtual reality or augmented reality. The 3D model of the cabin design system obtained from the FUGA tool, not only have been capped with virtualization but is also utilized for multiple analytic purposes thanks to the utilization of VTK and OCC as explained in sections 3.3 and 3.2. This mesh ability and multiblock set capability of cabin designs constructed in the FUGA tool enables the cabin model to undergo mass distribution, stress analysis, acoustic analysis, etc.

The middle overhead luggage compartment is configured to be parametric, hence any changes that are implemented in the configuration of seat numbers, pax capacity, and abreast, the overhead hatracks instantiate and adapt to the changes accordingly. The ceiling panels are parametric in nature as well as it is dependent upon the sidewall panels at the given panel position frames. In the absence of sidewall panels, the code automates the conversion from twin ceiling panels to a single panel that covers the whole roof. This provides flexibility and ease of operation for the configuration. An accurate model of twin-aisle configuration provides a new opportunity for the simulation of cabin-related analysis and comparison with different aisle configurations. This would benefit the analysis of wide-body aircraft. Multiple simulation data readings can also be taken with a complete configuration being presented.

The cargo liner is implemented for both single and twin-aisle configurations so that the panels for the cargo liners can be taken into consideration for mass calculation and analytical purposes. The liners are parametric, considering the total aircraft length owing box and the position of the landing gear and wing box. The liner instantiates only on the positions where the wing box is not located and it does not extend outside the frames, beams, and beam struts. In the case of the introduction of a hydrogen tank, the liners are modified to not extend beyond the wing box region for the allocation of liquid hydrogen tanks. Hence it is also partially compatible with LH2 tank integration models. The cargo liners also take into consideration the thickness profile of the structural components and reduce component interactions and non-feasibility. The liner intervals can also be increased or decreased to reduce computational time, but this affects the individual panel weight ratio and calculation when considering each panel. The cargo liners are instantiated in VTK format, hence it can be used to mesh and be taken into consideration for further analysis in weight as well as visual analysis.

6.3 Liquid Hydrogen Tank Retro-fit to existing fuselage

The cost of designing, simulating, and testing an aircraft design modeled around a liquid hydrogen tank integrated into the design requires time, manpower, and extensive financial strength. One of the practical alternative methods for testing the feasibility of liquid hydrogen tanks in the aviation industry is through retrofitting a hydrogen tank into the fuselage of existing commercial aircraft. Airbus A320-Neo configuration, being the top-of-the-line model in the current market for single-aisle configuration, and A330-Neo for twin-aisle configuration is taken as reference aircraft for the study. These 2 configuration would be suited to be an optimal candidate for retrofit thanks to their popularity and extensive availability throughout the European and Asian aviation market.

Some of the aircraft performance parameters are neglected such as the involvement of lift surfaces into account for the study. This is due to the primary focus on aircraft cabin parameters and fuselage structure. Since the FUGA tool is primarily built around the enhancement of fuselage components and cabin design, a study related to the performance variation in this window is highlighted in this study. Since different configurations were analyzed and resulted in producing different outcomes. Each aisle configuration would be further discussed in detail with the noted results.

6.3.1 Single Aisle Configuration

In the context of retrofitting hydrogen tanks being placed in the aft region of the aircraft, many critical factors must be accounted for. For the same single-aisle configuration, the pax configuration was varied with 132, 150, 168, and 192, and the tank configuration was varied between 1 to 4 tanks the datasheet for all the configurations can be seen in figure 45, 89, 90 and 91 and analysis of the critical factors are as follows,

Primarily starting with the Center of Gravity (C.G) variation, CGmax, and CGmin are measured when the hydrogen tank is full and when there is no fuel in the tank, the visual representation of the shift in the CG from the max to min position can be seen in figure 50 and 51, the Static margin Measured between the Neutral Point and CG variation can be referred in the datasheet.

The variation of the CGmax and CGmin is visualized in a box-whisker plot in figure 52, it shows that CGmax of 132 pax has a higher band (max and min value of the whisker) variation, this is because of the weight contribution from the hydrogen tank, the variation of the total tank weight (without fuel) can be seen in figure 50. Even though there is a decreasing trend in the CGmin variation, the percentage difference in the CGmin is around 0.97 percent.

CGmin is measured when there is no fuel and in most conventional aircraft designs the CGmin trend is to move much closer toward the neutral point, but in this case, due to the aft location of the fuel tank, it can be seen that the CGmin moves forward or further away from the neutral point. The Static Margin (SM) min and max measured (expressed as a percentage of MAC) can be referred to in the datasheet, consider 132 Pax capacity and the tank is varied from 2 to 4, there is a gradual increase in SMmin from 44.4% to 48.06% and the same can be seen in SMmax from 34% to 38.75%. Additionally, the lowest variation in SM in all the comparisons was seen in Pax capacity having a two-tank configuration and also the two-tank configuration can be preferred over the single tank due to two reasons, one mainly being the redundancy of the system, the second being the decrease in the sloshing when we consider more tanks.

On the other hand, considering more tanks for better redundancy of the system and the reduction of the sloshing effect in the system is not always the best solution, this is illustrated in figure 55, the figure shows the available total tank fuel and total range (at full load) for an increase of the number of tanks in a particular pax capacity, it is seen that with an increase of tank configuration, there is a decrease of fuel capacity and thereby decrease of the available range, this is mainly due to structural space occupied by the tank as you increase it. Additionally, the percentage dip or difference in range when you consider 2 to 3 tanks is 6.55% and 3 to 4 tanks are 2.07%, thereby the 2-tank configuration seems a more feasible option.

When it comes to the study on the range, the Payload range diagram would best illustrate the variation of range according to the payload, for Single-aisle configuration, Airbus A320neo was taken as the reference aircraft and the variation can be seen in figure 53, the rest can be found in the appendix section A.1.2, the percentage difference between the range at full capacity/load (for 132 pax) is around 128% or the there is the reduction of 78% of the range from the reference aircraft when the aircraft is retrofitted with hydrogen tank.

When the comparison of the range is conducted between the Pax configuration, the 192 and 168 Pax configurations would be the ones with the least range available due to the space available for the tank to be integrated. On the other hand when you do an intra-comparison between the range available in particular pax, an example of which is shown in figure 54, it is seen that the 2 tank configuration demonstrates the higher range capability, making it the more feasible option in all the pax configuration.

Weight distribution plays a major role, the inter and intra-comparison of results can be seen in figure 56 and 57. Consider 132 pax with different tank configurations as shown in figure 56, it is seen that the total Tank weight gradually decreases from 2 to 3 tanks, this is because when more tanks are considered, with increasing tanks there is an increase of non-usable space between the two tanks which can be referred as the dead space between the tanks, it contributes to two things, one being the decrease in the tank weight and other is the decrease in the fuel weight, in turn leading to decrease in range as discussed above.

6.3.2 Twin Aisle Configuration

The variation in performance and ability of the aircraft deviates noticeably in the twin-aisle configuration when compared to the single-aisle configuration. This behavior is due to longer fuselage length and larger diameter. This results in more volume inside the fuselage. In this section, a brief discussion about the performance character in the twin-aisle configuration due to retrofitting of the hydrogen tank and quantities of the tank will be seen.

The cumulative fuel weight that the aircraft can carry reduces as the number of liquid hydrogen tank quantities increases. This is observed due to the shape that the tank is made of. The optimized shape for a liquid hydrogen tank is a sphere that has no sharp edges which is not desirable for high pressure. Since the tank has smooth spherical ends, this occupies spaces at the ends and this reduces the total usable space for storing LH2 fuel. Additionally, it increases material consumption and the total weight of LH2 tanks as seen in figure 59. A noticeable change in the pattern is seen in the figure where the 1 Tank configuration has a lower tank weight by 6.68 % than LH2 2-tank configuration. This is due to the method in which the program is written in the FUGA tool where the fuselage cross-section of the ends of the tank is considered and lofted. This neglects the usable volume surrounding the loft that makes the curvature of the aft section of the fuselage. Hence lowered tank volume and weight. It is desirable for having more than 1 tank configuration, primarily for redundancy and safety. The study was conducted for different passenger configurations of 168 to 288. The most optimum passenger configuration chosen was 232. This is due to sufficient place provided for retrofitting of liquid hydrogen tank into the aircraft fuselage for the reference aircraft where the fuel volume for 2 tank configuration is greater than 1 tank configuration by 6.68 %. This configuration is not always optimum for other reference aircraft and will be subject to change based on the fuselage design of the reference aircraft.

The Total weight of the aircraft, as well as the range of the aircraft, differs with change in passenger capacity as seen in figure 61. 2 Tank configuration is decided as the optimum tank configuration due to the redundancy it provides followed by maximum range for the redundancy due to lower occupation of fuselage space and weight. With the increase in passenger capacity to 232, the space available for the liquid hydrogen tank reduces by 20.70 % with an increase in passenger by 10.3% when compared to 208 passenger capacity. This reduces the total range of the aircraft. In contrast, with the reduction of the LH2 tank, the total tank weight also reduces by 20.7 %, but this is compensated by an increase in cabin component weight and passenger weight. Hence, providing a maximum range of 164 passengers and a minimum range of 288 passengers, the optimum pax capacity chosen for this configuration is 232.

With different passenger capacities and tank configurations studied. The range for the configuration with constant passenger capacity increases with the increase in the number of tanks. This is seen in figure 62. With the increase in tank quantity, the total usable liquid hydrogen fuel weight reduces. This in turn reduces the range of the aircraft. The range of the aircraft is further reduced due to an increase in liquid hydrogen tank weight. This also further explains that 2 tank configuration is best opted for as the maximum range for the provided payload by up to 4.5 % compared to 1 tank configuration and 4 % more than 3 tank configuration. The study of different passenger capacities for a constant tank configuration is seen in figure 63. In this figure, the reference aircraft's payload range diagram is also illustrated. The payload range for the diagram for the liquid hydrogen tank does not exhibit 4 point graph, but a 2-point graph because the aircraft's structural design can incorporate more load and can fly at longer distances. The Maximum Take of weight (mTOW) of the aircraft is decided based on the aircraft's structural limitation and hence restricts the aircraft from carrying maximum payload and maximum fuel. The trade-off between payload and fuel results in 4 point diagram. The liquid hydrogen region falls well inside the aircraft's maximum capability, hence the aircraft configuration can carry maximum payload and fuel. From the payload range diagram, the range of the aircraft increases with a decrease in total payload, and the configuration with a maximum capacity of 232 passengers seems to be more feasible.

The center of gravity shift depending on the tank configuration is an interesting study as seen in figure 64. Minimum and maximum C.G. shift shrinks as the passenger capacity increases. This is due to the reduced fuel weight in the aircraft. The large deviation in the center of gravity is the result of the fuel tank being in the aft region of the fuselage. In an ideal aircraft design, the fuel of the aircraft is positioned inside the wings, but since the liquid hydrogen tank requirements state that the tank should be deprived of sharp edges, hence fuselage is an optimum part of the aircraft. Initially, the aircraft would have the C.G. towards the aft region at full fuel. As the plane cruises, the fuel burns, and since the fuel burns the weight that is present at the aft region of the fuselage begins to reduce, this triggers the transition of C.G. towards the front region of the fuselage. This deviation may be large at larger tanks and more fuel weight. It is seen in the figure that the average C.G. moves towards the rear region of the fuselage as the number of tanks increases, additionally, the deviation of C.G. also shrinks as the tank quantity increases. This behavior change is due to the reduction in fuel weight and increase in passenger capacity. Since a static weight that does not reduce/increase as the aircraft is in the air is being introduced, the static margin of the aircraft begins to stabilize and the shift of C.G. minimizes. The 2-tank configuration seems to be more forgiving for all passenger capacities where the front limit of C.G. remains relatively stable and the rear limit of C.G. shrinks with an increase in Passenger capacity. Hence 2 tank configuration exhibits more optimal results as required.

The total range of the aircraft is dependent on the fuel as well. figure 67 shows the relation of range with fuel on board. This also segregates individual fuel present in the tank depending on the configuration. An ideal multi-tank configuration has equal distribution of fuel in them, It is seen in 2 tank configuration, the fuel distribution is not even, which may lead to a favorable or unfavorable scenario in the event of single tank failure. While the optimum tank configuration as seen in the comparison is 4 tank configuration, this comes with a reduction in range and an increase in excess tank weight. When total weight distribution is taken into consideration for a constant passenger capacity of 232 and different tank configurations, there is similar weight distribution as seen in figure 66. There is a slight variation in tank weight and structural weight, which is reasonable for an increase in tank mass, but also a reduction in total fuel carried, hence compensating for the additional weight. When taking 2 tank configurations and varying the passenger capacity as seen in figure 67, there is a noticeable variation in weight distribution in the graph. It is noted that the structural weight contribution noticeably increases whereas the tank weight contribution reduces as the passenger capacity increases. It corresponds to the study where the fuselage area is occupied by the passengers with an increase in passenger capacity while the tank area reduces.

6.3.3 Aisle Configuration Comparison

The behavior of aircraft for single and twin-aisle configurations differs from the retrofit of liquid hydrogen tanks. In this section, a brief study of this difference in behavior is carried out.

For the same hydrogen tank configuration of 2 tanks for both single and twin-aisle configurations, there is a large difference in a shift of the center of gravity as seen in figure 68. This is because of a large difference in the total fuel weight the tank carries for both aisle configurations. Twin-aisle configuration carries approximately 82 % more fuel than single aisle configuration for the same passenger capacity of 168 and tanks configuration. Such a large increase in fuel weight is the result of a wider aircraft body and cabin volume available due to the multi-aisle property. This provides advantageous for twin-aisle configuration in terms of range but also increases overall weight. With more fuel weight included in the twin-aisle configuration, this also affects the center of gravity, in which a large shift in C.G. is experienced in the twin-aisle. In this region, the single-aisle configuration provides a more stable configuration where the shift in C.G. is not wide as in the twin-aisle, this helps the single-aisle to reduce trim drag it experiences during flight. The cruise efficiency of the twin-aisle configuration may be impacted due to extensive input of trim drag to balance the aircraft as fuel burns.

The payload range diagram for the same passenger capacity of 168 and different tank configurations is seen in figure 69. It is fast to notice that the twin-aisle configuration has a significantly longer range compared to the single-aisle counterpart for the same specification. This directly contributes to the larger fuel tank and fuel quantity in the twin-aisle. But for the same amount of payload that the twin-aisle configuration carries, it has a heavier weight of approximately 37.6 % more than the single-aisle configuration, out of which, the fuel weight is 82 % more than the single-aisle configuration. This shows that the total increase in the range comes with an increase in fuel weight.

The Operational Empty weight of the aircraft is the cumulative total of the structural, tank, and cabin mass of the aircraft. Twin-aisle, being a wider body with more structural material present is naturally heavier than single-aisle configuration as seen in figure 70. But since it has a larger weight and size, it requires a longer runway and compatible gate in the airport. While single aisle configuration is beneficial for regional airports and city hops, twin aisle is capable of carrying more passengers and covering larger distances but requires additional runway length for take-off and landing. Though the twin-aisle is approximately 46.3 % heavier in terms of Operational Empty weight, the average difference in tank weight is approximately 251 % for respective comparison, making twin-aisle configuration tanks 2.5 times larger than the single-aisle configuration and the range the twin-aisle configuration provides is approximately 69 % more than the single-aisle configuration.

Concluding the comparison of single and twin-aisle variants with the retrofit of the liquid hydrogen tank, the optimum design for single-aisle configuration would be 168 passenger and 2 tank configuration as single aisle is much suited for it's stable flight cruise performance and capability to land in smaller airports. With it's range of 1078 km, it is an optimal selection for domestic travel and city hops.

The optimum design point for a twin-aisle configuration retrofitted with a liquid hydrogen tank would be 2 tank configuration with 232 passenger capacity. This tank configuration enforces redundancy with reasonable passenger capacity and a reasonable range of 4180 km (at max capacity). This provides a balance between range and passenger capacity. Since the aircraft's maximum weight is well within the structural limitation, the aircraft is capable of carrying maximum passenger capacity with the full fuel tank.

7 Conclusions

The advantage of the digital mock-up model of the cabin design system reduces the total cost and time consumed for building prototypes and enables clients and customers to experience the immersive virtual and augmented reality of the cabin system in a digital environment. Though a digital model may be quicker and financially liberal when compared to building a mockup model, it consumes sufficient time and computational power to manually create each parameter of the cabin system. By introducing a knowledge-based engineering approach into the cabin design system, basic information such as seating positions, galley and lavatory positions, and also aircraft geometry is to be given, and various rulesets and pre-defined parameters create a feasible cabin design system that meets all the requested parameters well within the design constraints of the aircraft cabin. The Design and Engineering Engine method in KBE proves to be the most optimum currently available to utilize in creating a digital mockup. The FUGA tool's twin-aisle configuration is now extended and necessary advancement has been implemented to be feasible enough for scientific analysis and to provide a fully immersive digital experience.

With all features of the Kowledge-based engineering approach being added to the FUGA tool except for one. A user-friendly interface to have seamless interaction and usage of tools. It is now implemented into the FUGA tool through a web-based application. This bridges the gap between the usability of the tool for users with an elementary knowledge of Python programming since FUGA is built on the backbone of Python. Flask being the primary framework for running the Web-based application, additional programs such as C # and Python were integrated into Flask for simultaneous and parallel processing of data.

With a digital cabin design system capable of performing analytical solutions in both single and twin-aisle. FUGA tool has the capability of performing weight analysis based on the integration of non-integral hydrogen fuel tanks into the fuselage of the aircraft. Though the position of the hydrogen tank is designed to be placed in the rear region of the fuselage. This opened the door for many analyses and studies conducted with different tank configurations and passenger capacity. With such a diverse level of study, the potential of liquid hydrogen tank retrofit into existing fuselage has been briefly studied and local optimum designs were chosen based on the acquired data. With the results and outcomes that were taken from the study conducted in the thesis, the solution for individual research questions are concluded individually.

7.1 Research Question 1:

How can a web-based application effectively represent knowledge-based engineering, and to what extent can it be universally applied to KBE projects

A knowledge-based Engineering application can be represented using many ways, it

can be just using the limited data option which can be can be given to the user or it can be entirely dynamic where you can receive the data from the user in terms of files and objects, additionally, have a process of editing the data received making it more editable. The ”Representation” also depends on two things one is the GUI and the other is the WBA platform that is being used, in this case since the KBE and the tool run on Python, it was more sensible to use a Python-based framework making it more interactable. And With the GUI it depends on the HTML, CSS & JAVASCRIPT, also with interaction with the WBA frame being used, the interaction between these makes it more dynamic.
Effectiveness mainly depends on the modularity of the WBA, if the framework is modular, any improvements made to the KBE or any additional feature added can be easily adapted in the framework. It also depends on the interaction between the interface language and the KBE Language.

7.2 Research Question 2:

How knowledge-based engineering approach is utilized in parametric aircraft designing and cabin designing?

Knowledge-Based Engineering has multiple approaches that correspond to the purpose and outcomes desired from its usage. DEE methodology is used primarily for creating parametric aircraft cabin design and aircraft parametric modeling. Through the DEE method in the KBE approach, a repository of containers with design rulesets that creates inter depending rulesets to create the basic structure of the aircraft geometry and internal cabin design. Through the repository of information, the basic cabin design parameters are initialized, and depending on the input provided by the user/client, the corresponding rulesets are utilized to create the detailed cabin design or aircraft geometry. A workspace or repository of product data which includes crucial details about cabin design or aircraft geometry provides the necessary information for the ruleset to instantiate or create the design. The inference engine is responsible for triggering or utilizing the correct ruleset for the depending profiles. The dependency of each ruleset is visualized as nodes. Nodes which set initial conditions, either systematic or user-defined are initialization nodes while the node which provides the outcome or result are end nodes, and the node configuration for the FUGA tool is seen in Figure 6.

With the implementation of knowledge-based engineering, the introduction of new models such as the middle hatracks, and modification of existing model size and position such as the ceiling panels were easy to implement and positioned based on the positions of other components of the cabin. This is elaborated and seen in the Section 5.2.

7.3 Research Question 3:

What are the benefits of a liquid hydrogen tank being retrofitted into an existing fuselage? What are the trade-offs?

A liquid hydrogen tank is a step toward a green aviation path in which fossil fuel is replaced with liquid hydrogen to have steam and water as exhaust. A liquid hydrogen tank can be fitted into an existing aircraft through a non-integral tank. This would require a study about the effects of the additional tanks on the structure of the aircraft. The benefits of having a liquid hydrogen tank retrofitted into an existing fuselage than designing an aircraft with a liquid hydrogen tank are given below:

1. Retrofit of liquid hydrogen tank does not require extensive resources, manpower, and financial aid into research and development of new aircraft design incorporating the liquid hydrogen tank into the design.

2. Maintenance and repairs would be easier and existing aircraft can be converted into greener aircraft once detailed analysis and results are validated.

3. The tank can be positioned depending on the availability of space inside the aircraft at any time.

Introducing a completely new system into an existing fuselage design would require additional support systems and materials to ensure the safety and efficient operation of aircraft. This additional system induces more OEM into the aircraft at the region where the tank is being placed, hence a noticeable level of shift is seen in C.G. This would induce trim drag during the cruise, hence may reduce flight efficiency. Since a tank is being retrofitted into the aircraft, the available volume for fuel is drastically reduced, thereby affecting the range of the aircraft, and since the tank is being placed inside the fuselage of the aircraft, the total passenger capacity of the aircraft is less than the traditional cabin counterpart as seen in Figure 53 and 54 for single-aisle configuration and Figure 62 and 63 for twin-aisle configuration. The shift of C.G. is seen for different aisle configurations in Figure 68.

7.4 Research Question 4:

How does the performance of aircraft vary from single and twin-aisle with liquid hydrogen tanks?

The critical performance change that is noticeable from single and twin-aisle configurations is the range and payload capacity relationship along with C.G. Shift. single-aisle configuration having a narrow body fuselage does not provide the tank enough volume for range. This affects the total endurance of the configuration. Furthermore, since the tank volume is sufficient for city hops, it also is beneficial in terms of C.G. With lower tank volume, the shift of C.G. from rear to front is relatively low when compared to twin aisle configuration of low as 1.7 %. Though the configuration has a lower range, it has greater cruising efficiency with less trim drag. In the twin-aisle configuration, thanks to the wide-body fuselage, the fuselage provides a larger volume to store more fuel, hence a longer range is achieved. But since most of the fuel is stored in the rear region of the fuselage, the shift of C.G. from rear to front is quite significant when compared to a single-aisle configuration with up to 7.5%. The comparison of C.G. difference is seen in Figure 68. The OEM of a twin-aisle configuration is relatively more than a single-aisle configuration in general comparison, but taking into consideration of liquid hydrogen tank, the excess material and support system makes the overall OEM for twin-aisle configuration heavier. This is vividly seen in Figure 70. Liquid hydrogen tank retrofitted aircraft regardless of the aisle configurations, provide significantly lower range and payload capacity compared to their reference aircraft. But this is because the liquid hydrogen tank is positioned inside the fuselage region where passenger-occupied regions were replaced with the liquid hydrogen tank and its supporting systems.

7.5 Future Works

The possibilities for future development of the FUGA tool are large. With upcoming advancements in Technological and industrial fields, this would open many doors that would help FUGA to provide enhanced data and information. There is a lot of scope for improvement and future studies which can be conducted, they are as follows:

1. In the case of Web-Based Application, A more user-friendly web approach and a user experience survey can be conducted to know the types of applicability they require from the FUGA in the web app. An attempt to make the web app fail-proof by introducing some preliminary checks before executing.

2. The Integration of hydrogen tank retrofit into the fuselage in FUGA is a loft from one regional cross-section of the fuselage to another, this does not take into consideration the curvature of the fuselage and hence, the volume and area of the curvature is not taken into consideration for large hydrogen tank. This issue can be easily corrected and a more robust tank geometry can be designed for arriving at a more accurate result.

3. Additionally having both integral and non-integral (including external pod design) type tanks. Also having a so-called top tank design tank variant.

4. On the retrofit design analysis currently performed, an Extended fuselage tank analysis would give further understanding of the difference in both. Furthermore, introducing variable positioning of tanks in the fuselage as well as possibilities of placing them in the wing region would provide noticeable results.

References

[1] Krein A, Williams G. Flightpath 2050: Europeâs vision for aeronautics. Innovation for Sustainable Aviation in a Global Environment: Proceedings of the Sixth European Aeronautics Days, Madrid. 2012;30.

[2] Xie S, Tu Y. Rapid one-of-a-kind product development. The International Journal of Advanced Manufacturing Technology. 2006;27:421-30.

[3] Qin SF, Harrison R, West AA, Jordanov IN, Wright DK. A framework of web-based conceptual design. Computers in Industry. 2003;50(2):153-64.

[4] Walther JN, Hesse C, Biedermann J, Nagel B. Extensible aircraft fuselage model generation for a multidisciplinary, multi-fidelity context. In: 33rd Congress of the International Council of the Aeronautical Sciences (ICAS); 2022. .

[5] Skarka W. Application of MOKA methodology in generative model creation using CATIA. Engineering Applications of Artificial Intelligence. 2007;20(5):677-90.

[6] German Aerospace Center (DLR). HOME CPACS ELEMENTS; 2021-04-20. Accessed May 28, 2023. `https://www.cpacs.de/documentation/CPACS_3_4_0_Docs/html/89b6a288-0944-bd56-a1ef-8d3c8e48ad95.htm`.

[7] Walther JN, Kocacan B, Hesse C, Gindorf A, Nagel B. Automatic cabin virtualization based on preliminary aircraft design data. CEAS Aeronautical Journal. 2022;13(2):403-18.

[8] Lee DS, Fahey DW, Forster PM, Newton PJ, Wit RC, Lim LL, et al. Aviation and global climate change in the 21st century. Atmospheric environment. 2009;43(22-23):3520-37.

[9] Verhagen WJ, Bermell-Garcia P, Van Dijk RE, Curran R. A critical review of Knowledge-Based Engineering: An identification of research challenges. Advanced Engineering Informatics. 2012;26(1):5-15.

[10] Scherer J, Kohlgrüber D. Fuselage structures within the CPACS data format. Aircraft Engineering and Aerospace Technology: An International Journal. 2016;88(2):294-302.

[11] Gomez A, Smith H. Liquid hydrogen fuel tanks for commercial aviation: Structural sizing and stress analysis. Aerospace Science and Technology. 2019;95:105438.

[12] Munjulury RC, Staack I, Berry P, Krus P. A knowledge-based integrated aircraft conceptual design framework. CEAS Aeronautical Journal. 2016;7:95-105.

[13] Reddy EJ, Sridhar C, Rangadu VP. Knowledge based engineering: notion, approaches and future trends. American Journal of Intelligent Systems. 2015;5(1):1-17.

[14] Tu Y, Xie S. A WWW-based integrated product development information management system. IFAC Proceedings Volumes. 2000;33(20):469-74.

[15] Xie S. A decision support system for rapid one-of-a-kind product development. The International Journal of Advanced Manufacturing Technology. 2006;28:643-52.

[16] Baldwin R, Krugman P. Industrial policy and international competition in wide-bodied jet aircraft. In: Trade policy issues and empirical analysis. University of Chicago Press; 1988. p. 45-78.

[17] Ringqvist L. Extended range operation of twin-engined transport aircraft (ETOPS). In: Aircraft Design Systems and Operations Meeting; 1984. p. 2512.

[18] Sarkar A. Evolving green aviation transport system: a hoilistic approah to sustainable green market development. 2012.

[19] Verstraete D, Hendrick P, Pilidis P, Ramsden K. Hydrogen fuel tanks for subsonic transport aircraft. International journal of hydrogen energy. 2010;35(20):11085-98.

[20] Xu W, Li Q, Huang M. Design and analysis of liquid hydrogen storage tank for high-altitude long-endurance remotely-operated aircraft. International Journal of Hydrogen Energy. 2015;40(46):16578-86.

[21] Winnefeld C, Kadyk T, Bensmann B, Krewer U, Hanke-Rauschenbach R. Modelling and designing cryogenic hydrogen tanks for future aircraft applications. Energies. 2018;11(1):105.

[22] Verstraete D. On the energy efficiency of hydrogen-fuelled transport aircraft. International Journal of Hydrogen Energy. 2015;40(23):7388-94.

[23] Thilmany J. Open Cascade SAS. Mechanical Engineering-CIME. 2011;133(5):16-7.

[24] Hagberg A, Swart P, S Chult D. Exploring network structure, dynamics, and function using NetworkX. Los Alamos National Lab.(LANL), Los Alamos, NM (United States); 2008.

[25] Kitware. About Overview: VTK OVERVIEW; n.d. Accessed May 17, 2023. https://vtk.org/about/#overview.

[26] Kitware. About : Paraview; n.d. Accessed May 17, 2023. https://www.paraview.org/about/.

[27] Cooper D, LaRocca G. Knowledge-based techniques for developing engineering applications in the 21st century. In: 7th AIAA ATIO Conf, 2nd CEIAT Int'l Conf on Innov and Integr in Aero Sciences, 17th LTA Systems Tech Conf; followed by 2nd TEOS Forum; 2007. p. 7711.

[28] Ammar-Khodja S, Perry N, Bernard A. Processing knowledge to support knowledge-based engineering systems specification. Concurrent Engineering. 2008;16(1):89-101.

[29] Stokes M, et al. Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications. vol. 3. Professional Engineering Publishing London; 2001.

[30] Lovett P, Ingram A, Bancroft C. Knowledge-based engineering for SMEs a methodology. Journal of materials processing technology. 2000;107(1-3):384-9.

[31] Curran R, Verhagen WJ, Van Tooren MJ, Van Der Laan TH. A multidisciplinary implementation methodology for knowledge based engineering: KNOMAD. Expert Systems with Applications. 2010;37(11):7336-50.

[32] La Rocca G, Van Tooren M. Enabling distributed multi-disciplinary design of complex products: a knowledge based engineering approach. Journal of Design Research. 2007;5(3):333-52.

[33] Baxter D, Gao J, Case K, Harding J, Young B, Cochrane S, et al. An engineering design knowledge reuse methodology using process modelling. Research in engineering design. 2007;18:37-48.

[34] La Rocca G. Knowledge based engineering techniques to support aircraft design and optimization. 2011.

[35] Nagel B, Böhnke D, Gollnick V, Schmollgruber P, Rizzi A, La Rocca G, et al. Communication in aircraft design: Can we establish a common language. In: 28th International Congress of the Aeronautical Sciences. vol. 201; 2012. .

[36] Liersch CM, Hepperle M. A distributed toolbox for multidisciplinary preliminary aircraft design. CEAS Aeronautical Journal. 2011;2:57-68.

[37] Alder M, Moerland E, Jepsen J, Nagel B. Recent advances in establishing a common language for aircraft design with CPACS. 2020.

[38] Lukasczyk J, Liang X, Luo W, Ragan ED, Middel A, Bliss N, et al. A Collaborative Web-Based Environmental Data Visualization and Analysis Framework. In: EnvirVis@ EuroVis; 2015. p. 25-9.

[39] TIOBE. TIOBE TIOBE INDEX: The Python Programing Language; n.d. Accessed May 17, 2023. https://www.tiobe.com/tiobe-index/python/.

[40] Pallets Projects. About FLASK; n.d. Accessed May 19, 2023. https://flask.palletsprojects.com/en/2.3.x/quickstart/.

[41] Django Software Foundation. Documentation django; n.d. Accessed May 19, 2023. https://docs.djangoproject.com/en/4.2/.

[42] CherryPy team. Documentation CherryPy; n.d. Accessed May 19, 2023. https://docs.cherrypy.dev/en/latest/.

[43] Pylons Project. Documentation Pyramid; n.d. Accessed May 19, 2023. https://docs.pylonsproject.org/projects/pyramid/en/latest/.

[44] FriendFeed. Documentation Tornado Web Server; n.d. Accessed May 19, 2023. https://www.tornadoweb.org/en/stable/.

[45] SANIC. HOME SANIC FRAMEWORK; n.d. Accessed May 19, 2023. https://sanic.dev/en/.

[46] Ploty. HOME Dash Python User Guide; n.d. Accessed May 19, 2023. https://dash.plotly.com/.

[47] Aircraft Range: the Breguet Range Equation;. Accessed July 25, 2023. https://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node98.html.

[48] Maintenance Book Airbus A321; 2005. Accessed July 15, 2023. https://www.airbus.com/sites/g/files/jlcbta136/files/2021-11/Airbus-Commercial-Aircraft-AC-A321.pdf.

[49] Maintenance Book Airbus A330; 2005. Accessed July 15, 2023. https://www.airbus.com/sites/g/files/jlcbta136/files/2021-11/Airbus-Commercial-Aircraft-AC-A330.pdf.

A Appendix

A.1 Results

A.1.1 Advancement of Twin aisle coniguration

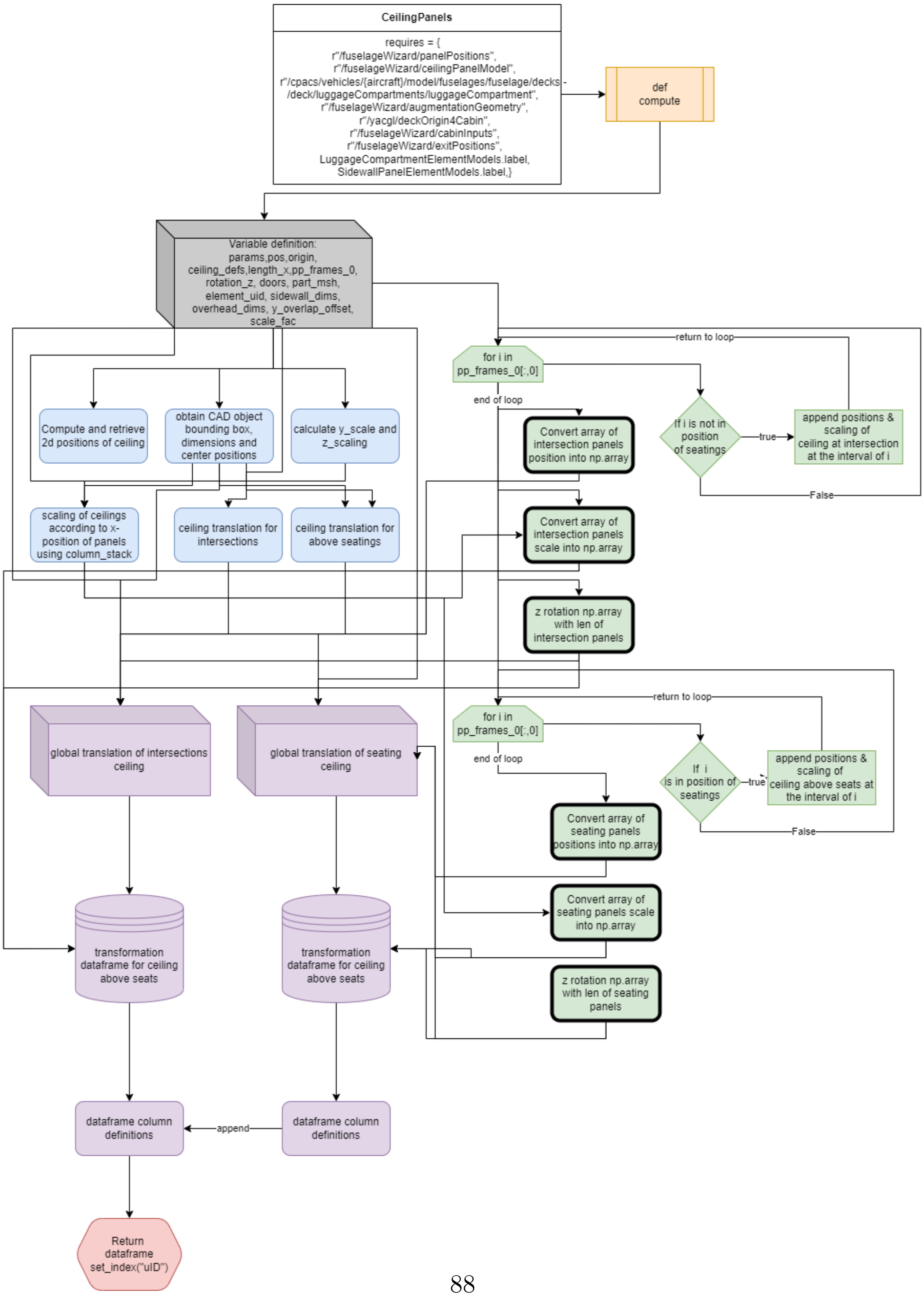

CeilingPanels
requires = {
r"/fuselageWizard/panelPositions",
r"/fuselageWizard/ceilingPanelModel",
r"/cpacs/vehicles/{aircraft}/model/fuselages/fuselage/decks -
/deck/luggageCompartments/luggageCompartment",
r"/fuselageWizard/augmentationGeometry",
r"/yacgl/deckOrigin4Cabin",
r"/fuselageWizard/cabinInputs",
r"/fuselageWizard/exitPositions",
LuggageCompartmentElementModels.label,
SidewallPanelElementModels.label,}
def compute
Variable definition:
params,pos,origin,
ceiling_defs,length_x,pp_frames_0,
rotation_z, doors, part_msh,
element_uid, sidewall_dims,
overhead_dims, y_overlap_offset,
scale_fac
Compute and retrieve 2d positions of ceiling
obtain CAD object bounding box, dimensions and center positions
calculate y_scale and z_scaling
scaling of ceilings according to x-position of panels using column_stack
ceiling translation for intersections
ceiling translation for above seatings
return to loop
for i in pp_frames_0[:,0]
end of loop
If i is not in position of seatings
true
append positions & scaling of ceiling at intersection at the interval of i
False
Convert array of intersection panels position into np.array
Convert array of intersection panels scale into np.array
z rotation np.array with len of intersection panels
return to loop
for i in pp_frames_0[:,0]
end of loop
If i is in position of seatings
true
append positions & scaling of ceiling above seats at the interval of i
False
Convert array of seating panels positions into np.array
Convert array of seating panels scale into np.array
z rotation np.array with len of seating panels
global translation of intersections ceiling
global translation of seating ceiling
transformation dataframe for ceiling above seats
transformation dataframe for ceiling above seats
dataframe column definitions
append
dataframe column definitions
Return dataframe set_index("uID")

Figure 71: Detailed flowchart of python code implemented in ceiling panel generation in twin-aisle configuration

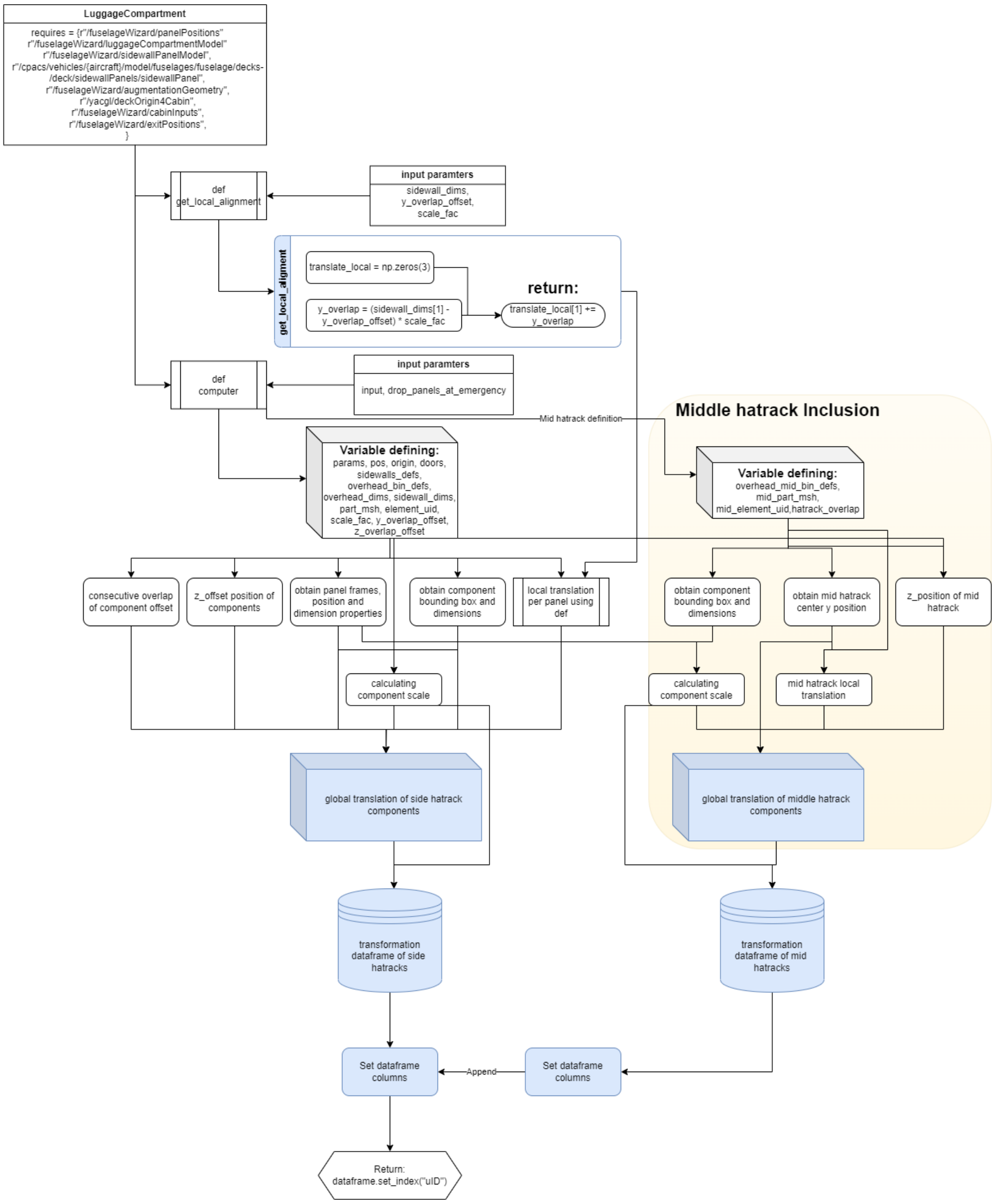


Figure 72: Detailed flowchart of python code implemented in hatracks generation in twin-aisle configuration

A.1.2 Single Aisle Configuration

In this section you can find the data sheets, 3D/2D layouts, Payload-Range Diagram, CG plots and weight distributions studies carried out on different variants in Single-aisle configuration.

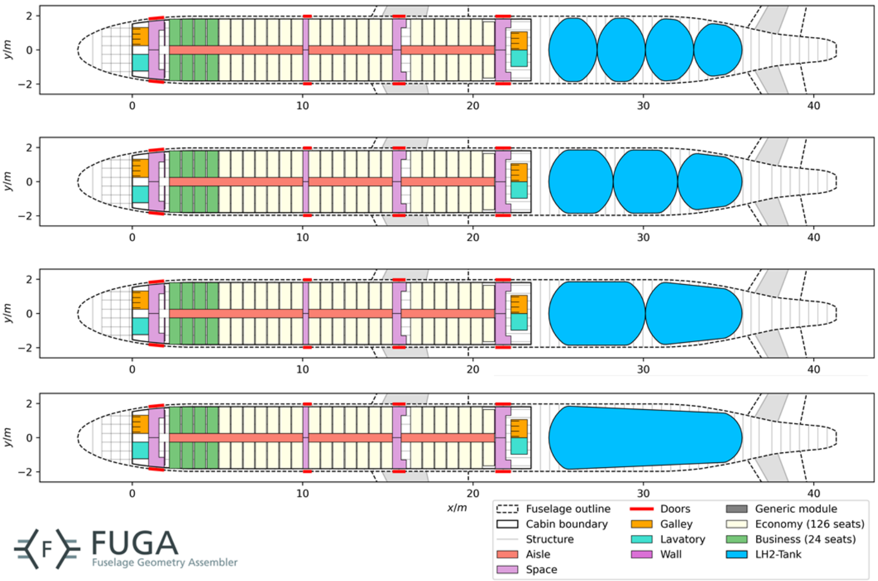


Figure 73: Variation of Tank Configuration for fixed Pax capacity (150 Pax) in 2D

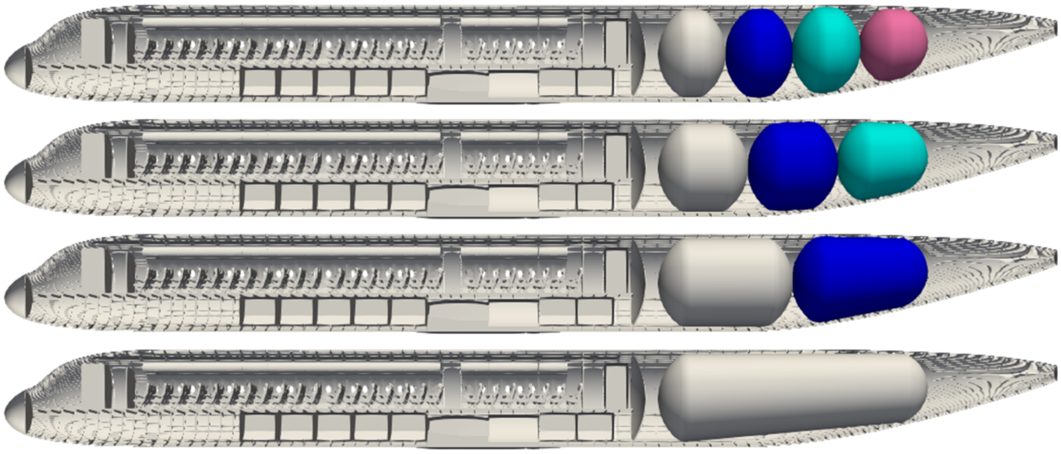


Figure 74: Variation of Tank Configuration for fixed Pax capacity (150 Pax) in 3D

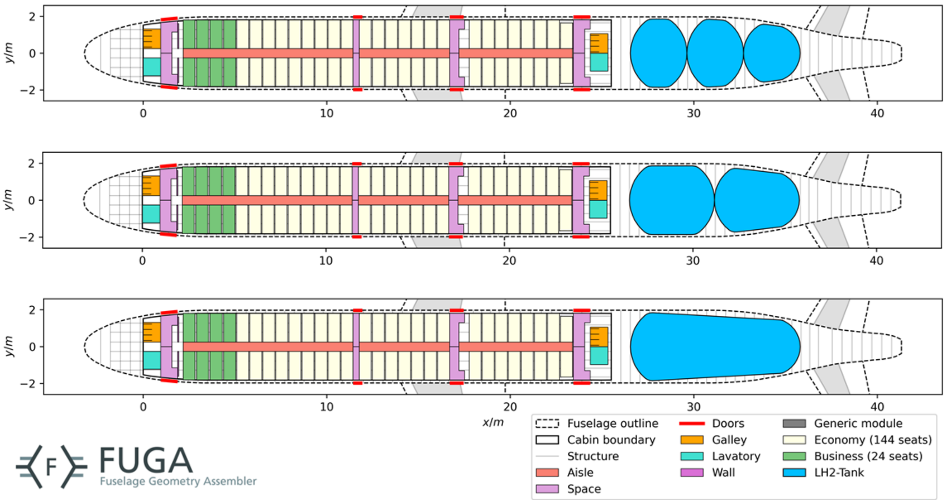


Figure 75: Variation of Tank Configuration for fixed Pax capacity (168 Pax) in 2D

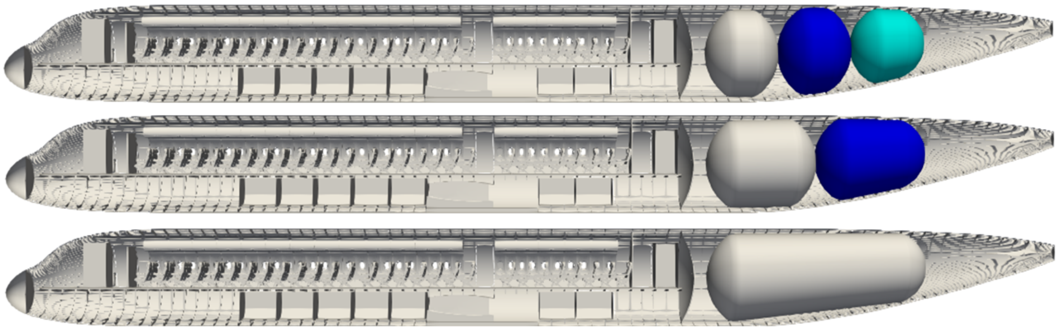


Figure 76: Variation of Tank Configuration for fixed Pax capacity (168 Pax) in 3D

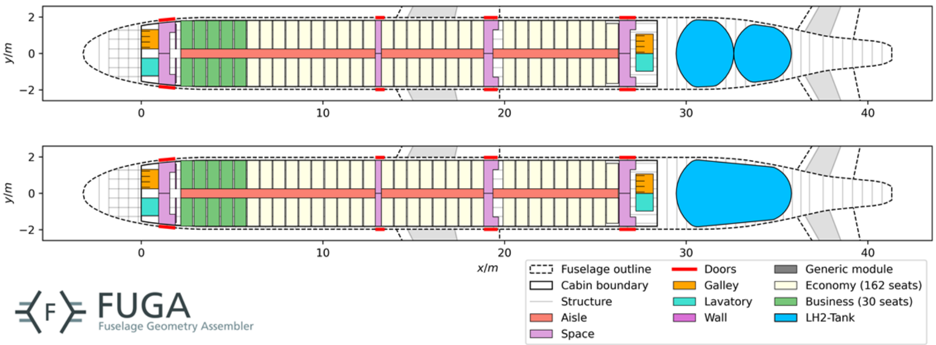


Figure 77: Variation of Tank Configuration for fixed Pax capacity (192 Pax) in 2D

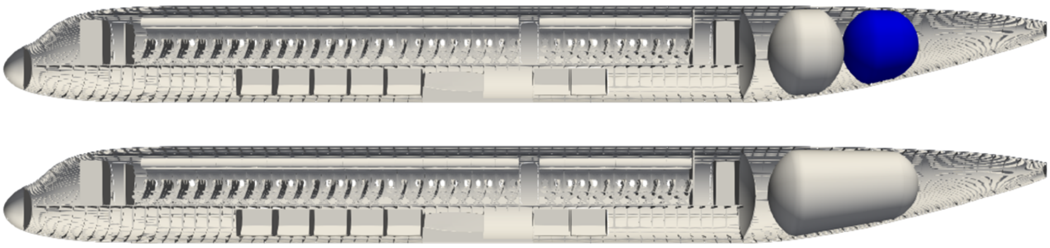


Figure 78: Variation of Tank Configuration for fixed Pax capacity (192 Pax) in 3D

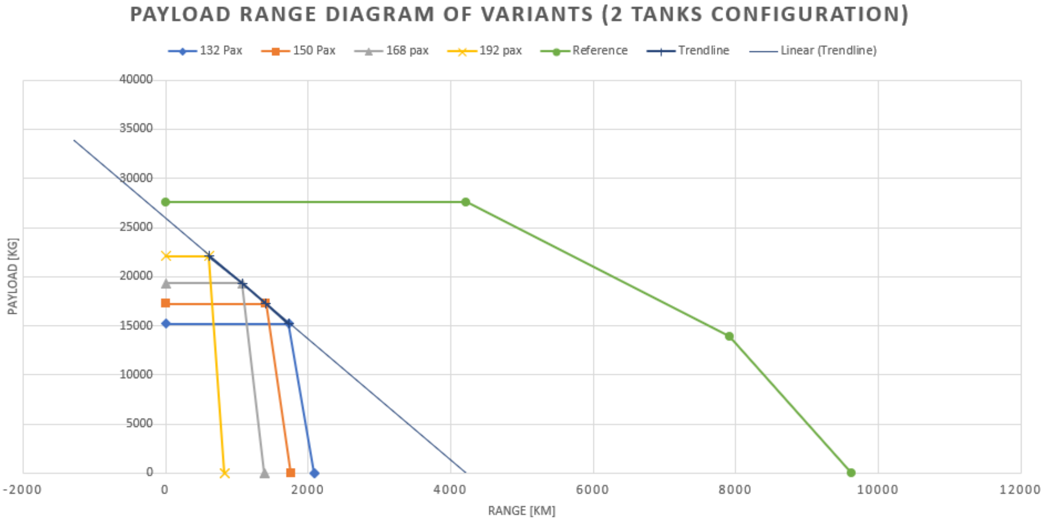


Figure 79: Variation of Payload Range Diagram for Pax capacity (132, 150, 168 and 192 Pax) in fixed tank configuration (2 Tank)

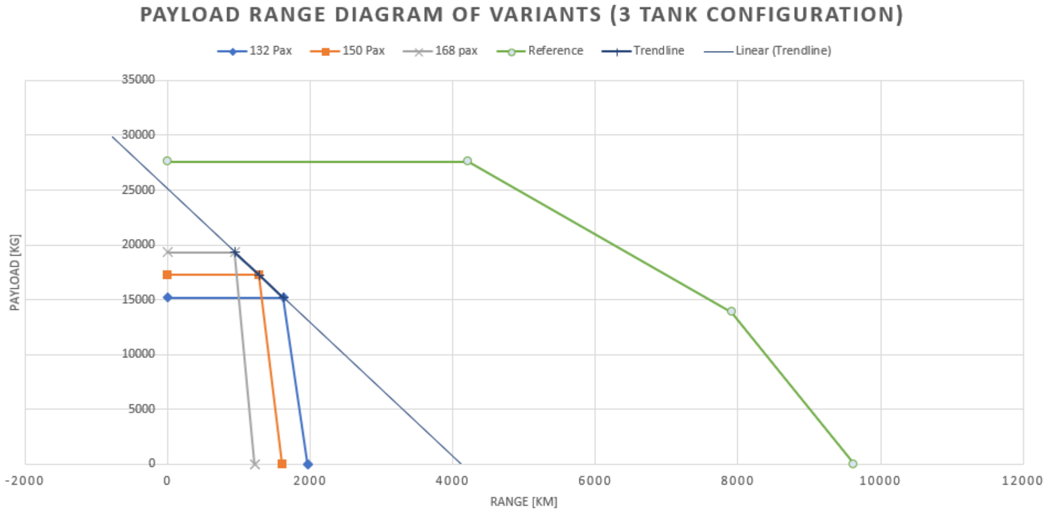


Figure 80: Variation of Payload Range Diagram for Pax capacity (132, 150, 168 and 192 Pax) in fixed tank configuration (3 Tank)

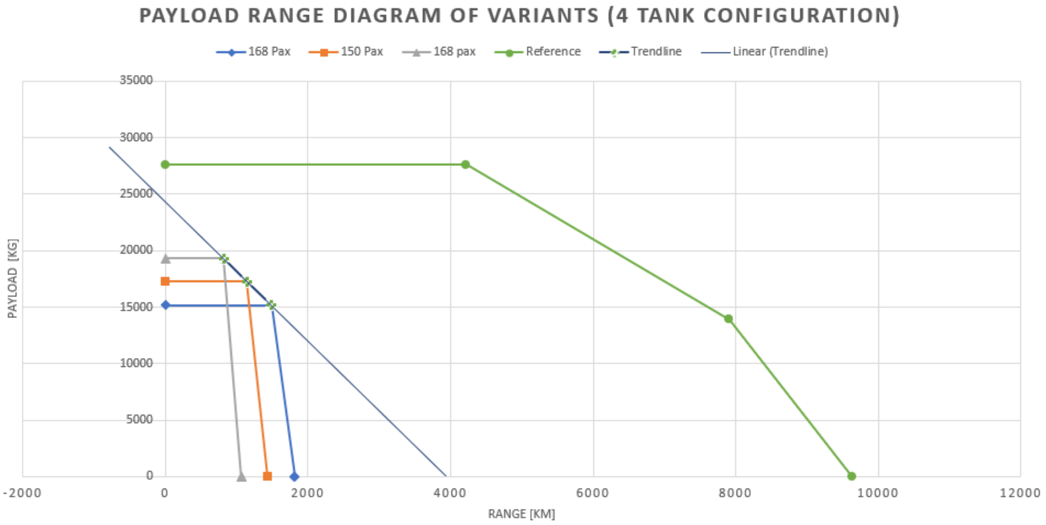
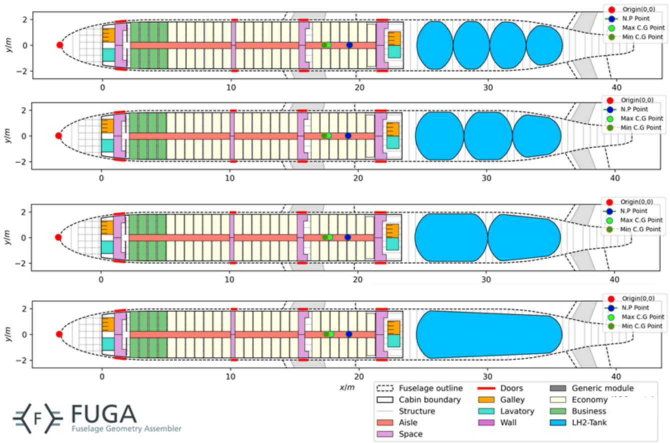
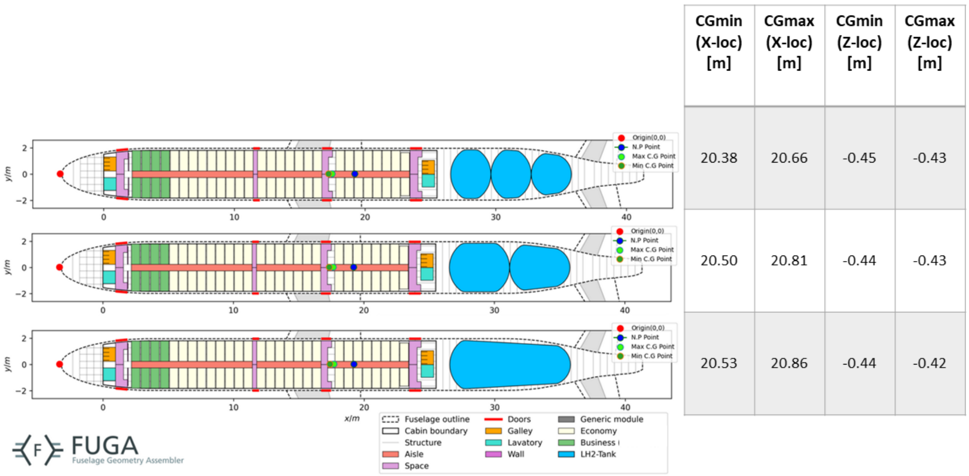


Figure 81: Variation of Payload Range Diagram for Pax capacity (132, 150, 168 and 192 Pax) in fixed tank configuration (4 Tank)



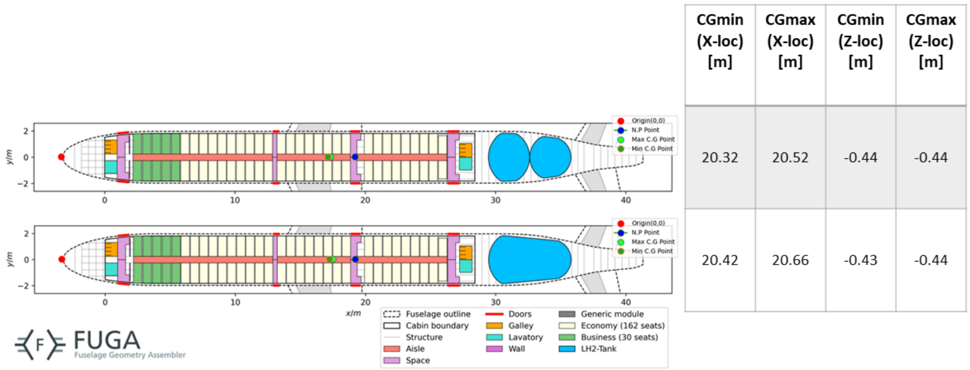
CGmin (X-loc) [m]	CGmax (X-loc) [m]	CGmin (Z-loc) [m]	CGmax (Z-loc) [m]
20.40	20.70	-0.45	-0.44
20.51	20.86	-0.44	-0.43
20.62	21.00	-0.44	-0.42
20.61	20.98	-0.43	-0.42

Figure 82: Variation C.G for 150 Pax capacity in tank configuration (1, 2, 3 and 4 Tanks)



CGmin (X-loc) [m]	CGmax (X-loc) [m]	CGmin (Z-loc) [m]	CGmax (Z-loc) [m]
20.38	20.66	-0.45	-0.43
20.50	20.81	-0.44	-0.43
20.53	20.86	-0.44	-0.42

Figure 83: Variation C.G for 168 Pax capacity in tank configuration (1, 2, 3 and 4 Tanks)



CGmin (X-loc) [m]	CGmax (X-loc) [m]	CGmin (Z-loc) [m]	CGmax (Z-loc) [m]
20.32	20.52	-0.44	-0.44
20.42	20.66	-0.43	-0.44

Figure 84: Variation C.G for 192 Pax capacity in tank configuration (1, 2, 3 and 4 Tanks)

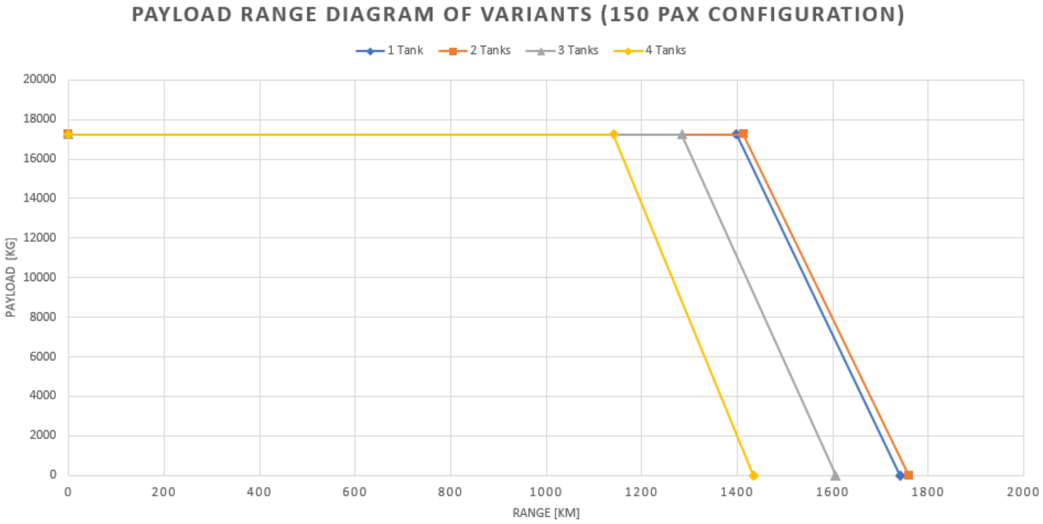


Figure 85: Variation of Payload Range Diagram for 150 Pax capacity in tank configuration (1, 2, 3 and 4 Tanks)

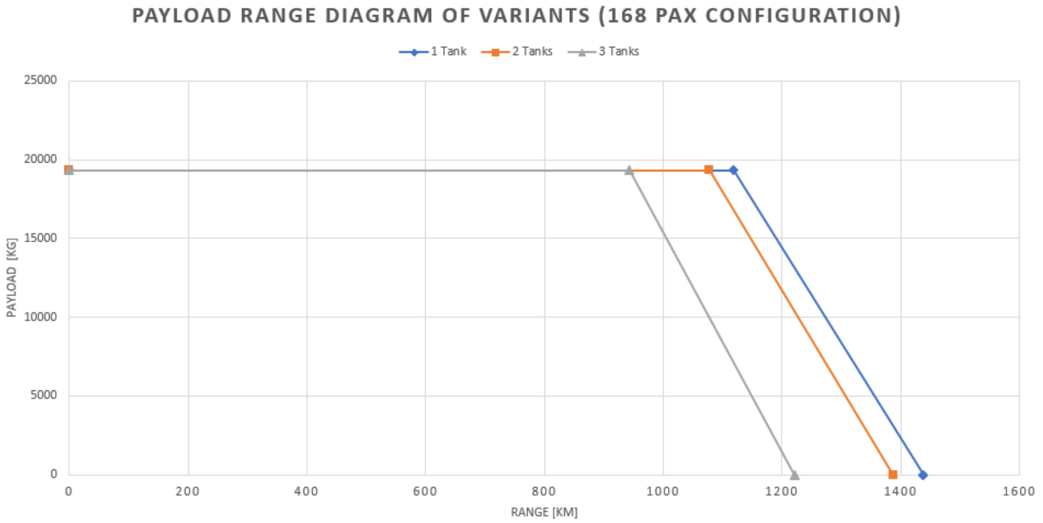


Figure 86: Variation of Payload Range Diagram for 168 Pax capacity in tank configuration (1, 2, 3 and 4 Tanks)

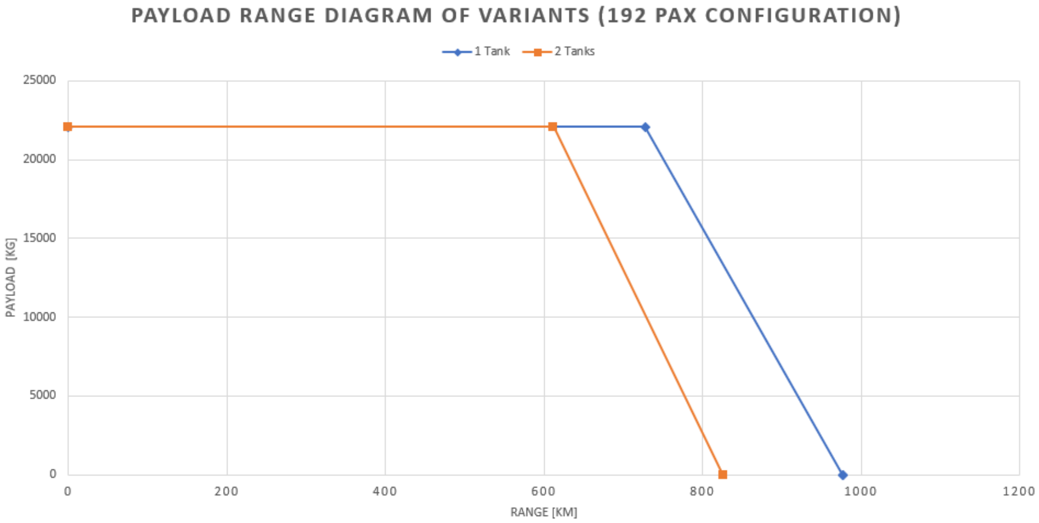


Figure 87: Variation of Payload Range Diagram for 192 Pax capacity in tank configuration (1, 2, 3 and 4 Tanks)

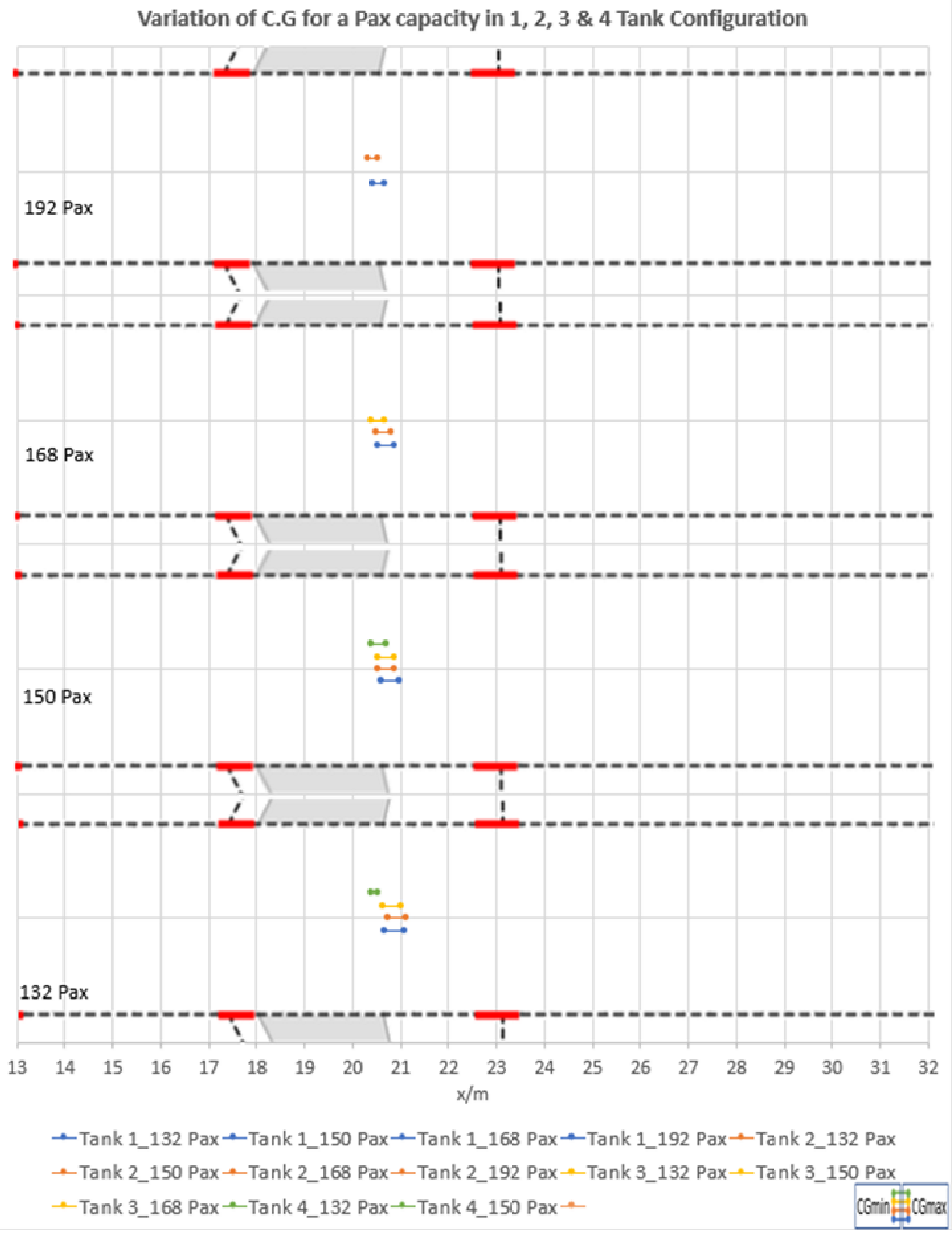
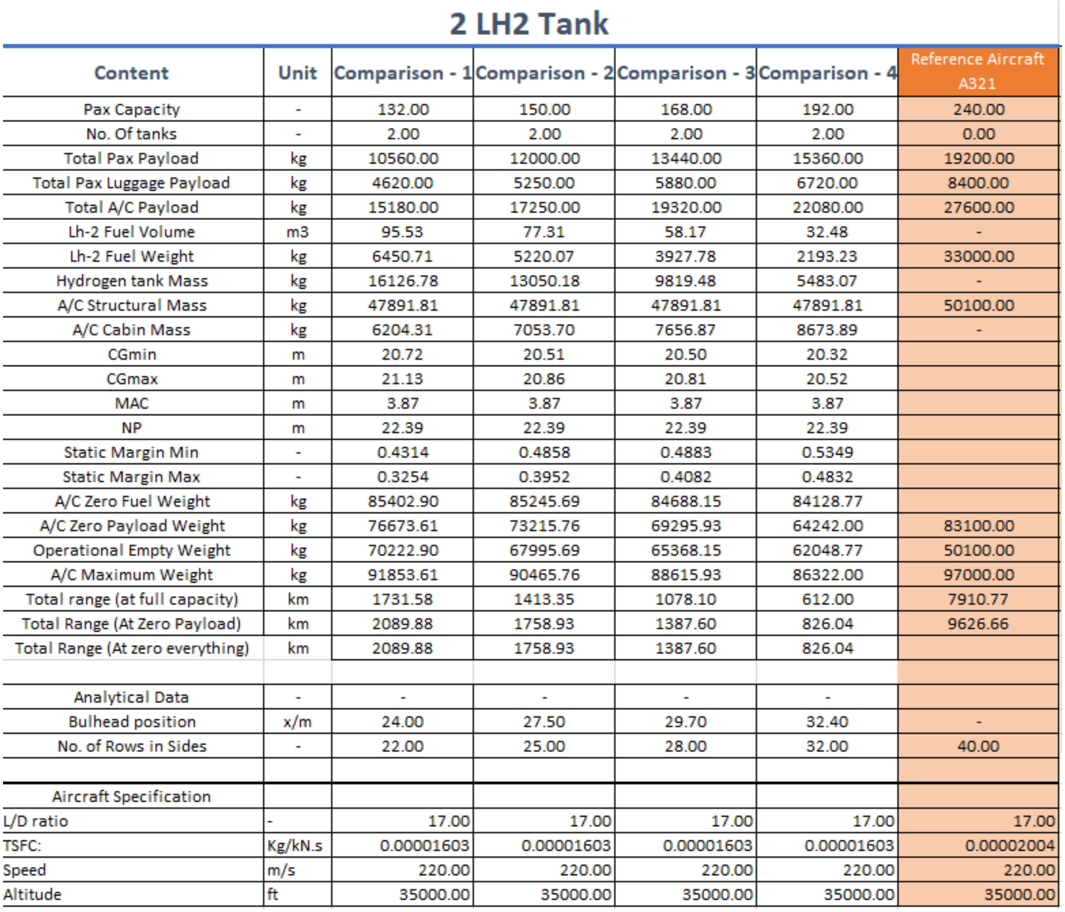


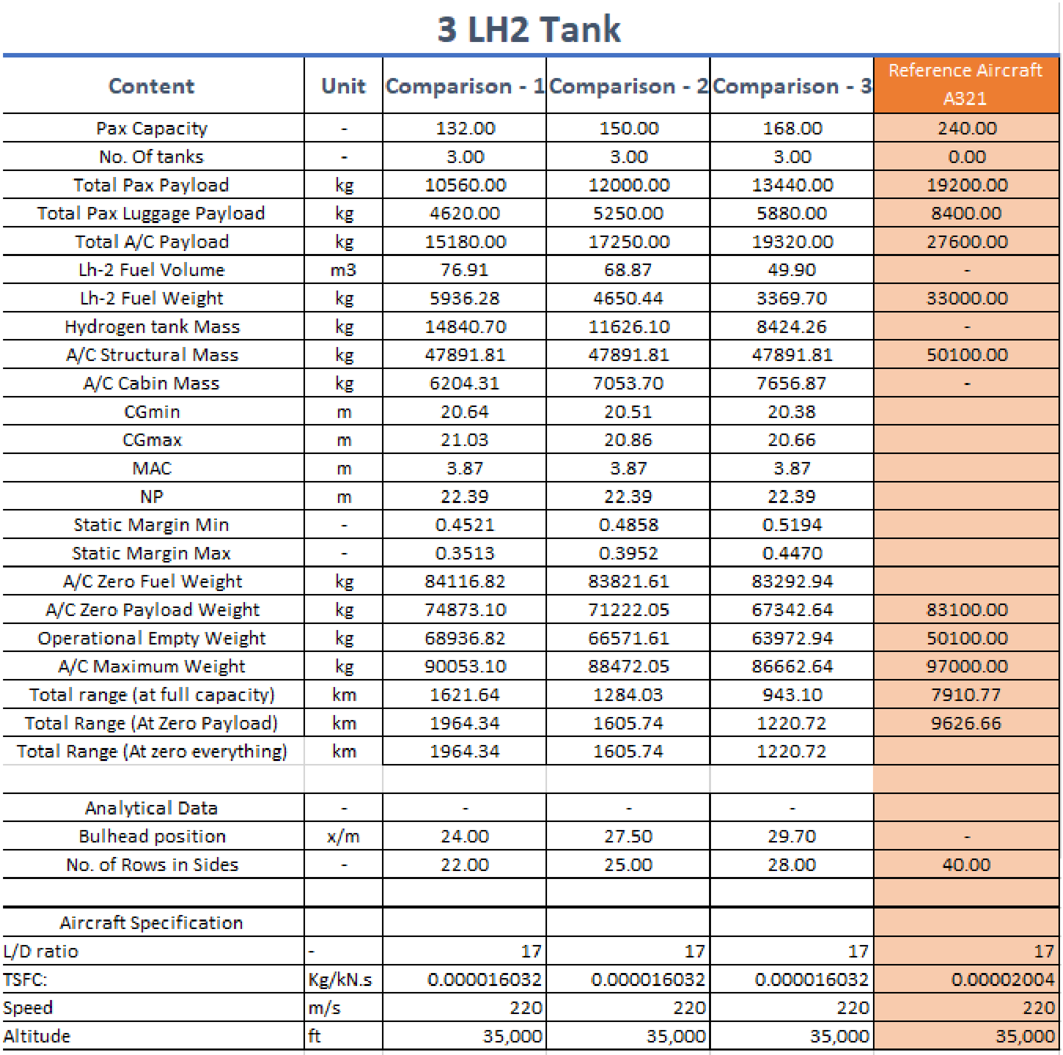
Figure 88: Variation of C.G for Pax capacity in 1,2,3 and 4 tank configurations



2 LH2 Tank

Content	Unit	Comparison - 1	Comparison - 2	Comparison - 3	Comparison - 4	Reference Aircraft A321
Pax Capacity	-	132.00	150.00	168.00	192.00	240.00
No. Of tanks	-	2.00	2.00	2.00	2.00	0.00
Total Pax Payload	kg	10560.00	12000.00	13440.00	15360.00	19200.00
Total Pax Luggage Payload	kg	4620.00	5250.00	5880.00	6720.00	8400.00
Total A/C Payload	kg	15180.00	17250.00	19320.00	22080.00	27600.00
Lh-2 Fuel Volume	m3	95.53	77.31	58.17	32.48	-
Lh-2 Fuel Weight	kg	6450.71	5220.07	3927.78	2193.23	33000.00
Hydrogen tank Mass	kg	16126.78	13050.18	9819.48	5483.07	-
A/C Structural Mass	kg	47891.81	47891.81	47891.81	47891.81	50100.00
A/C Cabin Mass	kg	6204.31	7053.70	7656.87	8673.89	-
CGmin	m	20.72	20.51	20.50	20.32	
CGmax	m	21.13	20.86	20.81	20.52	
MAC	m	3.87	3.87	3.87	3.87	
NP	m	22.39	22.39	22.39	22.39	
Static Margin Min	-	0.4314	0.4858	0.4883	0.5349	
Static Margin Max	-	0.3254	0.3952	0.4082	0.4832	
A/C Zero Fuel Weight	kg	85402.90	85245.69	84688.15	84128.77	
A/C Zero Payload Weight	kg	76673.61	73215.76	69295.93	64242.00	83100.00
Operational Empty Weight	kg	70222.90	67995.69	65368.15	62048.77	50100.00
A/C Maximum Weight	kg	91853.61	90465.76	88615.93	86322.00	97000.00
Total range (at full capacity)	km	1731.58	1413.35	1078.10	612.00	7910.77
Total Range (At Zero Payload)	km	2089.88	1758.93	1387.60	826.04	9626.66
Total Range (At zero everything)	km	2089.88	1758.93	1387.60	826.04	
Analytical Data	-	-	-	-	-	
Bulhead position	x/m	24.00	27.50	29.70	32.40	-
No. of Rows in Sides	-	22.00	25.00	28.00	32.00	40.00
Aircraft Specification						
L/D ratio	-	17.00	17.00	17.00	17.00	17.00
TSFC:	Kg/kN.s	0.00001603	0.00001603	0.00001603	0.00001603	0.00002004
Speed	m/s	220.00	220.00	220.00	220.00	220.00
Altitude	ft	35000.00	35000.00	35000.00	35000.00	35000.00

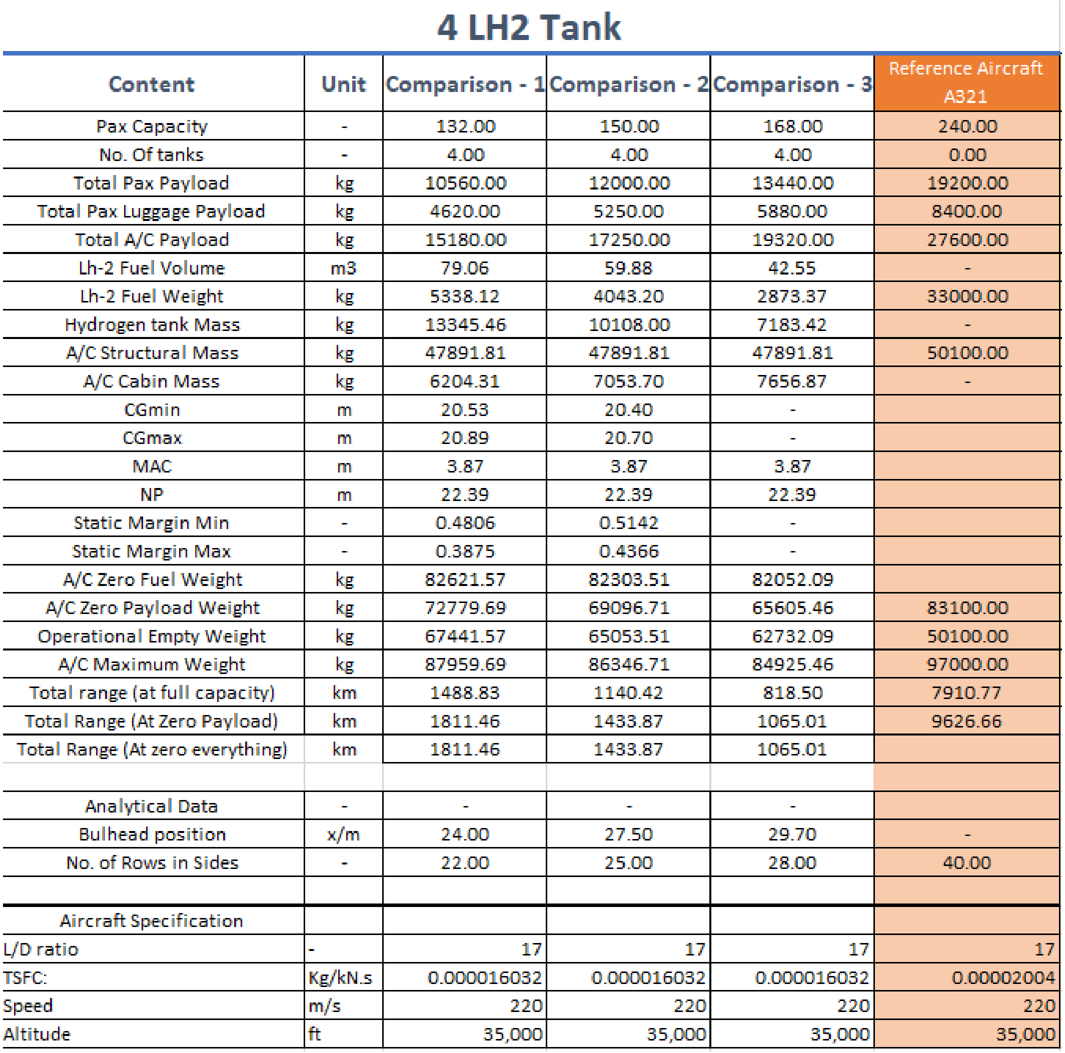
Figure 89: Data Sheet of Variation of Pax capacity (132, 150, 168 and 192 Pax) for fixed tank configuration (1 Tank)



3 LH2 Tank

Content	Unit	Comparison - 1	Comparison - 2	Comparison - 3	Reference Aircraft A321
Pax Capacity	-	132.00	150.00	168.00	240.00
No. Of tanks	-	3.00	3.00	3.00	0.00
Total Pax Payload	kg	10560.00	12000.00	13440.00	19200.00
Total Pax Luggage Payload	kg	4620.00	5250.00	5880.00	8400.00
Total A/C Payload	kg	15180.00	17250.00	19320.00	27600.00
Lh-2 Fuel Volume	m3	76.91	68.87	49.90	-
Lh-2 Fuel Weight	kg	5936.28	4650.44	3369.70	33000.00
Hydrogen tank Mass	kg	14840.70	11626.10	8424.26	-
A/C Structural Mass	kg	47891.81	47891.81	47891.81	50100.00
A/C Cabin Mass	kg	6204.31	7053.70	7656.87	-
CGmin	m	20.64	20.51	20.38	
CGmax	m	21.03	20.86	20.66	
MAC	m	3.87	3.87	3.87	
NP	m	22.39	22.39	22.39	
Static Margin Min	-	0.4521	0.4858	0.5194	
Static Margin Max	-	0.3513	0.3952	0.4470	
A/C Zero Fuel Weight	kg	84116.82	83821.61	83292.94	
A/C Zero Payload Weight	kg	74873.10	71222.05	67342.64	83100.00
Operational Empty Weight	kg	68936.82	66571.61	63972.94	50100.00
A/C Maximum Weight	kg	90053.10	88472.05	86662.64	97000.00
Total range (at full capacity)	km	1621.64	1284.03	943.10	7910.77
Total Range (At Zero Payload)	km	1964.34	1605.74	1220.72	9626.66
Total Range (At zero everything)	km	1964.34	1605.74	1220.72	
Analytical Data	-	-	-	-	
Bulhead position	x/m	24.00	27.50	29.70	-
No. of Rows in Sides	-	22.00	25.00	28.00	40.00
Aircraft Specification					
L/D ratio	-	17	17	17	17
TSFC:	Kg/kN.s	0.000016032	0.000016032	0.000016032	0.00002004
Speed	m/s	220	220	220	220
Altitude	ft	35,000	35,000	35,000	35,000

Figure 90: Data Sheet of Variation of Pax capacity (132, 150, 168 and 192 Pax) for fixed tank configuration (1 Tank)



4 LH2 Tank

Content	Unit	Comparison - 1	Comparison - 2	Comparison - 3	Reference Aircraft A321
Pax Capacity	-	132.00	150.00	168.00	240.00
No. Of tanks	-	4.00	4.00	4.00	0.00
Total Pax Payload	kg	10560.00	12000.00	13440.00	19200.00
Total Pax Luggage Payload	kg	4620.00	5250.00	5880.00	8400.00
Total A/C Payload	kg	15180.00	17250.00	19320.00	27600.00
Lh-2 Fuel Volume	m3	79.06	59.88	42.55	-
Lh-2 Fuel Weight	kg	5338.12	4043.20	2873.37	33000.00
Hydrogen tank Mass	kg	13345.46	10108.00	7183.42	-
A/C Structural Mass	kg	47891.81	47891.81	47891.81	50100.00
A/C Cabin Mass	kg	6204.31	7053.70	7656.87	-
CGmin	m	20.53	20.40	-	
CGmax	m	20.89	20.70	-	
MAC	m	3.87	3.87	3.87	
NP	m	22.39	22.39	22.39	
Static Margin Min	-	0.4806	0.5142	-	
Static Margin Max	-	0.3875	0.4366	-	
A/C Zero Fuel Weight	kg	82621.57	82303.51	82052.09	
A/C Zero Payload Weight	kg	72779.69	69096.71	65605.46	83100.00
Operational Empty Weight	kg	67441.57	65053.51	62732.09	50100.00
A/C Maximum Weight	kg	87959.69	86346.71	84925.46	97000.00
Total range (at full capacity)	km	1488.83	1140.42	818.50	7910.77
Total Range (At Zero Payload)	km	1811.46	1433.87	1065.01	9626.66
Total Range (At zero everything)	km	1811.46	1433.87	1065.01	
Analytical Data	-	-	-	-	
Bulhead position	x/m	24.00	27.50	29.70	-
No. of Rows in Sides	-	22.00	25.00	28.00	40.00
Aircraft Specification					
L/D ratio	-	17	17	17	17
TSFC:	Kg/kN.s	0.000016032	0.000016032	0.000016032	0.00002004
Speed	m/s	220	220	220	220
Altitude	ft	35,000	35,000	35,000	35,000

Figure 91: Data Sheet of Variation of Pax capacity (132, 150, 168 and 192 Pax) for fixed tank configuration (1 Tank)

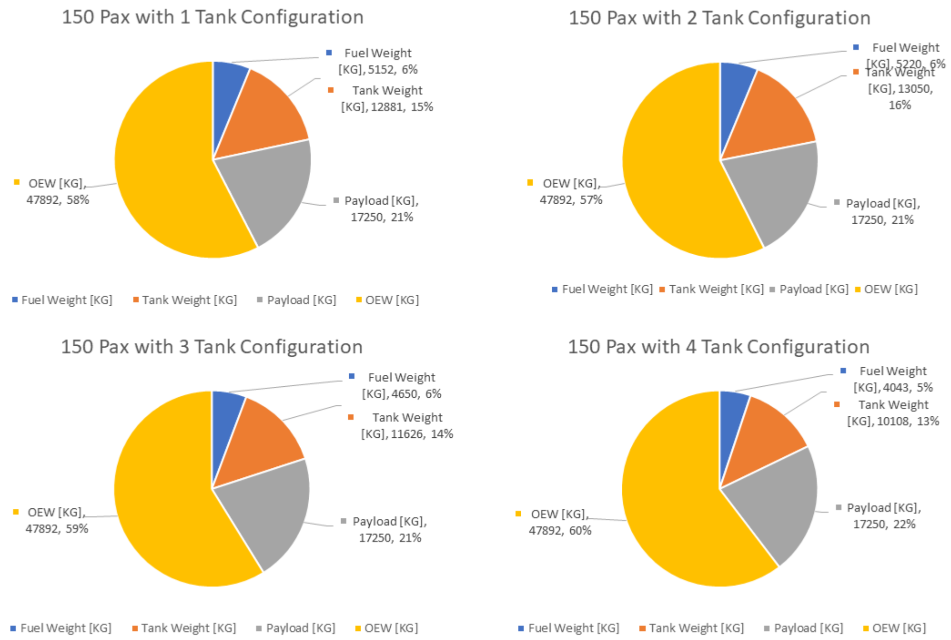


Figure 92: Weight distribution for 150 pax with different tank configurations

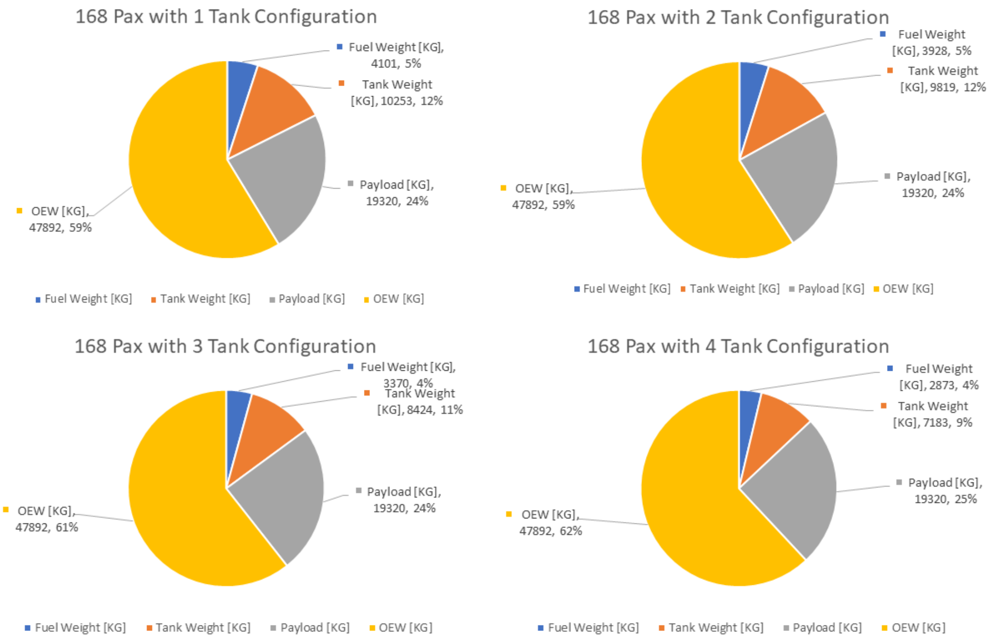


Figure 93: Weight distribution for 164 pax with different tank configurations

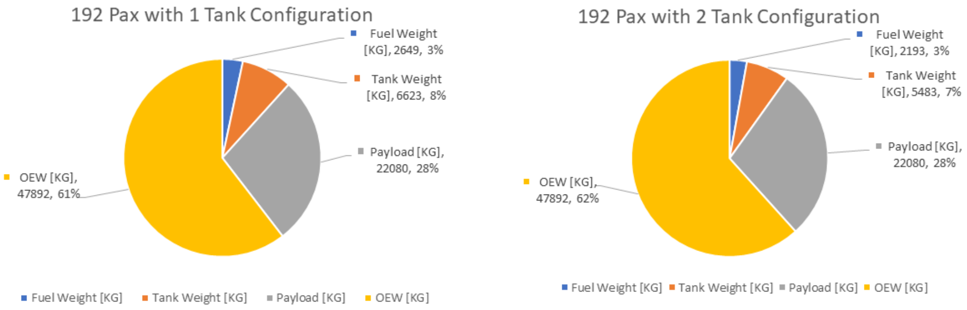
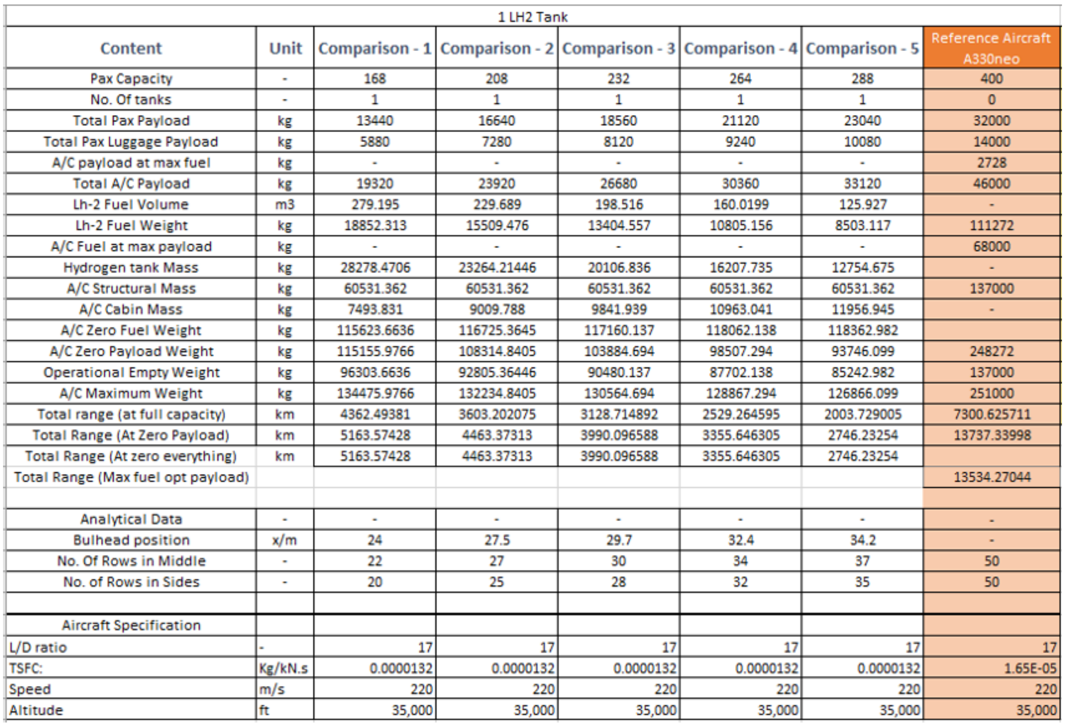


Figure 94: Weight distribution for 192 pax with different tank configurations

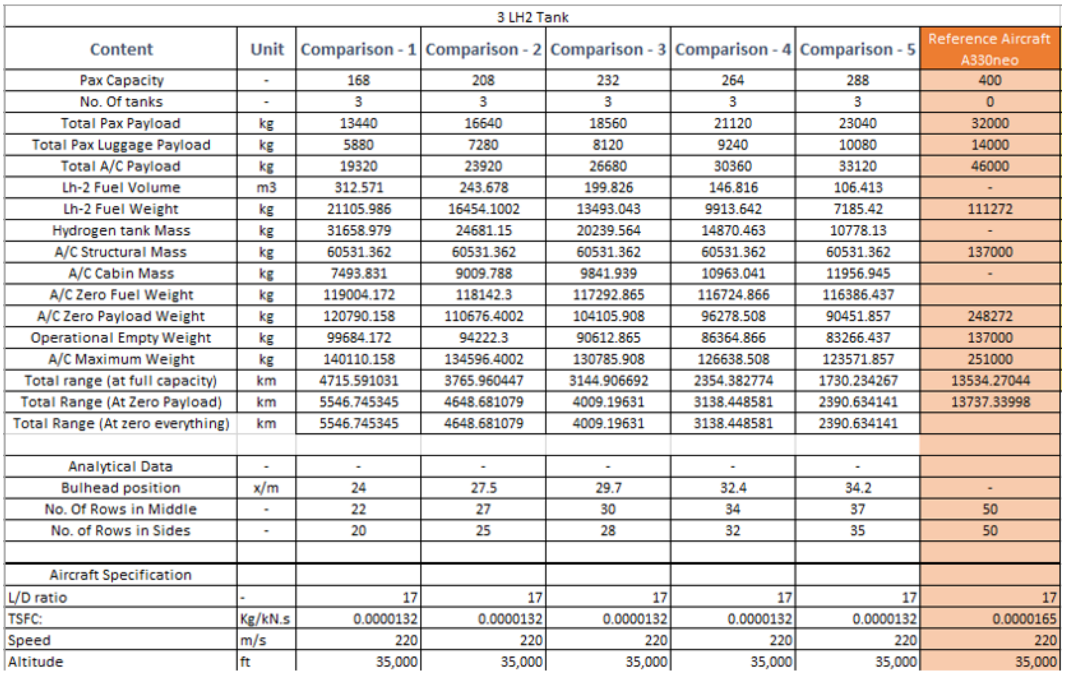
A.1.3 Twin Aisle Configuration

In this section you can find the data sheets, 2D layouts, Payload-Range Diagram and weight distributions studies carried out on different variants in twin-aisle configuration.



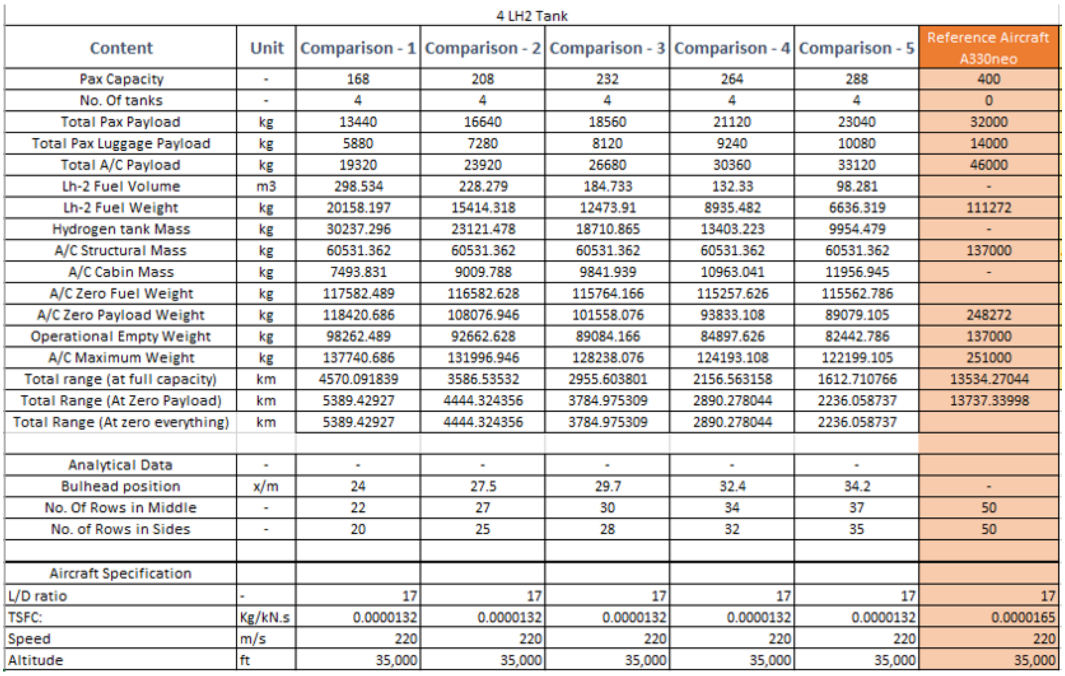
1 LH2 Tank							
Content	Unit	Comparison - 1	Comparison - 2	Comparison - 3	Comparison - 4	Comparison - 5	Reference Aircraft A330neo
Pax Capacity	-	168	208	232	264	288	400
No. Of tanks	-	1	1	1	1	1	0
Total Pax Payload	kg	13440	16640	18560	21120	23040	32000
Total Pax Luggage Payload	kg	5880	7280	8120	9240	10080	14000
A/C payload at max fuel	kg	-	-	-	-	-	2728
Total A/C Payload	kg	19320	23920	26680	30360	33120	46000
Lh-2 Fuel Volume	m3	279.195	229.689	198.516	160.0199	125.927	-
Lh-2 Fuel Weight	kg	18852.313	15509.476	13404.557	10805.156	8503.117	111272
A/C Fuel at max payload	kg	-	-	-	-	-	68000
Hydrogen tank Mass	kg	28278.4706	23264.21446	20106.836	16207.735	12754.675	-
A/C Structural Mass	kg	60531.362	60531.362	60531.362	60531.362	60531.362	137000
A/C Cabin Mass	kg	7493.831	9009.788	9841.939	10963.041	11956.945	-
A/C Zero Fuel Weight	kg	115623.6636	116725.3645	117160.137	118062.138	118362.982	
A/C Zero Payload Weight	kg	115155.9766	108314.8405	103884.694	98507.294	93746.099	248272
Operational Empty Weight	kg	96303.6636	92805.36446	90480.137	87702.138	85242.982	137000
A/C Maximum Weight	kg	134475.9766	132234.8405	130564.694	128867.294	126866.099	251000
Total range (at full capacity)	km	4362.49381	3603.202075	3128.714892	2529.264595	2003.729005	7300.625711
Total Range (At Zero Payload)	km	5163.57428	4463.37313	3990.096588	3355.646305	2746.23254	13737.33998
Total Range (At zero everything)	km	5163.57428	4463.37313	3990.096588	3355.646305	2746.23254	
Total Range (Max fuel opt payload)							13534.27044
Analytical Data	-	-	-	-	-	-	-
Bulhead position	x/m	24	27.5	29.7	32.4	34.2	-
No. Of Rows in Middle	-	22	27	30	34	37	50
No. of Rows in Sides	-	20	25	28	32	35	50
Aircraft Specification							
L/D ratio	-	17	17	17	17	17	17
TSFC:	Kg/kN.s	0.0000132	0.0000132	0.0000132	0.0000132	0.0000132	1.65E-05
Speed	m/s	220	220	220	220	220	220
Altitude	ft	35,000	35,000	35,000	35,000	35,000	35,000

Figure 95: Study data collected for 1 tank configuration and different pax capacities



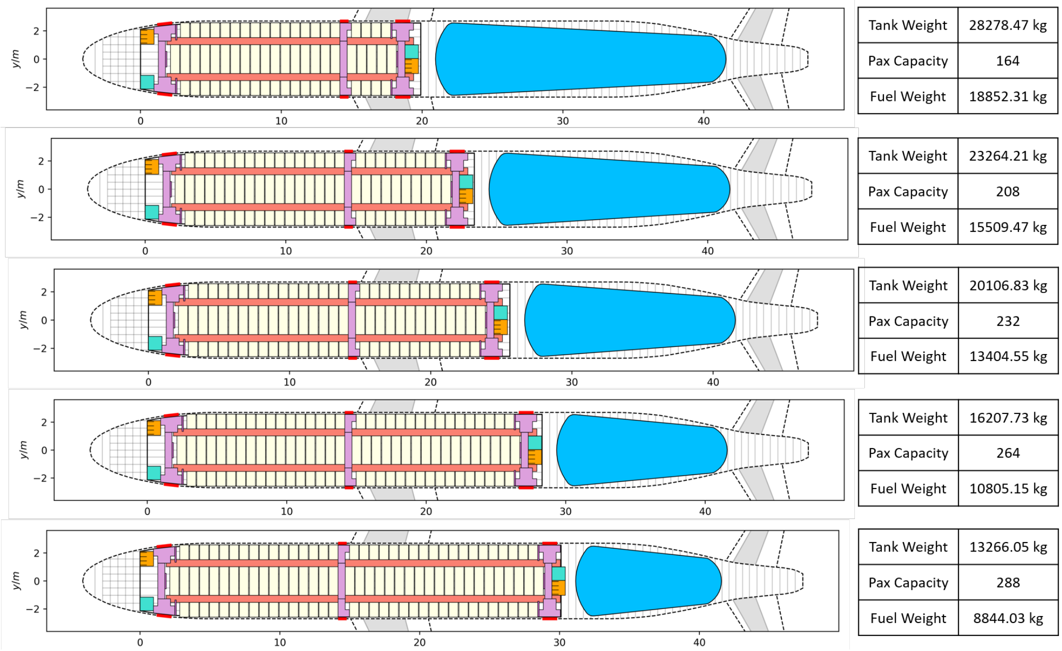
3 LH2 Tank							
Content	Unit	Comparison - 1	Comparison - 2	Comparison - 3	Comparison - 4	Comparison - 5	Reference Aircraft A330neo
Pax Capacity	-	168	208	232	264	288	400
No. Of tanks	-	3	3	3	3	3	0
Total Pax Payload	kg	13440	16640	18560	21120	23040	32000
Total Pax Luggage Payload	kg	5880	7280	8120	9240	10080	14000
Total A/C Payload	kg	19320	23920	26680	30360	33120	46000
Lh-2 Fuel Volume	m3	312.571	243.678	199.826	146.816	106.413	-
Lh-2 Fuel Weight	kg	21105.986	16454.1002	13493.043	9913.642	7185.42	111272
Hydrogen tank Mass	kg	31658.979	24681.15	20239.564	14870.463	10778.13	-
A/C Structural Mass	kg	60531.362	60531.362	60531.362	60531.362	60531.362	137000
A/C Cabin Mass	kg	7493.831	9009.788	9841.939	10963.041	11956.945	-
A/C Zero Fuel Weight	kg	119004.172	118142.3	117292.865	116724.866	116386.437	
A/C Zero Payload Weight	kg	120790.158	110676.4002	104105.908	96278.508	90451.857	248272
Operational Empty Weight	kg	99684.172	94222.3	90612.865	86364.866	83266.437	137000
A/C Maximum Weight	kg	140110.158	134596.4002	130785.908	126638.508	123571.857	251000
Total range (at full capacity)	km	4715.591031	3765.960447	3144.906692	2354.382774	1730.234267	13534.27044
Total Range (At Zero Payload)	km	5546.745345	4648.681079	4009.19631	3138.448581	2390.634141	13737.33998
Total Range (At zero everything)	km	5546.745345	4648.681079	4009.19631	3138.448581	2390.634141	
Analytical Data	-	-	-	-	-	-	
Bulhead position	x/m	24	27.5	29.7	32.4	34.2	-
No. Of Rows in Middle	-	22	27	30	34	37	50
No. of Rows in Sides	-	20	25	28	32	35	50
Aircraft Specification							
L/D ratio	-	17	17	17	17	17	17
TSFC:	Kg/kN.s	0.0000132	0.0000132	0.0000132	0.0000132	0.0000132	0.0000165
Speed	m/s	220	220	220	220	220	220
Altitude	ft	35,000	35,000	35,000	35,000	35,000	35,000

Figure 96: Study data collected for 3 tank configuration and different pax capacities



		4 LH2 Tank					
Content	Unit	Comparison - 1	Comparison - 2	Comparison - 3	Comparison - 4	Comparison - 5	Reference Aircraft A330neo
Pax Capacity	-	168	208	232	264	288	400
No. Of tanks	-	4	4	4	4	4	0
Total Pax Payload	kg	13440	16640	18560	21120	23040	32000
Total Pax Luggage Payload	kg	5880	7280	8120	9240	10080	14000
Total A/C Payload	kg	19320	23920	26680	30360	33120	46000
Lh-2 Fuel Volume	m3	298.534	228.279	184.733	132.33	98.281	-
Lh-2 Fuel Weight	kg	20158.197	15414.318	12473.91	8935.482	6636.319	111272
Hydrogen tank Mass	kg	30237.296	23121.478	18710.865	13403.223	9954.479	-
A/C Structural Mass	kg	60531.362	60531.362	60531.362	60531.362	60531.362	137000
A/C Cabin Mass	kg	7493.831	9009.788	9841.939	10963.041	11956.945	-
A/C Zero Fuel Weight	kg	117582.489	116582.628	115764.166	115257.626	115562.786	
A/C Zero Payload Weight	kg	118420.686	108076.946	101558.076	93833.108	89079.105	248272
Operational Empty Weight	kg	98262.489	92662.628	89084.166	84897.626	82442.786	137000
A/C Maximum Weight	kg	137740.686	131996.946	128238.076	124193.108	122199.105	251000
Total range (at full capacity)	km	4570.091839	3586.53532	2955.603801	2156.563158	1612.710766	13534.27044
Total Range (At Zero Payload)	km	5389.42927	4444.324356	3784.975309	2890.278044	2236.058737	13737.33998
Total Range (At zero everything)	km	5389.42927	4444.324356	3784.975309	2890.278044	2236.058737	
Analytical Data	-	-	-	-	-	-	
Bulhead position	x/m	24	27.5	29.7	32.4	34.2	-
No. Of Rows in Middle	-	22	27	30	34	37	50
No. of Rows in Sides	-	20	25	28	32	35	50
Aircraft Specification							
L/D ratio	-	17	17	17	17	17	17
TSFC:	Kg/kN.s	0.0000132	0.0000132	0.0000132	0.0000132	0.0000132	0.0000165
Speed	m/s	220	220	220	220	220	220
Altitude	ft	35,000	35,000	35,000	35,000	35,000	35,000

Figure 97: Study data collected for 4 tank configuration and different pax capacities



Tank Weight	28278.47 kg
Pax Capacity	164
Fuel Weight	18852.31 kg

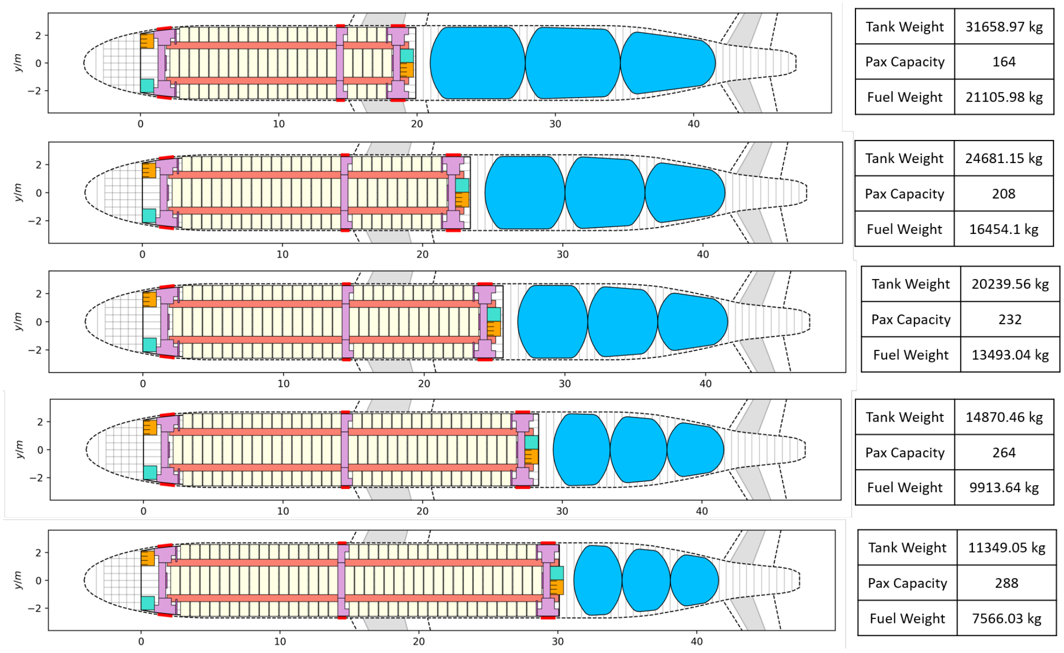
Tank Weight	23264.21 kg
Pax Capacity	208
Fuel Weight	15509.47 kg

Tank Weight	20106.83 kg
Pax Capacity	232
Fuel Weight	13404.55 kg

Tank Weight	16207.73 kg
Pax Capacity	264
Fuel Weight	10805.15 kg

Tank Weight	13266.05 kg
Pax Capacity	288
Fuel Weight	8844.03 kg

Figure 98: Pax variation with respect to 1 tank configuration.



Tank Weight	31658.97 kg
Pax Capacity	164
Fuel Weight	21105.98 kg

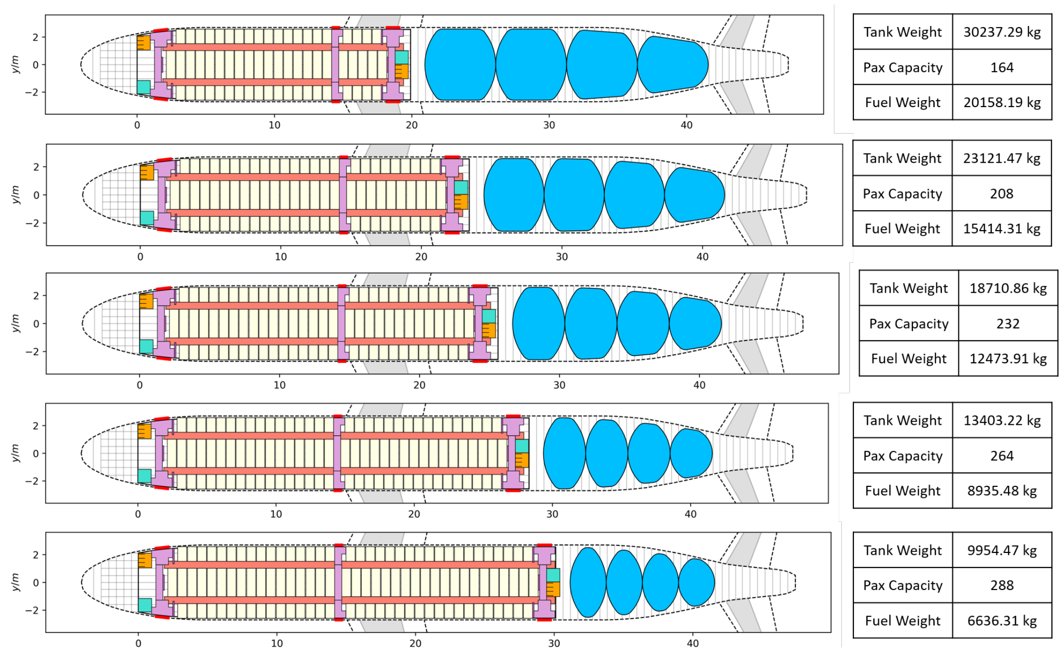
Tank Weight	24681.15 kg
Pax Capacity	208
Fuel Weight	16454.1 kg

Tank Weight	20239.56 kg
Pax Capacity	232
Fuel Weight	13493.04 kg

Tank Weight	14870.46 kg
Pax Capacity	264
Fuel Weight	9913.64 kg

Tank Weight	11349.05 kg
Pax Capacity	288
Fuel Weight	7566.03 kg

Figure 99: Pax variation with respect to 2 tank configuration.



Tank Weight	30237.29 kg
Pax Capacity	164
Fuel Weight	20158.19 kg

Tank Weight	23121.47 kg
Pax Capacity	208
Fuel Weight	15414.31 kg

Tank Weight	18710.86 kg
Pax Capacity	232
Fuel Weight	12473.91 kg

Tank Weight	13403.22 kg
Pax Capacity	264
Fuel Weight	8935.48 kg

Tank Weight	9954.47 kg
Pax Capacity	288
Fuel Weight	6636.31 kg

Figure 100: Pax variation with respect to 4 tank configuration.

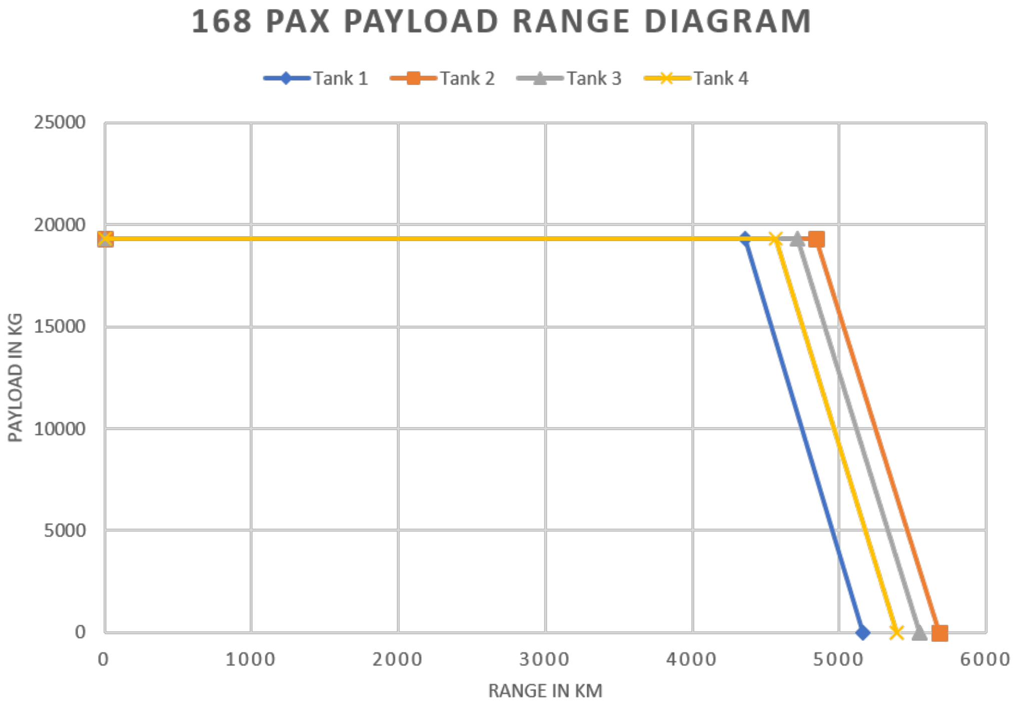


Figure 101: Payload Range variation with change in tank for 168 pax capacity

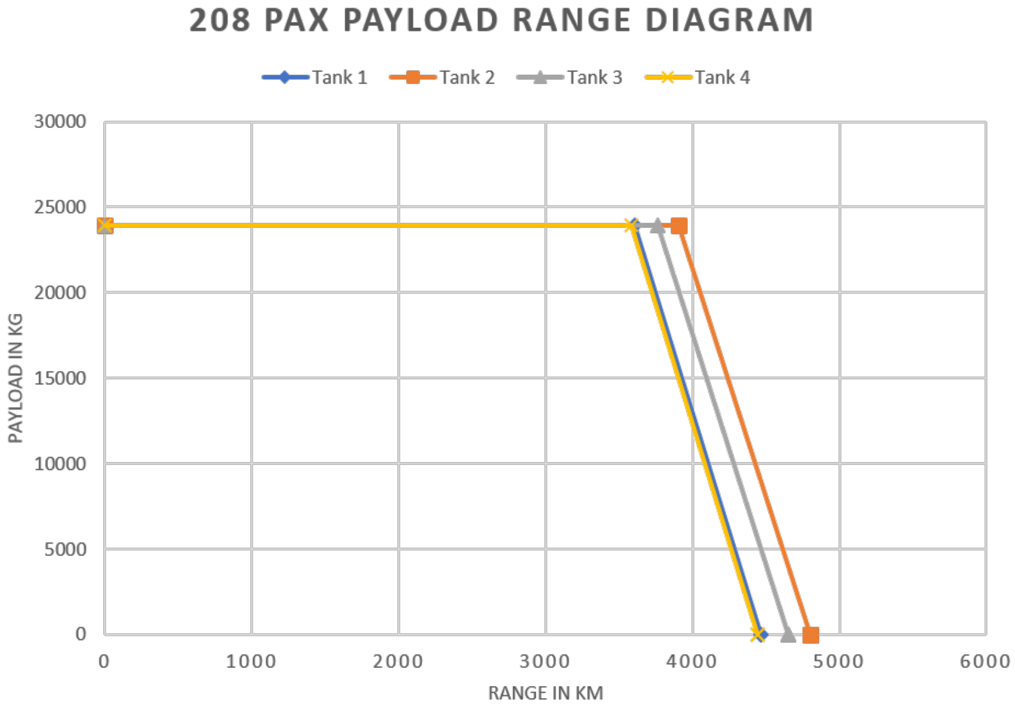


Figure 102: Payload Range variation with change in tank for 208 pax capacity

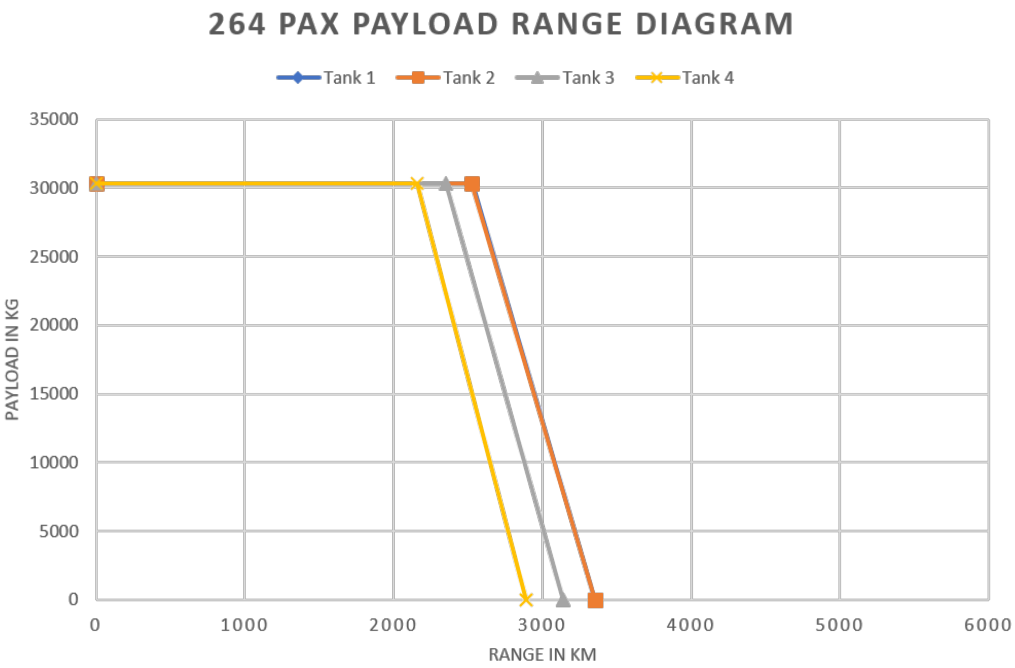


Figure 103: Payload Range variation with change in tank for 264 pax capacity

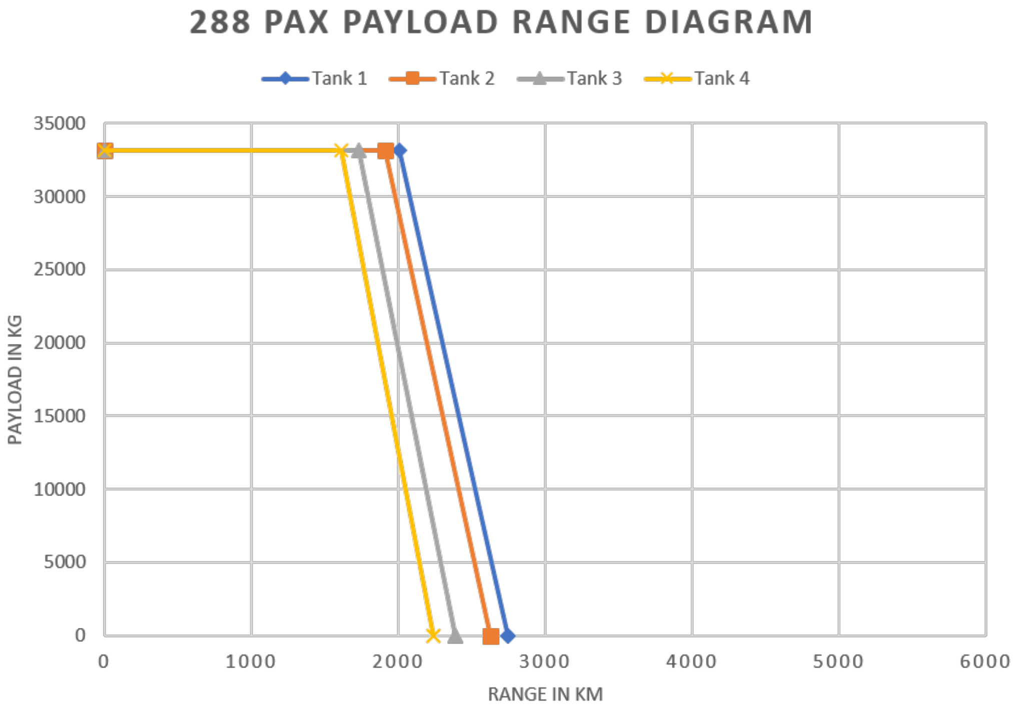
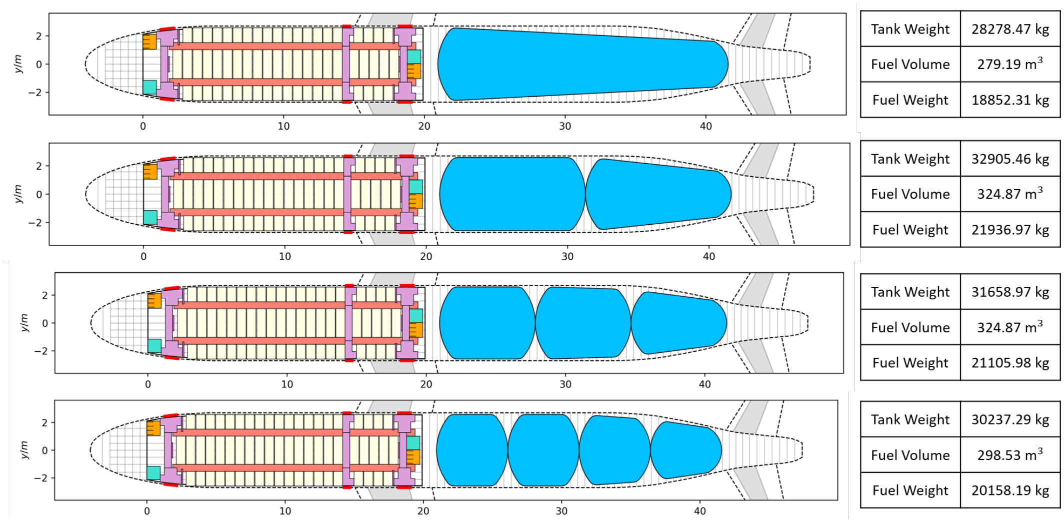


Figure 104: Payload Range variation with change in tank for 288 pax capacity



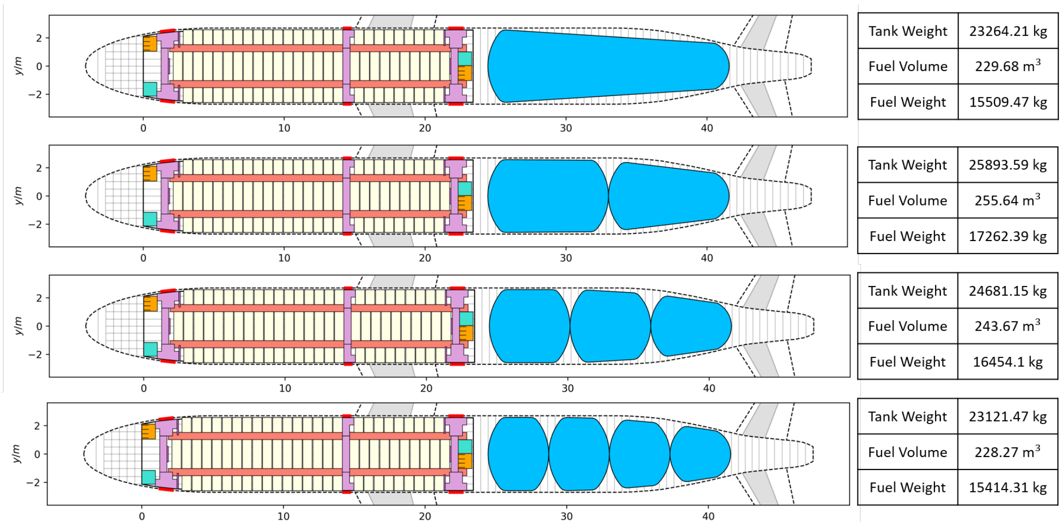
Tank Weight	28278.47 kg
Fuel Volume	279.19 m^3
Fuel Weight	18852.31 kg

Tank Weight	32905.46 kg
Fuel Volume	324.87 m^3
Fuel Weight	21936.97 kg

Tank Weight	31658.97 kg
Fuel Volume	324.87 m^3
Fuel Weight	21105.98 kg

Tank Weight	30237.29 kg
Fuel Volume	298.53 m^3
Fuel Weight	20158.19 kg

Figure 105: Cabin view for 168 pax and variation in tank



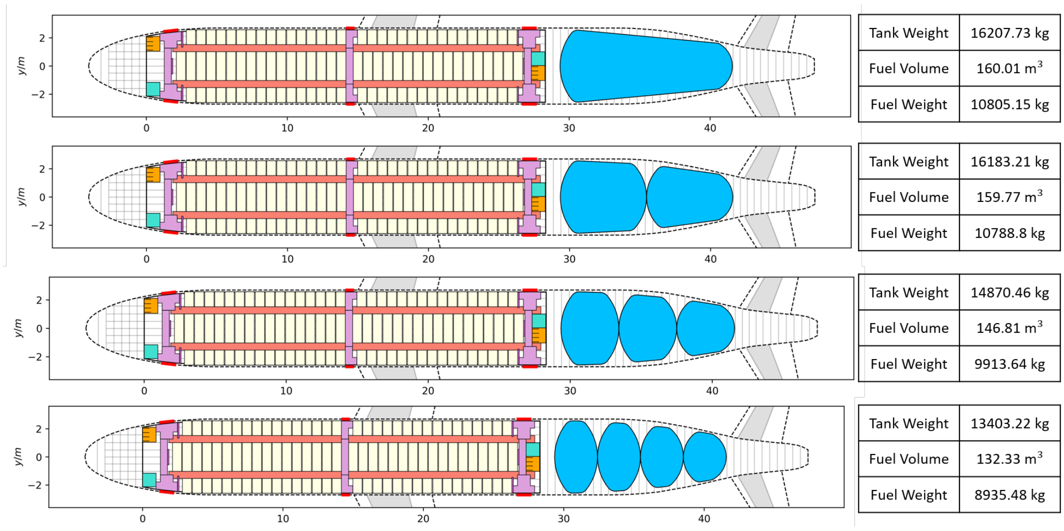
Tank Weight	23264.21 kg
Fuel Volume	229.68 m^3
Fuel Weight	15509.47 kg

Tank Weight	25893.59 kg
Fuel Volume	255.64 m^3
Fuel Weight	17262.39 kg

Tank Weight	24681.15 kg
Fuel Volume	243.67 m^3
Fuel Weight	16454.1 kg

Tank Weight	23121.47 kg
Fuel Volume	228.27 m^3
Fuel Weight	15414.31 kg

Figure 106: Cabin view for 208 pax and variation in tank



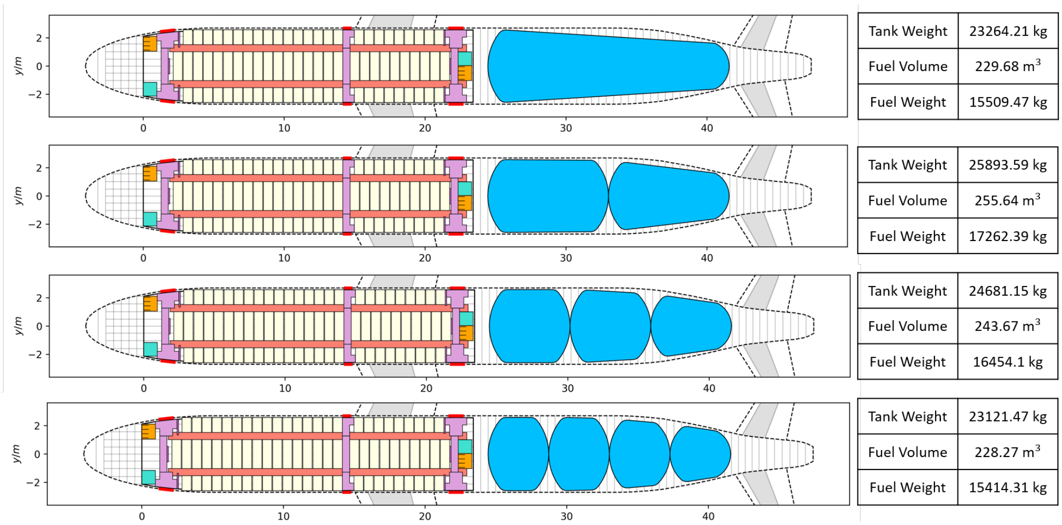
Tank Weight	16207.73 kg
Fuel Volume	160.01 m^3
Fuel Weight	10805.15 kg

Tank Weight	16183.21 kg
Fuel Volume	159.77 m^3
Fuel Weight	10788.8 kg

Tank Weight	14870.46 kg
Fuel Volume	146.81 m^3
Fuel Weight	9913.64 kg

Tank Weight	13403.22 kg
Fuel Volume	132.33 m^3
Fuel Weight	8935.48 kg

Figure 107: Cabin view for 264 pax and variation in tank



Tank Weight	23264.21 kg
Fuel Volume	229.68 m^3
Fuel Weight	15509.47 kg

Tank Weight	25893.59 kg
Fuel Volume	255.64 m^3
Fuel Weight	17262.39 kg

Tank Weight	24681.15 kg
Fuel Volume	243.67 m^3
Fuel Weight	16454.1 kg

Tank Weight	23121.47 kg
Fuel Volume	228.27 m^3
Fuel Weight	15414.31 kg

Figure 108: Cabin view for 208 pax and variation in tank

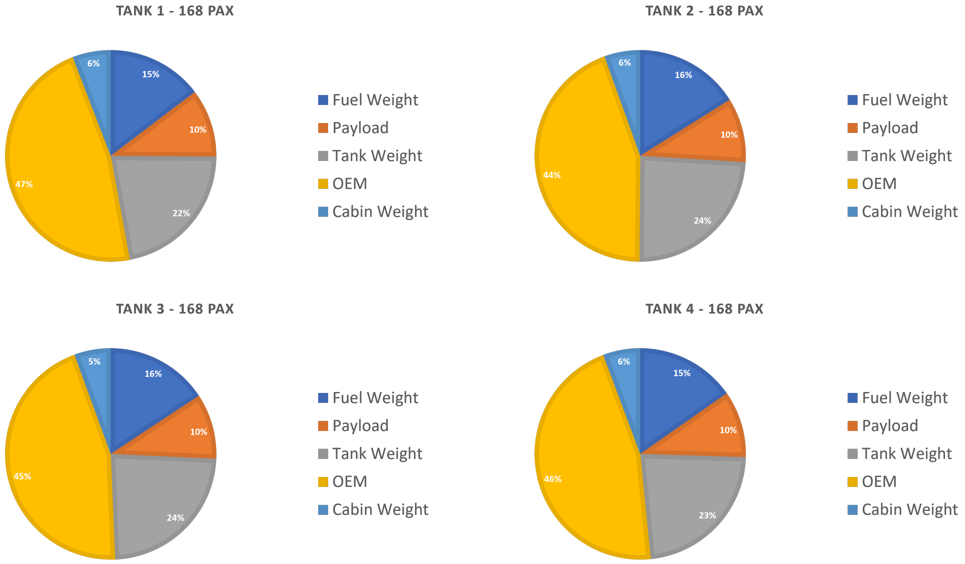


Figure 109: Weight distribution for 168 pax with different tank configurations

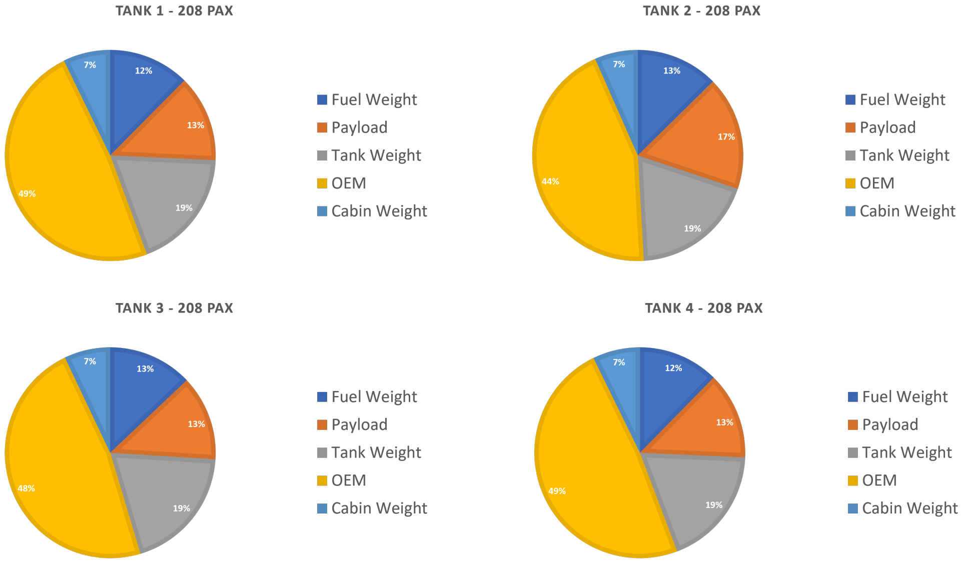


Figure 110: Weight distribution for 208 pax with different tank configurations

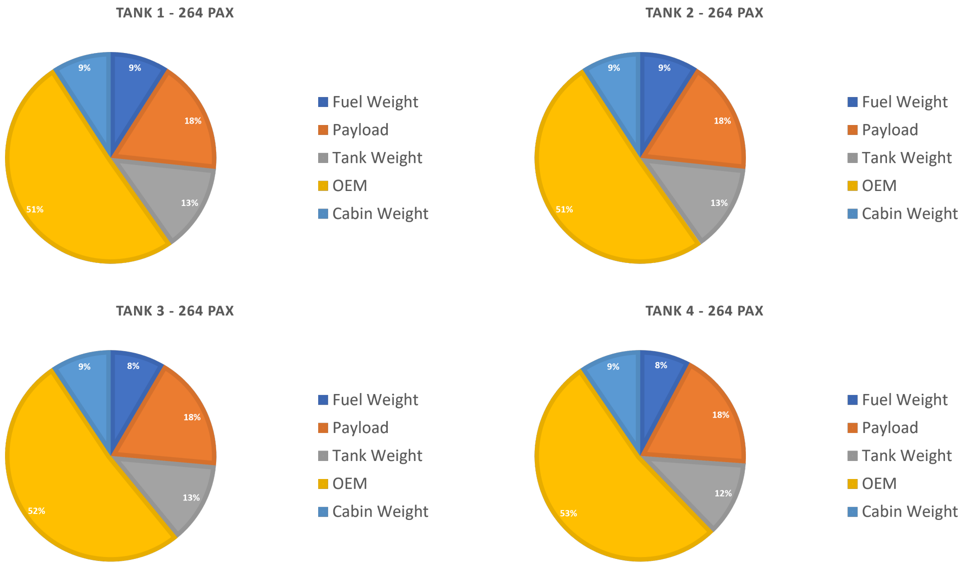


Figure 111: Weight distribution for 264 pax with different tank configurations

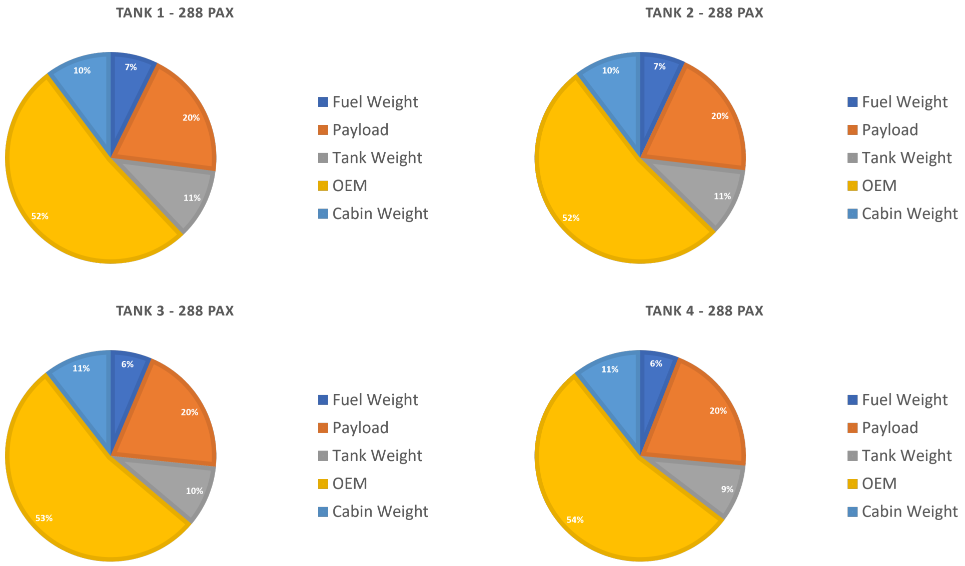


Figure 112: Weight distribution for 288 pax with different tank configurations