TECHNOLOGY ROADMAPPING

FOR A SUBORBITAL ROCKET WITH MULTIDISCIPLINARY DESIGN OPTIMISATION

Authors: Martin LEMAIRE
Ryan SVOBODA

Supervisor: Pr. Dr. Volker GASS
Co-Supervisor: Mathieu UDRIOT

Doc ID: 2024_C_SE_SP_TECH_ROADMAP_REPORT

Date: Spring 2024

EPFL ROCKET TEAM

Abstract

The EPFL Rocket Team (ERT) has set an ambitious goal of reaching space before the end of this decade. To accomplish this extraordinary feat, the team must adapt its current organisational structure to incorporate proactive strategic initiatives. These initiatives are crucial for ensuring the successful achievement of their objective, or at the very least, making significant progress towards it.

To set these strategic initiatives effectively, it is essential to develop a strong roadmapping framework. This report details the systems engineering process employed to create a comprehensive handbook designed for the application of technology roadmapping within a student-led technical association.

Additionally, there is an included case study that serves a dual purpose: it verifies the requirements established during the development process and provides a practical introduction to the application of the new framework for the association. By following the outlined framework, the ERT can enhance its strategic planning capabilities and move closer to realising its goal of reaching space.

[THIS PAGE INTENTIONALLY LEFT BLANK]

Acknowledgements

Technology Roadmapping and Systems Engineering are not common subjects in engineering schools, and nearly non-existent at the bachelor level anywhere around the world. As interesting as they can be, these topics are mostly covered by senior executives in large technology companies, and therefore not the target skills for engineering students.

We would like to start by thanking our supervisor, Pr. Dr. Volker Gass, for making this project possible, allowing us to pursue these subjects freely, and also for his guidance throughout the challenges that came with it. It has been a great learning opportunity for us and this journey revealed to be of very high value.

We would also like to thank our co-supervisor, Mathieu Udriot, from EPFL Space Center (eSpace), for his patience, kindness, and availability. Also being a former Systems Engineer of the EPFL Rocket Team (ERT), he had a comprehensive understanding of our problematics, and his supervision greatly helped along the project's milestones.

Finally, we would like to thank the Systems Engineers and the Head Management of the ERT for believing in the value of strategic planning for the future of our association.

[THIS PAGE INTENTIONALLY LEFT BLANK]

Contents

List of Figures

Abbreviations

- **SRR:** System Requirements Review
- **PDR:** Preliminary Design Review
- **CDR:** Critical Design Review
- **FRR:** Flight Readiness Review
- **PFR:** Post Flight Review
- **FOM:** Figure of Merit
- **DSM:** Dependency Structure Matrix or Design Structure Matrix
- **SVN:** Stakeholder Value Network
- **EPFL:** École Polytechnique Fédérale de Lausanne
- **ERT:** EPFL Rocket Team
- **eSpace:** EPFL Space Center
- **TCDC:** The Countdown Company
- **SKIL:** Student Kreativity and Innovation Laboratory
- **LPAC:** Laboratory for Processing of Advanced Composites
- **PH:** Physics Department of EPFL
- **ATPR:** Atelier de l'Institut de Production et Robotique
- **ATME:** Atelier de l'institut de génie Mécanique
- **EuRoC:** European Rocketry Challenge
- **MIT:** Massachussetts Institute of Technology
- **ESL:** Engineering Systems Laboratory, MIT
- **ATRA:** Advanced Technology Roadmap Architecture
- **ESA:** European Space Agency
- **CEO:** Chief Executive Officer
- **CTO:** Chief Technical Officer
- **SE:** Systems Engineer
- **MDO:** Multidisciplinary Design Optimisation
- **ARIS:** Akademische Raumfahrt Initiative Schweiz
- **DSD:** Design Solution Definition

[THIS PAGE INTENTIONALLY LEFT BLANK]

1 Introduction

The EPFL Rocket Team has set an ambitious goal: to launch a bi-liquid rocket into space before the decade concludes. Achieving such a feat requires not only bold innovation but also meticulous planning and engineering expertise. The project detailed in this report focuses on the development of a handbook and a suite of tools designed to offer a systematic approach to technology roadmapping. These resources aim to empower users with adaptable processes, facilitating more informed executive decision-making. Complemented by a case study, these tools serve as a robust foundation for future strategic planning work.

Our methodology throughout the development of these tools has been the application of the classical V-Model of Systems Engineering (See Figure 1), with adaptation to fit a Bachelor Project.

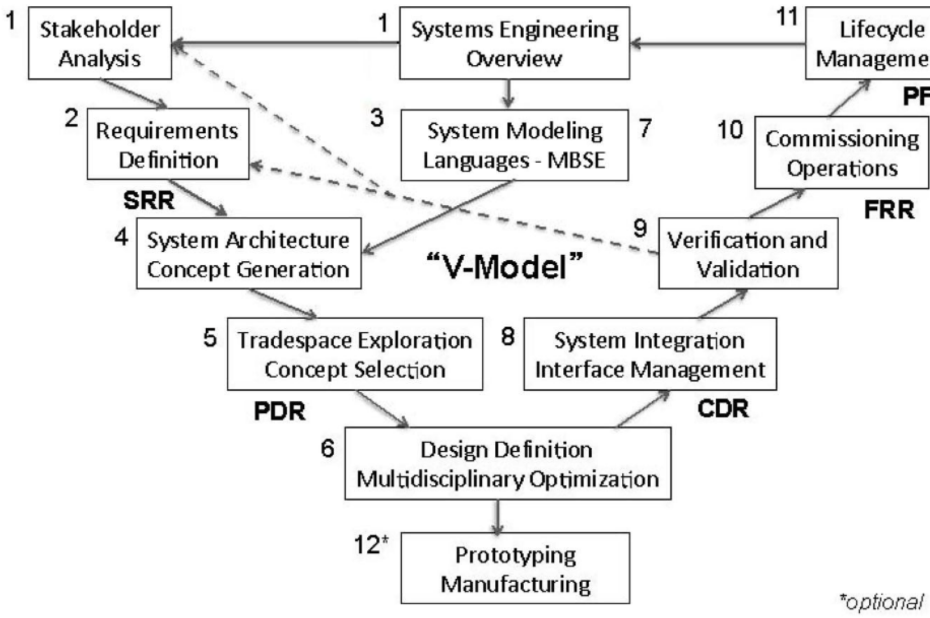


Figure 1: Traditional Systems Engineering's V-Model [1]

Following this model, the following timeline (See Figure 2) was applied, accompanied of the relevant reviews with the supervision committee (Pr. Dr. Volker Gass and Mathieu Udriot).

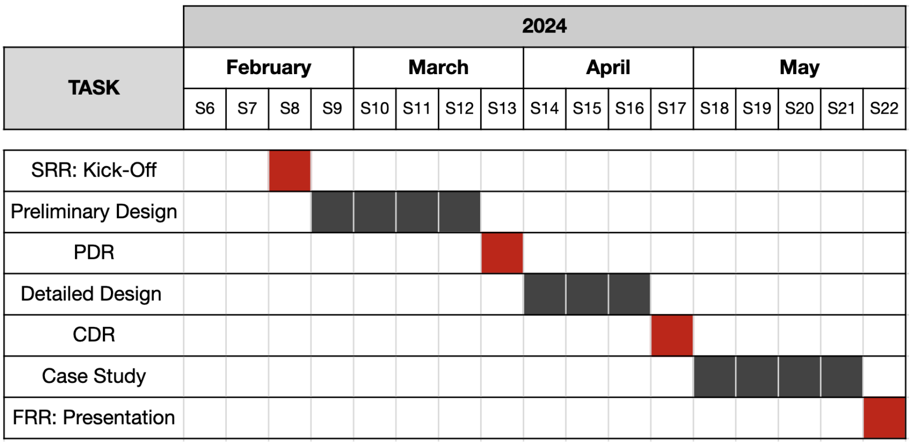


Figure 2: Gantt Chart of the Project

The structure of this report mirrors that of the V-model, encompassing a series of essential components. These include stakeholder analysis, requirements definition, system architecture, tradespace exploration, design definition, and a case study serving as verification and validation.

[THIS PAGE INTENTIONALLY LEFT BLANK]

2 Stakeholder Analysis

2.1 Purpose

To effectively shape project requirements in accordance with stakeholder expectations, a thorough stakeholder analysis is indispensable at project inception. This analysis goes beyond mere enumeration, instead delving into the intricate web of stakeholder relationships. By uncovering these connections, it enables a comprehensive understanding of the project's objectives and its contextual relevance. Therefore, the stakeholder analysis serves as a crucial initial step, ensuring that the project progresses in alignment with the varied interests and perspectives of its stakeholders.

2.2 Process Flowchart

The Figure 3 depicts the process used in this stakeholder analysis,

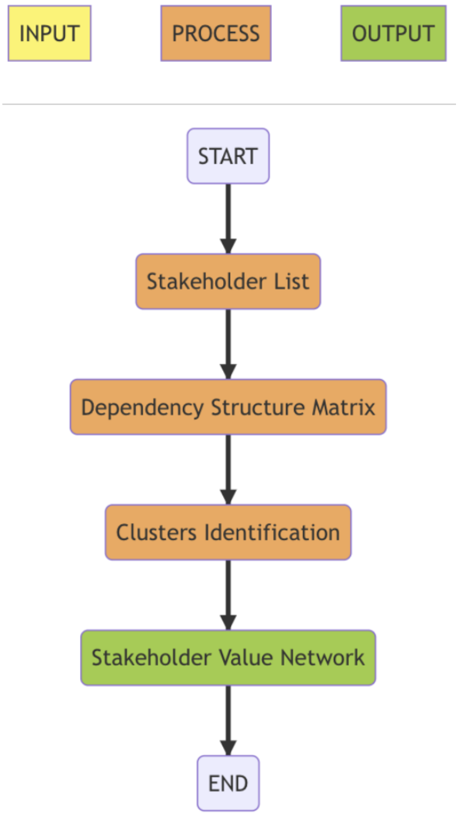


Figure 3: Flowchart of the Stakeholder Analysis Process

2.3 Stakeholders List

In this analysis, we'll consider that the project's scope encompasses the activities of the Systems Engineering team within the EPFL Rocket Team. Consequently, the primary focus will be on the Systems Engineers, while the remaining participants will be regarded as stakeholders. Additionally, to streamline oversight, the project's supervisory committee is integrated into the EPFL Space Center (eSpace) to prevent duplication.

To begin our analysis effectively, we need to gather data on all stakeholders associated with the EPFL Rocket Team's Systems Engineering team. This involves collecting information from the Sponsoring database, manufacturing partners, and academic collaborators. Once collected, we can generate a clear and comprehensive list of these entities (See list next page, 33 identified stakeholders).

- APCO Tech.
- EPFL MAKE
- Maxon
- Kistler
- 3D Precision
- CADFEM
- HeliAlps
- ARIS
- CF Turbo
- AVCO
- Meili Tech. SA
- SolidWorks
- Destinus
- Logitech
- Woodair
- Norelem
- Thyssenkrupp
- TCDC
- SKIL
- LPAC
- Section PH
- INARTIS
- Festo
- Loterie Romande
- ATPR & ATME
- Testing Facility
- EuRoC
- ESL (ATRA)
- ESA
- ERT Alumni
- eSpace
- ERT's CEO
- ERT's Technical Team

2.4 Dependency Structure Matrix

In order to facilitate manipulation and analysis of the network, we will map the previous list as a DSM (See Figure 4).

	APCO	3D_PREC	MAXON	KISTLER	CFTURBO	CADFEM	AVCO	SW	DESTINUS	NORELEM	THK	FESTO	LOGITECH	MEILI	TCDC	ESA	ALUMNI	ARIS	EUROC	ESL	PH_Dept	MAKE	SKIL	eSpace	SE	CEO	Tech Team	LPAC	ATELIER	INARTIS	HELIALPS	LOTERIE	WOODAIR	Test_Faci
APCO	0	1	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
3D_PREC	1	0	0	1	0	1	0	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
MAXON	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0
KISTLER	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
CFTURBO	0	0	0	1	0	1	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
CADFEM	0	1	1	1	1	0	0	0	1	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
AVCO	0	0	0	1	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
SW	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0
DESTINUS	1	0	0	1	0	1	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
NORELEM	1	0	0	1	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
THK	1	1	0	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
FESTO	1	1	0	1	1	1	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
LOGITECH	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
MEILI	1	1	0	0	0	1	0	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
TCDC	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
ESA	1	1	0	1	1	1	1	1	1	0	1	0	0	1	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
ALUMNI	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
ARIS	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
EUROC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
ESL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
PH_Dept	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0
MAKE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0
SKIL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0
eSpace	1	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1	1	1	0	1	1	1	0	1	0	0	0	0	0
SE	1	1	1	1	1	1	0	1	1	1	1	0	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1	0	1	1	1	0	0	1
CEO	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1	0	1	1	0	1	0	0	1	1	1	1	0
Tech Team	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	0	0	1
LPAC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0
ATELIER	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0
INARTIS	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
HELIALPS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
LOTERIE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
WOODAIR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Test_Faci	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

Figure 4: 34x34 Dependency Structure Matrix of the 33 Stakeholders and the SE team (Based on publicly available information, and ERT Sponsoring Team's information)

2.5 Stakeholder Clusters Identification

Numerous techniques are available for discerning clusters within a Dependency Structure Matrix, often referred to as an adjacency matrix. The approach we'll employ involves clustering through a Force Directed Diagram. To accomplish this, we'll utilise the Mermaid State Diagram tool, which enables graph generation through a scripted format. This tool inherently employs a Force Directed Layout algorithm, ideally suited for our purpose. By employing the Python function below, we can seamlessly convert a DSM from Excel into a Mermaid script, facilitating the visualisation of clusters.

````python
def write_dependency_lines(dsm_matrix, labels, output_file):
    with open(output_file, 'w') as f:
        f.write(f"```mermaid\nstateDiagram\n")
        for i, row in enumerate(dsm_matrix):
            for j, val in enumerate(row):
                if val == 1:
                    f.write(f"{labels[j]} --> {labels[i]}\n")
        f.write("```\n")
````

With the resulting graph, we can manually identify clusters and merge the concerned stakeholders using an OR logic.

We can iterate with this method, using the algorithm on the output until the network is fully connected (i.e. all cases of the DSM are equal to 1) and all stakeholders are classified (See Figure 5).

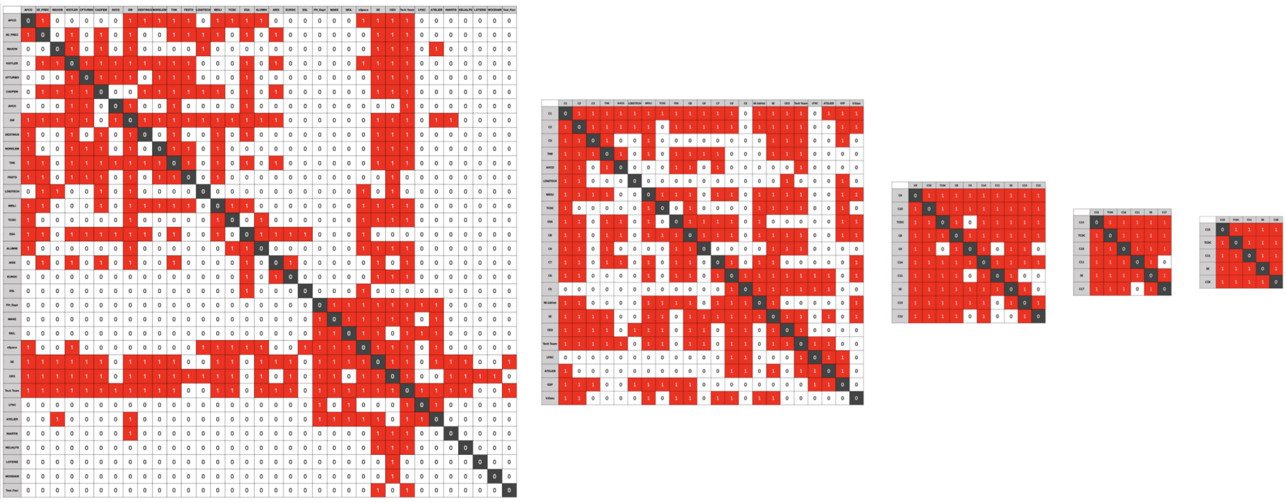


Figure 5: DSM clustering algorithm iterations (Left to right, purpose of illustration, not readability)

2.6 Stakeholder Value Network

With the clusters now identified, we can construct the Stakeholder Value Network (See Figure 6). So far, we haven't factored in the specifics of the connections due to the limitations of publicly available data. However, data provided by the ERT Sponsoring Team and our own knowledge enables us to precisely define these connections, particularly those directly linked to the ERT.

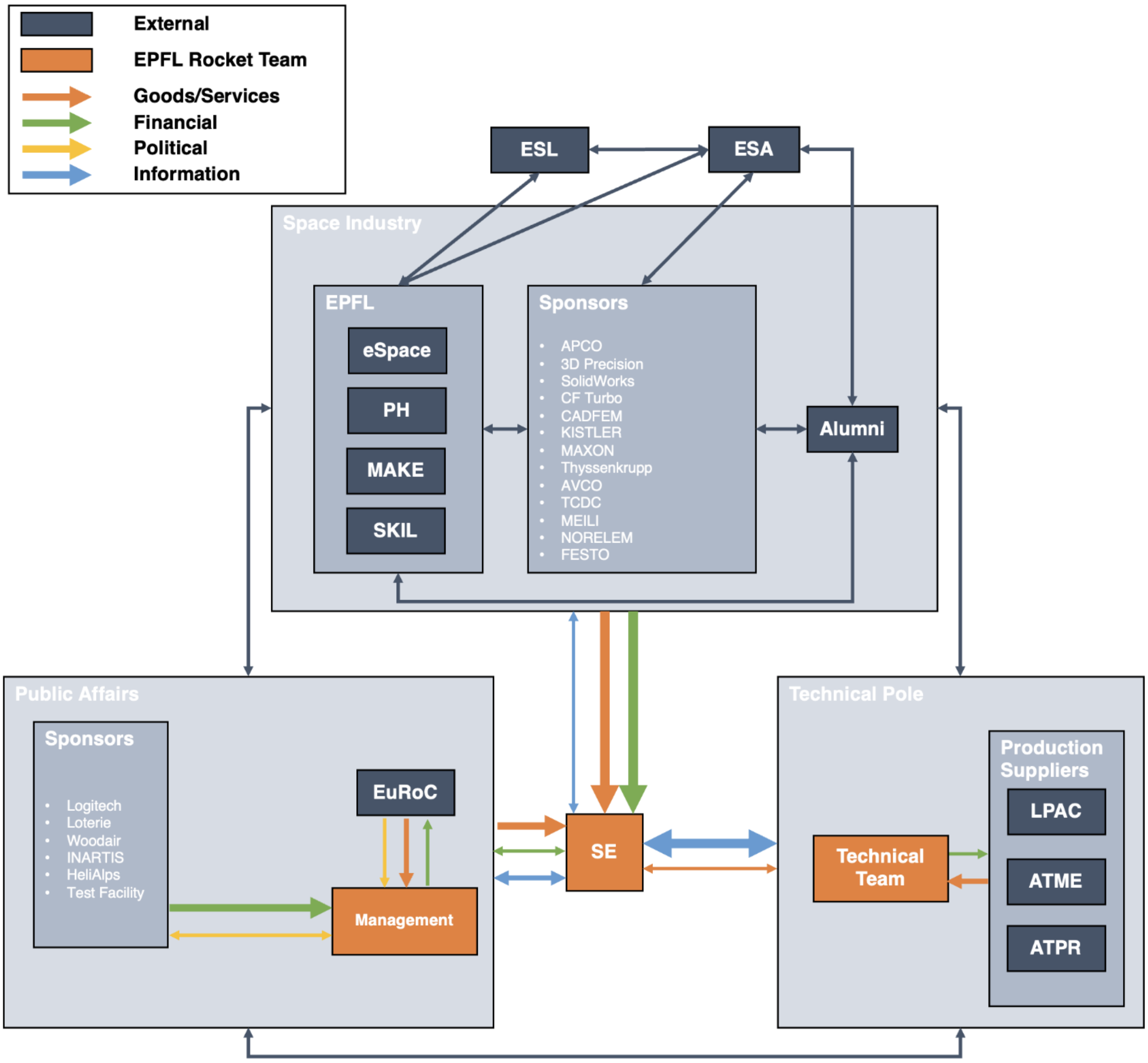


Figure 6: Stakeholder Value Network of the Project

3 Requirements Definition

- *Type* - **ID** - Short Description

  Full description:

- *Need* - **2024_SE_SP_NEED_01** - Need for technology roadmap

  Full description: The EPFL Rocket Team's systems engineering team needs a way to make informed long term plans. In order to be able to optimise resource and budget allocation in order to reach their objectives more effectively.

- *Goal* - **2024_SE_SP_GOAL_01** - User's workload (initial)

  Full description: It should take a single user less than 60 hours to establish a technology roadmap.

- *Goal* - **2024_SE_SP_GOAL_02** - Time efficient (automation)

  Full description: Establishing the initial roadmap should not require repetitive "manual" "boring" work, what can be automated should be automated

- *Goal* - **2024_SE_SP_GOAL_03** - Update work load

  Full description: It should take 16 hours or less for a SE to understand and update a roadmap

- *Goal* - **2024_SE_SP_GOAL_04** - Ease of use (Time)

  Full description: It should take a team member 2 hours or less to understand the outcomes of the roadmap for their specific subsystem.

- *Goal* - **2024_SE_SP_GOAL_05** - Filtered information

  Full description: It should take the CEO 2 hours or less to understand the future needs of the technical team

- *Goal* - **2024_SE_SP_GOAL_06** - Process Portability

  Full description: The process should be portable to other projects in different fields and of different scale.

- *Goal* - **2024_SE_SP_GOAL_07** - Proof of work

  Full description: It should be easy to show to an outsider what work went into establishing the technological roadmap

- *Goal* - **2024_SE_SP_GOAL_08** - Instruction clarity

  Full description: The instructions used to establish the roadmap shall be clear and justify why the the step should be done as instructed.

- *Goal* - **2024_SE_SP_GOAL_09** - Example

  Full description: An example for each step should be provided.

- *Goal* - **2024_SE_SP_GOAL_10** - Intuitive tools

  Full description: It should be as intuitive as possible to use the tools.

- *Goal* - **2024_SE_SP_GOAL_11** - Tool ease of access

  Full description: The tool should require the least amount of extra software to be able to use.

- *Goal* - **2024_SE_SP_GOAL_12** - Tool portability

  Full description: Tool should be able to work on MacOS and Windows.

- *Goal* - **2024_SE_SP_GOAL_13** - Information availability

  Full description: A single SE should be able to create a roadmap without needing to contact additional members of the association.

- *Goal* - **2024_SE_SP_GOAL_14** - Shareability (decisions)

  Full description: A document should be made for each step of the process.

- *Goal* - **2024_SE_SP_GOAL_15** - Shareability (time to understand)

  Full description: There should be a clear and concise graphs which summaries the results of each step of the process

- *Goal* - **2024_SE_SP_GOAL_16** - Information gathered (Financial cost)

  Full description: A ball park estimation for how much money it will cost to develop a technology on the chosen path shall be made

- *Goal* - **2024_SE_SP_GOAL_17** - Information gathered (Human resources cost)

  Full description: A ball park estimation for how many man hours it will cost to develop a technology on the chosen path shall be made

- *Goal* - **2024_SE_SP_GOAL_18** - Information gathered (criticality)

  Full description: An estimate on how critical/necessary a technology is for the chosen objective shall be made

- *Goal* - **2024_SE_SP_GOAL_19** - Information gathered (time)

  Full description: An estimate on how much time it will take to develop the technologies on the chosen path shall be made

- *Goal* - **2024_SE_SP_GOAL_20** - Information gathered (dependencies)

  Full description: The key dependancies in between technologies (and projects) on the chosen path shall be displayed on the roadmap.

- *Goal* - **2024_SE_SP_GOAL_21** - information gathered (facilities)

  Full description: The facilities needed for the development and use of a technology shall be displayed on the roadmap

- *Goal* - **2024_SE_SP_GOAL_22** - information gathered (manufacturing techniques)

  Full description: The main manufacturing techniques necessary for developing the technologies shall be displayed on the roadmap.

- *Goal* - **2024_SE_SP_GOAL_23** - information gathered (logistics capabilities)

  Full description: The main logistiques capabilities necessary for the development and use of a technology shall be displayed on the roadmap.

- *Goal* - **2024_SE_SP_GOAL_24** - information gathered (information transfer)

  Full description: The necessary skills/capabilitie/technologies to transfer to the future engineers of the rocket team.

- *Goal* - **2024_SE_SP_GOAL_25** - Information gathered (external dependencies)

  Full description: Show which technologies are dependant on stakeholders outside of rocket team.

- *Goal* - **2024_SE_SP_GOAL_26** - information gathered (sustainability)

  Full description: Evaluate and assess the sustainability impact of each technology included on the chosen path

- *Goal* - **2024_SE_SP_GOAL_27** - information gathered (technical skills required)

  Full description: Demonstrate which skills will be necessary for team members (understand what students are learning)

- *Goal* - **2024_SE_SP_GOAL_28** - Tool long-lifecycle (potential short comings)

  Full description: Be explicit on what are the potential short comings/risks associated with the roadmap methodologies and tools.

- *Goal* - **2024_SE_SP_GOAL_29** - Tool long-lifecycle (updatable)

  Full description: Allow the tools to be easily modifiable and updateable

- *Goal* - **2024_SE_SP_GOAL_30** - different types of technologies (software and hardware)

  Full description: Support for different types of dicplines (e.g., software development, infrastructure upgrades).

- *Goal* - **2024_SE_SP_GOAL_31** - quantitative recommendations

  Full description: Data-driven recommendations for resource allocation based on roadmap priorities.

- *Goal* - **2024_SE_SP_GOAL_32** - Roadmap validity (date)

  Full description: The roadmap outputs should clearly indicate when the roadmap was done or updated

- *Goal* - **2024_SE_SP_GOAL_33** - Roadmap validity (author)

  Full description: The roadmap outputs should clearly indicate the author (or who made the different decisions which impacted the roadmap)

- *Goal* - **2024_SE_SP_GOAL_34** - exploration breadth and depth

  Full description: The roadmap should clearly display which technologies were explored and to which extent.

- *Goal* - **2024_SE_SP_GOAL_35** - update instructions

  Full description: There should be clear instructions on how and when to update a roadmap.

- *Goal* - **2024_SE_SP_GOAL_36** - timeline shifting

  Full description: Capability to adjust timelines and priorities easily.

- *Goal* - **2024_SE_SP_GOAL_37** - Technology roadmap validity

  Full description: Assurance that technical roadmaps align with the association's overall strategic goals.

- *Goal* - **2024_SE_SP_GOAL_38** - feedback, improving the roadmap

  Full description: There should be a clear feedback procedure for the roadmap or project plans. Which can be done by any team member

- *Goal* - **2024_SE_SP_GOAL_39** - ease of collaboration

  Full description: The TM should be able to understand how the roadmap is structured.

- *Goal* - **2024_SE_SP_GOAL_40** - goal of roadmap

  Full description: The TM should be able to understand why the roadmap exists.

- *Goal* - **2024_SE_SP_GOAL_41** - progress report

  Full description: Generate reports for project progress, delays, and achievements.

- *Goal* - **2024_SE_SP_GOAL_42** - reason for developing a technology

  Full description: The TM should be able to understand why they are developing the technologies that they are.

- *Goal* - **2024_SE_SP_GOAL_43** - access to filtered information

  Full description: Clear understanding of their project's objectives, scope, and constraints.

- *Goal* - **2024_SE_SP_GOAL_44** - responsibilities linked to roadmap

  Full description: Clearly defined project milestones and deadlines.

- *Goal* - **2024_SE_SP_GOAL_45** - information displayed (risk)

  Full description: A list of the main risks associated with developing the technologies on the chosen path shall be displayed on the roadmap.

- *Goal* - **2024_SE_SP_GOAL_46** - risk mitigation

  Full description: There should be a way of displaying mitigation strategies for key risks associated to different technologies on the roadmap.

- *Goal* - **2024_SE_SP_GOAL_47** - outcome (objective)

  Full description: After having applied the process and used the tools developped, there should be a clear longterm objective which was chosen based on quatifiable measures

- *Goal* - **2024_SE_SP_GOAL_48** - outcome (paths)

  Full description: After having applied the process and used the tools developped, there should be a clear set of technologie paths which lead from the technologies which the rocket team have already developeded to the objective.

- *Goal* - **2024_SE_SP_GOAL_49** - outcome (optimal path)

  Full description: After having applied the process and used the tools developped, there should be a quantifiably optimal technologie path which leads from the technologies which the rocket team have already developeded to the objective.

- *Goal* - **2024_SE_SP_GOAL_50** - outcome (Investment strategy)

  Full description: After having applied the process and used the tools developped, there should be a quantifiably optimal resource investment strategy for the development of technologies along the optimal path to the objective.

4 System Architecture

4.1 Purpose

The system architecture intends to set the general structure of the generated product. In our case, we aim to create a handbook that is accompanied by several MATLAB tools to automate certain tasks. Using the ATRA as the basis for our work, most of the architecture is imposed by this framework.

4.2 Process Flowchart

The Figure 7 depicts the process used to determine the system architecture,

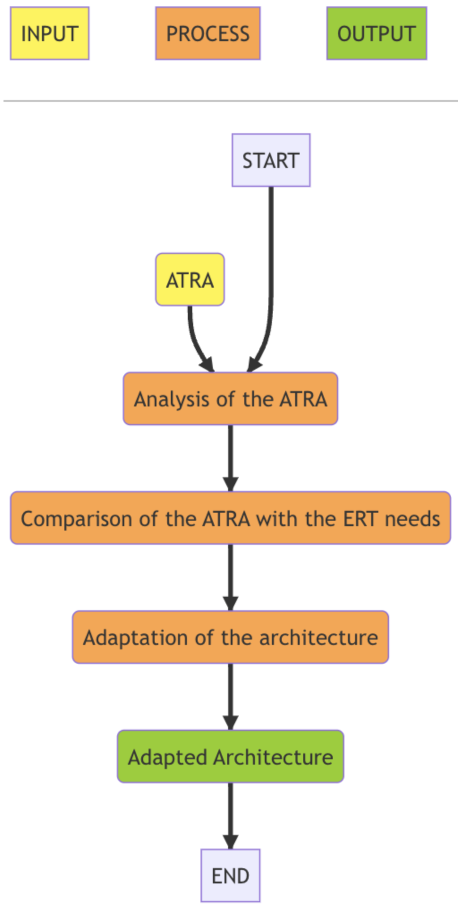


Figure 7: Flowchart of the System Architecture Process

4.3 Analysis of the MIT Advanced Technology Roadmap Architecture

The handbook is built upon the Advanced Technology Roadmap Architecture (ATRA, See Figure 8), derived from the book "Technology Roadmapping and Development: A Quantitative Approach to the Management of Technology" by Olivier L. De Weck [2], Apollo Program Professor of Astronautics and Engineering Systems at MIT.

The architecture comprises four primary axes:

1. Competitive Benchmarking
2. Systems Modeling
3. Scenario Analysis
4. Portfolio Optimisation

Competitive Benchmarking sets out to delineate performance benchmarks, aligning with stakeholder expectations and comparing the focal organisation against leading competitors in analogous sectors. Systems Modeling establishes a transfer function-like system to generate predefined performance benchmarks based on input parameters. Scenario Analysis utilises this model to craft pertinent scenarios conducive to the organisation's goals. Finally, with multiple scenario options available, Optimisation is undertaken to identify the most fitting technology portfolio.

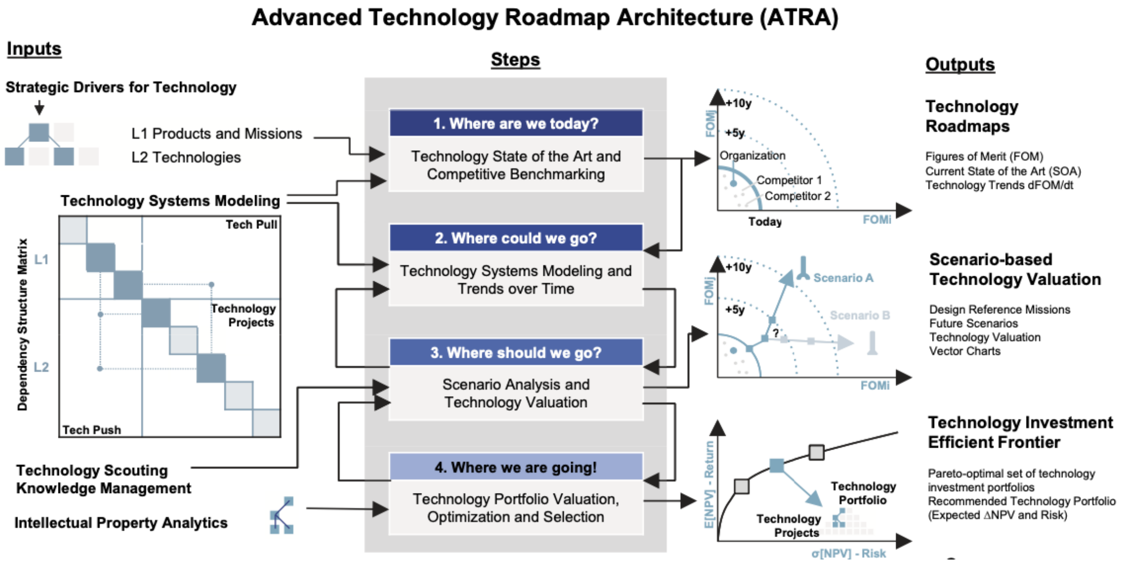


Figure 8: Advanced Technology Roadmap Architecture [2]

4.4 Comparison of the ATRA with ERT's needs

A few adaptation were made to make the process compliant with the ERT's needs and available resources.

The initial work package included in the Competitive Benchmarking part of the ATRA, which appears in the Figure 8, but also in the book (p.243), is:

- Figures of Merit
- State of the Art
- Current Best-in-class
- Technology Trends (dFOM/dt)
- Technology Readiness Level (TRL)
- Roadmap Maturity Score

Two key adjustments were implemented for this section. Firstly, the State of the Art analysis was refined into a Similar Missions Analysis. While similar to its predecessor, this approach differs in that it focuses on identifying technology advancements within a specific application context, with consideration of organisational resources.

Additionally, the maturity scoring for the roadmap was relocated to a new section dedicated to roadmap updates and enhancement instructions. This modification addresses the frequent turnover within the student association, ensuring that newcomers swiftly grasp the progress of ongoing work without undue effort deciphering prior assessments.

The system modeling component remains unchanged and integral as it stands. Its primary objective is to deliver precise Figures of Merit based on the design variables of the focal technology, a pivotal aspect of the process.

The package for the scenario analysis part is initially composed of:

- Design Reference Missions
- Project Value Propositions
- Technology Valuation
- Future Scenarios
- Vector Charts

This segment underwent the most significant alterations from the ATRA. Particularly in an industrial setting, scenario analysis entails thorough financial scrutiny and lengthy procedures. To align with the time constraints for roadmap generation, the focus of this section is narrowed down to pinpointing the degrees of freedom in technology development and estimating the time and effort necessary for each stage of progression.

Finally, the portfolio optimisation part is originally:

- Pareto-Optimal Set of Technology Portfolios
- Recommended Technology Portfolio (Expected financial return and risk)

The primary adjustments in this section pertain to its objectives and outcomes. While the optimisation process largely resembles its industrial counterpart, the objectives differ to accommodate the association's unique requirements. Unlike typical investment portfolio considerations, our focus extends to encompass human resource requirements, project relevance, and the delineation of clearer objectives for the sponsoring team. This involves providing them with the necessary financial and material resources to navigate the selected trajectory effectively.

4.5 Adapted System Architecture

Having pinpointed variances from the original ATRA version, we can now generate the definitive architecture employed in this project.

We can utilise a Design Structure Matrix (DSM) to depict the interactions among the various components of the handbook and the associated tools (See Figure 9).

	COMPETITIVE BENCHMARKING								MODELING	SCENARIO	TECHNOLOGY PORTFOLIO OPTIMISATION AND SELECTION								MAINTENANCE		
	FIGURES OF MERIT			SIMILAR MISSIONS ANALYSIS			TRL ASSESSMENT				PARETO OPTIMAL SET			RECOMMENDED PORTFOLIO			HR NEEDS		UPDATE	MATURITY	
	CHAPTER	MATLAB	TEMPLATE	CHAPTER	MATLAB	TEMPLATE	CHAPTER	TEMPLATE	CHAPTER	CHAPTER	CHAPTER	MATLAB	TEMPLATE	CHAPTER	MATLAB	TEMPLATE	CHAPTER	TEMPLATE	CHAPTER	CHAPTER	TEMPLATE
CHAPTER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MATLAB	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
TEMPLATE	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
CHAPTER	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MATLAB	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMPLATE	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CHAPTER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMPLATE	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CHAPTER	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
CHAPTER	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0
CHAPTER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MATLAB	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1	0	0
TEMPLATE	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0
CHAPTER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MATLAB	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0
TEMPLATE	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
CHAPTER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMPLATE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
CHAPTER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
CHAPTER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMPLATE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0

Figure 9: Design Structure Matrix representation of the System Architecture

A higher level representation of the system architecture can be made with a block diagram (See Figure 10), allowing easier visualisation of the interfaces.

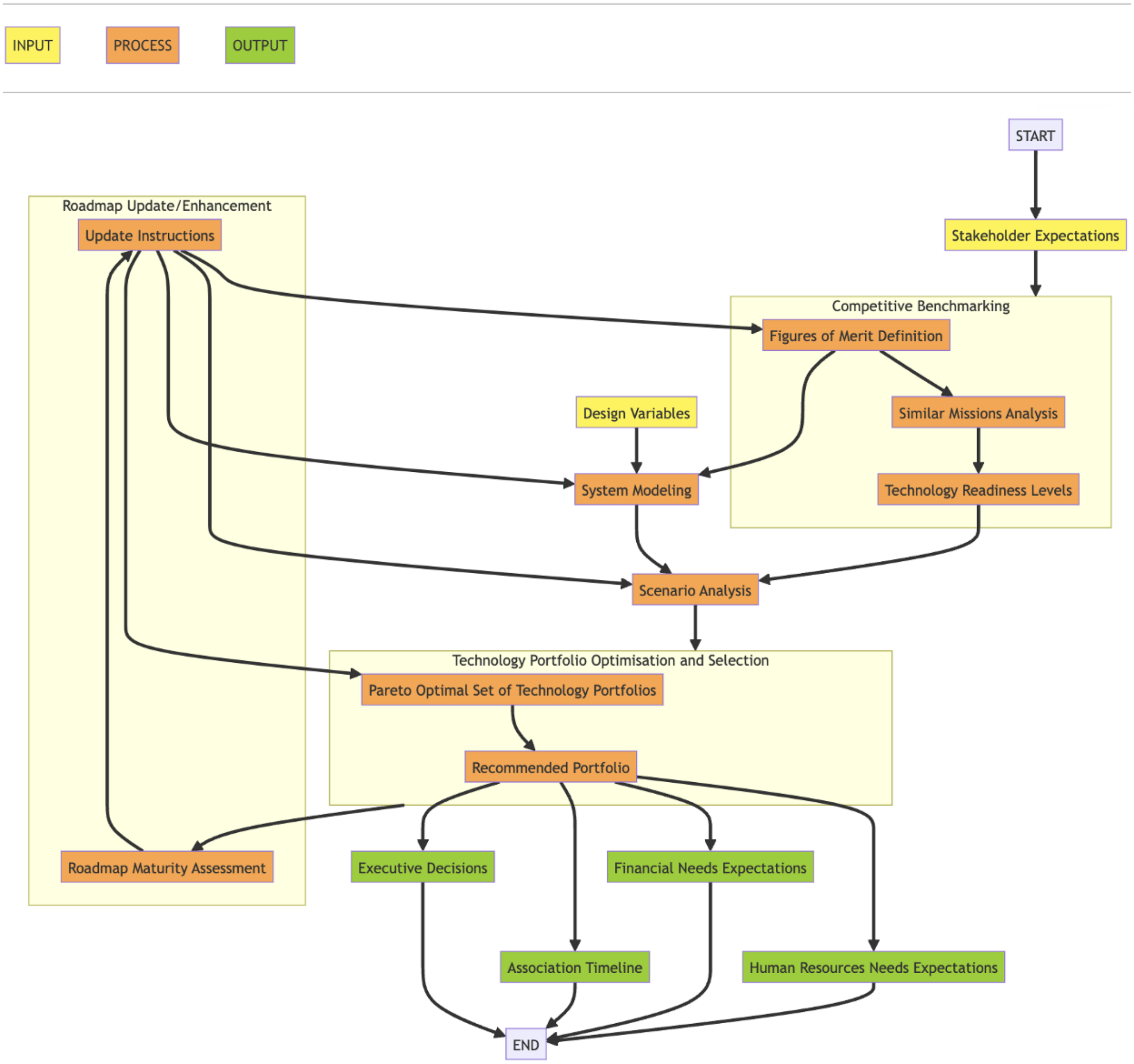


Figure 10: Adapted Version of the ATRA

5 Tradespace Exploration

5.1 Purpose

We have now defined the high level architecture of our system and can start the preliminary design phase of the project. This phase consists of the establishment of preliminary concepts and the exploration of possibilities that meet the project's requirements. This part of the project is concluded by the Preliminary Design Review (PDR), and is followed by a phase of detailed design.

5.2 Process Flowchart

The Figure 11 depicts the process used for the tradespace exploration,

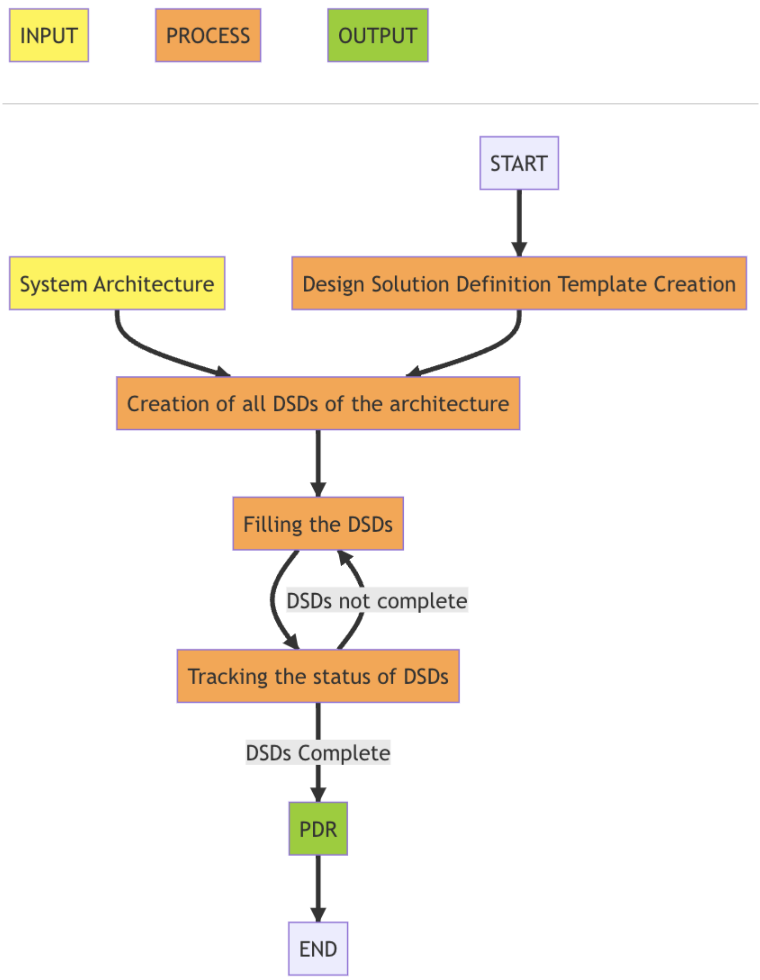


Figure 11: Flowchart of the Tradespace Exploration Process

5.3 Design Solution Definition

5.3.1 Purpose

With the high level components of the system defined, we can systematically fill a template document with predefined sections and features to ensure that the whole architecture is covered for the PDR. This document is called a Design Solution Definition document [3].

5.3.2 DSD Document

The template document (see Appendix B) consists of five parts: Introduction, Relevant Resources, Preliminary Guide Structure, Software Flowchart, and Output Documentation.

The introduction is divided into two sections. The first section includes a purpose paragraph, which requires the user to articulate their understanding of the component's goal. This is crucial for quickly assessing the component's maturity during the Preliminary Design Review (PDR). The second section contains abbreviations and definitions, making the document accessible to external readers and potential reviewers.

The Relevant Resources section serves as a repository for all books, articles, and other materials that provide pertinent information for the component's design. This section is important for maintaining a comprehensive bibliography for the final design.

The Preliminary Guide Structure consists of two parts. The first part is a flowchart outlining the process described in the chapter. This flowchart offers a quick overview of the chapter's structure, facilitating better understanding and easier change management for the author. It includes inputs, processes, and outputs, helping the user to prepare the necessary documents. After the flowchart is completed, each process block should have a section containing the preliminary content of the component.

The Software Flowchart is a high-level representation of the processes, inputs, and outputs involved. This part aids reviewers in understanding the component and simplifies modifications before detailed design. It is optional, as not all components require or allow for software creation.

The Output Documentation section requires the writer to review the various process steps and the component's purpose. This section helps create a documentation template for users following the described process and ensures that the component's purpose is fulfilled.

5.4 Tracking of the Status of DSDs

It is important to track the status of all the DSDs in real time to make sure that the documentation is ready for the different project's milestones. It also allows fast overview of the general status of the project without having to dive into each document.

In our case, using the Wiki (online workspace) of the ERT, all the links of the DSDs are regrouped on a single page (See Figure 12) and allow the overview of the DSDs.

1 - Technology Competitive Benchmarking

1.1 - DSD - Figures of Merit --> STATE: COMPLETED

1.2 - DSD - Similar Missions Analysis --> STATE: COMPLETED

1.3 - DSD - Technology Readiness Levels --> STATE: COMPLETED

2 DSD - Technology Systems Modeling --> STATE: COMPLETED

3 DSD - Scenario Analysis --> STATE: Started

4 - Technology Portfolio Optimization and Selection

4.1 - DSD - Pareto Optimal Set of Technology Investment Portfolios --> STATE: COMPLETED

4.2 - DSD - Recommended Technology Portfolio --> STATE: COMPLETED

4.3 - DSD - Human Resources Needs Expectations --> STATE: COMPLETED

5 - Roadmap Maintenance

5.1 - DSD - Roadmap Maturity Scoring --> STATE: COMPLETED

5.2 - DSD - Roadmap Update Instructions --> STATE: COMPLETED

Figure 12: Screenshot of the DSD tracking page on ERT's wiki

5.5 Preliminary Design Review (PDR)

The PDR signifies the conclusion of the preliminary design phase and qualifies the project to proceed to the detailed design phase. At this point, all DSDs must be finalised for presentation to the review committee, providing a preliminary version of the final product.

Following the PDR, feedback from the reviewers is incorporated, and necessary adjustments to the DSDs are made before commencing the detailed design phase.

[THIS PAGE INTENTIONALLY LEFT BLANK]

6 Design Definition

6.1 Purpose

Following the PDR, the project advances to the Design Definition phase. At this stage, the concepts for the handbook chapters are established, software concepts are validated, and the documentation output is defined. While the process is similar to the previous phase, the focus shifts from exploration and overview to the development of final, operational designs. This phase culminates in the Critical Design Review (CDR) and is succeeded by a phase dedicated to verification and validation.

6.2 Process Flowchart

The Figure 13 depicts the process used for the tradespace exploration,

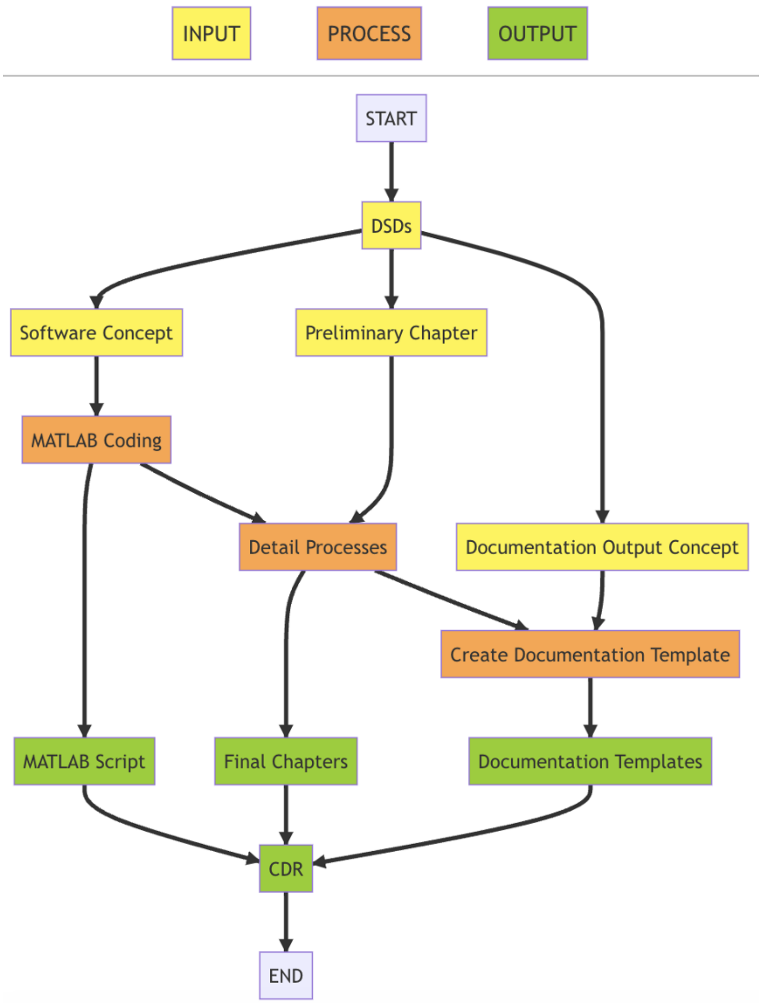


Figure 13: Flowchart of the Design Definition Process

6.3 Critical Design Review

The CDR marks the end of the design phase. During this review, the final version of the handbook is presented, including software and documentation templates. The feedback collected during the review is then implemented before entering the verification and validation phase, that will demonstrate that the system meets the project requirements.

[THIS PAGE INTENTIONALLY LEFT BLANK]

7 Verification and Validation

7.1 Purpose

At this stage, the handbook is complete, and all associated software and documentation templates are fully operational. However, the product cannot be delivered in its current form without a performance demonstration and sample output. To verify that the product meets the project requirements, a case study is conducted, applying the handbook's principles to a specific subject.

7.2 Process Flowchart

The Figure 14 depicts the process used for the tradespace exploration,

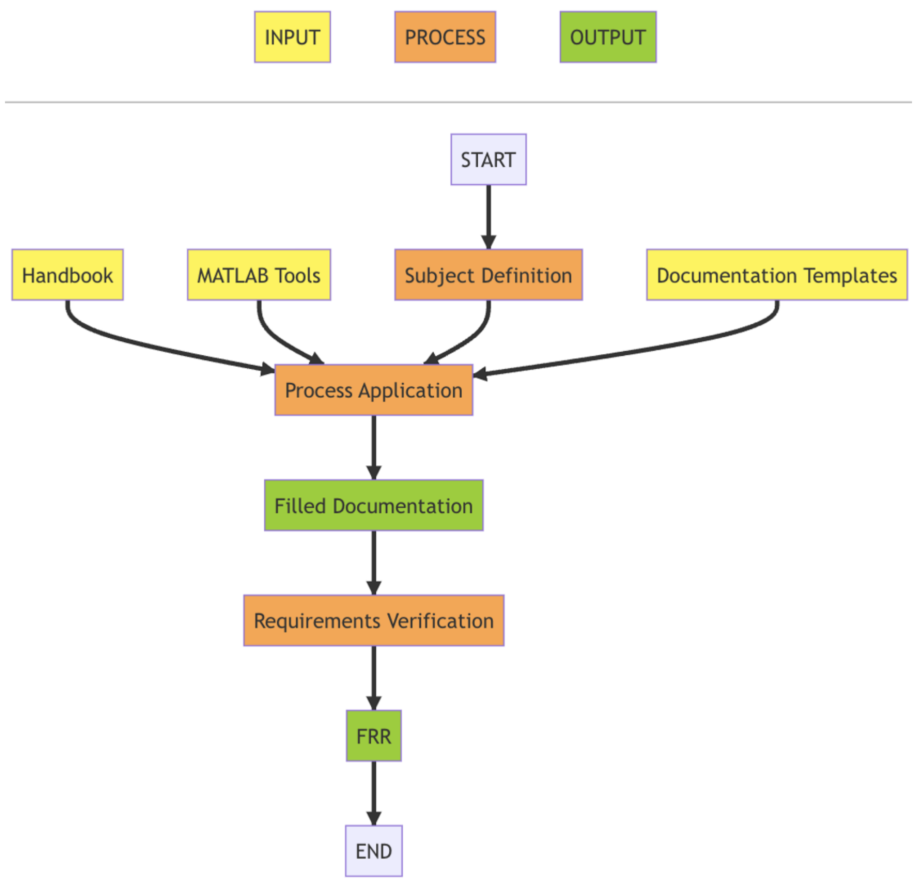


Figure 14: Flowchart of the Verification and Validation Process

7.3 Definition of the case

As the ultimate goal of the ERT is to reach space, performing a suborbital flight, we are going to apply the principles of the handbook to an initial roadmapping for the development of technologies allowing to reach space with a bi-liquid propulsion system.

Technology roadmapping aims to be applied to the whole portfolio of technologies to provide detailed insight for the general systems. Roadmaps should ultimately be established for every sub-assembly and every sub-system. However, an initial high level one has to be established to a certain extent, to guide the creation of other, lower level ones.

7.4 Results

The complete case study, which is composed of all the filled documentation templates for the defined subject can be found in the Appendix part of the handbook.

7.4.1 Competitive Benchmarking

For the first part of the process, the competitive benchmarking, 4 FOMs were defined: Peak Velocity, Altitude, Sustainability, and Cost. Data was available for most of these FOMs, with the exception of the sustainability rating. For this, a custom rating was established, ranging between 1 and 3, and representing the sustainability actions of the organisation. This represents an area where a more accurate model for sustainability should be established to obtain a better fitted FOM.

The main conclusion was that, if we only take the data of the FOMs of all the launch vehicles of the ERT and apply the technological s-curve prediction to it, meaning what will happen if the same progress rate continues until space is reached, we get a suborbital flight around 2037. This underlines the importance of setting stronger strategic initiatives in the association to reduce this issue.

The main difficulty regarding the comparison with competitors was the collection of data. Indeed, our participation at EuRoC allowed us to collect data from these flights, but they are limited by their categories. Therefore, in the future, data sources must be extended to obtain the performance of other competitors. For example, the Spaceport America Cup operates a 30km category, but specific flight data is hard to obtain.

7.4.2 Scenario Analysis

Due to the time limitations of this project, the scenario analysis was adapted and simplified. For this purpose, the stochastic model used to develop the portfolio optimization code was reused with parameters derived from the competitive benchmarking part of the case study. By applying a confidence interval to the predictions on the FOM, we were able to use the stochastic model within this interval. While this method does not guarantee only feasible solutions, it provides a high degree of confidence that we are close to reality. Consequently, we were able to generate a dataset of five different scenarios up to 2030.

7.4.3 Portfolio Optimisation

For the optimization part, the provided tool enabled full automation of the process, allowing us to obtain results very quickly and choose the recommended portfolio rapidly. This optimisation resulted in a portfolio expected to reach an altitude of 45 km by 2030, which is quite far from the initial goal of 100 km. This outcome highlights the critical importance of setting robust strategic planning to find a solution that leads to better performance growth.

7.5 Requirements Verification

Looking back at the requirements of the project, we were not able to verify a lot of these. This is probably due to the fact that we were not familiar with the topic at the beginning of the project and did not imagine that the project had such a large scale.

7.6 Flight Readiness Review - Final Presentation

The FRR marks the final milestone of our project, representing the qualification of the product to operate in its intended environment. For this review, the final version of the handbook with the integrated feedback from the CDR shall be ready, as well as the complete case study, with a summary of the results, to demonstrate that the requirements of the project are verified.

8 Conclusion

Our project aimed to create a handbook and associated tools to implement technology roadmapping for the EPFL Rocket Team. We adhered to the traditional V-Model of Systems Engineering, which structured our project into multiple phases, each concluded by an associated review.

Initially, the stakeholder analysis and requirements definition phase culminated in the System Requirements Review, which served as a kick-off meeting with the supervision committee. Next, the tradespace exploration phase, intended to generate a preliminary design of the handbook, was finalised by the Preliminary Design Review. Following this, the design definition phase, focused on completing the handbook, concluded with the Critical Design Review. Finally, the verification and validation phase, translated into a case study demonstrating the performance of the handbook and the associated tools, was concluded by the Flight Readiness Review (though not involving an actual flight), which served as the final presentation for the project.

Through these phases and reviews, we successfully developed a comprehensive system, laying a solid foundation for effective technology roadmapping within the EPFL Rocket Team.

[THIS PAGE INTENTIONALLY LEFT BLANK]

References

[1] O. L. D. Weck, “Fundamentals of systems engineering.” [Online]. Available: https://ocw.mit.edu/courses/16-842-fundamentals-of-systems-engineering-fall-2015/

[2] ——, “Technology roadmapping and development: A quantitative approach to the management of technology,” 2022. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-88346-1

[3] NASA, “Systems engineering handbook.” [Online]. Available: https://www.nasa.gov/reference/systems-engineering-handbook/

[THIS PAGE INTENTIONALLY LEFT BLANK]

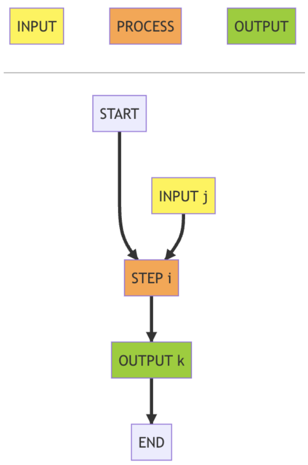
A Design Solution Definition Template

1. **Introduction:**
   (a) **Purpose:**
   ...
   (b) **Abbreviations:**
   - **FOM:** *Figure of Merit*

   (c) **Definitions:**
   - **Figure of Merit:** *Quantity used to characterise the performance of a device, system or method, relative to its alternatives. In engineering, figures of merit are often defined for particular materials or devices in order to determine their relative utility for an application.*

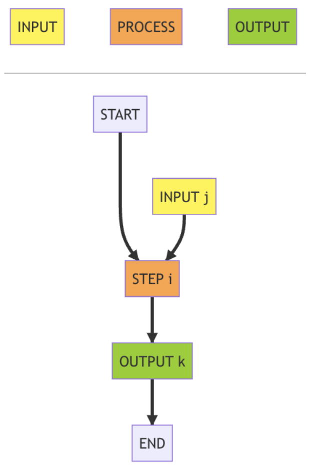
2. **Relevant Resources:**
   (a) *Author(s),Title,Organisation,Date: URL*

3. **Preliminary Guide Structure:**
   (a) Process Flowchart:



   (b) STEP i: ...
   ...

4. **Software Flowchart (if applicable):**



5. **Output Documentation:**
   ...

B Technology Roadmapping Handbook and Case Study

TECHNOLOGY ROADMAPPING

HANDBOOK

Authors: **Martin LEMAIRE**
Ryan SVOBODA

Supervisor: **Pr. Dr. Volker GASS**
Co-Supervisor: **Mathieu UDRIOT**

Doc ID: 2024_C_SE_SP_TECH_ROADMAP_HANDBOOK

Date: Spring 2024

Abstract

The EPFL Rocket Team faces the challenge of managing its growth and increasing project complexity as it strives to achieve a successful suborbital flight before 2030. To address this challenge, this handbook presents a systematic approach to technology roadmapping tailored to the needs of student-led associations. Drawing on the Advanced Technology Roadmap Architecture[1] developed by MIT Professor Olivier L. De Weck, the guidelines outlined in this handbook enable accurate executive decision-making, needs prediction, and resources allocation. By adopting this framework, student teams like the EPFL Rocket Team can effectively navigate the complexities of strategic planning, system modeling, and multiobjective optimisation.

Keywords: Technology Roadmapping, Strategic Planning, System Modeling, Multiobjective Optimisation, Multidisciplinary Design Optimisation

[THIS PAGE INTENTIONALLY LEFT BLANK]

Contents

[THIS PAGE INTENTIONALLY LEFT BLANK]

List of Figures

List of Tables

[THIS PAGE INTENTIONALLY LEFT BLANK]

Definitions and Abbreviations

Definitions

- **Figure of Merit:** Quantity used to characterise the performance of a device, system or method, relative to its alternatives. In engineering, figures of merit are often defined for particular materials or devices in order to determine their relative utility for an application. [3]
- **Similar Missions Analysis:** Systematic comparison and evaluation of projects or initiatives with similarities to a specific endeavor.
- **Technology Readiness Levels:** Technology Readiness Levels are a set of management metrics that enable the assessment of the maturity of a particular technology and the consistent comparison of maturity between different types of technology - all in the context of a specific system, application and operational environment.[4]
- **Pareto Optimal:** A Pareto set of solutions represents a collection of feasible options where no improvement in one aspect can be made without sacrificing another. It embodies the best trade-offs among competing criteria, aiding decision-makers in selecting the most suitable solutions.
- **Roadmap Maturity Scoring:** Roadmap Maturity Scoring is the process of establishing a benchmark applied to technology roadmaps that allows to assess different key characteristics, like granularity, depth, and clarity of the prescribed path.
- **Model**: A model is a mathematical object that has the ability to predict the behavior of a real system under a set of defined operating conditions and simplifying assumptions.
- **Simulation**: Simulation is the process of using a model for a particular set of inputs in order to predict the system response.
- **Design Variables**: These are the parameters which are to be optimized. They can be local, affecting only one discipline, or shared, impacting multiple disciplines.
- **Discipline Analysis**: The act of using a disciplinary model (as defined in the previous section) in order to get a set of a state variable values from a set of design variable values.
- **State Variables**: Result of a discipline analysis.
- **Coupling Variables**: To account for interactions between disciplines, certain variables (coupling variables) are exchanged between them. These represent the impact of one discipline on another.
- **Constraint**: A constraint is a condition or requirement that the design solution must satisfy. Constraints can arise from physical laws, performance requirements, safety standards, regulatory guidelines, or practical limitations. They play a crucial role in ensuring that the optimized design is feasible and meets all necessary criteria across the different disciplines involved.
- **Objective** An objectives refer to the goals or desired outcomes that the optimization process aims to achieve. These objectives are typically quantified as objective functions, which the optimization algorithm seeks to maximize or minimize. Objectives can vary widely depending on the specific application and the disciplines involved in the optimization problem.
- **Optimization Process**: The optimization process involves adjusting the design variables to achieve the best overall system performance while satisfying constraints from each discipline. This often requires iterative analysis and exchange of coupling variables until convergence is achieved.

Abbreviations

- **ATRA:** Advanced Technology Roadmap Architecture
- **FOM:** Figure of Merit
- **dFOM/dt:** Technology Trend
- **SMA:** Similar Missions Analysis
- **TRL:** Technology Readiness Levels
- **TRA:** Technology Readiness Assessment
- **NPV:** Net Profit Value
- $\triangle$**NPV:** Net Profit Value Difference
- **COTS:** Commercial Off The Shelf
- **R&D:** Research and Development
- **NRC:** Non-Recurring Cost
- **RC:** Recurring Cost
- **ROI:** Return On Investment
- **OPM**: Object Process Model
- **OPD**: Object Process Diagram
- **OPL**: Object Process Language

1 Introduction

This handbook is designed to offer a structured methodology for technology roadmapping tailored to the context of student-led rocketry associations. The goal is to establish a robust strategic planning framework to propel the association towards achieving the milestone of reaching the Kármán line by 2030.

Drawing from the Advanced Technology Roadmap Architecture[1], these guidelines are structured around four key steps: Competitive Benchmarking, System Modeling, Scenario Analysis and Portfolio Optimisation.

The initial phase involves competitive benchmarking, comprising three essential components. Figures of merit are utilised to quantify how well the association's specifications align with stakeholder expectations. Similar Missions Analysis assesses the current status of the association and comparable organisations pursuing similar objectives, laying the groundwork for future scenarios. Technology Readiness Levels offer insights into current technological capabilities, aiding in predicting how figures of merit might evolve in scenario analyses and verifying predictions.

The second step focuses on system modeling and optimization, serving as a transfer function that correlates design variables with associated figures of merit. Modeling plays an important role in the derivation of future performance based on certain technological investments, as it provides with the information on how well a system will perform with certain parameters.

Next, using the previously established model and Multidisciplinary Design Optimiser (MDO), the scenario analysis explores various strategies to achieve figures of merit goals within a specified timeframe. Exploring the possibilities and degrees of freedom for future decisions, a set of paths to target are determined, allowing to compare well defined solutions.

The fourth step aims to provide recommendations regarding the technology portfolio through two phases. The Pareto optimal set of technology portfolios addresses multi-objective optimisation problems based on previously identified scenarios. The recommended portfolio outlines the optimal path within the Pareto optimal set, accompanied by executive decisions, financial and human resource needs expectations, and an explicit scenario timeline.

Additionally, a section on maturity assessment offers insights into the accuracy of decisions and identifies areas for further development. Update instructions are provided, offering clear and concise directives tailored to the current maturity level of the roadmap for continual improvement.

The following flowchart (See Figure 1) depicts the adapted version of the Advanced Technology Roadmap Architecture [1] that is described in this handbook.

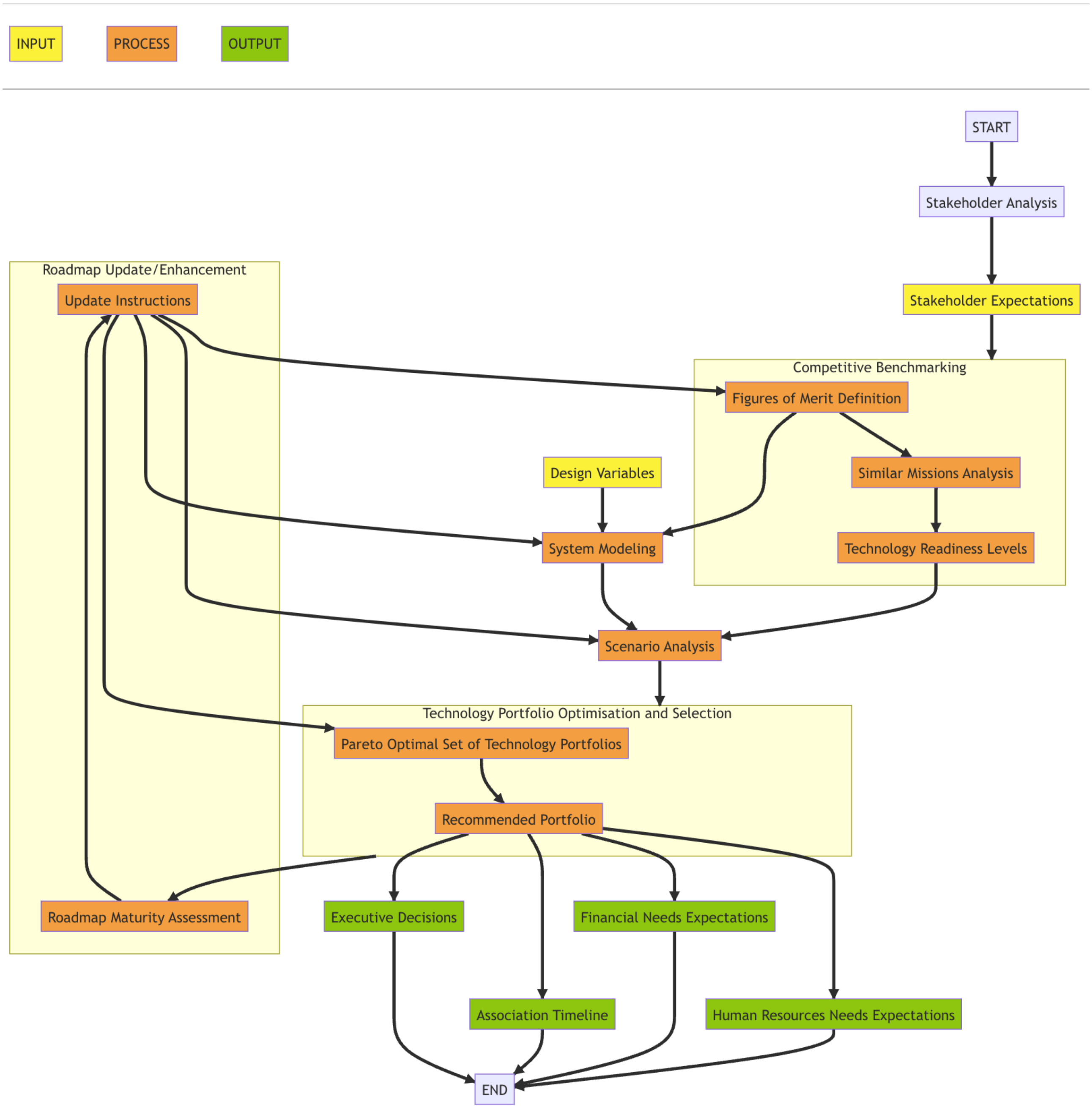


Figure 1: Adapted Version of the ATRA

2 Competitive Benchmarking

2.1 Figures of Merit

2.1.1 Purpose

In the field of technology roadmapping, our goal is to chart the most effective course towards fulfilling the long-term expectations of stakeholders. To gauge the progress towards meeting these expectations, it becomes essential to establish a set of pertinent indicators, known as Figures of Merit. These indicators serve as benchmarks, allowing us to compare technologies against existing standards and emerging trends. Through the utilisation of these metrics, organisations can make informed decisions regarding investments in research and development, strategically prioritising their efforts based on the stage of product development and the overarching objectives of the organisation.

2.1.2 Process Graph

The Figure 2 depicts the process that is described in this section,

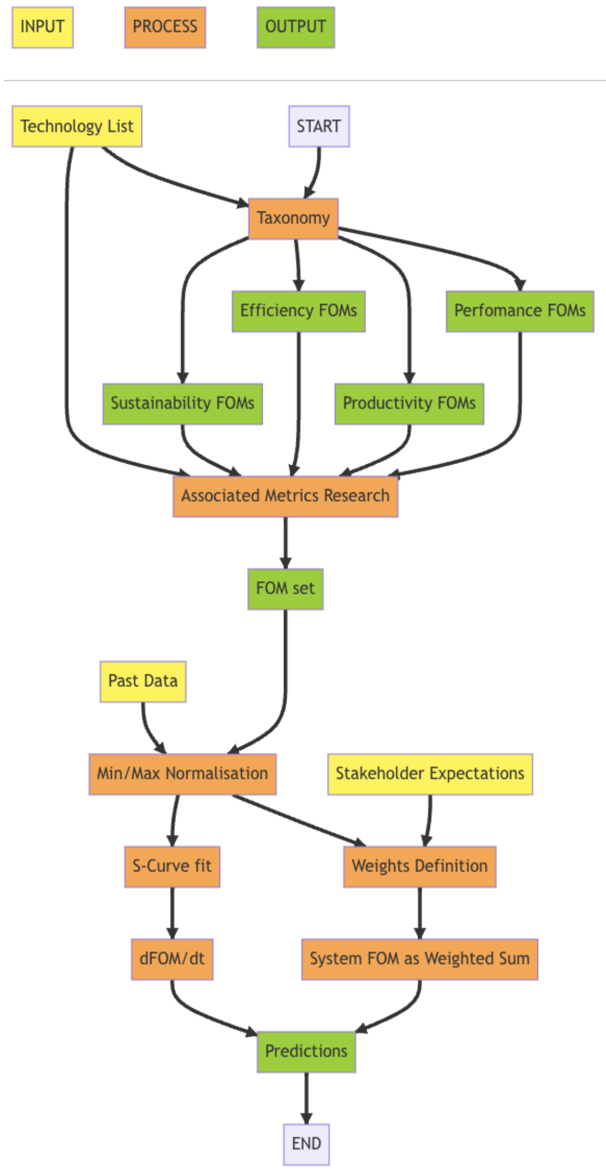


Figure 2: Flowchart of the Figures of Merit Definition Process

2.1.3 Step 1: Taxonomy of Technology

The process of defining general Figures of Merit (FOMs) often relies on existing sets that have been established for various types of technologies. Rather than devising new FOMs from scratch, we can leverage these pre-existing lists. To efficiently match the appropriate FOMs with the technologies encompassed within a system, we can employ a taxonomy of technology.

Technology taxonomy involves the systematic classification and grouping of different technological systems, methodologies, and artifacts based on shared characteristics, functions, or evolutionary relationships. Similar to how biological taxonomy organises living organisms into hierarchical groups according to their shared traits, technology taxonomy provides a structured framework for comprehending the expansive and diverse realm of human-made innovations.

This taxonomy can be systematically performed by filling a 3x3 technology matrix (See Table 1). Indeed, each of its cells is adequate for the definition of pertinent FOMs and the corresponding technologies are clearly identifiable from a well-defined list of technologies.

Technology Matrix	**Matter**	**Energy**	**Information**
Transforming			
Transporting			
Storing			

Table 1: 3x3 Technology Matrix

Example:

Examples of operands for the column classes: [5]

- **Matter:** packages, vehicles, crude oil, animals, plants, water, memorabilia
- **Energy:** potential, electrical, kinetic, thermal, nuclear
- **Information:** news reports, email, TV shows, voice conversations, books (content), bits

Technology Matrix	**Matter**	**Energy**	**Information**
Transforming	Basic open furnace (BOF) in steel making	Photovoltaic cells (PV) in a solar-electric farm	Graphical processing unit (GPU) in computing
Transporting	Transport aircraft in civil aviation	High-voltage electric transmission lines	Deep space network (DSN)
Storing	Storage tank for cryogenic hydrogen (LH_2)	Grid-level lithium-ion storage battery	Optical compact disk (CD) for data storage

Table 2: Example of Technology Matrix (3x3) for technology classification [1]

2.1.4 Step 2: FOM Set Definition

We can identify five general classes of FOMs, that enable us to generate a set that is pertinent with the previously classified technologies:

- **Performance:** "Amount of instantaneous power, capacity, speed, or precision that a technology can provide at a given point in time. Performance can be “peak performance” which can only be sustained for a short time, or “average performance” which is sustainable and assessed over longer periods of time." [1]
- **Productivity:** "Output of the system per unit time, measured in physical quantities or monetary value, as a function of the inputs into the system in terms of labor, capital, and technical factors (total factor productivity). In the simplest case of a linear production function, these are simple ratios." [1]
- **Efficiency:** "Outputs divided by inputs expressed in like units (energy, mass, information). Efficiency is nondimensional. We want efficiency to be as high as possible, but by definition it can never exceed unity, that is, a value of 1.0." [1]
- **Sustainability:** "Ratio of waste output per unit of useful or value-added output. We want this ratio to be as low as possible but it can never be lower than zero. The ratio of waste to actual value-added output can easily be a factor of 10 or larger, for example, in steel production from iron ore." [1]

We can therefore determine FOMs that are coherent with certain cells of the technology matrix and generate sets (See Table 3).

Technology Matrix	**Matter**	**Energy**	**Information**
Transforming	Power, Productivity, Mass Efficiency, Sustainability, Cost	Power, Energy, Efficiency, Sustainability, Cost	Productivity, Information, Efficiency, Cost
Transporting	Spped, Range, Sustainability, Cost	Sustainability, Cost	Speed, Cost
Storing	Capacity, Sustainability, Cost	Capacity, Sustainability, Cost	Capacity, Cost

Table 3: Prescripted FOM 3x3 Technology Matrix

It is important to note that the previous table provides categories of FOMs, which are not sufficient alone. Indeed, they do not always represent a definite metric. The table orients the user to minimise the time required to define a relevant set of FOMs, but further research is required to find the specific metric corresponding to a given FOM category for a predefined technology.

Also note that multiple FOMs might be found for the same category. They can all be taken into account, but it is important to keep this in mind for the weighted sum that defines the system. Indeed, stakeholder expectations might only cover the category, and therefore, the weight will be shared across these sub-FOMs.

Example:

For example, we can take the case of a rocket propulsion system and derive the FOMs as shown in Figure 3.

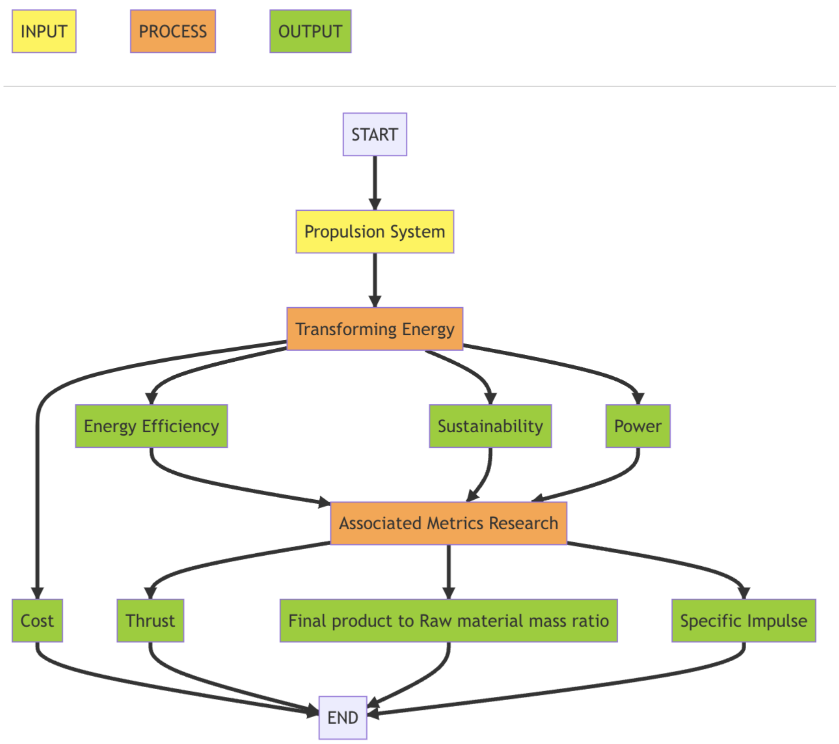


Figure 3: Example FOM Set Definition for a Rocket Propulsion System

In this example, the cost is considered a relevant metric by itself, which is why there is no research associated with it.

2.1.5 Step 3: Data Normalisation

In crafting a significant collection of FOMs, normalisation stands as a pivotal step, ensuring data conformity for precise weight allocation to each FOM. Employing Min-Max normalisation is instrumental, as it standardises data to a consistent range, spanning from 0 to 1. This normalisation method follows the formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Here, X represents the initial value, X_{min} denotes the dataset's minimum value, and X_{max} signifies the maximum value.

In our case, the maximum value is defined by the upper fundamental limit that a FOM can reach through a certain technology. The minimum value can also be derived as the lower fundamental limit or the value of the FOM for the least performing technology. In certain cases, it may be appropriate to set a limit different from the fundamental one, provided there is a justification. For example, while the fundamental limit of a rocket's velocity is far beyond the start of the hypersonic regime, entering the hypersonic regime introduces significantly more complexity. Therefore, it might be practical to set the hypersonic threshold as the limit.

The MATLAB function provided in Appendix B can be used to normalise a dataset. This function allows for inputting the fundamental limit instead of taking the maximum of the dataset.

Example:

In the case of the specific impulse of a bi-liquid engine, we start by determining a theoretical physical limit. An online literature review provides us with the Figure 4.

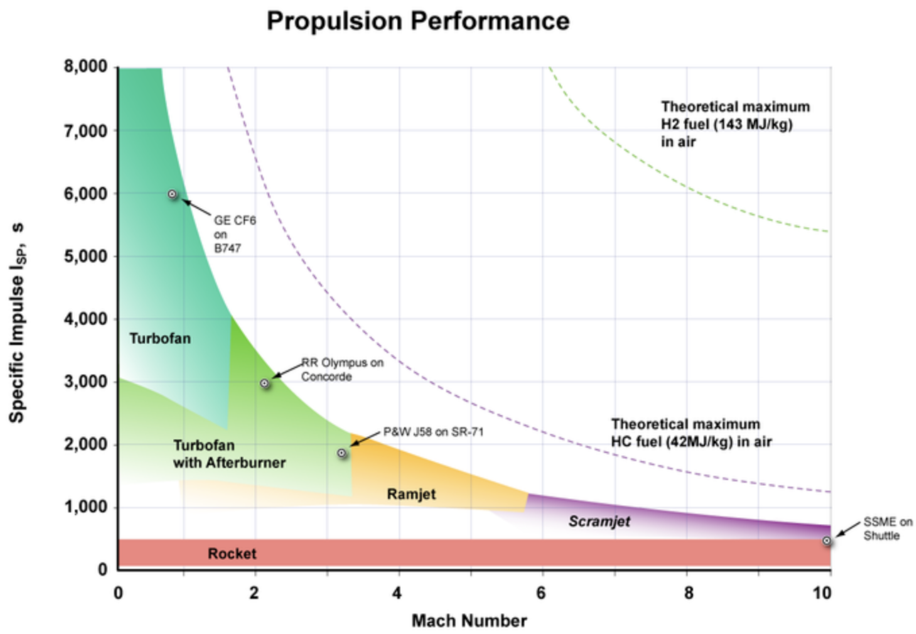


Figure 4: Propulsion Performance of different propulsion systems (Specific Impulse vs. Mach Number)

We can observe that the maximum specific impulse for a rocket propulsion system is about 500 and that the minimum is 0. In this case, the normalisation formula is therefore $X_{norm} = \frac{X}{500}$.

2.1.6 Step 4: System FOM Definition

We now have a set of meaningful FOMs that realistically represent the expectations that are related to a certain technology portfolio. We must analyse the stakeholders' expectations to derive weights for each of these FOMs. The goal is to use a weighted sum method to obtain a general FOM that will allow us to simplify the portfolio optimisation in the fourth step of the roadmapping process.

The most widely used method for multiobjective optimisation is the weighted sum method. The method transforms multiple objectives into an aggregated scalar objective function by multiplying each objective function by a weighting factor and summing up all contributors: [6]

$$J_{sum} = w_1 J_1 + w_2 J_2 + ... + w_N J_N$$

where $0 \leq w_i \leq 1$ and $\sum_{i=1}^{N} w_i = 1$

For example, acceleration is the most important objective element in the sports car market segment, but it has low preference in other market segments. For a motor company, it is crucial to identify the most important objective elements for different market segment, since they have the highest sensitivity to the aggregate performance of the individual vehicle.[1]

2.1.7 Step 5: Future FOM Evolutions

In the field of technology roadmapping, understanding and analysing technology trends serves as a foundational pillar for strategic planning. These trends encompass the evolving landscape of innovations within various domains, providing invaluable insights into the trajectory of technological development. By discerning patterns and foreseeing potential future directions, technology trends enable organisations to identify emerging opportunities, and proactively adapt their strategies and product portfolios.

Once the list of FOMs that we are considering is established, we can proceed to the analysis of the data. Technological progress is not linear over time and can be modeled as a logistic function, also called the technological S-curve(See Figure 5).

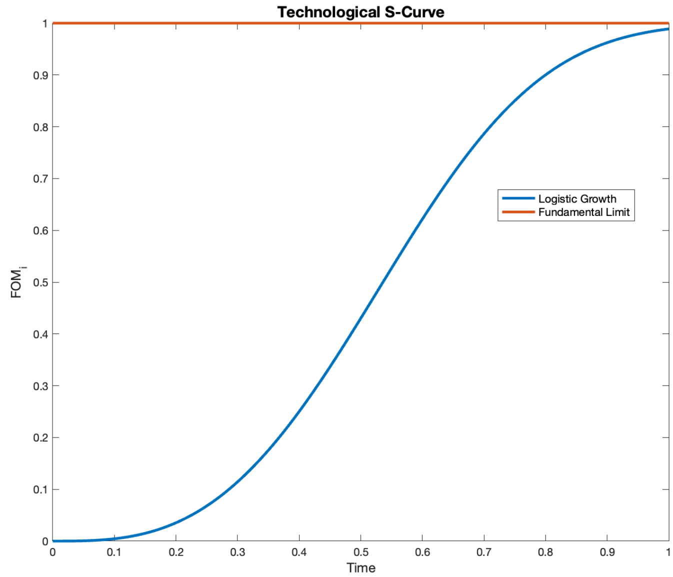


Figure 5: Technological S-Curve

Therefore, to predict the technological progress over time, we can apply a logistic fit to the past data, and then expand the timeframe using the logistic parameters that were found.

For example, the analysis of Solar Cell Efficiency allows us to produce the prediction of the evolution over time (See Figure 6).

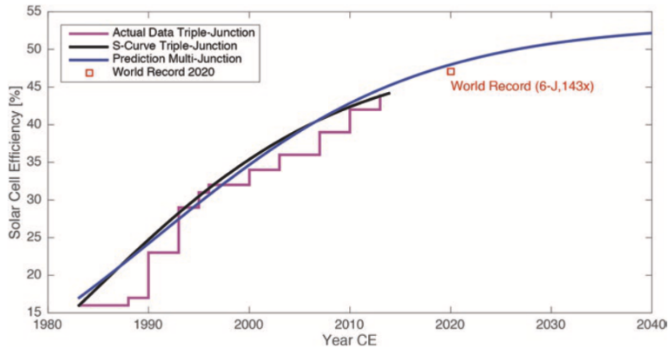


Figure 6: Actual vs. S-curve Model for Multijunction Solar Cell Efficiency [%] [1]

The MATLAB toolbox provided in Appendix B offers four types of predictions, allowing the user to compare models and make a decision on the preferred one. This is key because, we may not always have the entire S-curve, and we may also want to predict on shorter term. For example, if we know that we are in the end of the development cycle and that we do not have early data, we may prefer the logarithmic prediction instead of the S-curve or the exponential. The different possible predictions are linear, exponential, logarithmic and logistic(See Figure 7).

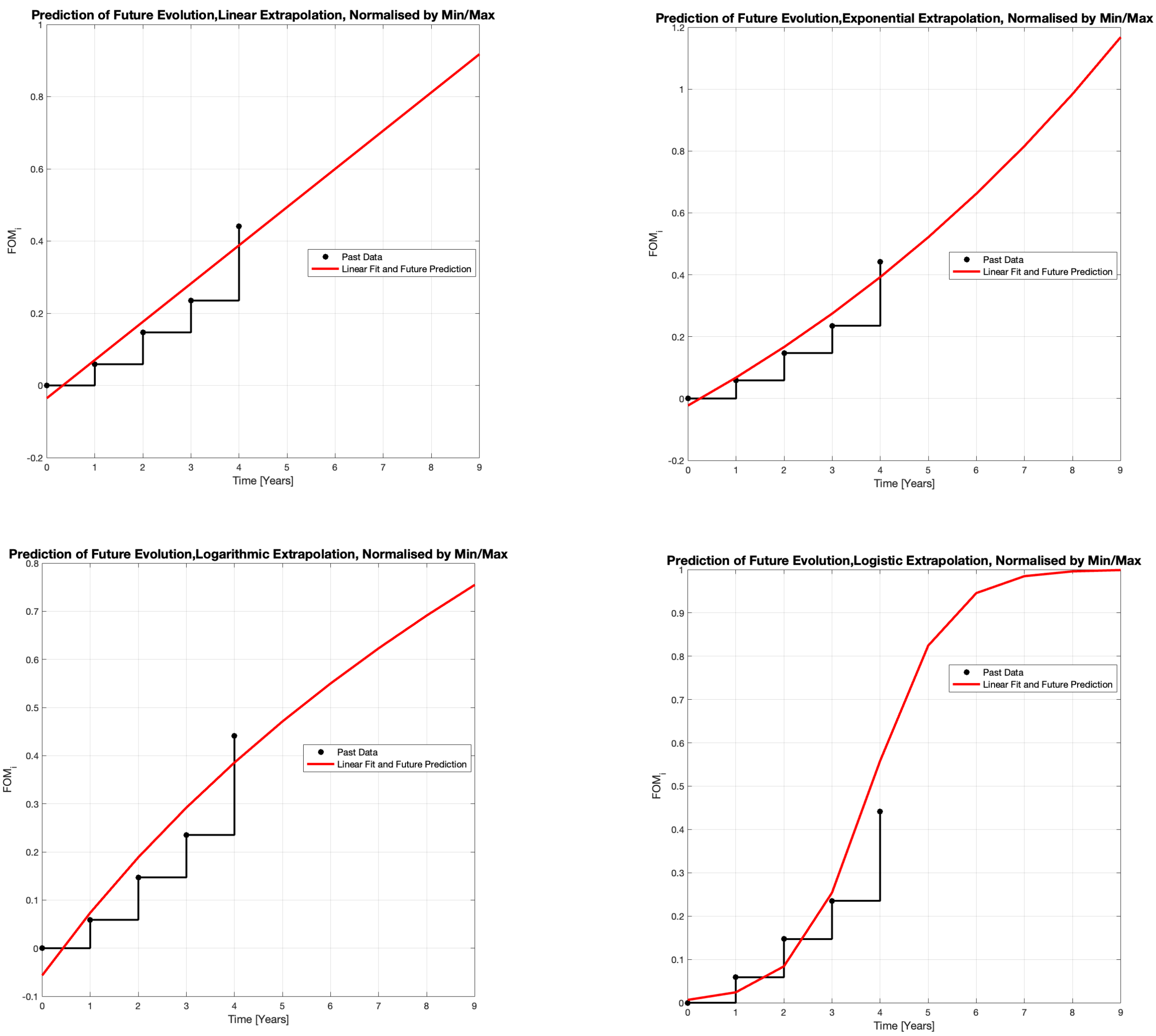


Figure 7: Example Prediction Plots Generated with the MATLAB toolbox

2.2 Similar Missions Analysis

2.2.1 Purpose

The purpose of conducting a Similar Missions Analysis is to comprehensively evaluate missions that share similarities with the project or initiative under consideration. This involves identifying existing missions, projects, or endeavors within a related domain or industry and analysing their approaches, outcomes, and lessons learned. By examining similar missions, stakeholders can gain valuable insights into the strategies, challenges, and successes experienced by others in comparable endeavors. This process serves as a crucial step in strategic planning, as it provides a foundation for understanding the existing landscape, informs decision-making, and facilitates the identification of best practices and potential pitfalls. Additionally, by benchmarking against similar missions, organisations can better position themselves within the competitive landscape, capitalise on existing knowledge, and identify opportunities for innovation and differentiation.

2.2.2 Process Graph

The Figure 8 depicts the process[7] that is described in this section,

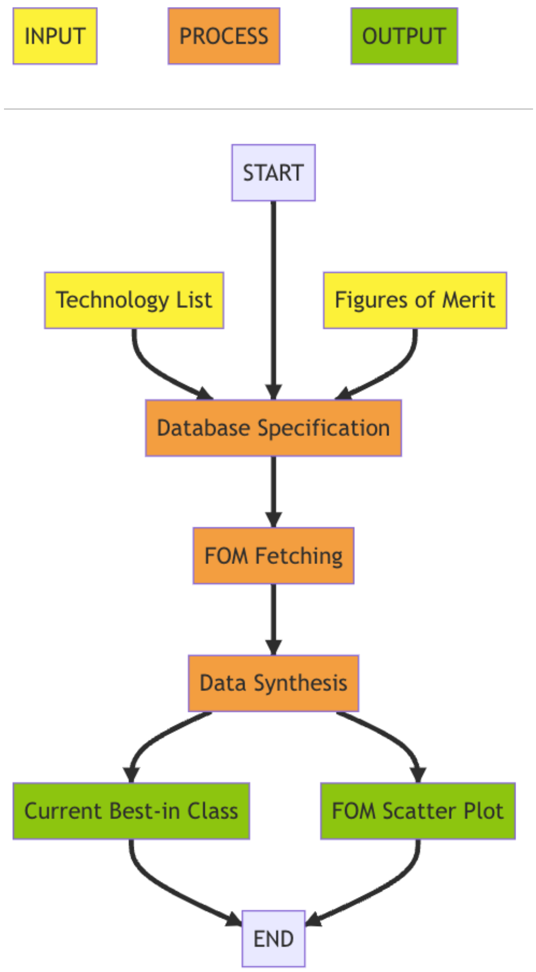


Figure 8: Flowchart of the Similar Missions Analysis Process

2.2.3 Step 1: Database Specification

The first stage of SMA involves the pivotal process of data collection, a critical step in ensuring the analysis's quality and reliability.

The Three Steps of Data Collection:

1. **Defining Technology/System Boundaries:** The initial step in the data collection process entails the clear definition of the boundaries surrounding the focal technology or system under analysis. This involves delineating the scope of the analysis, specifying the parameters, components, and characteristics that define the technology or system in question.
2. **Defining High-Level Specifications:** It's important to note that a technology's Figures of Merit (FOM) can vary depending on the broader system context. Therefore, it's crucial to define specific specification intervals to ensure that only relevant categories are examined. For instance, if the project involves developing a suborbital launch vehicle with limited payload capacity, a similar missions analysis for the propulsion system wouldn't consider engines like the SpaceX Raptor, as they're not aligned with the project's scope and requirements.
3. **Data sources:** Ultimately, we will establish precise and dependable data reservoirs to uphold the integrity and accuracy of our data collection efforts. These sources may encompass comprehensive records from analogous student rocketry groups or up-to-date insights from emerging micro launcher enterprises like Maiaspace or Sirius. Such sources could offer technological parallels closer to suborbital flights rather than conventional 3km Solid launcher systems. Research papers within comparable technological parameters can also be scrutinised, thus scientific repositories such as ResearchGate or laboratory websites can serve as highly dependable sources.

2.2.4 Step 2: FOM Fetching

Now that we've assembled a dependable collection of documentation, we can advance to investigating the data of interest, namely the Figures of Merit (FOMs). At this juncture, two scenarios may arise. The first scenario, the simpler of the two, involves directly locating the numerical value of the desired FOM either in the system documentation or within relevant papers, allowing us to store it alongside its source information. The second scenario poses a greater challenge, necessitating the utilisation of the system model, which constitutes the second step of the ATRA. In this instance, the specific numerical value of the focal FOM isn't readily available. However, we have access to the technical specifications, enabling us to determine the FOM value through the model.

2.2.5 Step 3: Data Synthesis

Upon completion of the data collection and meticulous organisation process, we move into the synthesis phase, where we aim to understand the current state of a specific technology. This involves plotting the various Figures of Merit (FOMs) in a multidimensional space, carefully exploring all possible combinations of two FOMs to create bi-dimensional graphs. For each pair of FOMs identified, we determine the current best-performing instance to establish the current Pareto front for each pair of FOMs, highlighting optimal trade-offs. Finally, we evaluate the values of the system FOM and assess whether the distribution of weights aligns with the observed results.

Example:

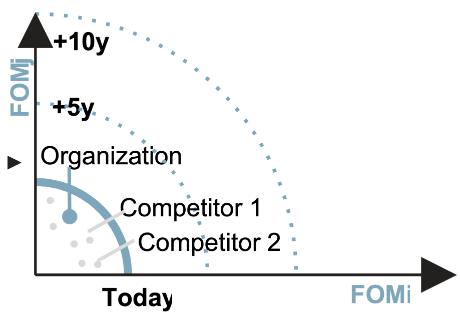


Figure 9: Example Bi-Dimensional Graph and Pareto Front [1]

2.3 Technology Readiness Levels

2.3.1 Purpose

Technology Readiness Levels (TRLs) are pivotal benchmarks in assessing technological preparedness and competitiveness. They serve two primary functions: firstly, by indicating how close a technology is to being ready for implementation in future missions, thereby guiding strategic decision-making. Secondly, TRLs enable the validation of previously projected technological trends, ensuring the accuracy of forecasting efforts. By systematically evaluating TRLs, stakeholders gain clarity on a technology's readiness and its potential impact, facilitating informed decision-making and strategic planning.

2.3.2 Process Graph

The Figure 10 depicts the process that is described in this section,

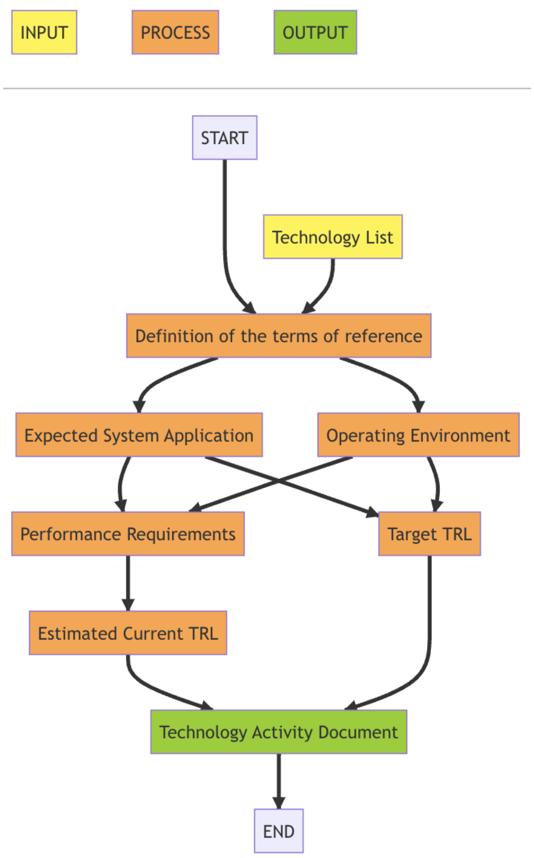


Figure 10: Flowchart of the Technology Readiness Assessment Process

2.3.3 Step 1: Performance Requirements and Current TRL Estimation

To initiate the assessment process, we lay down precise performance criteria, derived from both the anticipated system application and the operational context. These incremental requirements serve as thresholds against which specific TRLs are validated. Meeting these requirements entails reaching developmental milestones and finalising associated documentation. To facilitate this, we will rely on the criteria outlined by the ECSS commission (See Table 4).

Technology Readiness Level	Milestone achieved for the element	Work achievement (documented)
TRL 1 - Basic principles observed and reported	Potential applications are identified following basic observations but element concept not yet formulated.	Expression of the basic principles intended for use. Identification of potential applications.
TRL 2 - Technology concept and/or application formulated	Formulation of potential applications and preliminary element concept. No proof of concept yet.	Formulation of potential applications. Preliminary conceptual design of the element, providing understanding of how the basic principles would be used.
TRL 3 - Analytical and experimental critical function and/or characteristic proof-of-concept	Element concept is elaborated and expected performance is demonstrated through analytical models supported by experimental data/characteristics.	Preliminary performance requirements (can target several missions) including definition of functional performance requirements. Conceptual design of the element. Experimental data inputs, laboratory-based experiment definition and results. Element analytical models for the proof-of-concept.
TRL 4 - Component and/or breadboard functional verification in laboratory environment	Element functional performance is demonstrated by breadboard testing in laboratory environment.	Preliminary performance requirements (can target several missions) with definition of functional performance requirements. Conceptual design of the element. Functional performance test plan. Breadboard definition for the functional performance verification. Breadboard test reports.
TRL 5 - Component and/or breadboard critical function verification in a relevant environment	Critical functions of the element are identified and the associated relevant environment is defined. Breadboards not full-scale are built for verifying the performance through testing in the relevant environment, subject to scaling effects.	Preliminary definition of performance requirements and of the relevant environment. Identification and analysis of the element critical functions. Preliminary design of the element, supported by appropriate models for the critical functions verification. Critical function test plan. Analysis of scaling effects. Breadboard definition for the critical function verification. Breadboard test reports.
TRL 6: Model demonstrating the critical functions of the element in a relevant environment	Critical functions of the element are verified, performance is demonstrated in the relevant environment and representative model(s) in form, fit and function.	Definition of performance requirements and of the relevant environment. Identification and analysis of the element critical functions. Design of the element, supported by appropriate models for the critical functions verification. Critical function test plan. Model definition for the critical function verifications. Model test reports.
TRL 7: Model demonstrating the element performance for the operational environment	Performance is demonstrated for the operational environment, on the ground or if necessary in space. A representative model, fully reflecting all aspects of the flight model design, is build and tested with adequate margins for demonstrating the performance in the operational environment.	Definition of performance requirements, including definition of the operational environment. Model definition and realisation. Model test plan. Model test results.
TRL 8: Actual system completed and accepted for flight ("flight qualified")	Flight model is qualified and integrated in the final system ready for flight.	Flight model is built and integrated into the final system. Flight acceptance of the final system.
TRL 9: Actual system "flight proven" through successful mission operations	Technology is mature. The element is successfully in service for the assigned mission in the actual operational environment.	Commissioning in early operation phase. In-orbit operation report.

Table 4: TRL summary: Milestone and work achievement (reproduced from ISO 16290:2013)

For student associations, there may be instances where certain requirements, particularly regarding documentation, are not fully met. Consequently, strictly adhering to the criteria outlined in the table above could potentially result in a lower assigned TRL than what is truly warranted. Thus, it's crucial to not only analyse existing data but also engage in interviews with relevant individuals to ascertain the actual milestones achieved. This approach ensures more precise and accurate results are obtained.

2.3.4 Step 2: Target TRL Definition

In traditional technological companies or agencies, technologies are developed until TRL 8 or 9 before being implemented on flight systems. However, in the case of a student led association, such rigorous readiness levels are not always possible to attain are not critical to the minimum operational requirements completion. Therefore, we have to establish for each technology at what point in its development it can be implemented on flight systems and therefore be accounted in the scenario analysis that follows.

The Figure 11 delineates explicit milestones corresponding to each TRL, facilitating a clearer comprehension of the performance levels they signify. Alongside these milestones, we will also consider the significance of the studied technology within its associated system. Technologies that are critical to flight completion and safety will require higher TRLs than other less critical ones.

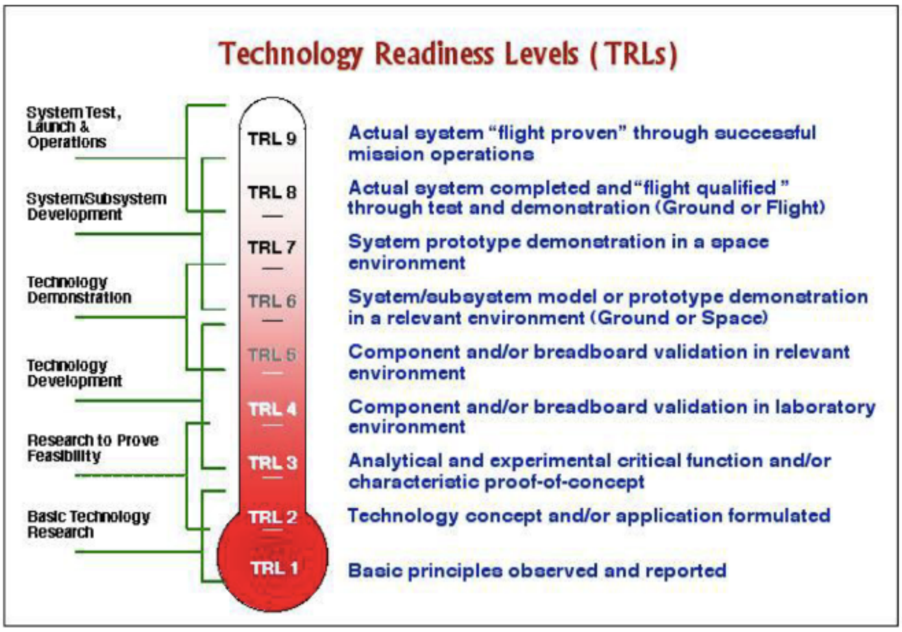


Figure 11: TRLs Thermometer Representation

3 Systems Modeling

3.1 Introduction

3.1.1 Purpose

The goal of system modelling is to be able to establish a model (definition below) which gives the mathematical relations in between the design variables of the technologies which make up the product, and the FOMs of the product. This allows you to set target values for the design variables of your technologies from the desired values of a FOM and vice-versa. For example, if you wanted to increase the range of an existing airplane by a set amount, by using a model, you would be able to know how much each design variable (such as: engine efficiency, fuel weight, airplane's empty weight, ...) must change in order to reach the target range.[1]

In the context of technology roadmaps, system models play a pivotal role in translating overarching objectives into quantifiable values for various components within the system. Instead of merely stating the objective as 'The rocket team needs to create a rocket reaching an apogee of 100km by 2030,' through an iterative process utilising a system model, the objective can become "by 2030, the rocket team should aim for an engine with a specific impulse exceeding X, a maximum thrust above X, a structural weight less than X, and a finesse within X% of X". This process entails inputting a multitude of design variable combinations into the system model and selecting the configuration that most closely aligns with the desired Figures of Merit (FOMs).

3.1.2 Model Vocabulary

[2][8]

- **Physical Model**: Start with governing equations
- **Empirical Model**: Based on observations/data
- **Fidelity**: How "precise" are the mathematical equations, high fidelity (CFD, FEM, ...), intermediate fidelity (beam theory, vortex lattice, ...), low fidelity (empirical models).
- **Breadth**: How much of the system is included in the model, small breadth (focus on a subsystem), medium breadth (all critical constraints), large breadth (complete system),.

3.2 Preliminary Guide Structure

Models come in all shapes and sizes, they can have different degrees of fidelity and they can vary in breadth (definitions above). A model's degree of fidelity and breadth is directly proportional to the time invested in its establishment and the utility and trustworthiness of the information obtained from its use. This means that there are trade-offs to be made when deciding how accurate and complete you want your model to be.

Your goal is to create a mathematical model of a large and complex system, to do so you start off by creating a conceptual model using a modelling language. The conceptual model serves as an intermediate step in between the high-level block diagram of the entire system and the mathematical model, meaning that as opposed to your mathematical model, your conceptual model will be organised by sub-systems and assemblies, and as opposed to a block diagram, your conceptual model will contain the numerical variables which serve as inputs and outputs of the design process. Establishing a conceptual model before trying to make the mathematical model should help you understand what the constituents of the system is and how they interact with each other. In our case we will use OPM (a system modeling language) as it is fairly simple given that it only has one graph (Object Process Diagram) which can be written out as text (Object Process Language), it also happens that OPM became ISO standard 19450 in 2015.

Once you have the conceptual model of your system, you need to prepare for the programming part of the system modelling, to do so you will start off by filling out the Master Table which enumerates all of the inputs and outputs of your model. In order to try and reduce the amount of time it takes to establish the model, you should try to find existing models/simulation software which you will be able to implement in your model. You will then fill out a Design Structure Matrix (DSM) which tracks all of the interactions in between each of the elements of the Master Table, it also serves as an index for your governing equations. Once you have identified all of the inputs and outputs of your model and how they interact with each other, it's time to start planning how your code will be structured by creating groups of governing equations with clear inputs and outputs which are called modules. The interactions in between modules will be tracked in the N^2 Diagram. Finally you will need to code, test and integrate your modules.

The following state diagram depicts the process that is described below,

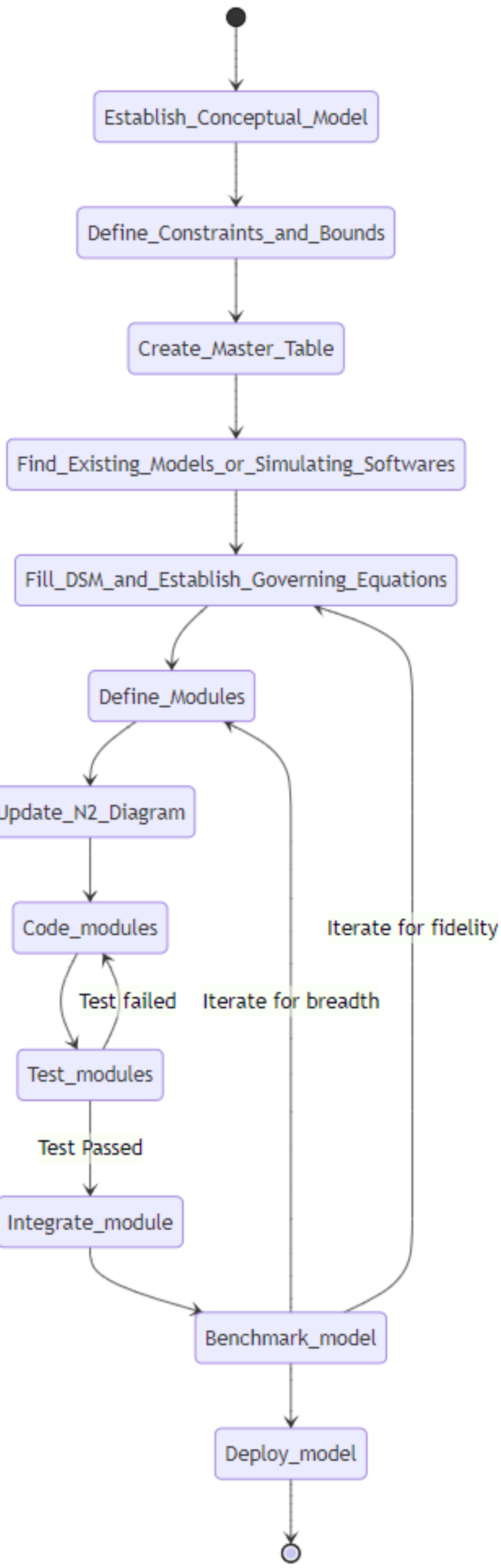


Figure 12: System Modeling Flowchart

3.2.1 Step 1: Establish Conceptual Model

In this step, the objective is to develop a conceptual model of the system by identifying various elements that constitute or interact within it. The function(s) and form(s) constituents (as explained in the table below) are entirely contextual and do not directly contribute to the final mathematical model. Instead, they serve as tools to facilitate a systematic approach to system modeling. This process ensures that the model creator can effectively organise and categorise the design variable(s), fixed parameter(s) and figure of merit(s) which, as opposed to the function(s) and form(s), can be given numerical values and represent the variables, constants and unknowns in the mathematical model.

To create the conceptual model using OPM you must first understand what the Building Blocks, Links and Relations of OPM are. Then, you will do a top down description of the system following one form (defined in the table below) at a time. For each form, define the design variables and fixed parameters which affect the form, then define the FOMs.[9][10]

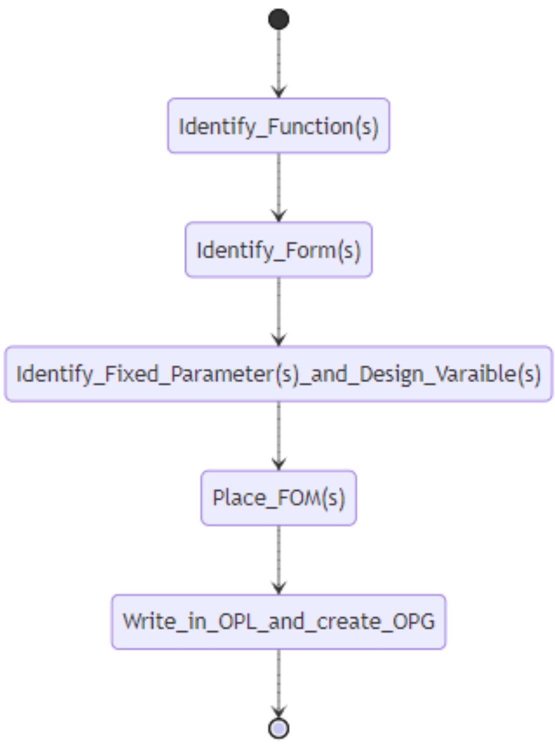


Figure 13: Conceptual Model Flowchart

Conceptual Model Vocabulary [9][10]

- **Function**: The main processes that the product carries out to create values. (example for airplanes: flying) (can not be assigned a numerical value)
- **Form**: The main objects which enable the function of the product. (example for airplanes: wings, engines, ...) (Can not be assigned a numerical value)
- **Fixed parameter**: Characteristics of the environment which can not be changed, basically the physics' constants. (example for airplanes: gravity, air density, ...) (Can be assigned a numerical value)
- **Design variables**: The characteristic of the form or function which a product or system designer can freely choose. (example for airplanes: weight at the start of the mission, weight at the end of the mission, specific fuel consumption, ...) (Can be assigned a numerical value)
- **Figures Of Merit**: Typically the main value creating characteristics of the function(s). (example for airplanes: range, speed, ...)(Can be assigned a numerical value) [1]

[1] Sometimes fixed parameters can be turned into design variables to enlarge the design space. Sometimes fixed parameters are former design variables that were fixed at some value because they were found not to affect any of the FOMs J or because their optimal level was predetermined.

OPM Entities: The Building Blocks

	Visual Representation	Textual Form	Definition	Description
Entities	Object	Nouns; first letter in every word is capitalized	*An **object** is a thing that has the potential of stable, uncon-ditional physical or mental existence.*	Objects are static things, which can be generated, changed, or consumed only by processes.
Entities	Process(ing)	Nouns in gerund form; first letter in every word is capitalized	*A **process** is a pattern of trans-formation that an object undergoes.*	Processes are dynamic things. They generate, change or consume objects.
Entities	Object state	Nouns, adjectives or adverbs; non-capitalized	*A **state** is a situation an object can be at.*	An object is at some state. A process can change an object's state.

Figure 14: OPM entities

Procedural Links: Connect Objects to Processes

Name	Symbol	OPD	OPL	Description
Con-sumptio		Processing ◁— Object	**Processing** consumes **Object.**	Process uses object up entirely during its occurrence.
Result	——▷	Processing —▷ Object	**Processing** yields **Object.**	Process creates an entirely new object during its occurrence.
Input & Output		Object (input state, output state), Processing	**Processing** changes **Object** from **input state** to **output state.**	The object is at input state prior to the process occurrence, and at output state as a result of its occurrence.
Effect	◁——▷	Processing ◁—▷ Object	**Processing** affects **Object.**	Process changes the state of the object in an unspecified manner.
Agent	——●	Object ——● Processing	**Object** handles **Processing.**	Object is a human that is not changed by the process; process needs the agent object in order to occur.
Instru-ment	——○	Object ——○ Processing	**Processing** requires **Object.**	Object is a non-human that is not changed by the process; process needs the instrument object in order to occur.
Invo-cation	——▷▷	X Processing → Y Processing	**X Processing** invokes Y **Processing.**	First process directly starts up a second process, without an intermediate object.

Figure 15: OPM Links

Fundamental Structural Relations: Reveal Entity Structure

Full Name (Shorthand Name in **bold**)	Symbol	OPD	OPL	Description
Aggregation-Participation			A consists of B, C, and D.	B, C and D are parts of the whole A.
Exhibition-Characterization			A exhibits B, C, and D.	B, C and D are attributes of A. (If B is a process, it is an operation of A.)
Generalization-Specialization			B, C, and D are As.	B, C and D are types of A.
Classification-**Instantiation**			B, C, and D are instances of A.	B, C and D are unique objects of the class A.

Figure 16: OPM Structural Realtions

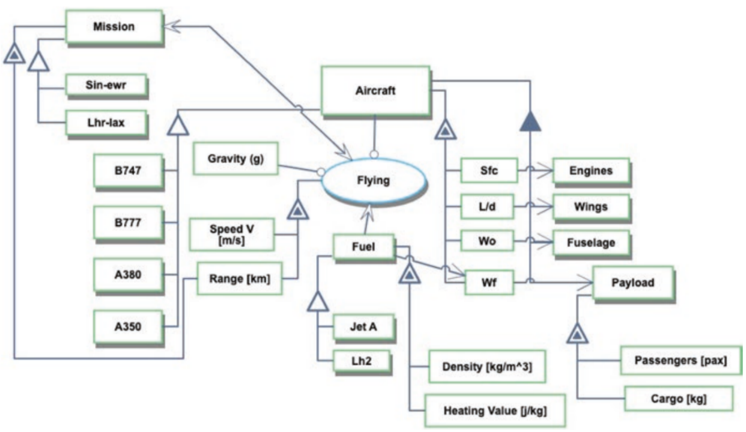


Figure 17: OPM Model Example: Long-Range Aircraft (simplified)

Building Blocks, Links and Relations (OPM)

3.2.2 Step 2: Define Constraints and Bounds

Constraints serve as boundaries within the design space, often arising from limited resources or technological limitations of certain design variables.

For example, you could but a constraint on the diameter of your rocket based on the production methods/tools you have at your disposal.

3.2.3 Step 3: Create Master Table

In simulation software, a 'master table' usually denotes a centralized data structure that holds essential parameters, settings, or configurations for the simulation model. This table functions as a central repository of information, utilized by the simulation software to execute simulations and perform calculations.

D	Antenna Diameter	[m]	design var.
A	Solar Panel Area	[m^2]	design var.
p	Orbital Period	[min]	design var.
R	Data Rate	[bps]	constraint
C	Cost	[$]	objective
P_t	Transmitter Power	[W]	dependent
P_{bus}	Bus Power	[W]	dependent
θ_a	Sun incidence angle	[deg]	parameter
$\eta_{a,t,}$	array/xmit efficiencies	[%]	parameter
S	Orbital altitude	[km]	dependent
α	constant	[-]	parameter
W_o	Solar constant	[W/m^2]	parameter

Figure 18: Master Table Example: Communication Satellite (simplified)

3.2.4 Step 4: Find Existing Models or Simulating Software

In order to reduce the amount of time it takes to establish the model, your goal is to find and implement existing models or simulation software (such as: RasAerp, RocketPy, ERT-Sim, ...) .

Identify all of the inputs and outputs of the software and update the Master Table if need be.

3.2.5 Step 5: Fill DSM and Establish Governing Equations

The DSM shows: which design variables affect which FOMs, which fixed parameters affects which FOMs and which constraints and bounds affect which design variables. Filling in the DSM allows for a methodological approach to establishing governing equations and to "index" the governing equations.

Governing Equations can vary in fidelity (definition above).

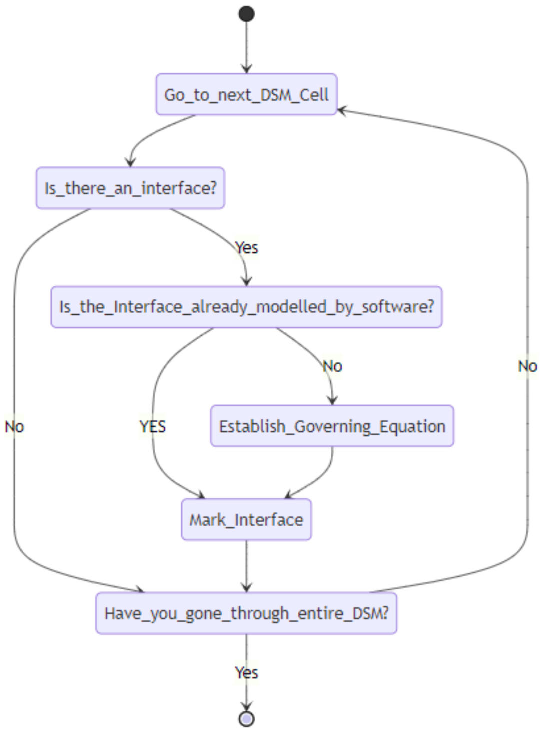


Figure 19: DSM and Governing Equations Flowchart

You should end up with a matrix with a FOM or constraint or boundary per row and one design variable per column, each cell should either be empty (no influence), contain a number which references the governing equation or contain a name which explains which software models the link.

3.2.6 Step 6: Define Modules

The modules are constituents of our model.

"A module in multidisciplinary system design optimization is a finite group of tightly coupled mathematical relationships who are under the responsibility of a particular individual or organization, and where some variables represent independent inputs while others are dependent outputs. The module frequently appears as a “black box” to other individuals or organisations."[2]

Each external software will be an independent module in our model.

Governing equations which are not implemented in the external software are created by clustering the DSM. Each module shall have a name, a clear set of inputs, outputs and equations which link the inputs to the outputs.

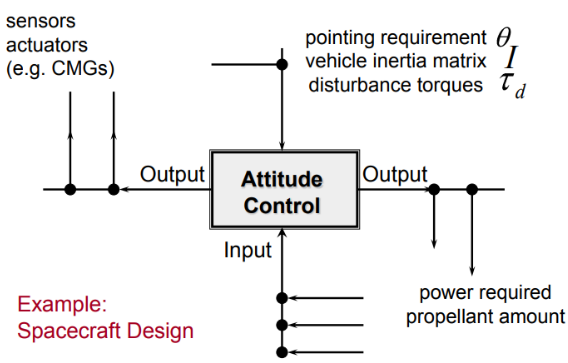


Figure 20: Module Example: Attitude Control[2]

3.2.7 Step 7: Update N^2 Diagram

"Each module within the simulation architecture is placed along the diagonal. Provides a visual representation of the flow of information through the simulation architecture. Helps to identify critical modules that have many inputs and outputs. The fidelity of critical modules should be thoroughly tested and verified. Explicitly defines all inputs and outputs for macro-modules and modules."

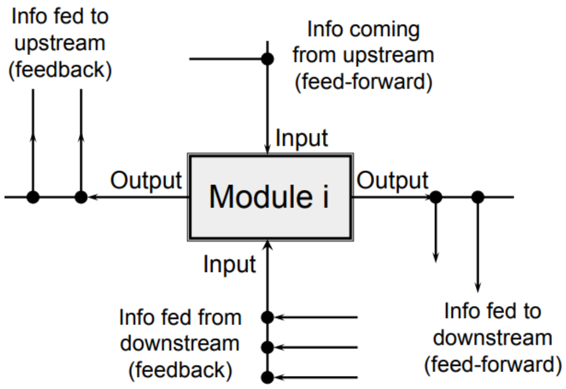


Figure 21: Inputs and Outputs of a Module[2]

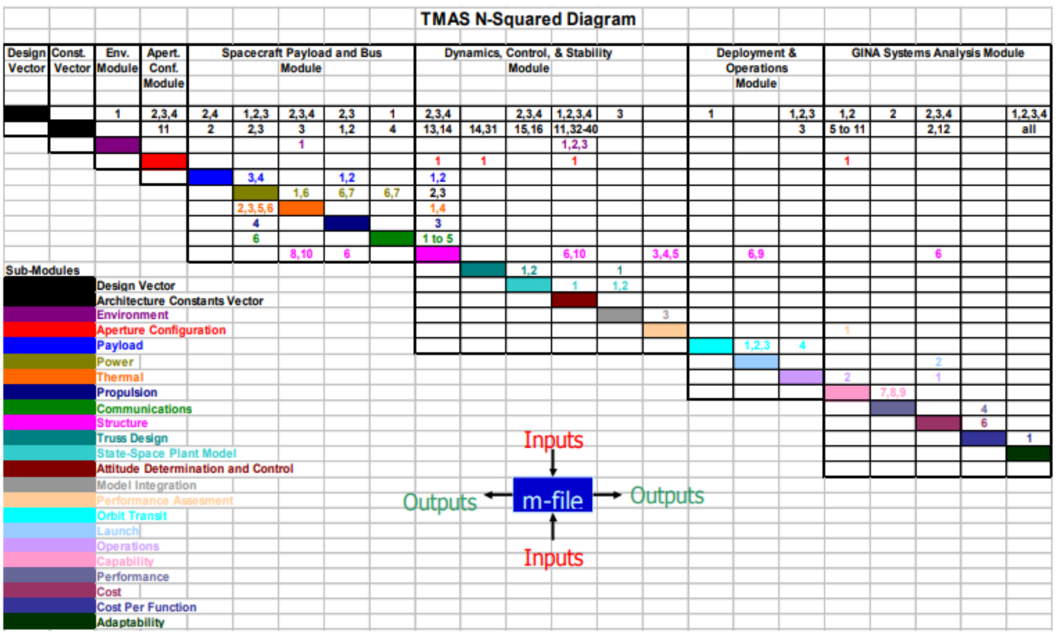


Figure 22: N^2 Diagram Example[2]

3.2.8 Step 8: Code Modules

The overarching architecture will probably resemble something like this, where "α, β, γ" are different software/programs, ",. B, C, D, E" are different modules and "*" are interface files.

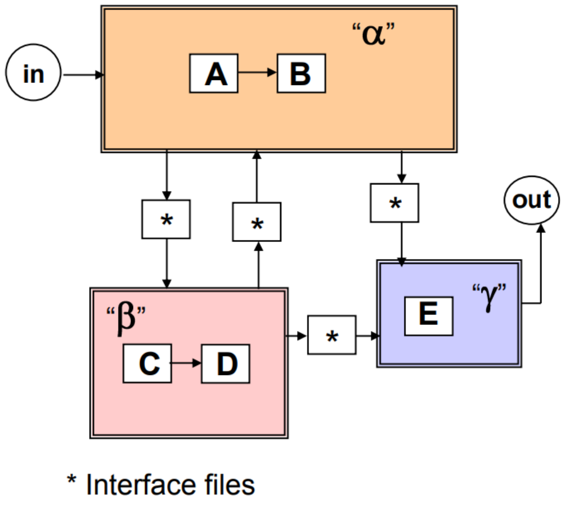


Figure 23: Module Interface Diagram[2]

3.2.9 Step 9: Test Modules

Given that each module has a clear set of inputs and outputs, they can and should be tested individually before integrating into the model. Special attention should be given to modules with a lot of connections of other modules.

3.2.10 Step 10: Integrate Modules

Once a module has been tested, it can be integrated into the model. Although there are several ways of integrating new modules into an existing model, here is a graph depicting what we believe is this simplest method for integrating modules relating to a specific sub-systems into the vehicle's model:

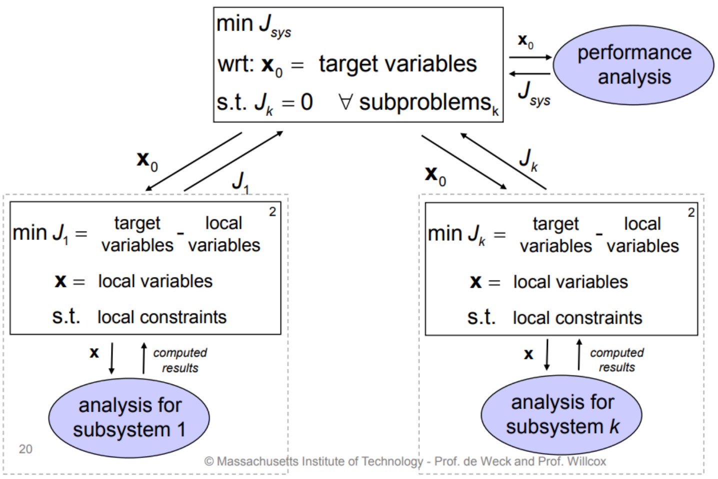


Figure 24: Module Architecture Example

Without going into too much detail, new modules can be treated as independent models whose target FOMs are the optimised design variables values of the higher level model. This allows the sub-systems to perform their own lower level roadmaps using MDO without changing the higher level vehicle's target design variables (which could affect other sub-systems).

3.2.11 Step 11: Benchmark Model

Once a model has been established it should be benchmarked with against values found during real life tests in order to understand the fidelity of the model.

3.2.12 Step 12: Deploy Model

Once all of the previous steps are complete it is time to use the model in order to find the optimal combination of design variables which lead to the desired FOM values.

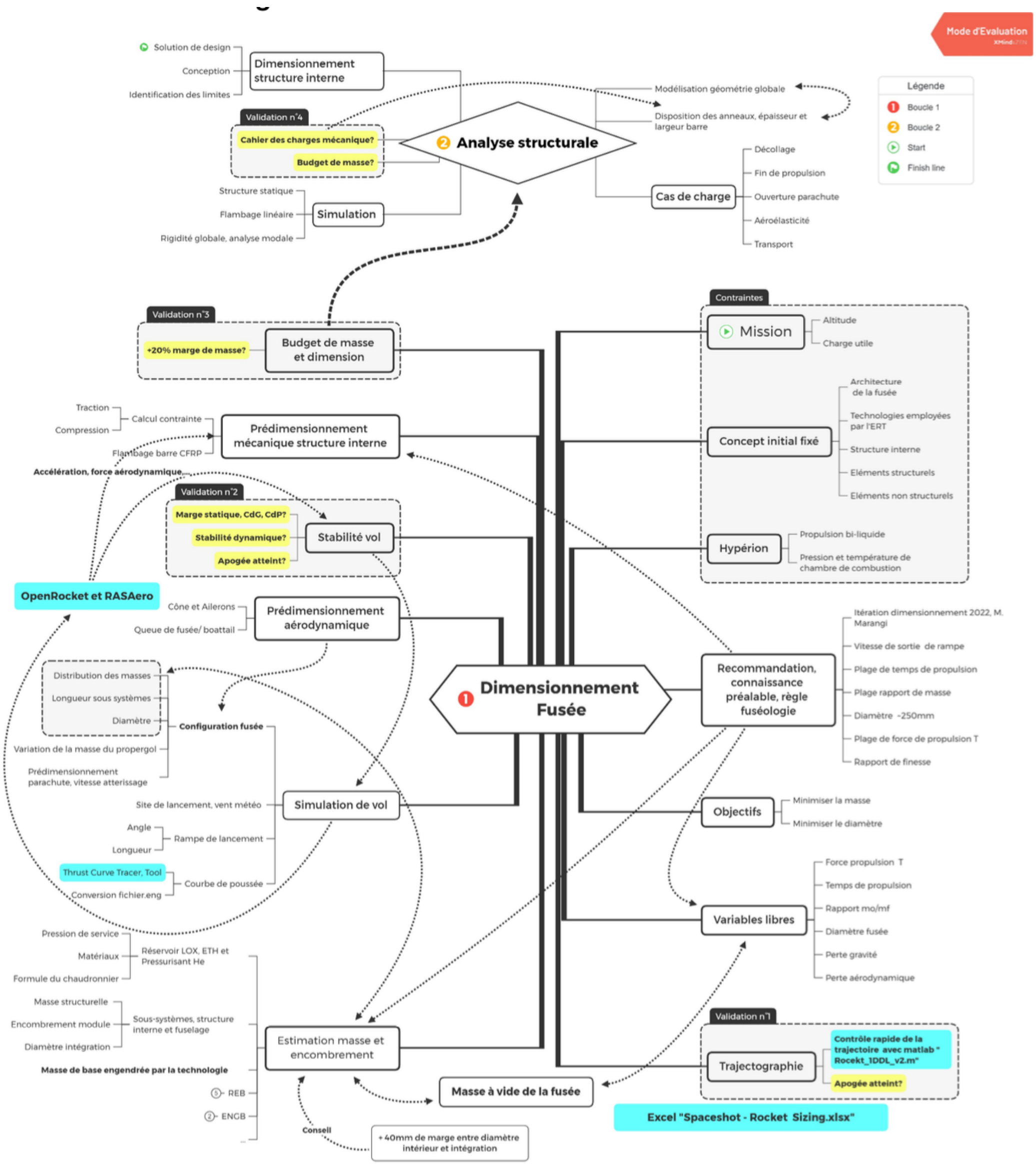


Figure 25: Model Example of internal structure for the EPFL rocket team's space shot rocket

3.3 Output Documentation

Conceptual Model Model of the system which shows how the system's main constituents interact which each other.

Master Table Table which enumerates all of the inputs and outputs of the model.

DSM Matrix which tracks which inputs affects which outputs, which constraints affect which inputs and how the links are computed.

N^2 **Diagram** Matrix which tracks all of the interfaces in between the modules of the model.

3.4 Output Software

After having completed this step, the roadmap maker should have a model with a chosen breadth and fidelity. The modules of the model will have been tested and the model will have been benchmarks.

[THIS PAGE INTENTIONALLY LEFT BLANK]

4 Multidisciplinary Design Optimization

4.1 Introduction

The goal of this chapter is to give an introduction to MDO by explaining what MDO is and how it is to be used in the context of technology roadmaps.

Set the foundation for future work by giving a list of useful resources, giving a brief description of how MDO has been used in a similar case and providing a suggestion as to how the rocket team could develop a tool which would allow them to apply MDO in the context of technology roadmapping.

4.2 Introduction to MDO in the context of technology roadmapping

Multidisciplinary design optimization (MDO) is an engineering approach which applies numerical optimization technique to the design of engineering systems involving multiple disciplines. It does so by using several discipline specific models and optimization techniques at different levels of the system in order to converge towards a set of feasible and optimized design variables.

MDO provides a structured framework to handle the complexity of designing systems with multiple interacting parts, leveraging computational power and advanced algorithms to find the best solutions that traditional sequential design approaches might miss.

MDO is especially important in the context of technology roadmapping as it allows us to set quantitative technical objectives with a higher degree of confidence compared to empirical methods.

4.3 Simplified MDO algorithm

1. **Initialization**:
   - Define the problem, including objectives, design variables, constraints, and the relationships between disciplines.
2. **Discipline Analyses**:
   - Perform simulations for each discipline independently to calculate state variables and initial coupling variables.
3. **Exchange of Coupling Variables**:
   - Share the coupling variables between disciplines to ensure interactions are accurately represented.
4. **Optimization Loop**:
   - Adjust design variables to improve the objective function (e.g., minimize weight, maximize performance) while satisfying all constraints.
   - Recalculate state variables and update coupling variables.
   - Iterate until convergence criteria are met, ensuring that all disciplines' requirements are balanced and the global optimum is achieved.
5. **Final Solution**:
   - The result is an optimized design that meets the needs of all disciplines involved, ensuring an integrated and efficient overall system.

4.4 MDO Architectures

4.4.1 What is a MDO Architecture ?

An MDO architecture is a structured framework that defines how the optimization problem is formulated, how the various disciplines interact and are coordinated throughout the optimization process. Different architectures are used to manage the complexity and computational demands of MDO problems, ensuring efficient convergence to an optimal solution.

4.4.2 Key Considerations When Choosing an Architecture

When choosing a Multidisciplinary Design Optimization (MDO) architecture, key considerations encompass the ease of: integrating "black-box" discipline models and analysis programs, updating discipline models, switching optimization algorithms, adding discipline models, and the computational resources required for each iteration.

4.4.3 Monolithic vs Distributed Architectures

Monolithic and Distributed MDO architectures represent two distinct approaches to solving Multidisciplinary Design Optimization problems. In monolithic architectures, a single, centralized optimization loop handles all disciplines simultaneously, encompassing all variables, constraints, and objective functions in one large problem. This approach, while capable of managing strong couplings between disciplines and often yielding high-quality solutions, suffers from high computational complexity, making it less feasible for large-scale problems. Conversely, distributed architectures decompose the overall problem into smaller, more manageable subproblems, each solved separately, often in parallel. This decentralized method reduces computational demands and leverages parallel processing, making it more scalable and efficient for weakly coupled systems. However, ensuring effective coordination and convergence between subproblems is crucial, as it directly impacts the quality of the overall solution. The choice between these architectures depends on the specific problem characteristics, including the degree of coupling between disciplines and the available computational resources.

4.4.4 Architectures Examples

Monolithic Architectures

- **All-at-Once (AAO)**: In this approach, all disciplines and the optimization problem are solved simultaneously in a single optimization loop. This architecture is straightforward but is computationally expensive for large, complex problems.
- **Multidisciplinary Feasible (MDF)**: Each discipline's analysis is fully resolved within each iteration of the optimization loop. This ensures feasibility with respect to all discipline constraints at every step. It is intuitive but may require significant computational resources due to repeated analyses.
- **Individual Discipline Feasible (IDF)**: The optimization process handles each discipline's analysis separately, allowing for parallel execution. However, this approach requires additional consistency constraints to ensure that the solutions across disciplines align properly.

Distributed Architectures

- **Collaborative Optimization (CO)**: This architecture splits the optimization into system-level and discipline-level problems. The system-level optimizer coordinates the design by adjusting the shared variables, while each discipline independently solves its local optimization problem. This method reduces the computational burden but requires effective coordination between levels.
- **Bi-Level Integrated System Synthesis (BLISS)**: This approach divides the optimization problem into system-level and discipline-level sub-problems, similar to CO, but employs a specific coordination strategy to ensure efficient convergence. The system-level problem optimizes global objectives while the discipline-level problems handle local constraints and objectives.
- **Analytical Target Cascading (ATC)**: This method decomposes the problem hierarchically, propagating targets (desired values) from the system level to subsystems and components. Each level adjusts its targets based on feedback from lower levels to ensure overall system optimization.
- **Concurrent Subspace Optimization (CSSO)**: This architecture divides the optimization problem into sub problems corresponding to different disciplines or components, which are solved concurrently. A coordinating mechanism ensures consistency and convergence towards the global optimum.

4.5 Example

An example of MDO applied to the multidisciplinary design of a rocket can be found in the bibliography. Here are some of the characteristics of the study.

- **Written in** : 1996
- **MDO Architecture** : Collaborative Optimisation (CO)
- **Number of Disciplinary Models** : 3
- **Types of Disciplinary Models** : Propulsion, Trajectory and Cost
- **Number of Design Variables** : 95
- **Number of Interdisciplinary Design Variables** : 23
- **Number of Constraints** : 16
- **Number of Objective Functions Tested** : 4
- **Types of Objective Functions Tested** : Minimum development cost, minimum dry weight, minimum gross liftoff weight and minimum ΔV.
- **Computation Time Required to Find Initial set of Optimised Values** : 1-3 weeks

4.6 Future Work

Two main types of work packages which need to be undertaken in order to develop a robust MDO program is:

1. **Create Baseline MDO program** :
   - **Software Architecture Design**: Design or select the architecture of the MDO software, including modules for optimization algorithms, discipline models, coupling strategies, and the user interface.
   - **Choose Disciplinary Partitioning**: Determine how to partition the various disciplines within the program to facilitate efficient optimization and interaction.
   - **Baseline Low Fidelity Disciplinary Models**: Develop initial low fidelity models for each discipline as a proof of concept.
   - **Initial Set of Interdisciplinary Design Variables**: Identify and define the initial set of design variables shared across different disciplines.
   - **Initial Set of Constraints and Values**: Establish preliminary constraints and their corresponding values to guide the optimization process.
2. **Create Disciplinary Higher Fidelity Models and Optimizers** :
   - Internal structure
   - Propulsion system
   - Trajectory
   - Recovery
   - ...

[THIS PAGE INTENTIONALLY LEFT BLANK]

5 Technology Paths

5.1 Introduction

The goal of this section is to use the model and MDO program created in the previous section in order to generate: a set of feasible and optimised design variable values which meet the target figures of merit and a set of intermediate missions which lead from the current FOM values to the target FOM values and follow the FOM trends found in section 1.

This section provides a preliminary pseudo algorithm which is to be when writing the final algorithm.

5.2 Process

5.2.1 General

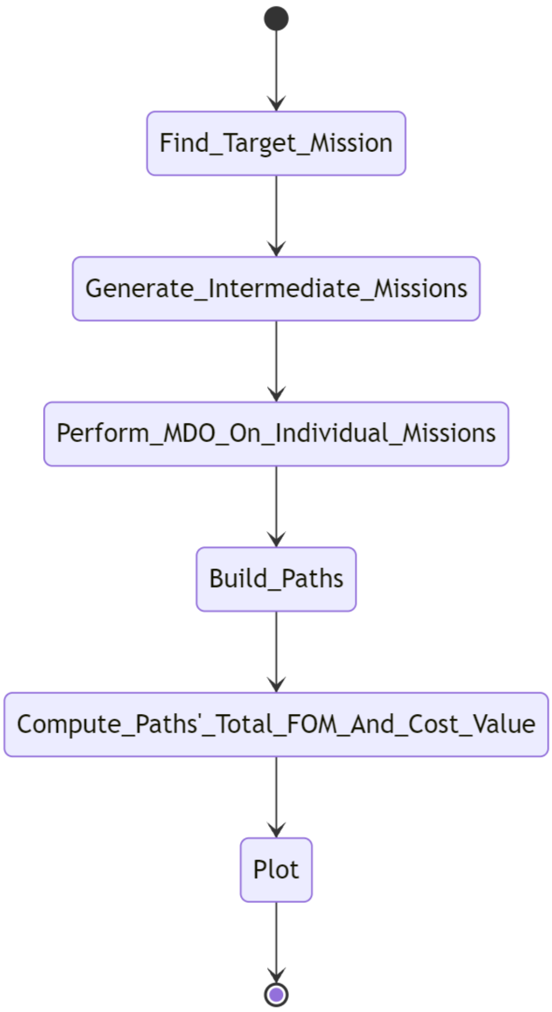


Figure 26: General Technology Path Flowchart

5.2.2 Find Target Mission

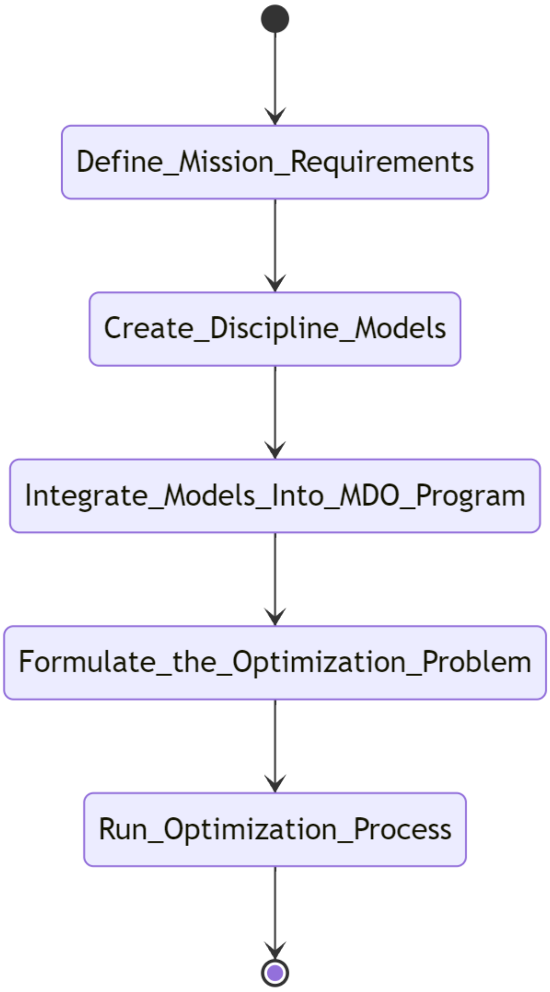


Figure 27: Find Target Mission Flowchart

5.2.3 Generate Intermediate Missions

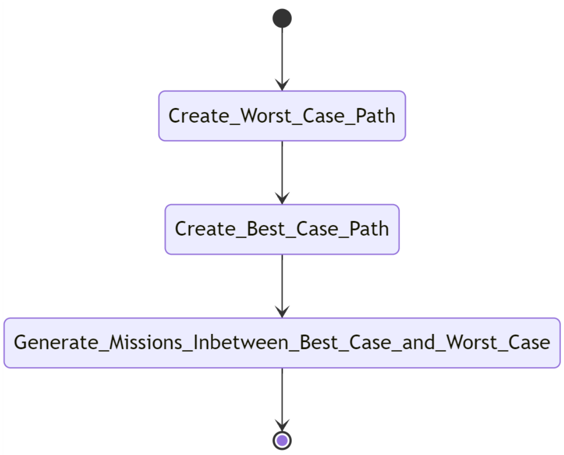


Figure 28: Generate Intermediate Missions Flowchart

Goal: Generate numerous "technology paths" which lead from the current FOM to the target FOM passing by intermediate missions.

5.2.4 Perform MDO On Individual Missions

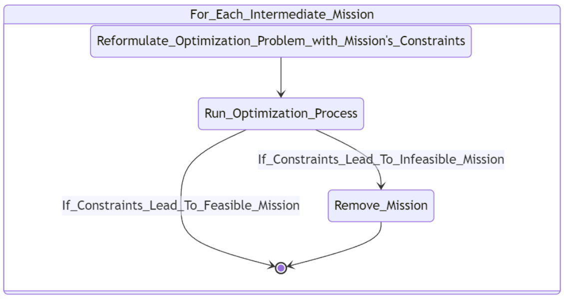


Figure 29: Perform MDO On Individual Missions Flowchart

5.2.5 Build Paths

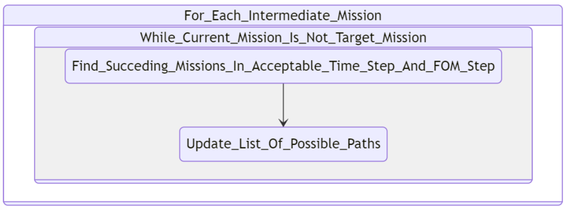


Figure 30: Build Paths Flowchart

5.2.6 Compute Paths' Total FOM And Cost Value

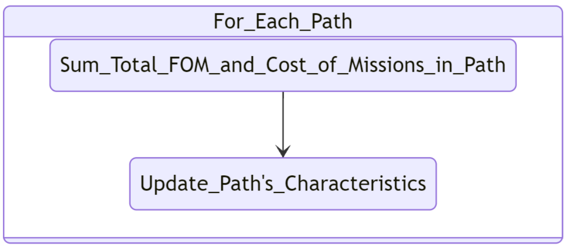


Figure 31: Compute Paths' Total FOM And Cost Value Flowchart

5.2.7 Plot

Goal: Present all of the information gathered. Output: - Future scenarios: .txt which presents for each mission: design variables, FOM, target year, Possible mission one step back, possible mission one step forward. - Vector chart: For each Mission: Plot (Σ(FOM), Year), Plot link in between each possible succession of missions. - Design reference mission: document which shows main vehicle specification, technology growth, number of steps) for each possible program

[THIS PAGE INTENTIONALLY LEFT BLANK]

6 Technology Portfolio Optimisation and Selection

6.1 Pareto Optimal Set of Technology Investments Portfolio

6.1.1 Purpose

In the previous section, all possible scenarios in which the general expectations are met were found, and we can now proceed to the optimisation of the technology portfolio. The goal of this part is to provide a set of Pareto Optimal technology portfolios that will allow us to identify the most favorable portfolio combinations, offering the highest potential returns with acceptable levels of risk.

6.1.2 Process Graph

The Figure 32 depicts the process that is described in this section,

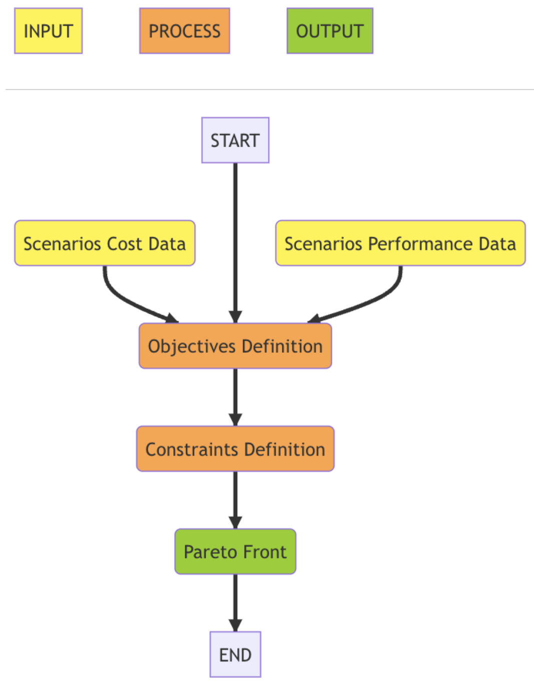


Figure 32: Flowchart of the Pareto Optimal Set of Technology Investments Portfolio Process

6.1.3 Step 1: Objectives Definition

In the industry, technology road mapping typically pursues two key objectives: maximising the increase in Net Profit Value (Δ NPV) while minimising associated risks, often quantified through the standard deviation of NPV. NPV encompasses various components that are concurrently optimised. Essentially, profits result from subtracting costs from total value, a metric closely tied to product performance, especially within the context of technology companies.

In our scenario, being a student-led association and a non-profit organisation, we must revise the ΔNPV to distinguish between two essential components: performance and cost. Performance is gauged through a weighted aggregation of FOMs, as outlined in the Competitive Benchmarking section of the handbook. Conversely, cost analysis mirrors the approach taken in the industrial version of ATRA. Utilising this data, our aim is twofold: to minimise total costs ($\sum(C)$) and to reduce the standard deviation of costs ($\sigma(C)$), thereby assessing the risk inherent in our portfolio. For instance, while a portfolio might boast lower overall costs, a singular, substantial investment could jeopardise the association's financial stability, potentially leading to bankruptcy.

Therefore, the optimisation problem can be modeled as,

$$\textbf{Maximise } Z_1 = E(FOM)$$

$$\textbf{Minimise } Z_2 = \sum(C)$$

$$\textbf{Minimise } Z_3 = \sigma(C)$$

Subject to,

- Maximum cost constraint
- Minimum FOM Level

The maximum cost is determined by estimating, based on past data of the sponsoring cash flows, the maximum budget available to the association. Here, we are not targeting to plan this maximum cost, as one of the objectives is to minimise it. However, a scenario that is below this estimated maximum cost is considered feasible.

On the flip side, it's crucial to ensure that every scenario within the set meets a minimum FOM level, reflecting the focal organisation's objectives and meeting stakeholders' expectations.

To maintain consistency in optimisation, whether minimising or maximising, we employ the relationship:

$$max(f) = -min(-f)$$

Typically, we omit the negative sign for simplicity while still preserving the intended portfolio identification.

Thus, the objectives of the optimisation problem can be reformulated as follows:

$$\textbf{Minimise } Z_1 = -E(FOM)$$

$$\textbf{Minimise } Z_2 = \sum(C)$$

$$\textbf{Minimise } Z_3 = \sigma(C)$$

6.1.4 Step 2: Objectives Computation

The multiobjective optimisation problem that we are left with comprises 3 objectives. For each of the scenarios that we got from the Scenario Analysis, we have on the one hand the performance data and its associated system FOM values along the scenarios timeline. On the other hand, we have the costs associated with the path, which are in the same vector dimension as the performance data and is representing the budget needed from a step to the next.

Computation of the System FOM

To compute the first objective Z_1, we simply take the last value of the System FOM vector, still using the same weights that were defined in the Competitive Benchmarking part.

Computation of the total cost

To compute the total cost associated with a technology portfolio, we sum all the components of the cost vector. Once all the total costs are computed, we normalise the set using Min/Max normalisation.

Computation of the Financial Risk

To compute the financial risk associated with the proposed portfolios, we will compute the standard deviation of the cost vector. The formula to compute the standard deviation (σ) of a vector of n elements $x_1, x_2, ..., x_n$ is:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}}$$

Where:

- $\bar{x}$ is the arithmetic mean of the vector, given by $\frac{\sum_{i=1}^{n} x_i}{n}$.
- x_i represents each individual element of the vector.

6.1.5 Step 3: Pareto Front Definition

By definition, the Pareto front represents the set of solutions along which it is not possible to improve one objective without degrading another. Using the MATLAB tool provided this handbook, it is possible to give the scenarios data and obtain the Pareto Optimal Set of Technology Portfolios.

The software will provide with all the 2 objectives Pareto fronts, showing the best tradeoffs between the 2 indicators (see Figure 33)

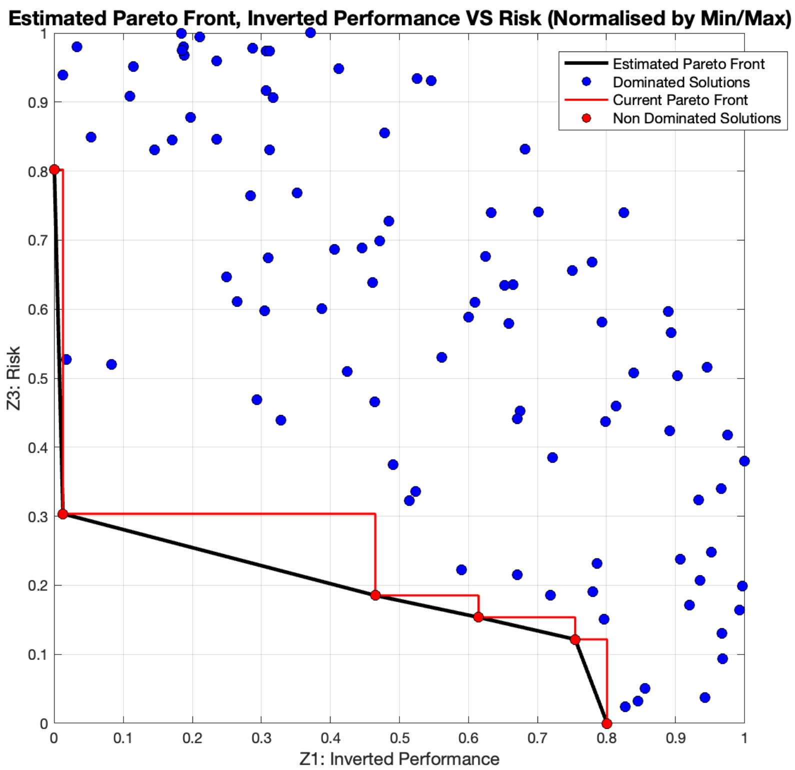


Figure 33: Estimated Bi-Objective Pareto Front (Inverted Performance vs. Risk), Normalised by Min/Max

Note that the gap between the current Pareto Front (red line) and the Estimated True Pareto Front represents an area where potential solutions may exist but were not found. Therefore, these gaps must be fed back into the Scenario Analysis to identify possible areas of improvement.

It will also provide the mesh of the 3 objectives Pareto front (see Figure 34),

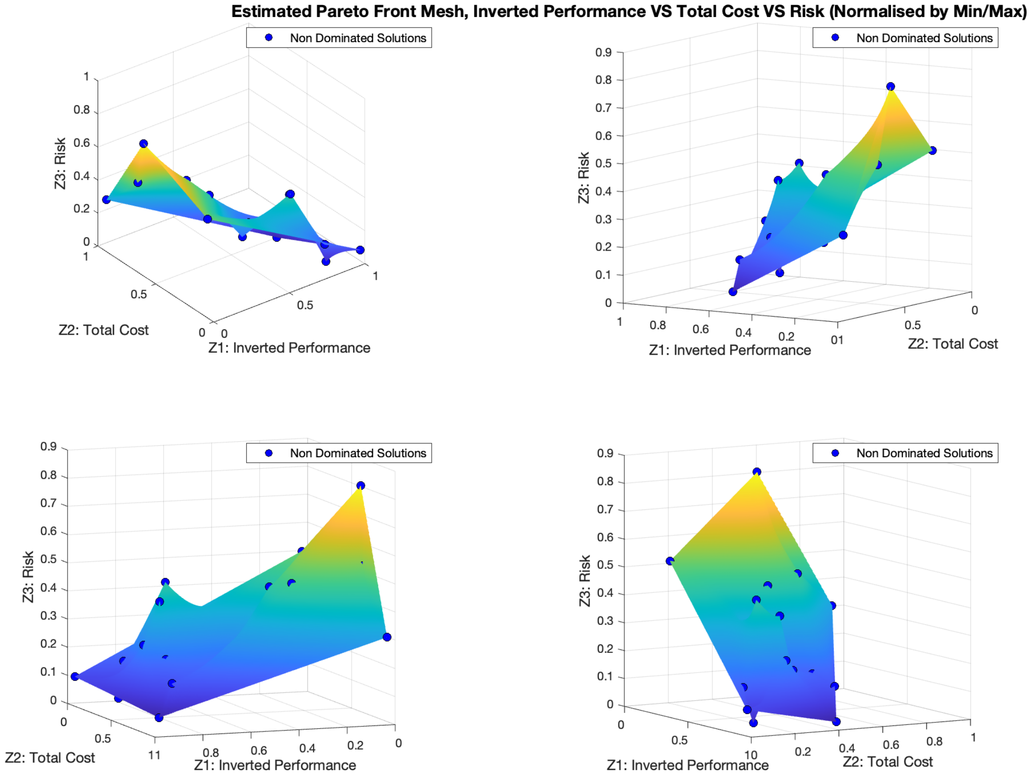


Figure 34: Estimated Tri-Objective Pareto Front (Inverted Performance vs. Total Cost vs. Risk), Normalised by Min/Max

6.2 Recommended Technology Portfolio

6.2.1 Purpose

We're now at the pivotal juncture of this process, tasked with selecting the optimal choice from our previously generated options. In the earlier phase, we pinpointed a spectrum of Pareto optimal technology portfolios. Our current objective is to identify the portfolio that best aligns with the overarching goals of the organisation. This decision holds immense significance as it will shape the trajectory of technological progress and resource allocation, guaranteeing that our chosen path optimally serves the organisation's objectives and long-term vision.

6.2.2 Process Graph

The Figure 35 depicts the process that is described in this section,

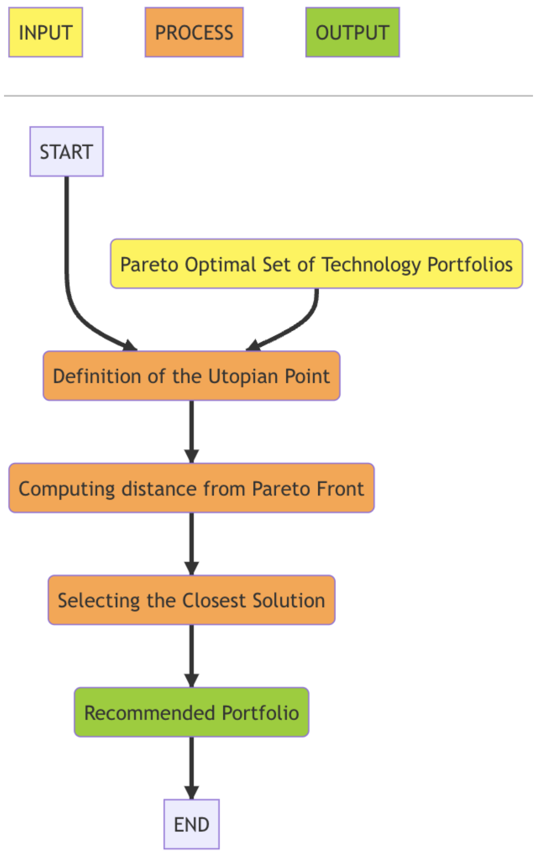


Figure 35: Flowchart of the Recommended Technology Portfolio Process

6.2.3 Step 1: Definition of the Utopian Point

In multiobjective optimisation, our aim is to meet all defined objectives to the fullest extent possible. Ideally, a perfect solution would maximise each objective. However, achieving this ideal is rare and often deemed unrealistic.

We can conceptualise this ideal scenario by pinpointing a point in our objective space that embodies this Utopian concept. By considering the boundaries of our feasible set, we can identify theoretical boundary Pareto Optimal scenarios. These scenarios, delineated by the Pareto front, illustrate extreme trade-offs where one objective reaches its maximum while the other reaches its minimum. By examining these boundary values, we can designate their intersection as the Utopian point (See Figure 36).

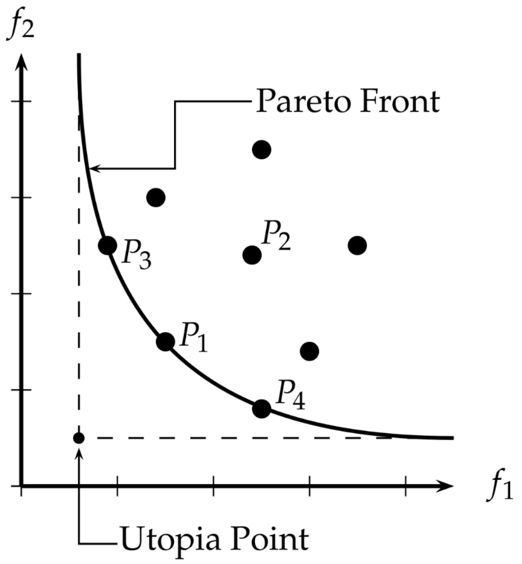


Figure 36: Utopia Point Representation in a 2 Objectives space

6.2.4 Step 2: Selection of the Recommended Portfolio

Having established the Utopian point, we can now discern among the Pareto Optimal solutions. The optimal choice among these solutions is the one that closely approaches the Utopian point. To quantify this, we utilise the Euclidean distance formula:

$$d = \sqrt{Z_i^2 + Z_j^2 + Z_k^2}$$

We then select the solution with the minimum distance, which yields our recommended portfolio (See Figure 37).

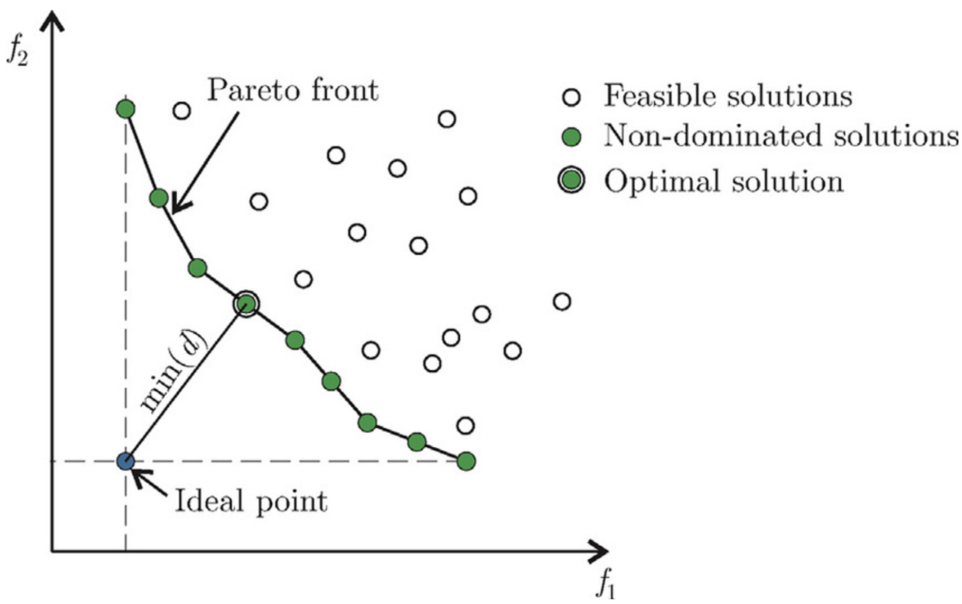


Figure 37: Illustration of the selection of the Optimal Solution, Ideal point is equivalent to the Utopian point

In a three dimensional case, we can directly plot the normalised distance versus the index of the portfolio to observe the optimal solution (See figure 38).

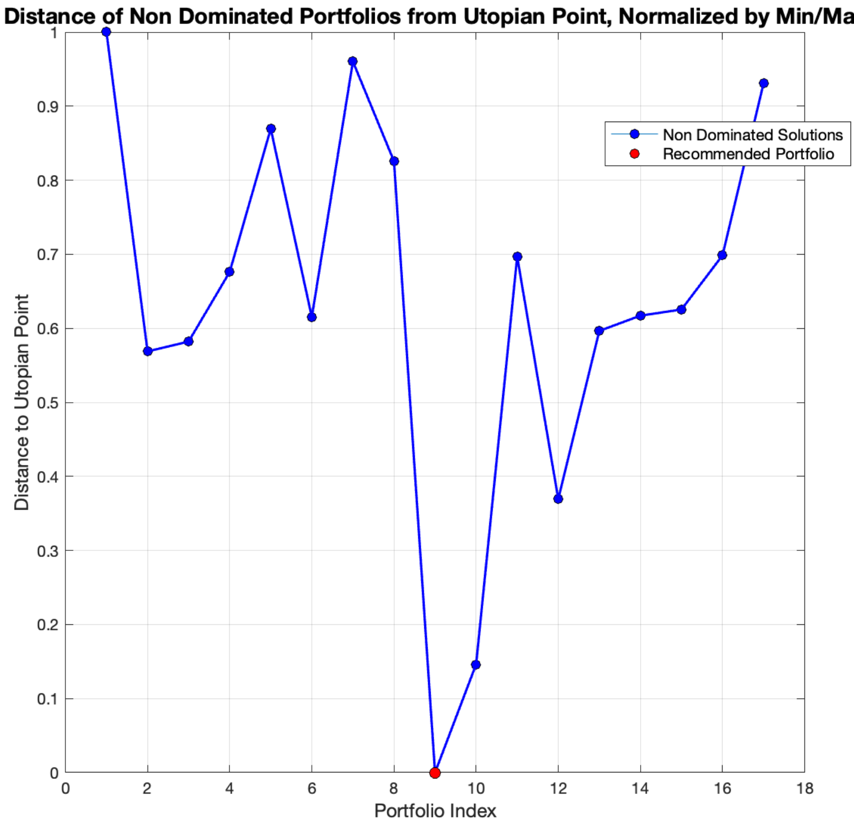


Figure 38: Plot of the 3D distances from Utopian Point of the Non-Dominated Solutions, normalised by Min/Max

The Recommended Portfolio is finally represented as in the Figure 39.

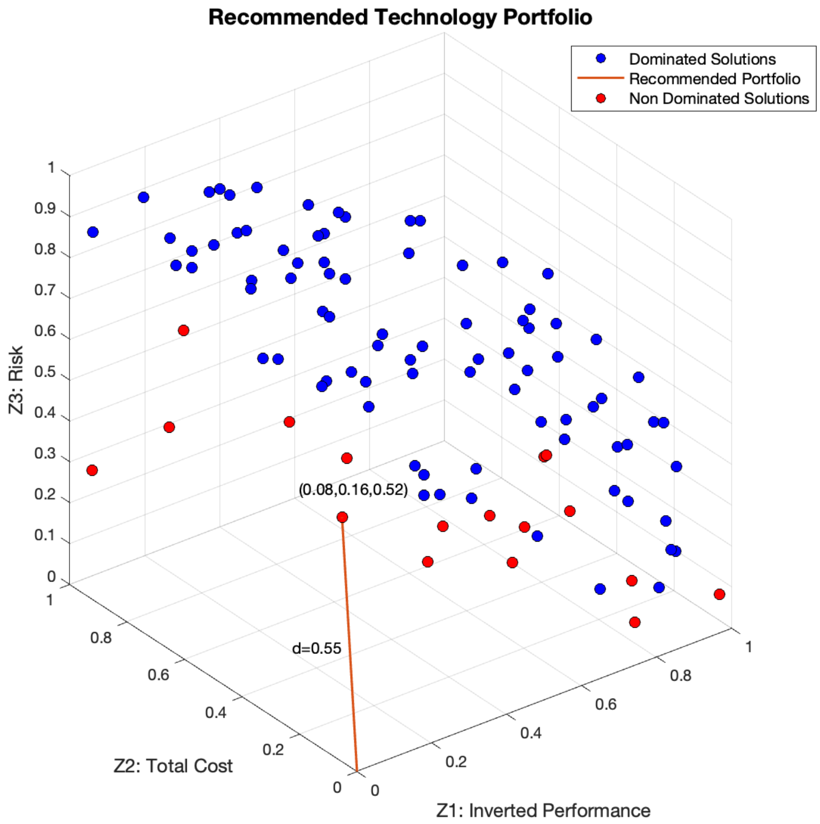


Figure 39: Plot of the Solution Space with the Recommended Portfolio Highlighted. Distance from the Utopia point to the Recommended Portfolio is specified and the coordinates of the recommended portfolio are indicated as well, in the following order: (Z1,Z2,Z3). Normalised by Min/Max

6.3 Human Resources Needs Expectations

6.3.1 Purpose

Establishing effective human resource practices within a student-led association presents unique challenges distinct from those encountered in traditional industry settings. The annual turnover of personnel poses a particular hurdle, especially during this transitional phase from one-year projects to multi-year endeavors, coupled with the integration of more advanced technological projects. To navigate this transition successfully, especially in endeavors like technology roadmapping, where we aim to tailor tools for student-led associations, it's imperative to address human resource needs meticulously. This entails devising recruitment strategies that account for consecutive turnovers and implementing comprehensive training programs to ensure seamless adaptation to evolving project demands and technological advancements.

6.3.2 Process Graph

The Figure 40 depicts the process that is described in this section,

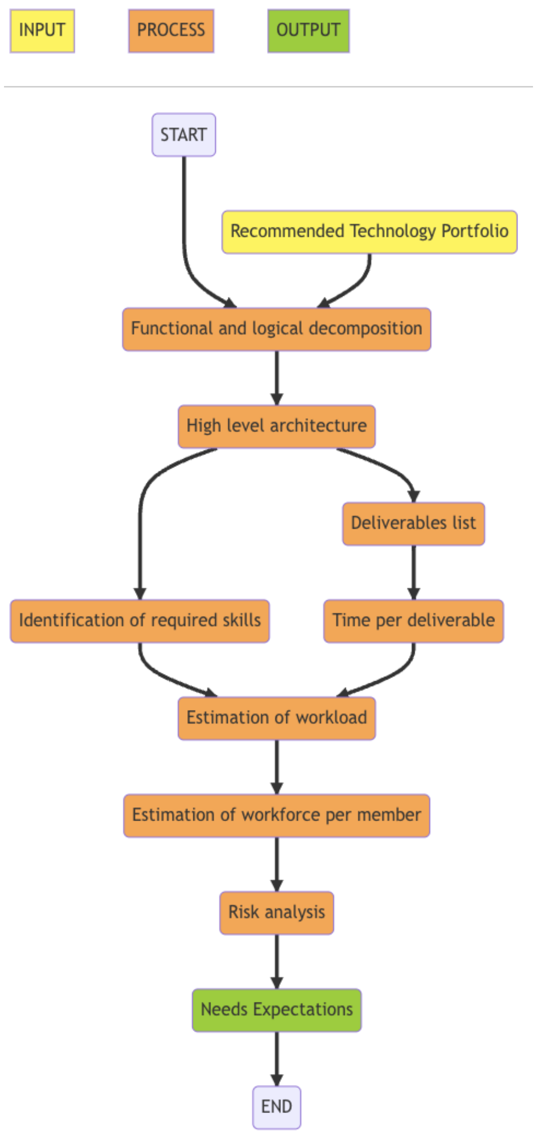


Figure 40: Flowchart of the Human Resources Needs Expectations Process

6.3.3 Step 1: Identification of required skills

Every technology is comprised of associated skills that are required in order to efficiently develop it. From programming languages to hardware engineering, each facet of technological advancement demands a unique set of skills. A good identification of these required skills is crucial to compose an efficient team that will successfully bring the technology to fruition.

In instances where there's a significant need in a specific area that isn't commonly found among engineering students, proactive measures must be taken. It becomes necessary to recruit motivated individuals and allocate them to training projects or on sub-critical assemblies. These projects should be tailored with specific requirements aimed at honing the required skill set. By providing targeted training and hands-on experience, organisations can bridge skill gaps and cultivate the expertise needed to propel technological innovation forward. Through strategic recruitment and skill development initiatives, teams can ensure they possess the necessary capabilities to tackle emerging challenges head-on.

6.3.4 Step 2: Estimation of workload

Once we've identified the necessary skill set for a project, the next step is to assess the overall workload involved in advancing from one TRL to the next. This evaluation is crucial for effectively planning and managing the project's progression.

One approach to estimating this workload is drawing from past project experiences or leveraging connections with organisations engaged in comparable developments. By examining historical data and insights from similar endeavors, we can gain valuable perspective on the tasks and resources required for navigating the transition between TRLs.

At a systems engineering level, a systematic method involves establishing a comprehensive roadmap for the specific technology in question. This roadmap serves as a guide for outlining the project's trajectory and breaking it down into distinct deliverables. Each deliverable is accompanied by a "Time-to-complete" interval, providing a clear timeline for accomplishing key milestones.

By decomposing the project into manageable components and assigning timeframes to each, we can effectively gauge the overall workload associated with advancing through the TRLs. This systematic approach not only helps in estimating the effort required but also facilitates effective project management and resource allocation throughout the developmental journey.

6.3.5 Step 3: Estimation of workforce per member

To estimate the necessary manpower for a project, a critical starting point involves assessing the commitment levels of potential team members. Understanding the varying degrees of dedication across different roles is essential for constructing a cohesive and effective team.

Consider, for instance, the discrepancy in commitment between a Team Leader and a Team Member. Typically, a Team Leader invests more time and demonstrates a higher level of motivation compared to a Team Member. These distinctions in commitment are crucial factors to consider when determining staffing requirements.

The process of evaluating commitment levels is empirical in nature. One practical approach involves soliciting input directly from team members. By asking individuals to complete a form detailing their time commitment per week, we can gather valuable data to inform our decision-making process.

This method not only provides us with quantifiable information but also allows us to establish a range of time commitments across different roles within the project. Armed with this data, we can then proceed to estimate the required number of individuals to recruit, ensuring that the team composition aligns with the project's needs and objectives.

6.3.6 Step 4: Risks Mitigations

Skills Unavailability

As projects demand advanced skills not typically covered in engineering curricula, initiatives like the Space Race (Level 1 Model Rocket Competition) within the ERT demonstrate a promising path forward. Tailoring project requirements to specific skill sets ensures alignment between project objectives and available expertise, fostering a conducive environment for skill development. Furthermore, the creation of pedagogic semester projects, guided by specialised professors, offers a structured approach to address skill gaps while maximising learning potential. These initiatives not only enhance members' skill sets but also empower the association to undertake ambitious projects with confidence and proficiency.

Motivation Loss

Mitigating the risk of motivation loss is paramount in a student-led association where the workforce relies entirely on the members' dedication without contractual obligations or financial incentives. Given this context, motivation loss poses a significant threat to the project's success. To address this challenge, leveraging past data becomes imperative. By analysing historical trends of member attrition throughout the project's lifecycle, proactive measures can be implemented. This includes adjusting recruitment strategies to incorporate margins that anticipate potential dropouts. By adopting this approach, the association can better safeguard against motivation loss and sustain momentum towards achieving its objectives.

Labor Shortage

Mitigating the risk of labor shortage is crucial in a student-led association, particularly when facing challenges such as supply shortages in the workforce due to insufficient demand from the student population. In the short term, COTS solutions can be leveraged to address immediate staffing gaps. However, for sustainable mitigation in the long term, reinforcing recruitment efforts is essential. This could involve broadening recruitment channels, enhancing marketing strategies, and establishing partnerships with relevant entities. Moreover, in cases where significant recruitment issues arise, it becomes imperative to integrate new constraints into the technology roadmapping process. By proactively addressing labor shortage risks through short-term solutions and long-term recruitment strategies, the association can ensure continuity and resilience in achieving its objectives.

[THIS PAGE INTENTIONALLY LEFT BLANK]

7 Roadmap Maintenance

7.1 Roadmap Maturity Scoring

7.1.1 Purpose

By definition, a technology roadmap can be very vague or extremely detailed, and the time needed to create it might go from a few days to several years according to its accuracy and maturity. It is also clear that the more accurate and detailed a roadmap is, the more value can be taken out of it. We therefore need a method to assess the maturity of the roadmap to make sure that the produced roadmaps are compliant with the expectations.

7.1.2 Process

We are going to use the Technology Roadmapping Maturity Scale (See Table 5) to assess the maturity of the roadmaps portfolio. From this scale we can identify three types of characteristics that correspond to different steps of the ATRA. The scale is the following,

Maturity level	Name	Characteristics at that level
I	Exploration	Partial list of only the most important technologies
		Focus mainly on technology scouting and finding "blind spots"
		Uneven format, quality, and depth of roadmaps
		Not used at all for decision-making, just for information
II	Canvassing	Complete list of roadmaps across the firm
		Centralized project inventory mapped to roadmaps
		Standardization of format and dedicated roadmap owners
		"Flat" list of technologies, no explicit link to products
III	Evaluation	Explicit hierarchy of roadmaps with link to products or missions
		Clear definition of FOMs and setting of targets
		Anticipated entry-in-service dates are used to set pace
		Find and exploit synergies across business units
IV	Prescription	Roadmaps are the main way to decide on R&D investments
		Value for money is calculated for sustaining technologies
		Quantified route-to-target options (vector charts) evaluated with risk levels in products where multiple technologies are used
		Clearly prioritized and ranked list of R&D projects in each roadmap
V	Optimization	Calculation of FOM targets and value with calibrated technical models for each product, including the mapped technologies
		Validated multiyear cost models for NRC and RC
		Prioritization of R&D investments across product divisions
		Portfolio optimization for value versus risk to maximize the NPV for the firm with explicit expectations on ROI of the R&D portfolio

Table 5: Technology Roadmapping Maturity Scale [1]

For each level, its associated maturity level can be found in the previous table.

7.1.3 Step 1: Precision of "Where are we now?"

The maturity levels for the "Where are we now?" part of the ATRA can be found below:

1. Partial list of only the most important technologies (LVL 1)
2. Focus mainly on technology scouting and finding "blind spots" (LVL 1)
3. Complete list of roadmaps across the firm (LVL 2)
4. Centralised project inventory mapped to roadmaps (LVL 2)
5. "Flat" list of technologies, no explicit link to products (LVL 2)
6. Explicit hierarchy of roadmaps with link to products or missions (LVL 3)
7. Clear definition of FOMs and setting of targets (LVL 3)
8. Calculation of FOM targets and value with calibrated technical models for each product, including the mapped technologies (LVL 5)

7.1.4 Step 2: Precision of "Where should we go?"

The maturity levels for the "Where should we go?" part of the ATRA can be found below:

1. Uneven format, quality, and depth of roadmaps (LVL 1)
2. Standardisation of format and dedicated roadmap owners (LVL 2)
3. Find and exploit synergies across business units (LVL 3)
4. Value for money is calculated for sustaining technologies (LVL 4)
5. Quantified route-to-target options (vector charts) evaluated with risk levels in products where multiple technologies are used (LVL 4)

7.1.5 Step 3: Precision of "Where we are going"

The maturity levels for the "Where we are going" part of the ATRA can be found below:

1. Not used at all for decision-making, just for information (LVL 1)
2. Anticipated entry-in-service dates are used to set pace (LVL 3)
3. Roadmaps are the main way to decide on R&D investments (LVL 4)
4. Clearly prioritised and ranked list of R&D projects in each roadmap (LVL 4)
5. Validated multiyear cost models for NRC and RC (LVL 5)
6. Prioritisation of R&D investments across product divisions (LVL 5)
7. Portfolio optimisation for value versus risk to maximise the NPV for the firm with explicit expectations on ROI of the R&D portfolio (LVL 5)

7.2 Roadmap Update Instructions

7.2.1 Purpose

The technology roadmapping process requires ongoing maintenance and updates, rather than being a one-time endeavor. As we analyze prospective scenarios, there comes a point where the scenarios or decisions outlined in the roadmaps may lose relevance.

7.2.2 Process Graph

The Figure 41 depicts the process that is described in this section,

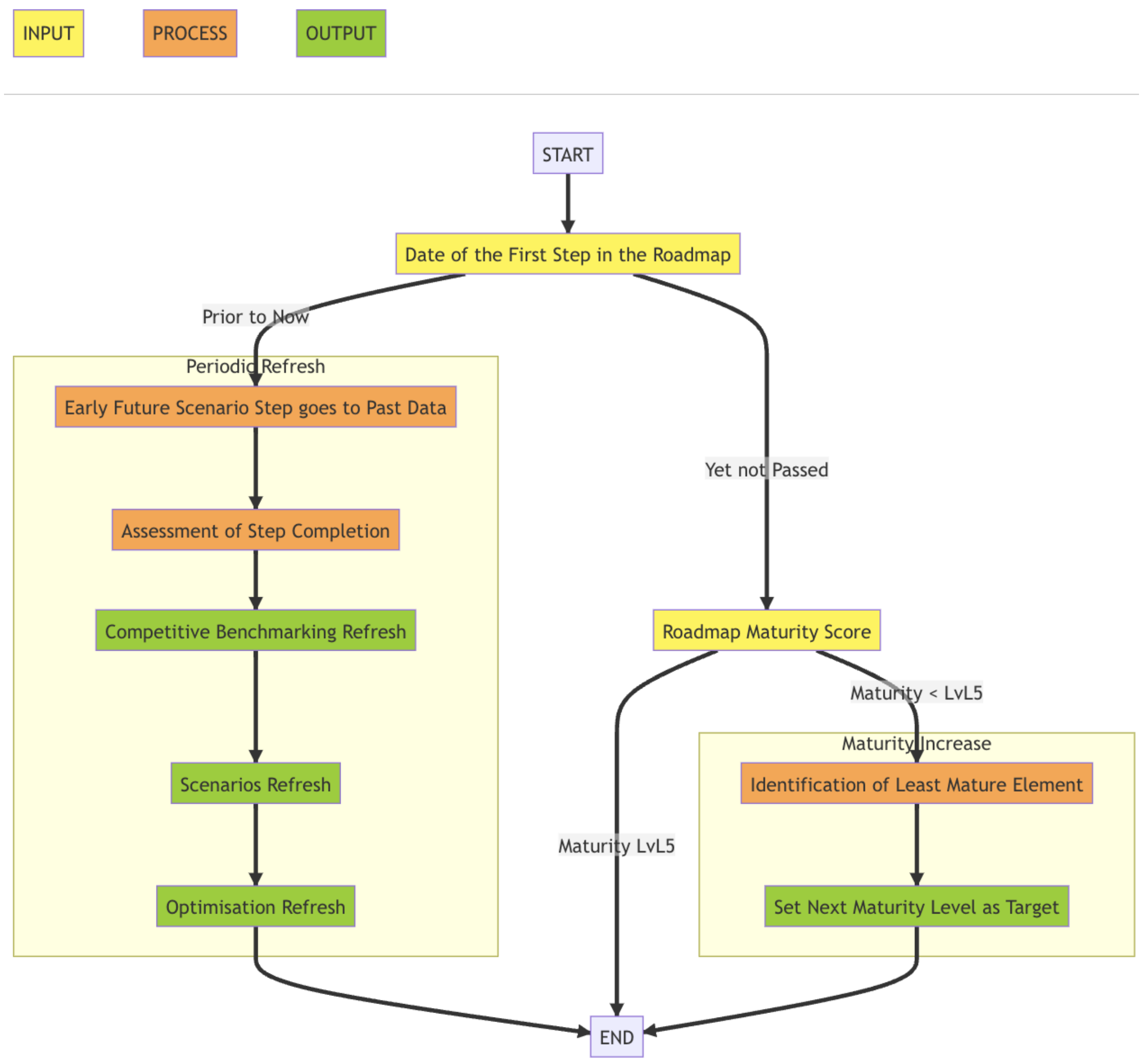


Figure 41: Flowchart of the Roadmap Update Process

7.2.3 Periodic Refresh Instructions

The scenarios outlined in a roadmap consist of sequential stages, which may, for instance, be annual advancements in the TRL. As each stage progresses, transitioning from the "future" scenario category to historical data, it moves from scenario analysis to competitive benchmarking within the roadmap process detailed in this handbook.

Two potential outcomes emerge in this scenario. Firstly, if the prediction proves accurate and the target is precisely met, the future scenarios remain relevant and applicable. However, such exact outcomes are infrequent, and minor deviations are likely. Consequently, a comprehensive refresh of all roadmap components becomes necessary to accommodate the new circumstances.

This refresh does not signify an increase in maturity but rather ensures the foundational integrity of the roadmap remains intact.

7.2.4 Maturity Increase Instructions

If the roadmap is current and there is potential for further maturation, we may proceed with enhancing it. Here, we leverage the pre-established maturity scoring to pinpoint areas warranting prioritised enhancements. The objective is to ascend to the subsequent maturity level within the least developed segment of the roadmap. For precise delineation of the next level, consult the Maturity Scoring section.

8 Conclusions

The EPFL Rocket Team is a student association that is currently seeking a high pace technological development, to accomplish the goal of reaching the Kármán line by 2030. This goes directly upfront with the nature of a student led association, which is composed of a workforce that generally lacks experience in advanced space technologies. To this intent, strong strategic planning processes must be implemented to the framework of the association.

This handbook provides a solid basis for the integration of the Advanced Technology Roadmap Architecture in the executive decisions of the organisation.

The first part of these guidelines, the competitive benchmarking, allowed the definition of clear performance markers, the Figures of Merit, to quantitatively rate the completion of the stakeholders expectations by the focal organisation. Drawing from this, the Figures of Merit of the main competitors of the organisation, which generally have similar stakeholders expectations, allow to define how competitive the focal organisation is in its activity field. A set of MATLAB tools are provided to help the user as much as possible.

In the second step, a detailed guideline is provided for establishing a system model, alongside an introduction to Multidisciplinary Design Optimization (MDO) and an initial guide for developing an MDO program.

The third step focuses on the utilization of the system model and MDO program in the context of technology roadmapping in order to identify long-term technological targets and formulate a range of viable technology trajectories, bridging current technological capabilities with projected progress rates towards the long-term target.

The last part of the process, which aims to provide with the recommended portfolio and the key executive decisions to be made in the future or to take into account for current decisions, gives a nearly fully automated portfolio optimisation system. The MATLAB tool that is appended to this handbook, helps the user to systematically optimise its decisions, taking the right objectives into account.

Finally, the roadmapping process is only efficient if it is maintained and enhanced over time. To this intent, a crucial part is added, to allow the user to efficiently assess the maturity of a roadmap, helping him draw conclusions on how much the outputs can be trusted, but also where major improvements can be made.

References

[1] O. L. D. Weck, “Technology roadmapping and development: A quantitative approach to the management of technology,” 2022. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-88346-1

[2] d. W. O. Willcox K., “Multidisciplinary design optimization.” [Online]. Available: https://ocw.mit.edu/courses/ids-338j-multidisciplinary-system-design-optimization-spring-2010/resources/mitesd_77s10_lec03/

[3] G. M. E. Alejandro C. Olivieri, “in practical three-way calibration,” 2014. [Online]. Available: https://www.sciencedirect.com/topics/chemistry/figure-of-merit

[4] SBFI, “Trl definition.” [Online]. Available: https://www.sbfi.admin.ch/dam/sbfi/en/dokumente/technology_readinesslevelstrl.pdf.download.pdf/technology_readinesslevelstrl.pdf

[5] C. L. M. Olivier L. De Weck, “Complex system classification,” 2004. [Online]. Available: http://strategic.mit.edu/docs/3_36_INCOSE2004_complexsys.pdf

[6] I. Y. K. Olivier L. De Weck, “Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation,” 2005. [Online]. Available: http://strategic.mit.edu/docs/2_11_SMO_AWS_multiobjective.pdf

[7] S. J. M. Renu Sabharwal, “An intelligent literature review: adopting inductive approach to define machine learning applications in the clinical domain,” 2022. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00605-3

[8] D. Dori, “Model-based systems engineering with opm and sysml.”

[9] ——, “Object-process methodology a holistic systems paradigm.”

[10] N. R. Soderborg, “Representing systems through object-process methodology and axiomatic design.” [Online]. Available: https://core.ac.uk/download/pdf/4399822.pdf

A Documentation Templates

A.1 Figures of Merit Documentation Template

1. **System ID: ...**
2. **Roadmap ID: ...**
3. **Author: ...**
4. **Version:** *1.0.0*
5. **Technology List:** *(Input of the process)*
   - *Technology i*
6. **Taxonomy of Technology:** *(Place the elements of the technology list in the right cells)*

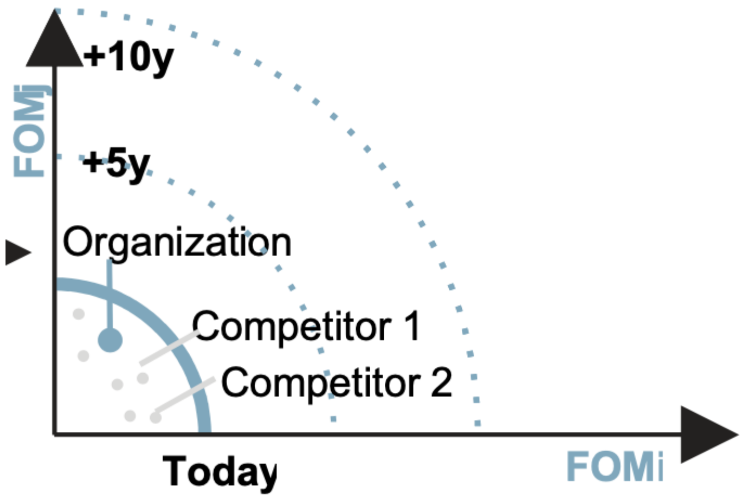
Technology Matrix	Matter	Energy	Information
Transforming			
Transporting			
Storing			

7. **Figures of Merit Set:**
   (a) *FOMi:*
   - Class: *(Performance, Efficiency,...), Directly prescripted via the taxonomy in the handbook*
   - Unit: *(Specify the unit of the FOM), Requires further research* $\rightarrow$ *User-Defined*
   - Limit: *(Numerical value + URL to justification), Requires further research* $\rightarrow$ *User-Defined*
8. **Data Set Description:** *(Provide with the sources)*
9. **Weight Choice Justification:**
   (a) *FOMi:*
   - Weight: *(Numerical value* $\in [0,1]$*)* $\rightarrow$ *User-Defined*
   - Stakeholder Expectation: *(From Stakeholder Analysis)*
   - Related Stakeholder Cluster: *(From Stakeholder Analysis)*
10. **Technology Trends:** *(Also provide with the slope numerical value given by the MATLAB Tool)*
    (a) *FOMi: (Graph of S-Curve Prediction, using the provided MATLAB Tool)*

Interpretation of the graph: *(Write a paragraph explaining your interpretation of the graphs and trends)*

A.2 Similar Missions Analysis Documentation Template

1. **System ID:** ...
2. **Roadmap ID:** ...
3. **Author:** ...
4. **Version:** *1.0.0*
5. **Technology/System Definition:** *(Define the nature of the focal system in one sentence)*
6. **Technology/System Boundaries:** *(Give contextual information like being a student association)*
7. **High Level Specifications:** *(Give details that are important to filter the research results)*
   - ...
8. **Data Sources:**
   - (i) Author, Title, Date: URL
9. **FOMs Numerical Values:**
   (a) *FOMi:*
      - Organisation j, FOM Value
10. **Multidimensional FOM Map:**
   (a) *FOMi/FOMj: (Graph of FOM values and current Pareto Front), Use the provided MATLAB Tool for 2D Pareto Front Plot*



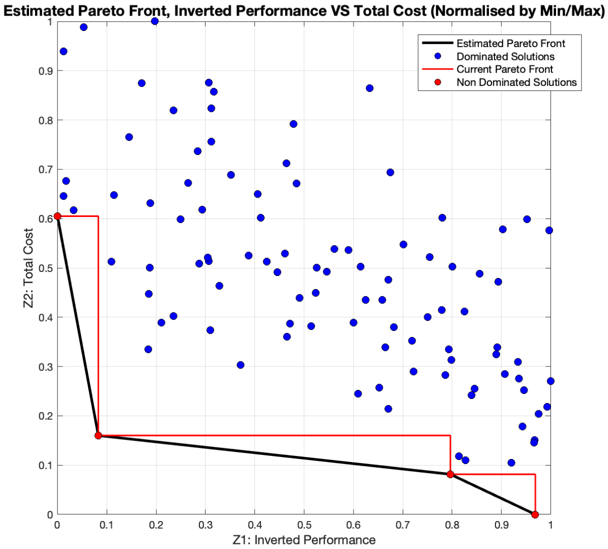
Interpretation of the graph: *(Write a paragraph explaining your interpretation of the graphs and trends)*

A.3 Technology Readiness Levels Documentation Template

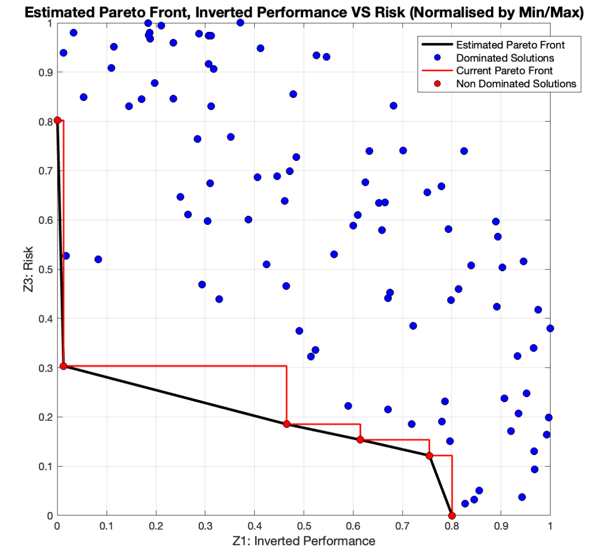
1. **System ID:** ...
2. **Roadmap ID:** ...
3. **Author:** ...
4. **Version:** *1.0.0*
5. **Activity Description:** *(Succintely define the focal system and its application/environment)*
6. **Estimated Current TRL:** *(Give the value and a short justification)*
7. **Target TRL:** *(Give the value and a short justification)*

A.4 Pareto Optimal Set of Technology Investment Portfolios Documentation Template

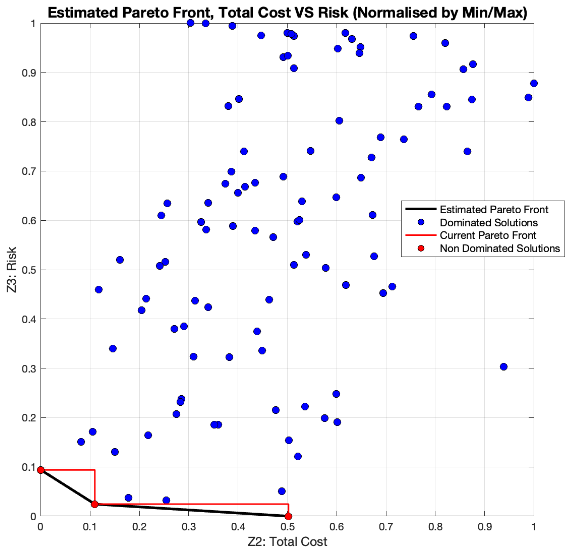
1. **System ID: ...**
2. **Roadmap ID: ...**
3. **Author: ...**
4. **Version:** *1.0.0*
5. **Pareto Front Plots:** *(Fully automated with the provided MATLAB Tool, all graphs given as output)*
   - Z_1 VS Z_2 Solution Space *(The plot below serves as an example, solutions are generated using stochastic methods and are therefore not real)*



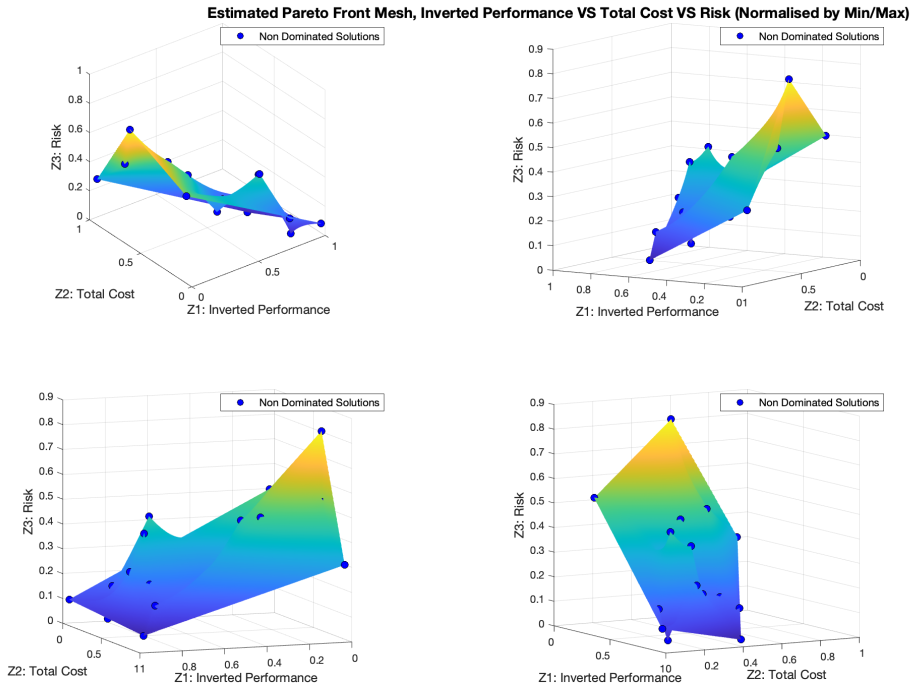
   - Z_1 VS Z_3 Solution Space *(The plot below serves as an example, solutions are generated using stochastic methods and are therefore not real)*



- Z_2 VS Z_3 Solution Space *(The plot below serves as an example, solutions are generated using stochastic methods and are therefore not real)*

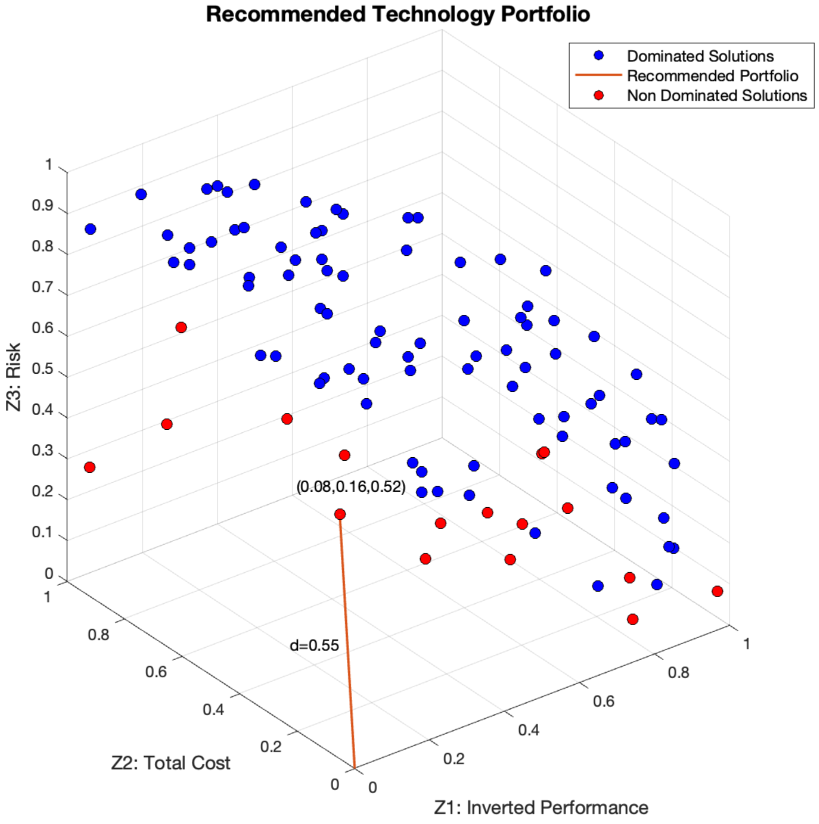


- Z_1 VS Z_2 VS Z_3 Solution Space *(The plot below serves as an example, solutions are generated using stochastic methods and are therefore not real)*

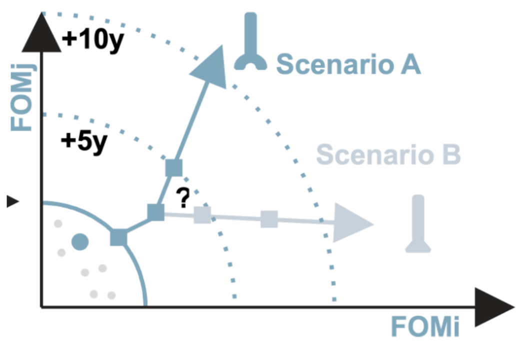


A.5 Recommended Technology Portfolio Documentation Template

1. **System ID:** ...
2. **Roadmap ID:** ...
3. **Author:** ...
4. **Version:** *1.0.0*
5. **Multidimensional Plot of the Recommended Portfolio:** *(Fully automated with the provided MATLAB Tool, graph given as output) (The plot below serves as an example, solutions are generated using stochastic methods and are therefore not real)*



6. **Vector Charts of the Recommended Portfolio:**



7. **Valuation of the Recommended Portfolio:***(Define it with the cost vector, no further work required)*
8. **Gantt Chart of the Recommended Portfolio:***(Define it with the FOM vectors, no further work required)*

A.6 Human Resources Needs Expectations Documentation Template

1. **System ID:** ...
2. **Roadmap ID:** ...
3. **Author:** ...
4. **Version:** *1.0.0*
5. **List of Relevant Skills:** *(Qualitative paragraph)*
6. **Estimated Workload:***(Qualitative paragraph)*
7. **Estimated Workforce Requirements:***(Qualitative paragraph)*

A.7 Roadmap Maturity Scoring Documentation Template

1. **System ID:** ...
2. **Roadmap ID:** ...
3. **Author:** ...
4. **Version:** *1.0.0*
5. **Maturity of the Competitive Benchmarking:** *(Prescripted by the handbook)*
6. **Maturity of the Models and Scenarios:** *(Prescripted by the handbook)*
7. **Maturity of the Portfolio Optimisation:** *(Prescripted by the handbook)*

B Competitive Benchmarking MATLAB Toolbox

Function for the value of the Technology Trend ($\frac{dFOM}{dt}$)

```matlab
%% technology_trend.m
%
% Author:   Martin LEMAIRE
%
% Project: [ME-314] - Technology Roadmapping for a Suborbital Rocket with
%           Multidisciplinary Design Optimisation
%
% Date:     Spring 2024
%
% Version: 1.0.0
%
% Inputs:
%            data:
%                type: array
%                description: normalised past data of the focal FOM
%
% Output:    slope:
%                type: float
%                description: slope of the linear regression line

function slope = technology_trend(data)
    % Generate x values (indices)
    x = 1:numel(data);

    % Perform linear regression
    p = polyfit(x, data, 1); % Linear fit

    % We simply return the slope
    slope = p(1);
end
```

Function for the Linear Prediction

```matlab
%% linear_prediction.m
%
% Author:   Martin LEMAIRE
%
% Project: [ME-314] - Technology Roadmapping for a Suborbital Rocket with
%           Multidisciplinary Design Optimisation
%
% Date:     Spring 2024
%
% Version: 1.0.0
%
% Inputs:
%            data:
%                type: array
%                description: normalised past data of the focal FOM
%
%            nb_expansion_points:
%                type: int
%                description: number of years of prediction after the last
%                             given point
%
% Output:    [x,y]:
%                type: array,array
%                description: coordinates of the points of the prediction
%                             line

function [x, y] = linear_prediction(data, nb_expansion_points)
    % Generate x values (indices)
    x = 1:numel(data);

    % Perform linear regression
```

```
    p = polyfit(x, data, 1); % Linear fit
    x = 1:numel(data)+nb_expansion_points;

    % Generate y values corresponding to the linear regression line
    y = polyval(p, x);
end
```

Function for the Exponential Prediction

```
%% exponential_prediction.m
%
% Author:  Martin LEMAIRE
%
% Project: [ME-314] - Technology Roadmapping for a Suborbital Rocket with
%          Multidisciplinary Design Optimisation
%
% Date:    Spring 2024
%
% Version: 1.0.0
%
% Inputs:
%           data:
%               type: array
%               description: normalised past data of the focal FOM
%
%           nb_expansion_points:
%               type: int
%               description: number of years of prediction after the last
%                            given point
%
% Output:   [x,y]:
%               type: array,array
%               description: coordinates of the points of the prediction
%                            curve

function [x, y] = exponential_prediction(data, nb_expansion_points)

    % Linearisation of the data, to be able to perform the regression
    data = log(data+1);

    % Generate x values (indices)
    x = 1:numel(data);

    % Perform linear regression
    p = polyfit(x, data, 1); % Linear fit
    x = 1:numel(data)+nb_expansion_points;

    % Generate y values corresponding to the linear regression line
    y = polyval(p, x);

    % We can now exponentiate the data
    y = exp(y)-1;
end
```

Function for the Logarithmic Prediction

```
%% logarithmic_prediction.m
%
% Author:  Martin LEMAIRE
%
% Project: [ME-314] - Technology Roadmapping for a Suborbital Rocket with
%          Multidisciplinary Design Optimisation
%
% Date:    Spring 2024
%
% Version: 1.0.0
%
```

```
% Inputs:
%           data:
%               type: array
%               description: normalised past data of the focal FOM
%
%           nb_expansion_points:
%               type: int
%               description: number of years of prediction after the last
%                            given point
%
% Output:   [x,y]:
%               type: array,array
%               description: coordinates of the points of the prediction
%                            curve

function [x, y] = logarithmic_prediction(data, nb_expansion_points)

    % We linearise the data to be able to perform the linear regression
    data = exp(data);

    % Generate x values (indices)
    x = 1:numel(data);

    % Perform linear regression
    p = polyfit(x, data, 1); % Linear fit

    x = 1:numel(data)+nb_expansion_points;

    % Generate y values corresponding to the linear regression line
    y = polyval(p, x);

    % We can now apply the logarithmic transformation to the data
    y = log(y);
end
```

Function for the Logistic Prediction

```
%% logistic_prediction.m
%
% Author:   Martin LEMAIRE
%
% Project:  [ME-314] - Technology Roadmapping for a Suborbital Rocket with
%           Multidisciplinary Design Optimisation
%
% Date:     Spring 2024
%
% Version:  1.0.0
%
% Inputs:
%           data:
%               type: array
%               description: normalised past data of the focal FOM
%
%           nb_expansion_points:
%               type: int
%               description: number of years of prediction after the last
%                            given point
%
% Output:   [x,y]:
%               type: array,array
%               description: coordinates of the points of the prediction
%                            curve

function [x, y] = logistic_prediction(data, nb_expansion_points)

    % We linearise the data to be able to perform a linear regression, this
    % the logit function, inverse of the logistic function
    data = log(data./(1-data));

```

```
    % Values close to 0 or 1 give Inf, so we simply turn them into
    % sufficiently large values, which are still manageable for the program
    data(data==Inf)=6;
    data(data==-Inf)=-6;

    % Generate x values (indices)
    x = 1:numel(data);

    % Perform linear regression
    p = polyfit(x, data, 1); % Linear fit
    x = 1:numel(data)+nb_expansion_points;

    % Generate y values corresponding to the linear regression line
    y = polyval(p, x);

    % We can now apply the backwards transformation to obtain the logistic
    % function
    y = 1./(1+exp(-y));
end
```

Function for the FOM Prediction Plots

```
%% fom_prediction_plot.m
%
% Author:  Martin LEMAIRE
%
% Project: [ME-314] - Technology Roadmapping for a Suborbital Rocket with
%          Multidisciplinary Design Optimisation
%
% Date:    Spring 2024
%
% Version: 1.0.0
%
% Inputs:
%           data:
%               type: array
%               description: normalised past data of the focal FOM
%
%           prediction_data_x:
%               type: array
%               description: Time vector of the prediction curve
%
%           prediction_data_y:
%               type: array
%               description: FOM vector of the prediction curve
%
%           method:
%               type: string
%               description: string for the title of the plot describing %
                               the extrapolation method
%
% Output:   None

function fom_prediction_plot(data, prediction_data_x, prediction_data_y, method)

    % Stairs pattern
    % To represent the state of the FOM at any point in time, we cannot
    % directly link the points together, because it would be wrong.
    % Therefore, instead, we trace a stairs pattern, characterising each
    % point as an impulse.
    x_data = 1:numel(data);
    if length(x_data)>1
        x_stairs_data = [];
        y_stairs_data = [];
        for i = 1: length(x_data)-1
            x_stairs_data = [x_stairs_data, x_data(i)];
            y_stairs_data = [y_stairs_data, data(i)];
            x_stairs_data = [x_stairs_data, x_data(i+1)];
            y_stairs_data = [y_stairs_data, data(i)];
```

```
        end
        x_stairs_data = [x_stairs_data, x_data(length(x_data))];
        y_stairs_data = [y_stairs_data, data(length(data))];
    else
        x_stairs_data = x_data;
        y_stairs_data = data;
    end

    % In MATLAB, indices start at 1, but we start at year 0
    x_stairs_data = x_stairs_data - 1;
    x_data = x_data - 1;
    prediction_data_x = prediction_data_x - 1;

    % Graph
    figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
    set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
    set(gcf, 'Color', 'w');
    set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
    plot(x_stairs_data,y_stairs_data, 'Color','k', 'lineWidth', 2);
    hold all;
    scatter(x_data,data, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'k');
    plot(prediction_data_x,prediction_data_y, '-', 'lineWidth', 2.5, 'Color','r');
    grid on;
    xlabel('Time [Years]', 'FontSize', 12);
    ylabel('FOM_i', 'FontSize', 12);
    legend('','Past Data','Linear Fit and Future Prediction', 'Location', 'best', 'FontSize',
        10);
    title(strcat('Prediction of Future Evolution, ',method,' Extrapolation, Normalised by Min/
        Max'), 'FontSize', 14);
    axis square
end
```

Function for the Similar Missions Analysis Plots

```
%% similar_missions_analysis.m
%
% Author:  Martin LEMAIRE
%
% Project: [ME-314] - Technology Roadmapping for a Suborbital Rocket with
%          Multidisciplinary Design Optimisation
%
% Date:    Spring 2024
%
% Version: 1.0.0
%
% Inputs:
%           FOMI_norm:
%               type: array
%               description: normalised past data of the abscissa FOM
%
%           FOMJ_norm:
%               type: array
%               description: normalised past data of the ordinates FOM
%
%           ERT_indices:
%               type: array
%               description: Indices of the rockets that were developped by the ERT
%
%           FOMI_string:
%               type: string
%               description: label of the x_axis
%
%           FOMJ_string:
%               type: string
%               description: label of the y_axis
%
%           title_string:
%               type: string
%               description: string for the title of the plot
```

```
%
% Output:   None

function similar_missions_analysis(FOMI_norm,FOMJ_norm,ERT_indices,FOMI_string,FOMJ_string,
    title_string)

    % Calculating the distance of the best in class from the origin to
    % determine the radius of the Pareto Front
    distances = sqrt(FOMI_norm.^2+FOMJ_norm.^2);
    distance = max(distances);
    theta = linspace(0,pi/2,100);
    x = distance*cos(theta);
    y = distance*sin(theta);

    % Definition of the balanced trade-offs line to help visualise and
    % avoid extreme decisions oriented towards the progressino of only one
    % objective.
    equilibrium_x = linspace(0,distance,100);

    figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
    set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
    set(gcf, 'Color', 'w');
    set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
    plot(x,y, '-', 'lineWidth', 3, 'Color', 'black');
    hold all;
    scatter(FOMI_norm, FOMJ_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b
        ', 'SizeData', 50);
    scatter(FOMI_norm(ERT_indices), FOMJ_norm(ERT_indices), 'Marker', 'o', 'MarkerEdgeColor',
        'k', 'MarkerFaceColor', 'r', 'SizeData', 50);
    plot(equilibrium_x,equilibrium_x, '-.', 'lineWidth', 1, 'Color', 'k');
    grid on;
    xlabel(FOMI_string, 'FontSize', 12);
    ylabel(FOMJ_string, 12);
    legend('Estimated Current Pareto Front', 'Other Organisations Past Technologies', 'Past 
        ERT Technologies', 'Balanced Trade-Offs Line', 'Location', 'best', 'FontSize', 10);
    title(title_string, 'FontSize', 14);
    axis square
end
```

C Portfolio Optimisation MATLAB Script

```
%% portfolio_optimiser.m
%
% Author:  Martin LEMAIRE
%
% Project: [ME-314] - Technology Roadmapping for a Suborbital Rocket with
%          Multidisciplinary Design Optimisation
%
% Date:    Spring 2024
%
% Version: 1.0.0

% Provides with the data along the steps of the program if 1, else make it 0
DISPLAY = 0;

% Use a random data sample, allowing to test the program if 1, else make it 0
TEST = 1;

%% Dataset Definition

% Matrix containing the System FOM vectors of each scenario, with the FOM
% values in chronological order
sys_FOM_matrix = [
    0.4 0.6 0.64 0.7 0.8;
    0.4 0.46 0.58 0.69 0.76
    ];

% Matrix containing the Cost vectors of each scenario, with the cost values
% in chronological order
cost_matrix = [
    30000 26000 15000 42000 36000;
    20000 22000 28000 24000 32000
    ];

%% Verification Data
% This part aims to generate a sample of data, that is random to test in
% several different environments. The aim is also to bound the randomness
% by applying stochastic probabilities methods, in order to generate a realistic
% sample of data.

if TEST==1
    % Define parameters
    num_vectors = 100; % Number of scenarios in the matrix
    vector_length = 20; % Number of years of data

    % Generate the matrix
    sys_FOM_matrix = zeros(num_vectors, vector_length);

    % Generate vectors
    for i = 1:num_vectors
        % Generate first value at 0.3
        sys_FOM_matrix(i, 1) = 0.3;

        % Generate last value strictly in [0.6, 1]
        sys_FOM_matrix(i, end) = rand*(1-0.6) + 0.6;

        % Generate strictly increasing values in between
        for j = 2:(vector_length-1)
            sys_FOM_matrix(i, j) = sys_FOM_matrix(i, j-1) + rand*(sys_FOM_matrix(i, end) -
                sys_FOM_matrix(i, j-1));
        end
    end

    % Display the matrix
    if DISPLAY==1
        disp('Generated␣matrix:');
        disp(sys_FOM_matrix);
    end

    % We estimate that a larger increase in the system FOM means a larger
```

```
    % associated cost.

    % Initialise the variations matrix
    variations_sys_FOM = zeros(size(sys_FOM_matrix, 1), size(sys_FOM_matrix, 2) - 1);

    % Compute variations for each row
    for i = 1:size(sys_FOM_matrix, 1)
        variations_sys_FOM(i, :) = diff(sys_FOM_matrix(i, :));
    end

    % Normalisation of the variations to format them as a probabilities
    probability = (variations_sys_FOM - min(variations_sys_FOM, [], 'all')) / (max(
        variations_sys_FOM, [], 'all') - min(variations_sys_FOM, [], 'all'));

    % Display the second matrix
    if DISPLAY==1
        disp('Generated␣variations␣matrix:');
        disp(variations_sys_FOM);
    end

    % Initialise the matrix
    cost_matrix = zeros(size(variations_sys_FOM));

    % Generate values for each vector
    for i = 1:size(variations_sys_FOM, 1)
        for j = 1:size(variations_sys_FOM, 2)

            peak_position = 1.3*probability(i,j); % Empirically determined factor to make the
                peak position more realistic
            cost = gaussian_random(peak_position);

            % Generate random value between min_val and max_val
            cost_matrix(i, j) = cost*40000+20000;
        end
    end

    % Display the second matrix
    if DISPLAY==1
        disp('Generated␣second␣matrix:');
        disp(cost_matrix);
    end
end

%% System FOM Objective, Z1, Computation

% Negated vector to turn the objective into a minimisation and taking the
% last value as it is the expected final FOM
Z1 = -sys_FOM_matrix(:, size(sys_FOM_matrix, 2));

% Display the result
if DISPLAY==1
    disp('System␣FOM␣Objective␣Values:');
    disp(Z1);
end

%% Total Cost Objective, Z2, Computation

% Initialise the vector to store sum of elements
Z2 = zeros(size(cost_matrix, 1), 1);

% Loop through each row of the matrix
for i = 1:size(cost_matrix, 1)
    % Calculate the sum of elements in the current row
    Z2(i) = sum(cost_matrix(i, :));
end

% Display the result
if DISPLAY==1
    disp('Total␣Cost␣Objective␣Values:');
    disp(Z2);
end

```

```
%% Risk Objective, Z3, Computation

% Initialise the vector to store standard deviations
Z3 = zeros(size(cost_matrix, 1), 1);

% Loop through each row of the matrix
for i = 1:size(cost_matrix, 1)
    % Calculate the standard deviation of elements in the current row
    Z3(i) = std(cost_matrix(i, :));
end

% Display the result
if DISPLAY==1
    disp('Risk Objective Values:');
    disp(Z3);
end

%% Objectives Normalisation

Z1_norm = (Z1 - min(Z1)) / (max(Z1) - min(Z1));
Z2_norm = (Z2 - min(Z2)) / (max(Z2) - min(Z2));
Z3_norm = (Z3 - min(Z3)) / (max(Z3) - min(Z3));

if DISPLAY==1
    disp('Normalised System FOM Objective Values:')
    disp(Z1_norm);
    disp('Normalised Total Cost Objective Values:')
    disp(Z2_norm);
    disp('Normalised Risk Objective Values:')
    disp(Z3_norm);
end

%% Solution Space Graph

% Create a 3D plot
figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
scatter3(Z1_norm, Z2_norm, Z3_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor',
    'b', 'SizeData', 50);
grid on;
xlabel('Inverted Performance', 'FontSize', 12);
ylabel('Total Cost', 'FontSize', 12);
zlabel('Risk', 'FontSize', 12);
legend('Solutions', 'Location', 'best', 'FontSize', 10);
title('Solution Space of Portfolios, Performance VS Total Cost VS Risk (Normalised by Min/Max)
    ', 'FontSize', 14);
axis square
saveas(gcf, 'plot1.png');

% Create a 2D plot of Performance VS Cost
figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
scatter(Z1_norm, Z2_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', '
    SizeData', 50);
grid on;
xlabel('Inverted Performance', 'FontSize', 12);
ylabel('Total Cost', 'FontSize', 12);
legend('Solutions', 'Location', 'best', 'FontSize', 10);
title('Performance VS Total Cost (Normalised by Min/Max)', 'FontSize', 14);
axis square
saveas(gcf, 'plot2.png');

% Create a 2D plot of Performance VS Risk
figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
```

```
scatter(Z1_norm, Z3_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', '
    SizeData', 50);
grid on;
xlabel('Inverted Performance', 'FontSize', 12);
ylabel('Risk', 'FontSize', 12);
legend('Solutions', 'Location', 'best', 'FontSize', 10);
title('Performance VS Risk (Normalised by Min/Max)', 'FontSize', 14);
axis square
saveas(gcf, 'plot3.png');

% Create a 2D plot of Total Cost VS Risk
figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
scatter(Z2_norm, Z3_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', '
    SizeData', 50);
grid on;
xlabel('Total Cost', 'FontSize', 12);
ylabel('Risk', 'FontSize', 12);
legend('Solutions', 'Location', 'best', 'FontSize', 10);
title('Total Cost VS Risk (Normalised by Min/Max)', 'FontSize', 14);
axis square
saveas(gcf, 'plot4.png');

%% 2D Pareto Front Approximation - Z1 VS Z2

pareto_Z1=Z1_norm;
pareto_Z2=Z2_norm;

indices_to_remove = [];

% Finding Non Pareto Optimal scenarios
for i = 1:length(pareto_Z1)
    for n = 1:length(pareto_Z1)
        if pareto_Z1(n)<pareto_Z1(i) && pareto_Z2(n)<pareto_Z2(i) && ~any(indices_to_remove ==
            i)
            indices_to_remove = [indices_to_remove, i];
        end
    end
end

% Removing the Non Pareto Optimal scenarios
pareto_Z1(indices_to_remove)=[];
pareto_Z2(indices_to_remove)=[];

figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
scatter(Z1_norm, Z2_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', '
    SizeData', 50);
hold all;
plot(sort(pareto_Z1),sort(pareto_Z2, 'descend'), '-', 'lineWidth', 1.5, 'MarkerFaceColor', 'r'
    );
scatter(pareto_Z1, pareto_Z2, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'r', '
    SizeData', 50);
grid on;
xlabel('Inverted Performance', 'FontSize', 12);
ylabel('Total Cost', 'FontSize', 12);
legend('Solutions', 'Estimated Pareto Front', 'Pareto Optimal Solutions', 'Location', 'best',
    'FontSize', 10);
title('Estimated Pareto Front, Performance VS Total Cost (Normalised by Min/Max)', 'FontSize',
     14);
axis square
saveas(gcf, 'plot5.png');

%% 2D Pareto Front Approximation - Z1 VS Z3

pareto_Z1=Z1_norm;
pareto_Z3=Z3_norm;

```

```
indices_to_remove = [];

% Finding Non Pareto Optimal scenarios
for i = 1:length(pareto_Z1)
    for n = 1:length(pareto_Z1)
        if pareto_Z1(n)<pareto_Z1(i) && pareto_Z3(n)<pareto_Z3(i) && ~any(indices_to_remove == i)
            indices_to_remove = [indices_to_remove, i];
        end
    end
end

% Removing the Non Pareto Optimal scenarios
pareto_Z1(indices_to_remove)=[];
pareto_Z3(indices_to_remove)=[];

figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
scatter(Z1_norm, Z3_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', 'SizeData', 50);
hold all;
plot(sort(pareto_Z1), sort(pareto_Z3,'descend'), '-', 'lineWidth', 1.5, 'MarkerFaceColor', 'r');
scatter(pareto_Z1, pareto_Z3, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'r', 'SizeData', 50);
grid on;
xlabel('Inverted␣Performance', 'FontSize', 12);
ylabel('Risk', 'FontSize', 12);
legend('Solutions', 'Estimated␣Pareto␣Front','Pareto␣Optimal␣Solutions', 'Location', 'best', 'FontSize', 10);
title('Estimated␣Pareto␣Front,␣Performance␣VS␣Risk␣(Normalised␣by␣Min/Max)', 'FontSize', 14);
axis square
saveas(gcf, 'plot6.png');

%% 2D Pareto Front Approximation - Z2 VS Z3

pareto_Z2=Z2_norm;
pareto_Z3=Z3_norm;

indices_to_remove = [];

% Finding Non Pareto Optimal scenarios
for i = 1:length(pareto_Z2)
    for n = 1:length(pareto_Z2)
        if pareto_Z2(n)<pareto_Z2(i) && pareto_Z3(n)<pareto_Z3(i) && ~any(indices_to_remove == i)
            indices_to_remove = [indices_to_remove, i];
        end
    end
end

% Removing the Non Pareto Optimal scenarios
pareto_Z2(indices_to_remove)=[];
pareto_Z3(indices_to_remove)=[];

figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
scatter(Z2_norm, Z3_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', 'SizeData', 50);
hold all;
plot(sort(pareto_Z2), sort(pareto_Z3,'descend'), '-', 'lineWidth', 1.5, 'Color', 'r');
scatter(pareto_Z2, pareto_Z3, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'r', 'SizeData', 50);
grid on;
xlabel('Total␣Cost', 'FontSize', 12);
ylabel('Risk', 'FontSize', 12);
legend('Solutions', 'Estimated␣Pareto␣Front','Pareto␣Optimal␣Solutions', 'Location', 'best', 'FontSize', 10);
```

```
title('Estimated␣Pareto␣Front,␣Total␣Cost␣VS␣Risk␣(Normalised␣by␣Min/Max)', 'FontSize', 14);
axis square
saveas(gcf, 'plot7.png');

%% 3D Pareto Front Approximation - Z1 VS Z2 VS Z3

pareto_Z1=Z1_norm;
pareto_Z2=Z2_norm;
pareto_Z3=Z3_norm;

indices_to_remove = [];

% Finding Non Pareto Optimal scenarios
for i = 1:length(pareto_Z1)
    for n = 1:length(pareto_Z1)
        if pareto_Z1(n)<pareto_Z1(i) && pareto_Z2(n)<pareto_Z2(i) && pareto_Z3(n)<pareto_Z3(i)
             && ~any(indices_to_remove == i)
            indices_to_remove = [indices_to_remove, i];
        end
    end
end

% Removing the Non Pareto Optimal scenarios
pareto_Z1(indices_to_remove)=[];
pareto_Z2(indices_to_remove)=[];
pareto_Z3(indices_to_remove)=[];

figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
scatter3(Z1_norm, Z2_norm, Z3_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor',
    'b', 'SizeData', 50);
hold all;
scatter3(pareto_Z1, pareto_Z2, pareto_Z3, 'Marker', 'o', 'MarkerEdgeColor', 'k', '
    MarkerFaceColor', 'r', 'SizeData', 50);
grid on;
xlabel('Inverted␣Performance', 'FontSize', 12);
ylabel('Total␣Cost', 'FontSize', 12);
zlabel('Risk', 'FontSize', 12);
legend('Solutions','Pareto␣Optimal␣Solutions', 'Location', 'best', 'FontSize', 10);
title('Estimated␣Pareto␣Front,␣Performance␣VS␣Total␣Cost␣VS␣Risk␣(Normalised␣by␣Min/Max)', '
    FontSize', 14);
axis square
saveas(gcf, 'plot8.png');

% Pareto Front Surface Representation
figure('Units', 'inches', 'Position', [0, 0, 15, 12]);
set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
F = scatteredInterpolant(pareto_Z1,pareto_Z2,pareto_Z3,'natural','none');
sgr = linspace(min(pareto_Z1),max(pareto_Z1),2000);
ygr = linspace(min(pareto_Z2),max(pareto_Z2),2000);
[XX,YY] = meshgrid(sgr,ygr);
ZZ = F(XX,YY);
subplot(2,2,1)
mesh(XX,YY,ZZ)
hold on
scatter3(pareto_Z1,pareto_Z2,pareto_Z3, 'Marker', 'o', 'MarkerEdgeColor', 'k', '
    MarkerFaceColor', 'b', 'SizeData', 50);
hold off
grid on;
xlabel('Inverted␣Performance', 'FontSize', 12);
ylabel('Total␣Cost', 'FontSize', 12);
zlabel('Risk', 'FontSize', 12);
legend('','Pareto␣Optimal␣Solutions', 'Location', 'best', 'FontSize', 10);
axis square
title('Estimated␣Pareto␣Front␣Mesh,␣Performance␣VS␣Total␣Cost␣VS␣Risk␣(Normalised␣by␣Min/Max)'
    , 'FontSize', 14,'HorizontalAlignment','left');
subplot(2,2,2)
mesh(XX,YY,ZZ)
```

```
hold on
scatter3(pareto_Z1,pareto_Z2,pareto_Z3, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', 'SizeData', 50);
hold off
view(-148,8)
grid on;
xlabel('Inverted␣Performance', 'FontSize', 12);
ylabel('Total␣Cost', 'FontSize', 12);
zlabel('Risk', 'FontSize', 12);
legend('','Pareto␣Optimal␣Solutions', 'Location', 'best', 'FontSize', 10);
axis square
subplot(2,2,3)
mesh(XX,YY,ZZ)
hold on
scatter3(pareto_Z1,pareto_Z2,pareto_Z3, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', 'SizeData', 50);
hold off
view(-200,8)
grid on;
xlabel('Inverted␣Performance', 'FontSize', 12);
ylabel('Total␣Cost', 'FontSize', 12);
zlabel('Risk', 'FontSize', 12);
legend('','Pareto␣Optimal␣Solutions', 'Location', 'best', 'FontSize', 10);
axis square
subplot(2,2,4)
mesh(XX,YY,ZZ)
hold on
scatter3(pareto_Z1,pareto_Z2,pareto_Z3, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', 'SizeData', 50);
hold off
view(-300,8)
grid on;
xlabel('Inverted␣Performance', 'FontSize', 12);
ylabel('Total␣Cost', 'FontSize', 12);
zlabel('Risk', 'FontSize', 12);
legend('','Pareto␣Optimal␣Solutions', 'Location', 'best', 'FontSize', 10);
axis square
saveas(gcf, 'plot9.png');

%% Distance from Utopian Point Graph

% Computation of Euclidean distance
distances=zeros(length(pareto_Z1), 1);
for i = 1:length(pareto_Z1)
    distances(i)=sqrt(pareto_Z1(i)*pareto_Z1(i)+pareto_Z2(i)*pareto_Z2(i)+pareto_Z3(i)*pareto_Z3(i));
end
% MinMax Normalisation of the distances
distances_norm = (distances-min(distances))/(max(distances)-min(distances));

figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
plot(distances_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b');
hold all;
plot(distances_norm, '-', 'lineWidth', 1.5, 'Color','b');
scatter(find(distances_norm==0),0, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'r', 'SizeData', 50);
grid on;
xlabel('Portfolio␣Index', 'FontSize', 12);
ylabel('Distance␣to␣Utopian␣Point', 'FontSize', 12);
legend('Pareto␣Optimal␣Solutions','', 'Recommended␣Portfolio', 'Location', 'best', 'FontSize', 10);
title('Distance␣of␣Pareto␣Optimal␣Portfolios␣from␣Utopian␣Point,␣Normalised␣by␣Min/Max', 'FontSize', 14);
axis square
saveas(gcf, 'plot10.png');

%% Recommended Portfolio

% Deriving the scenario closest to Utopia
```

```
recommended_portfolio_Z1=pareto_Z1(distances_norm==0);
recommended_portfolio_Z2=pareto_Z2(distances_norm==0);
recommended_portfolio_Z3=pareto_Z3(distances_norm==0);

figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
set(gcf, 'Color', 'w');
set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
scatter3(Z1_norm, Z2_norm, Z3_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', 'SizeData', 50);
hold all;
plot3([0,recommended_portfolio_Z1], [0,recommended_portfolio_Z2], [0,recommended_portfolio_Z3], '-', 'lineWidth', 1.5, 'MarkerFaceColor', 'r');
scatter3(pareto_Z1, pareto_Z2, pareto_Z3, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'r', 'SizeData', 50);
grid on;
xlabel('Inverted Performance', 'FontSize', 12);
ylabel('Total Cost', 'FontSize', 12);
zlabel('Risk', 'FontSize', 12);
legend('Solutions', 'Recommended Portfolio','Pareto Optimal Solutions', 'Location', 'best', 'FontSize', 10);
title('Recommended Technology Portfolio', 'FontSize', 14);
axis square
saveas(gcf, 'plot11.png');

%% FUNCTIONS
function random_value = gaussian_random(peak_position)
    % Set parameters
    mu = peak_position; % Mean (peak position)
    sigma = 0.1; % Standard deviation (controls spread)

    % Generate random value from Gaussian distribution
    random_value = normrnd(mu, sigma);

    % Clip value to be within [0, 1]
    random_value = max(0, min(1, random_value));
end
```

D Case Study for a Suborbital Rocket

D.1 Competitive Benchmarking

D.1.1 Figures of Merit

1. **System ID:** 1_LAUNCH_VEHICLE
2. **Roadmap ID:** 1_LAUNCH_VEHICLE
3. **Author:** Martin LEMAIRE
4. **Version:** *1.0.0*
5. **Technology List:**
   - *Launch Vehicle*
6. **Taxonomy of Technology:**

Technology Matrix	Matter	Energy	Information
Transforming			
Transporting	Launch Vehicle		
Storing			

7. **Figures of Merit Set:**
   (a) *Peak Velocity:*
   - Class: Speed
   - Unit: $[m/s]$
   - Limit: 1'715 $[m/s]$ (Hypersonic regime starts, much higher complexity)

   (b) *Altitude:*
   - Class: Range
   - Unit: $[m]$
   - Limit: 150'000 $[m]$ (50% over the Kármán line goal)

   (c) *Sustainability Metric (S):*
   - Class: Sustainability
   - Unit: [no unit], $S \in [1, 3] \cap \mathbb{N}$, Qualitative rating of the Sustainability related actions of the association:

   1 = No actions   2 = Actions being setup   3 = Concrete actions in place

   - Limit: 3 [no unit] (maximum score)

   (d) *Cost:*
   - Class: Cost
   - Unit: [CHF]
   - Limit: 0 [CHF], Minimisation (meaning best case is the lower bound) and cannot make profits

8. **Data Set Description:**
   We use the past documentation of the ERT for the Spaceport America Cup and EuRoC, which is privately stored on the Google Drive of the ERT, to collect the data of past ERT launch vehicles.

$$\overrightarrow{Velocity}_{max} = (100, 230, 250, 250, 228, 250, 460, 225, 225, 510, 900)^T$$

$$\overrightarrow{Altitude}_{AGL} = (600, 2'800, 3'300, 3'100, 2'900, 3'000, 8'500, 3'200, 3'200, 10'000, 33'000)^T$$

$$\overrightarrow{Sustainability} = (1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2)^T$$ *(Life Cycle Assessment initiated in 2023)*

$$\overrightarrow{Cost} = (16'000, 16'000, 16'500, 27'000, 25'000, 25'000, 30'000, 34'000, 34'000, 27'400, 27'400)^T$$

9. **Weight Choice Justification:**
   (a) *Peak Velocity:*
   - Weight: 0.2
   - Main Stakeholder Expectations: Maximum Performance, and High Technical Level
   - Main Related Stakeholder Clusters: Space Industry Sponsors and Technical Team

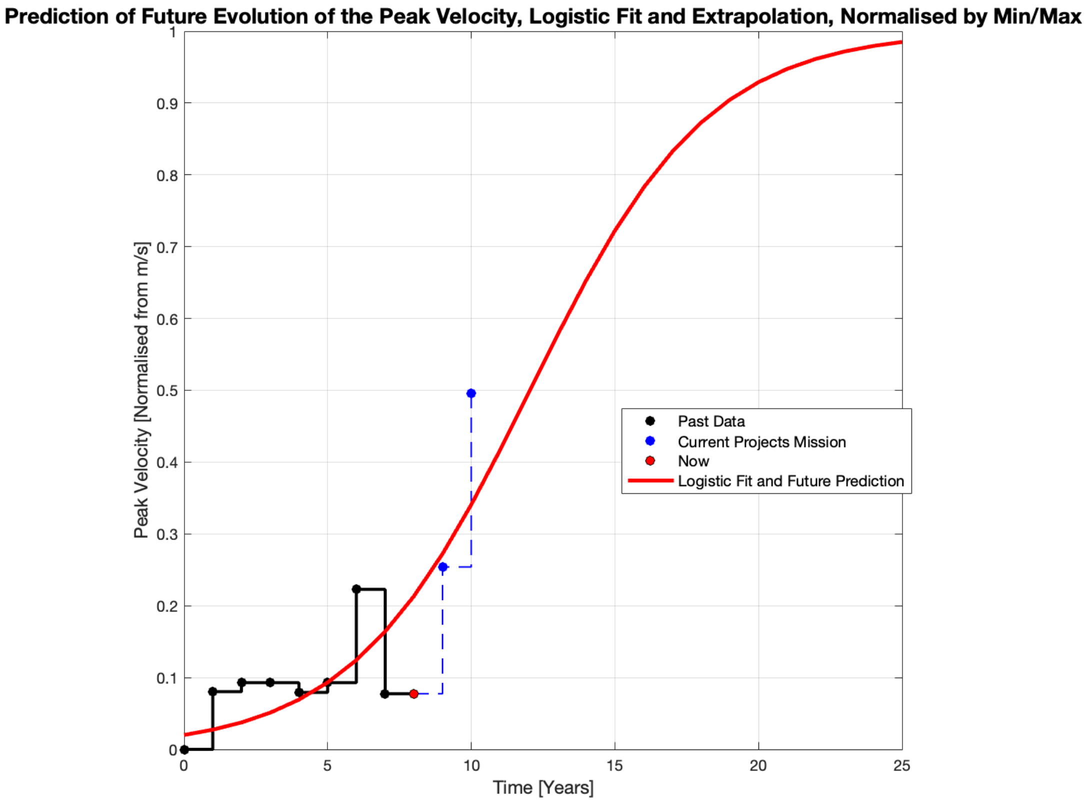
   (b) *Altitude:*
   - Weight: 0.5
   - Main Stakeholder Expectations: Maximum Performance, High Technical Level, and **Reach Space**
   - Main Related Stakeholder Cluster: Space Industry Sponsors

   (c) *Sustainability:*
   - Weight: 0.3
   - Main Stakeholder Expectations: Sustainable Image
   - Main Related Stakeholder Cluster: EPFL

   (d) *Cost:* (Cost is taken as a separate objective on purpose, as described in the Portfolio Optimisation part)
   - Weight: 1
   - Main Stakeholder Expectations: Low Cost
   - Main Related Stakeholder Cluster: Sponsors

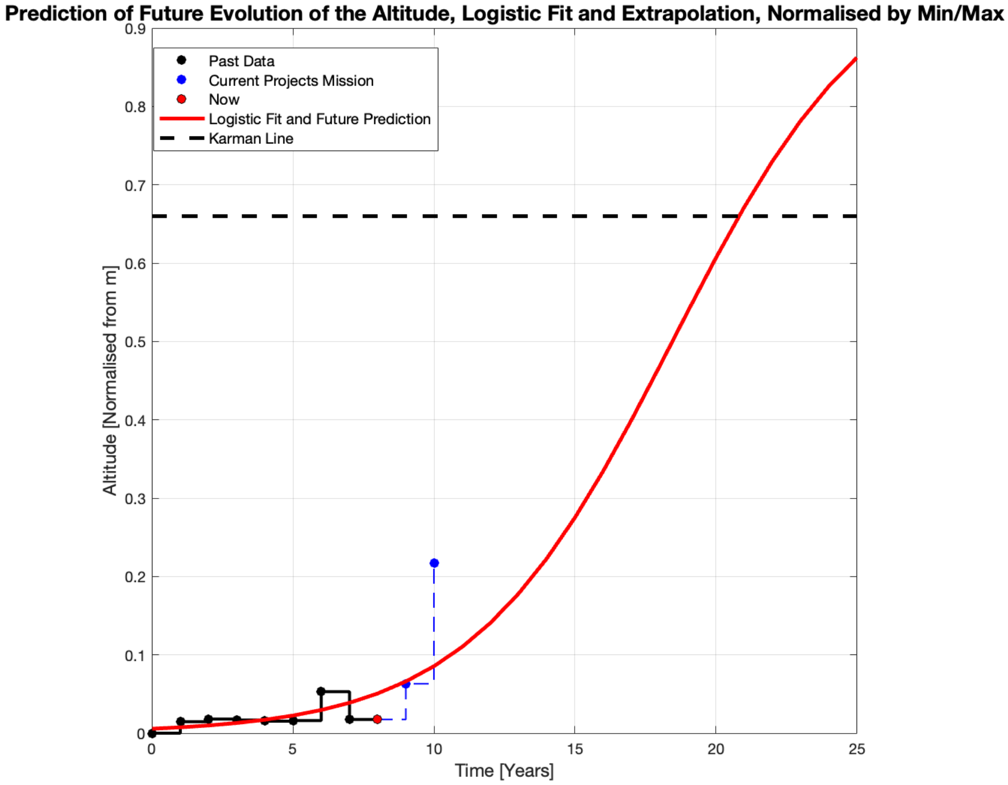
   In this part, the weights were defined on a purely qualitative basis from the stakeholder expectations.

10. **Technology Trends:**
   (a) *Peak Velocity:* Trend = 0.0294



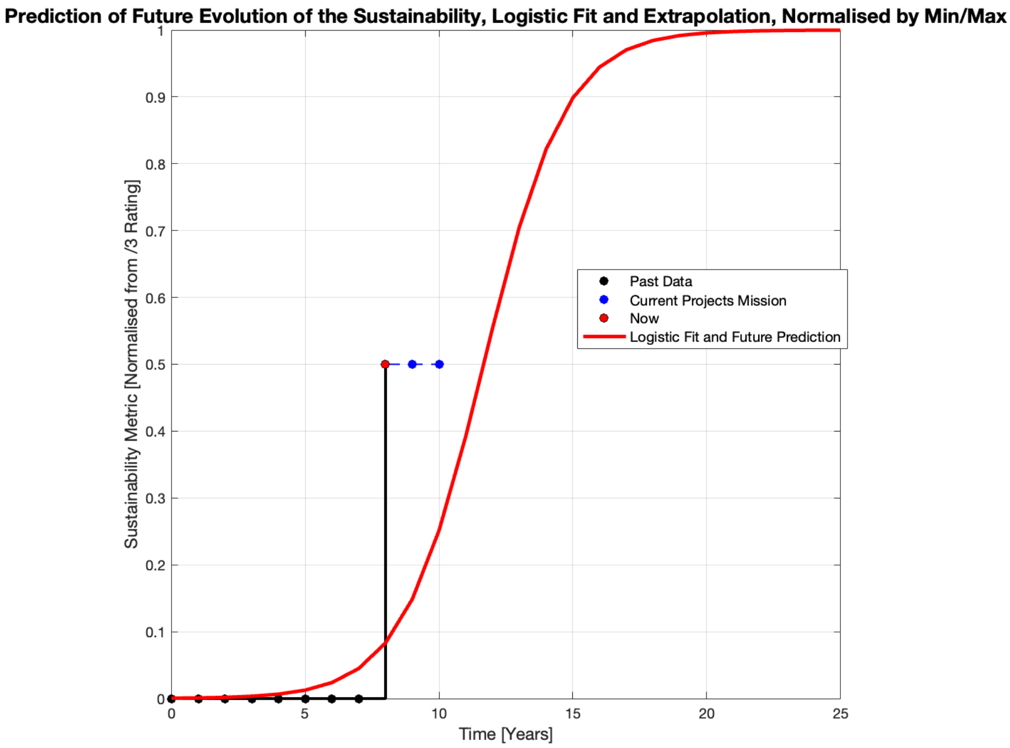
Interpretation: We can observe the decrease in the FOM at year 7. This corresponds to the rocket Nordend (3km Bi-Liquid), which followed the rocket Wildhorn (9km Solid). Indeed Wildhorn was supersonic and Nordend operated in the transonic regime. This is due to the transition from solid propulsion to bi-liquid, which wasn't ready enough to continue on the same path.

(b) *Altitude:* Trend = 0.0119



Interpretation:
The same interpretation as the previous one can be made on this graph. Moreover, we can directly say that at the current progress rate, the suborbital flight will not be possible until at least 13 years (i.e. 2037)

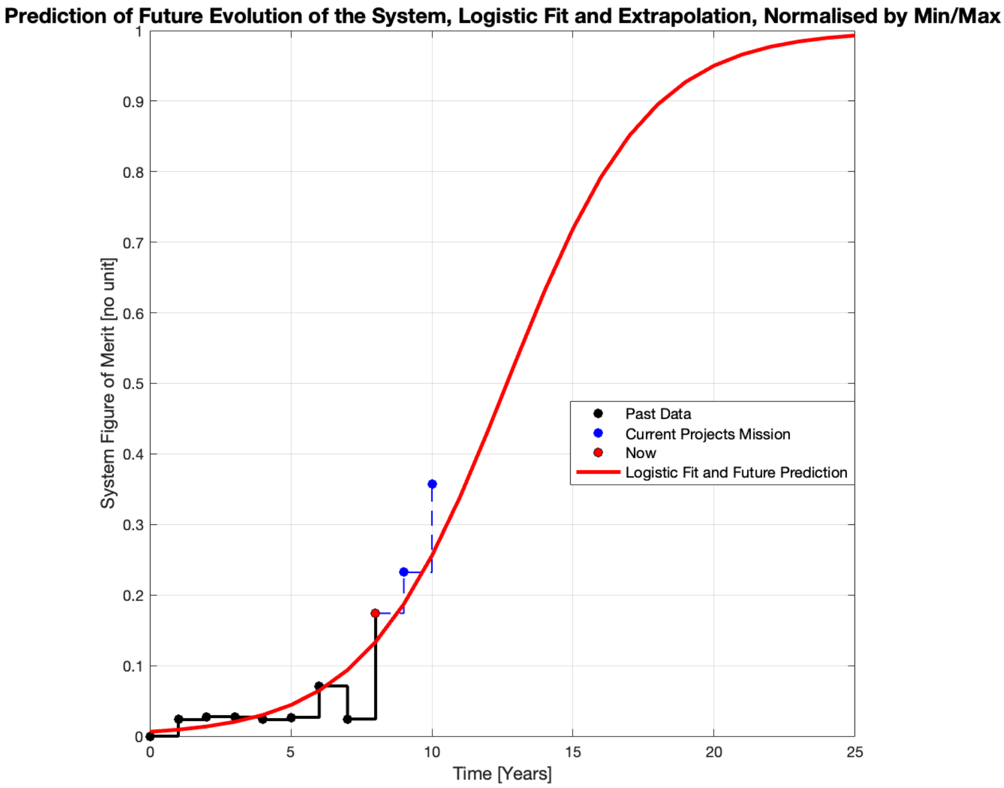
(c) *Sustainability:* Trend = 0.0409



Interpretation:

We can observe that we shouldn't expect concrete sustainability actions in the development of the rockets before the year 15 (i.e. 2031). It is also quite explicit that the current FOM does not offer accurate enough insights, and that further work should be pursued to establish a better model with an output offering more granularity and precision.

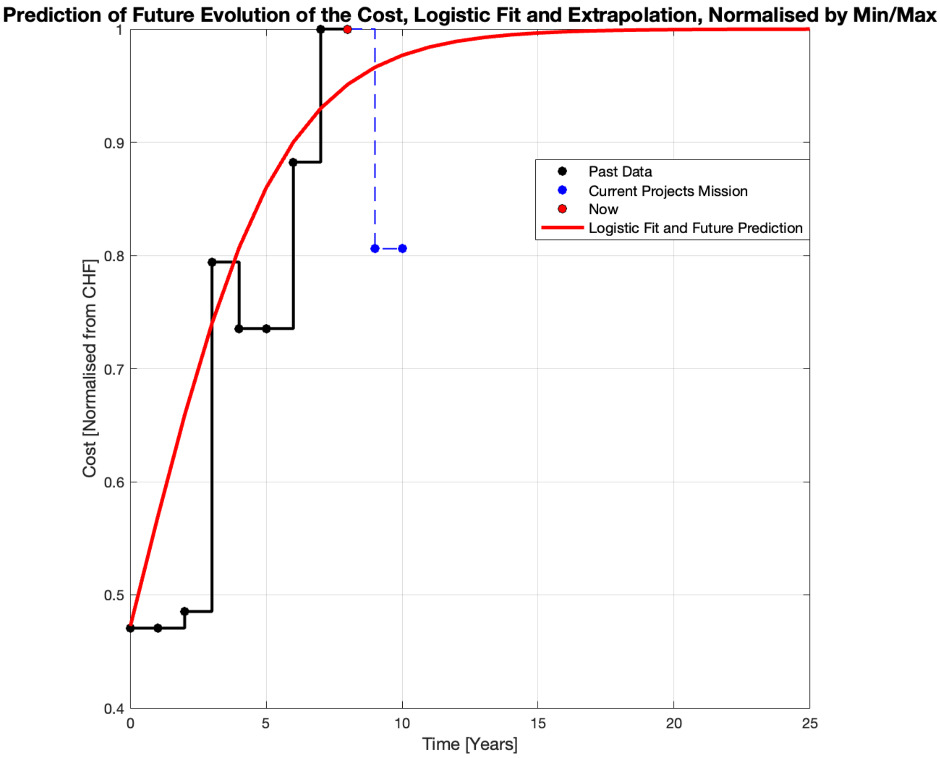
(d) *System FOM:* Trend = 0.0241



Interpretation:

The application of the weights on the FOMs give the general progress of the technology towards the stakeholders expectations. We can again see that the project Nordend, intended to implement the Bi-Liquid propulsion technology on a lower rocket class, was probably not the best solution and that the implementation of Roadmapping practice will allow to avoid this kind of disruption in the future.

(e) *Total Cost:* Trend = 0.0466



Interpretation:

We can see that the projects are currently operating at the maximum costs that the sponsoring team is able to provide. However, the current 1 in the cost FOM could change in the case of increased sponsoring capacities.

D.1.2 Similar Missions Analysis

1. **System ID:** 1_LAUNCH_VEHICLE
2. **Roadmap ID:** 1_LAUNCH_VEHICLE
3. **Author:** Martin LEMAIRE
4. **Version:** *1.0.0*
5. **Technology/System Definition:**

   The focal system, a launch vehicle, is a complex assembly of components designed to overcome Earth's gravity and reach the desired trajectory.
6. **Technology/System Boundaries:**
   - Student led organisation
   - Flights are strictly under the orbital classification (i.e. suborbital or less)
7. **High Level Specifications:**
   - Solid, hybrid, or Bi-liquid Rocket Propulsion System
8. **Data Sources:**
   - Cat Systems, EuRoC Teams Data, 2023: https://github.com/catsystems/euroc23-team-data/tree/main

   We are not able to acquire data on the sustainability ratings of the organisations and neither their financial status. We will therefore limit ourselves to the Peak Velocity and the Altitude for the comparison.

   It would have been preferable to obtain flight data on Launch Vehicle that flew at the Spaceport America Cup in higher altitude categories (10km and 30km), but these are not publicly available.
9. **FOMs Numerical Values:**

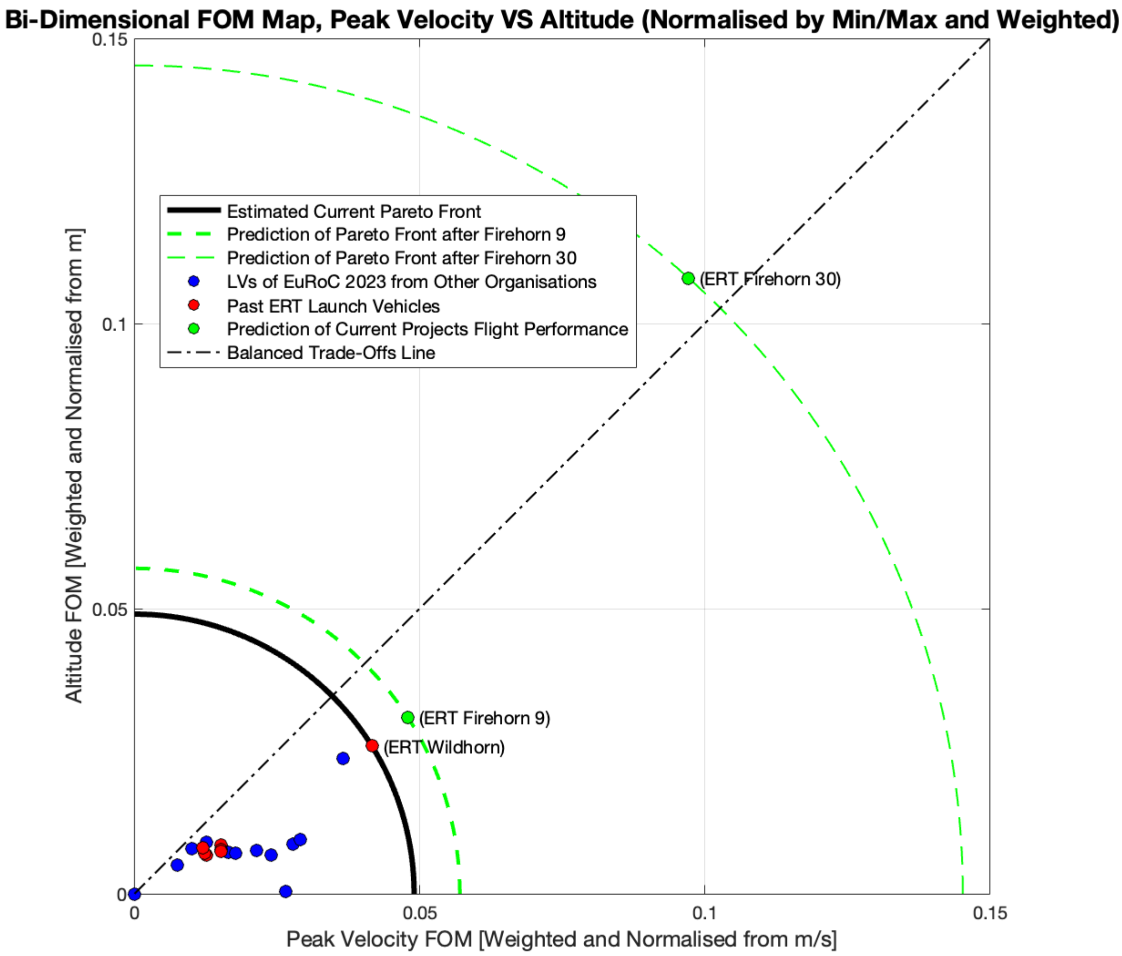
   (a) *Peak Velocity:*
   - ASTG, 230 [m/s], Solid 3km
   - AIR ESEIA, 260 [m/s], Solid 3km
   - ASAT, 190 [m/s], Solid 3km
   - AESIR, 350 [m/s], Solid 3km
   - ERT RORO, 230 [m/s], Solid 3km
   - ERT Matterhorn, 250 [m/s], Solid 3km
   - ERT Eiger, 250 [m/s], Solid 3km
   - ERT Bella Lui I, 228 [m/s], Solid 3km
   - ERT Bella Lui II, 250 [m/s], Hybrid 3km
   - ERT Wildhorn, 460 [m/s], Solid 9km
   - ERT Nordend, 225 [m/s], Bi-Liquid 3km
   - FARADAY, 270 [m/s], Solid 3km
   - BRISTOL, 210 [m/s], Solid 3km
   - NS, 340 [m/s], Solid 3km
   - POLITO, 320 [m/s], Solid 3km
   - NTNU, 130 [m/s], Solid 3km
   - PUT, 420 [m/s], Solid 9km
   - RED, 300 [m/s], Solid 3km
   - STAR, 360 [m/s], Solid 3km

   (b) *Altitude:*
   - ASTG, 3'466 [m], Solid 3km
   - AIR ESEIA, 2'918 [m], Solid 3km
   - ASAT, 2'296 [m], Solid 3km
   - AESIR, 3'385 [m], Solid 3km
   - ERT RORO, 2'800 [m], Solid 3km
   - ERT Matterhorn, 3'300 [m], Solid 3km
   - ERT Eiger, 3'100 [m], Solid 3km
   - ERT Bella Lui I, 2'900 [m], Solid 3km
   - ERT Bella Lui II, 3'000 [m], Hybrid 3km
   - ERT Wildhorn, 8'500 [m], Solid 9km
   - ERT Nordend, 3'200 [m], Bi-Liquid 3km
   - FARADAY, 2'914 [m], Solid 3km
   - BRISTOL, 3'122 [m], Solid 3km
   - NS, 870 [m], Solid 3km
   - POLITO, 2'789 [m], Solid 3km
   - NTNU, 772 [m], Solid 3km

- PUT, 7’840 [m], Solid 9km
- RED, 3’014 [m], Solid 3km
- STAR, 3’600 [m], Solid 3km

10. **Multidimensional FOM Map:**

(a) *Peak Velocity / Altitude:*



Interpretation:

The previous graph allows us to compare the past performance of the ERT to the perfomance of the rockets that flew at EuRoC 2023. We can again observe that the Wildhorn rocket sets the current highest performance rating. The representation of future rockets that haven’t flown yet but are already in development (Firehorn 9 and Firehorn 30) allows a clear understanding of the progress that these missions bring to the field of competitive student rocketry in the European landscape.

It is also important to note that the direct transition from Wildhorn to Firehorn 9 would have been the better scenario in the transition from solid propulsion to a bi-liquid technology but that in reality there was another launch vehicle between the 2, Nordend, which in this graph lies in the cluster of past ERT rockets, which are all 3km and subsonic launch vehicles.

D.1.3 Technology Readiness Assessment

1. **System ID:** 1_LAUNCH_VEHICLE
2. **Roadmap ID:** 1_LAUNCH_VEHICLE
3. **Author:** Martin LEMAIRE
4. **Version:** *1.0.0*
5. **Activity Description:**

   The goal of the mission is to operate a launch vehicle beyond the Kármán line. This implies the operation of certain subsystems in a void environment and atmosphere reentry phenomenons.
6. **Estimated Current TRL:**

   The current Launch vehicle is intended to verify multiple subsystems at a higher technological level and in conditions that are closer to the ones of a suborbital flight. We can take the example of the ground electronics that will increase in range.

   Following the ECSS milestones for defining the current TRL, we can estimate that the system is rated as a TRL 5. This implies:

   "Critical functions of the element are identified and the associated relevant environment is defined. Breadboards not full-scale are built for verifying the performance through testing in the relevant environment, subject to scaling effects."
7. **Target TRL:**

   The goal is the operation of the system for a suborbital mission. This corresponds to the TRL 8:

   "Actual system completed and 'flight qualified' through test and demonstration (Ground or Flight)"

D.1.4 MATLAB Script of the Competitive Benchmarking

```
%% competitive_benchmarking.m
%
% Author:   Martin LEMAIRE
%
% Project:  [ME-314] - Technology Roadmapping for a Suborbital Rocket with
%           Multidisciplinary Design Optimisation
%
% Date:     Spring 2024
%
% Version:  1.0.0

%% Fundamental Limits Definition

peak_velocity_fundamental_limit = 1715; % [m/s], Hypersonic Regime Start
altitude_fundamental_limit = 150000; % [m], 50% margin over the goal
sustainability_fundamental_limit = 3; % [no unit], /3 rating of sustainable development
    efforts and implementation
cost_fundamental_limit = 0; % [CHF], Cannot Make Money

%% Data Collection

peak_velocity = [100,230,250,250,228,250,460,225,225,510,900]; % [m/s]
altitude = [600,2800,3300,3100,2900,3000,8500,3200,3200,10000,33000]; % [m]
sustainability = [1,1,1,1,1,1,1,1,2,2,2]; % [no unit], qualitative rating
cost = [16000,16000,16500,27000,25000,25000,30000,34000,34000,27400,27400]; % [CHF]

%% Data Normalisation

normalised_peak_velocity = min_max_normalise(peak_velocity,peak_velocity_fundamental_limit);
normalised_altitude = min_max_normalise(altitude,altitude_fundamental_limit);
normalised_sustainability = min_max_normalise(sustainability,sustainability_fundamental_limit)
    ;
normalised_cost = -min_max_normalise(-cost,cost_fundamental_limit)+1;

```

```
disp("Normalised Peak Velocity FOM:");
disp(normalised_peak_velocity);
disp("Normalised Altitude FOM:");
disp(normalised_altitude);
disp("Normalised Sustainability FOM:");
disp(normalised_sustainability);
disp("Normalised Cost FOM:");
disp(normalised_cost);

%% System FOM computation

% Definition of the weights for each FOM
weight_peak_velocity = 0.2;
weight_altitude = 0.5;
weight_sustainability = 0.3;

% Check that the weights add up to 1
if (weight_peak_velocity+weight_altitude+weight_sustainability ~= 1)
    disp("Warning: the weights don't add up to 1")
end

% Applicationof the defined weights to the system FOM
system_fom = weight_peak_velocity.*normalised_peak_velocity+weight_altitude.*
    normalised_altitude+weight_sustainability.*normalised_sustainability;

%% Technology Trends Values

% Computation of the trends
trend_peak_velocity = technology_trend(normalised_peak_velocity);
trend_altitude = technology_trend(normalised_altitude);
trend_sustainability = technology_trend(normalised_sustainability);
trend_system_fom = technology_trend(system_fom);
trend_total_cost = technology_trend(normalised_cost);

% Displaying the numerical values
disp("Trend of the Peak Velocity FOM:");
disp(trend_peak_velocity);
disp("Trend of the Altitude FOM:");
disp(trend_altitude);
disp("Trend of the Sustainability FOM:");
disp(trend_sustainability);
disp("Trend of the System FOM:");
disp(trend_system_fom);
disp("Trend of the Cost:");
disp(trend_total_cost);

%% Future Predictions

% Definition of the number of years over which the prediction is made
expansion = 15;

% Peak Velocity FOM
[x_logistic,y_logistic]=logistic_prediction(normalised_peak_velocity,expansion);
plotter(normalised_peak_velocity, x_logistic, y_logistic, "Prediction of Future Evolution of
    the Peak Velocity, Logistic Fit and Extrapolation, Normalised by Min/Max", "Peak Velocity
    [Normalised from m/s]")

% Altitude FOM
[x_logistic,y_logistic]=logistic_prediction(normalised_altitude,expansion);
plotter(normalised_altitude, x_logistic, y_logistic, "Prediction of Future Evolution of the
    Altitude, Logistic Fit and Extrapolation, Normalised by Min/Max", "Altitude [Normalised
    from m]")

% Sustainability FOM
[x_logistic,y_logistic]=logistic_prediction(normalised_sustainability,expansion);
plotter(normalised_sustainability, x_logistic, y_logistic, "Prediction of Future Evolution of
    the Sustainability, Logistic Fit and Extrapolation, Normalised by Min/Max", "
    Sustainability Metric [Normalised from /3 Rating]")

% Sustainability FOM
[x_logistic,y_logistic]=logistic_prediction(system_fom,expansion);
```

```
plotter(system_fom, x_logistic, y_logistic, "Prediction of Future Evolution of the System,
    Logistic Fit and Extrapolation, Normalised by Min/Max", "System Figure of Merit [no unit
    ]")

% Cost FOM
[x_logistic,y_logistic]=logistic_prediction(normalised_cost,expansion);
plotter(normalised_cost, x_logistic, y_logistic, "Prediction of Future Evolution of the Cost,
    Logistic Fit and Extrapolation, Normalised by Min/Max", "Cost [Normalised from CHF]")

%% Similar Missions Analysis

% Data Collection
peak_velocity = [230,260,190,350,230,250,250,228,250,460,225,270,210,340,320,130,420,300,360];
altitude =
    [3466,2918,2296,3385,2800,3300,3100,2900,3000,8500,3200,2914,3122,870,2789,772,7840,3014,3600];

% Index of the ERT
ERT_indices = 5:11;
WH_index = 10; % Wildhorn, best performance among all the rockets of the ERT

% Adding the representation of the current projects' flight performance predictions
Firehorn_velocity = weight_peak_velocity*([510,900]-min(peak_velocity))/(
    peak_velocity_fundamental_limit-min(peak_velocity));
Firehorn_altitude = weight_altitude*([10000,33000]-min(altitude))/(altitude_fundamental_limit-
    min(altitude));

% Normalisation of the data
normalised_peak_velocity = min_max_normalise(peak_velocity,peak_velocity_fundamental_limit);
normalised_altitude = min_max_normalise(altitude,altitude_fundamental_limit);

% Display Numerical Results
disp("Normalised Peak Velocity FOM:");
disp(normalised_peak_velocity);
disp("Normalised Altitude FOM:");
disp(normalised_altitude);

disp("Application of the weight for the Peak Velocity FOM:")
weighted_normalised_peak_velocity = normalised_peak_velocity*weight_peak_velocity;
disp(weighted_normalised_peak_velocity)
disp("Application of the weight for the Altitude FOM:")
weighted_normalised_altitude = normalised_altitude*weight_altitude;
disp(weighted_normalised_altitude)

% Plotting the FOMs of the competitors and the ERT's past LVs for
% visualisation of the differences
SMA_plot(weighted_normalised_peak_velocity,weighted_normalised_altitude,ERT_indices,WH_index,
    Firehorn_velocity,Firehorn_altitude)

%% Similar Missions Analysis plot function

function SMA_plot(FOMI_norm,FOMJ_norm,ERT_indices,WH_index,Firehorn_velocity,Firehorn_altitude
    )

    distances = sqrt(FOMI_norm.^2+FOMJ_norm.^2);
    distance = max(distances);
    distance_2 = sqrt(Firehorn_velocity(1)^2+Firehorn_altitude(1)^2);
    distance_3 = sqrt(Firehorn_velocity(2)^2+Firehorn_altitude(2)^2);
    theta = linspace(0,pi/2,100);
    x = distance*cos(theta);
    y = distance*sin(theta);
    x_2 = distance_2*cos(theta);
    y_2 = distance_2*sin(theta);
    x_3 = distance_3*cos(theta);
    y_3 = distance_3*sin(theta);

    equilibrium_x = linspace(0,0.15,100);

    figure('Units', 'inches', 'Position', [0, 0, 10, 8]);
    set(gca, 'FontName', 'Helvetica', 'FontSize', 12);
    set(gcf, 'Color', 'w');
    set(gca, 'Color', 'w', 'LineWidth', 1.5, 'TickLength', [0.02 0.02]);
```

```
    plot(x,y, '-', 'lineWidth', 3, 'Color', 'black');
    hold all;
    plot(x_2,y_2, '--', 'lineWidth', 2, 'Color', 'green');
    plot(x_3,y_3, '--', 'lineWidth', 1, 'Color', 'green');
    scatter(FOMI_norm, FOMJ_norm, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'b', 'SizeData', 50);
    scatter(FOMI_norm(ERT_indices), FOMJ_norm(ERT_indices), 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'r', 'SizeData', 50);
    scatter(Firehorn_velocity,Firehorn_altitude, 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'SizeData', 50);
    plot(equilibrium_x,equilibrium_x, '-.', 'lineWidth', 1, 'Color', 'k');
    text(FOMI_norm(WH_index)+0.002 ,FOMJ_norm(WH_index),"(ERT Wildhorn)", 'FontSize', 10)
    text(Firehorn_velocity(1)+0.002 ,Firehorn_altitude(1),"(ERT Firehorn 9)", 'FontSize', 10)
    text(Firehorn_velocity(2)+0.002 ,Firehorn_altitude(2),"(ERT Firehorn 30)", 'FontSize', 10)
    grid on;
    xlabel('Peak␣Velocity␣FOM␣[Weighted␣and␣Normalised␣from␣m/s]', 'FontSize', 12);
    ylabel('Altitude␣FOM␣[Weighted␣and␣Normalised␣from␣m]', 'FontSize', 12);
    legend('Estimated␣Current␣Pareto␣Front','Prediction␣of␣Pareto␣Front␣after␣Firehorn␣9','Prediction␣of␣Pareto␣Front␣after␣Firehorn␣30', 'LVs␣of␣EuRoC␣2023␣from␣Other␣Organisations', 'Past␣ERT␣Launch␣Vehicles','Prediction␣of␣Current␣Projects␣Flight␣Performance', 'Balanced␣Trade-Offs␣Line', 'Location', 'best', 'FontSize', 10);
    title('Bi-Dimensional␣FOM␣Map,␣Peak␣Velocity␣VS␣Altitude␣(Normalised␣by␣Min/Max␣and␣Weighted)', 'FontSize', 14);
    axis square
end
```

D.2 Scenarios Generation

To generate meaningful scenarios, we will use the predictions from competitive benchmarking to define parameters for the stochastic model, ensuring a close-to-reality solution set.

We will focus on the graphs of the system FOM and cost, which are crucial for optimization. Since the primary goal of the roadmap is to establish strategic planning until 2030, we will take the predicted values for 2030 and use a confidence interval to determine five different portfolios. For the cost, points 1 and 2 are not plotted on the figure below but are determined as the symmetries of points 5 and 4 around 1.

From the figure below, we can derive the following parameters for the stochastic model used in the MATLAB multiobjective optimization script:

Minimum value of the system FOM:

$$min(E(FOM)) = 0.65$$

Initial FOM value (Firehorn 30 used in this case):

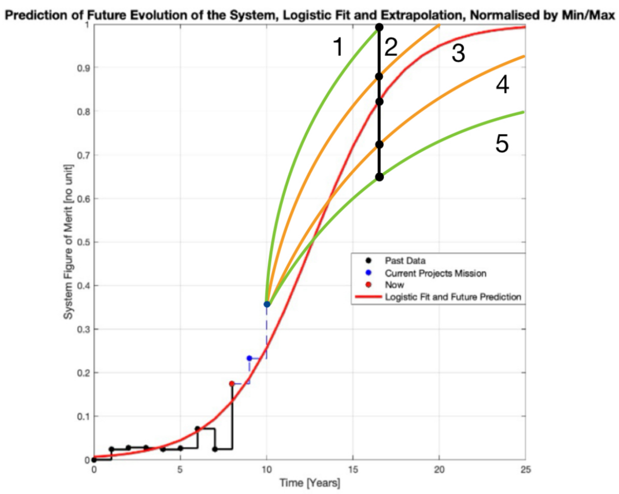
$$FOM = 0.35$$

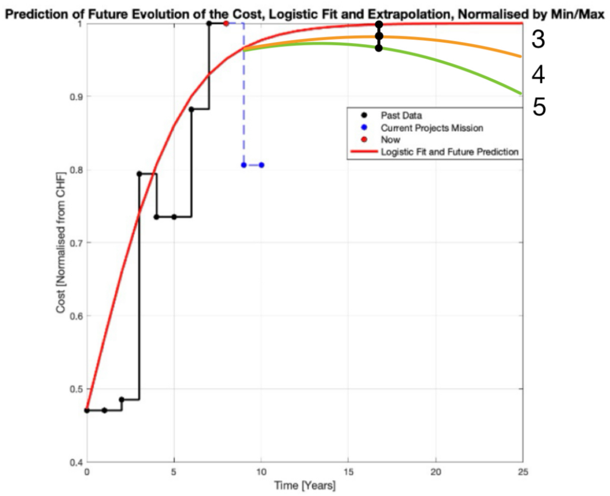
Minimum cost:

$$min(E(C)) = 0.95$$

Step from the minimum cost to the maximum cost:

$$\Delta E(C) = 0.1$$

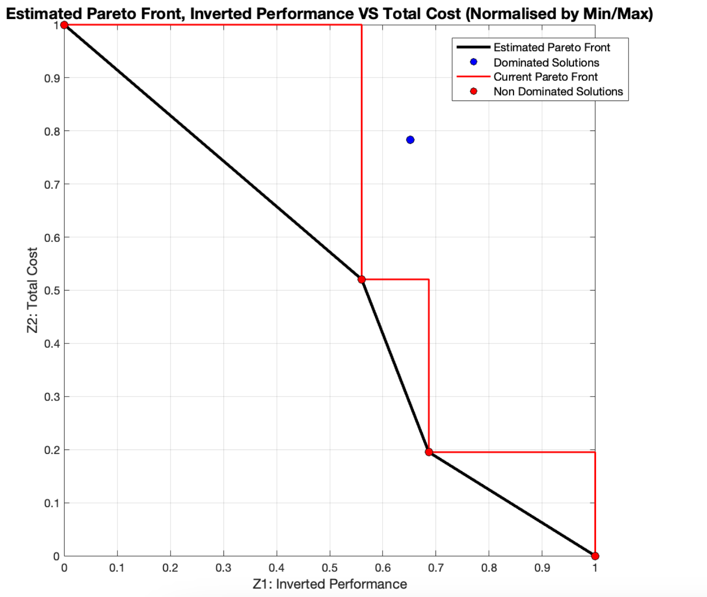




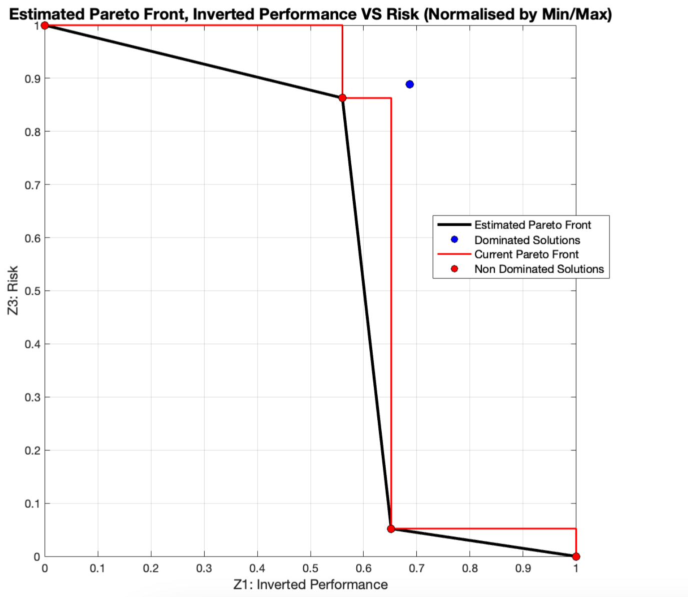
D.3 Portfolio Optimisation

D.3.1 Pareto Optimal Set of Technology Investment Portfolios Documentation Template

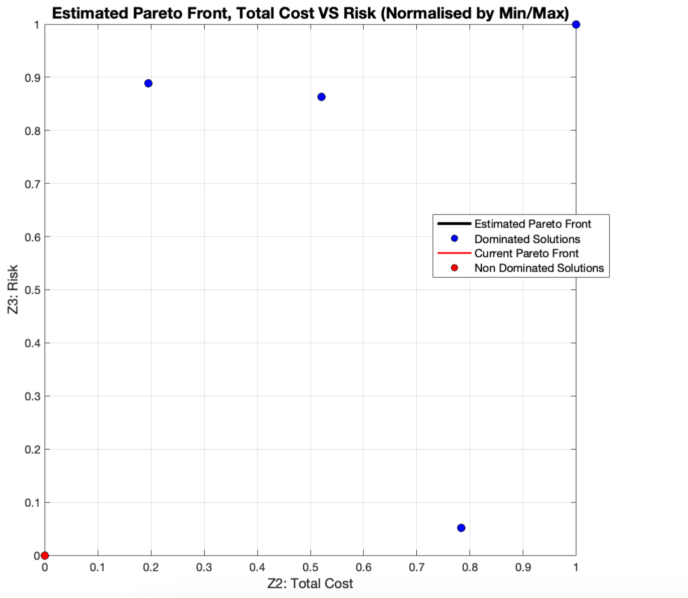
1. **System ID:** 1_LAUNCH_VEHICLE
2. **Roadmap ID:** 1_LAUNCH_VEHICLE
3. **Author:** Martin LEMAIRE
4. **Version:** *1.0.0*
5. **Pareto Front Plots:**
   - Z_1 VS Z_2 Solution Space



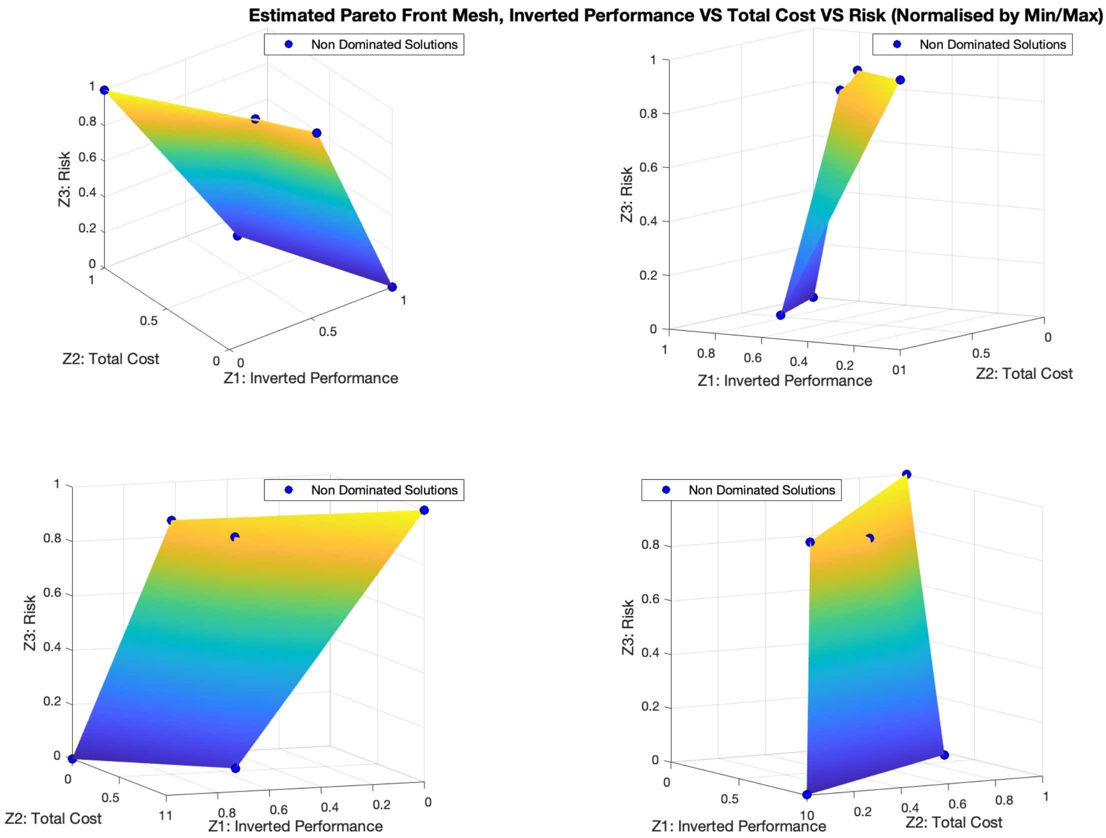
   - Z_1 VS Z_3 Solution Space



- Z_2 VS Z_3 Solution Space

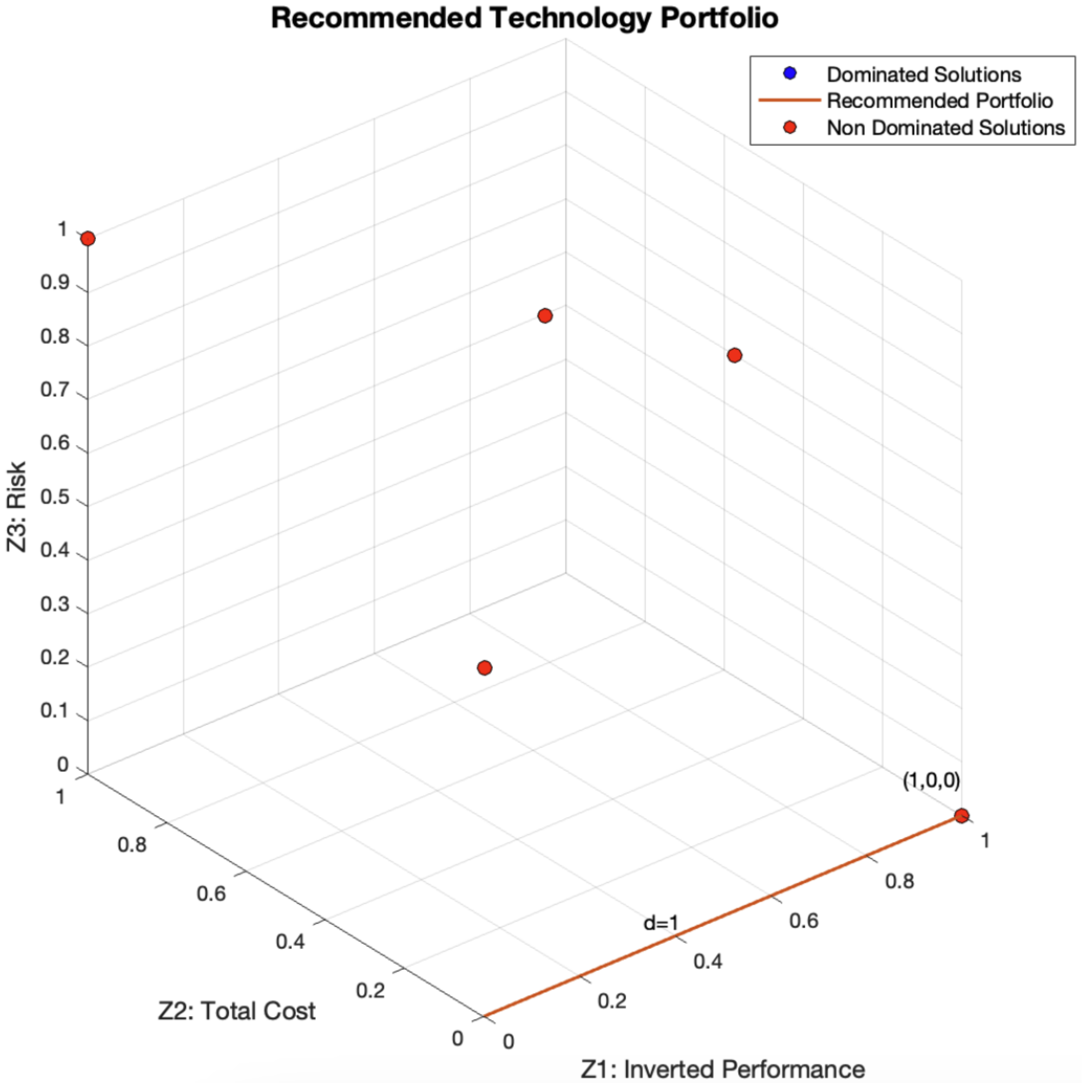


- Z_1 VS Z_2 VS Z_3 Solution Space



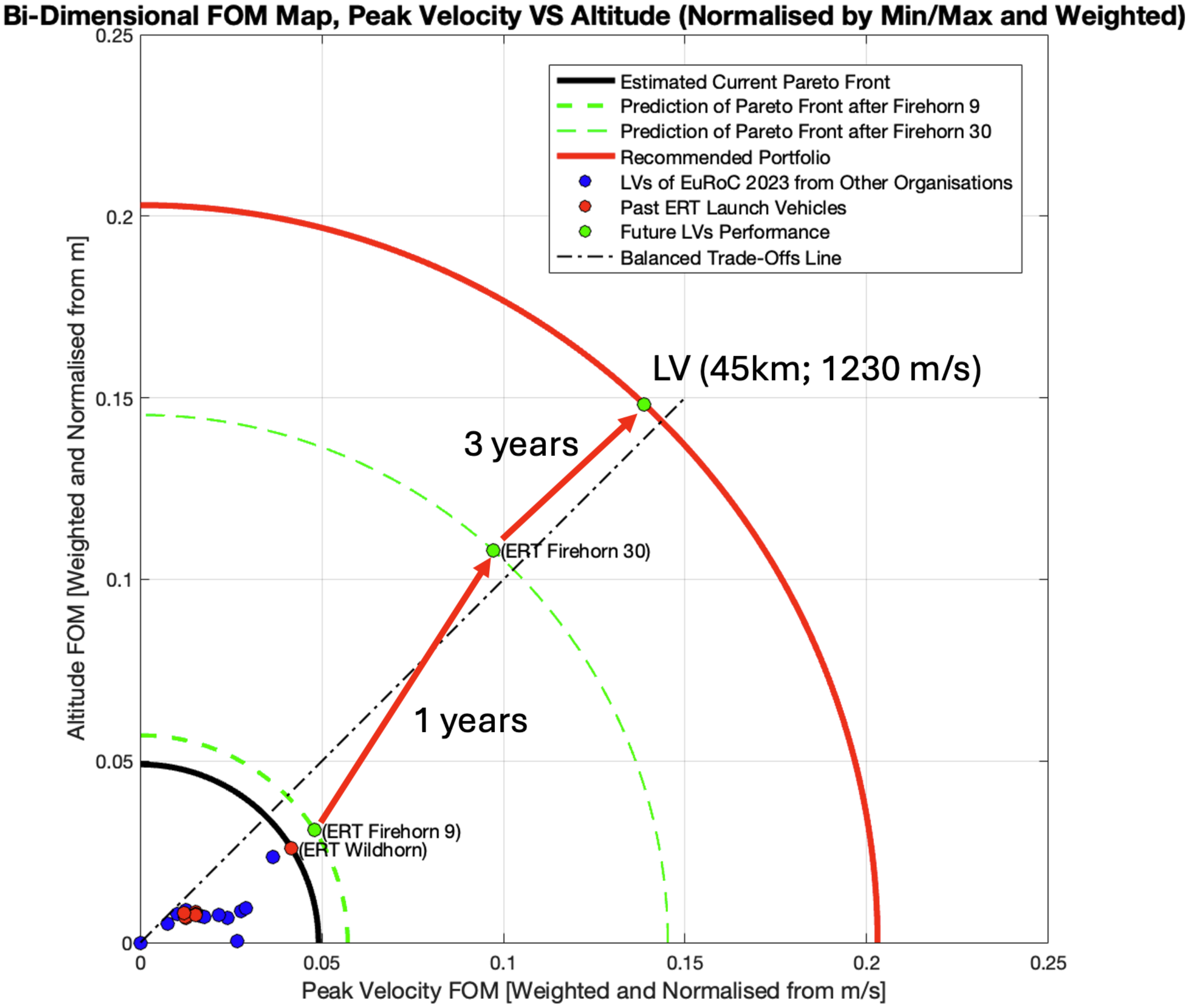
D.3.2 Recommended Technology Portfolio Documentation Template

1. **System ID:** 1_LAUNCH_VEHICLE
2. **Roadmap ID:** 1_LAUNCH_VEHICLE
3. **Author:** Martin LEMAIRE
4. **Version:** *1.0.0*
5. **Multidimensional Plot of the Recommended Portfolio:**



6. **Vector Charts of the Recommended Portfolio:**

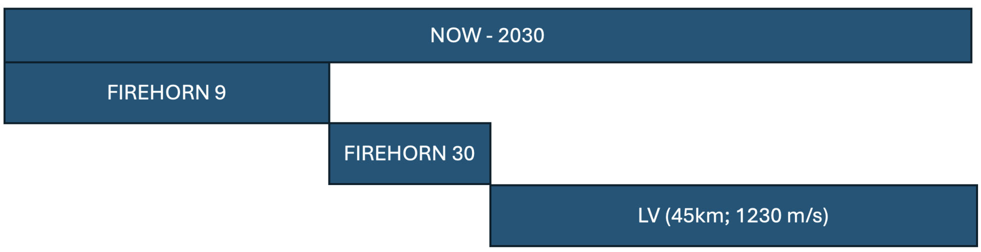
   From the multidimensional plot, it's evident that the recommended portfolio exhibits the lowest cost and risk, albeit with comparatively lower performance. It's important to note that being ranked lowest in performance among the solutions doesn't imply poor performance. Rather, it signifies that, while still aligning with long-term goals and current evolutionary trends, there is a lower performance comparatively to other, riskier, solutions.



As it appears, the recommended portfolio gives a balanced solution, that is close to the balanced trade-offs line.

7. **Valuation of the Recommended Portfolio:** According to the cost vector, the cost is announced to be about 30'000 [CHF] per year, which is lower than the maximum past budget.

8. **Gantt Chart of the Recommended Portfolio:**



To sum up, it is clear that, at the current rate of progress with a bi-liquid propulsion system, expecting a suborbital flight before 2030 is overly optimistic. The recommended portfolio, which extends to 2030, achieves an altitude of 45km. This is already a significant accomplishment, especially considering that this portfolio represents the least risky option.

D.3.3 Human Resources Needs Expectations Documentation Template

1. **System ID:** 1_LAUNCH_VEHICLE
2. **Roadmap ID:** 1_LAUNCH_VEHICLE
3. **Author:** Martin LEMAIRE
4. **Version:** *1.0.0*
5. **Estimated Workforce Requirements:** At this stage of the roadmapping process, there are insufficient details about the technologies involved at lower levels. Consequently, it is challenging to provide a precise estimation. However, considering the less risky recommended portfolio, this issue is likely less critical in this instance. This means that we can probably focus on improving other areas like pedagogy.

D.4 Roadmap Maturity Scoring Documentation Template

1. **System ID:** 1_LAUNCH_VEHICLE
2. **Roadmap ID:** 1_LAUNCH_VEHICLE
3. **Author:** Martin LEMAIRE
4. **Version:** *1.0.0*
5. **Maturity of the Competitive Benchmarking:**
   Clear definition of FOMs and setting of targets: LVL 3
6. **Maturity of the Models and Scenarios:**
   Uneven format, quality, and depth of roadmaps: LVL 1
7. **Maturity of the Portfolio Optimisation:**
   Not used at all for decision-making, just for information: LVL 1