Thesis report - AE5810

TU Delft

Impact of Valispace on space system design project performance

F. Weijand

TU Delft

Thesis report - AE5810

TU Delft

by

F. Weijand

Student Name	Student Number
Floris Weijand	4478762

Supervisor: B. Zandbergen
Place: TU Delft

TU Delft

Preface

Ever since doing the bachelor thesis, the Design Synthesis Exercise (DSE), I have noticed how difficult it is to maintain communication, exchange technical details and make design choices effectively during a system design process. It was suggested to me that Valispace was readily used by the Delfi program at the TU Delft, as well as by another team performing the DSE.

The topic of this thesis is to determine the added value of using Valispace in space system design projects. This is in the interest of young engineers willing to find a better platform for such projects rather than the typical combination of Excel sheets, Python scripts, and some communication application. It is also in my interest to learn how such a tool can add value, and what can be learned from doing a project like the DSE with and without Valispace.

I would like to thank my supervisor Ir. B.T.C. Zandbergen for his continuous guidance and support, as well as the TU Delft for the providing the resources needed for this thesis, and the learning opportunity of writing this thesis. I would also like to thank my partner Chris and brother Ruben for providing valuable feedback, and my friends and family for keeping me focused and motivated throughout this thesis, despite all extra effort and time it took to write.

F. Weijand
Delft, May 2025

Executive summary

Efficient, consistent and accurate space system design is one of the key challenges imposed by the rising number of commercialized space activities. There is a need for a standardized space system design platform for improved project performance. One such platform could be Valispace, but its potential benefits require additional research.

In this thesis, three research questions pertaining to the usefulness of Valispace for space system design projects are explored: *(1) In what ways does the integration of Valispace into the small satellite design process influence project performance?* Project performance is assessed here by the time efficiency, design accuracy and risk mitigation of the design process. *(2) To what extent can a satellite designed using Valispace be effectively verified?* This analysis considers both traditional verification methods and potential new approaches. *(3) How can Valispace be used to identify potential improvements to the project 'Small Satellites for Gravitational Waves Observation'?* This analysis assesses the potential of Valispace to improve space system design projects by examining an existing project.

To answer these questions, a before-and-after case study analyses was done. In this case study the Design Synthesis Exercise (DSE) 'Small Satellites for Gravitational Waves Observation' was considered before Valispace implementation and after. This project was only done for space system design pre-phase A, limiting the research to this phase.

First, the project was recreated in Valispace. Then, a verification procedure was done to check the system outputs were identical to those of the DSE, and the inputs were filled in similarly for the Firesat example in Space Mission Analysis and Design (SMAD) to further verify the design. Finally, experimentation using Valispace was done with some of the features of Valispace, such as components, margins, sensitivity analyses, and system analyses.

For the first research question, the project performance was investigated per step in the design process of the DSE project, including the mission design step, the system design step, the verification step and the sensitivity analysis step.

It was determined that during the mission design phase Valispace has a positive effect on the design accuracy and risk mitigation, while reducing time efficiency slightly due to the introduction of a requirements tree structure. The requirement tree structure proved especially useful for tracking the requirement status and identifying requirements at lower levels during later stages of the design process.

During the system design time efficiency was lower due to the necessity to structure the component tree and connect system properties, as well as connecting the requirements to these properties, but the design accuracy and risk mitigation are increased because of the inter-connectivity and flexibility of the design tree.

The verification phase's project performance metrics depend on the verification method selected. This report concludes that for unit testing the preferred method is assignment of a test engineer who subscribes to all identified units, and for system testing the existing system should by copied and the independent copy should be tested thoroughly. The unit testing approach reduces time efficiency while increasing design accuracy and mitigating risks, while the system testing approach greatly increases time efficiency and mitigates risk.

Sensitivity analyses project performance improved greatly, increasing time efficiency and design accuracy, while mitigating the risk of failing to account for snowballing effects on design budgets. However, a new risk of failing to undo changes for sensitivity analyses is introduced and must be taken into account.

For the second research question, unit testing and system testing were considered for verification. Three possible unit testing methods were identified in Valispace: copying units to an independent sys-

tem tree, scripting externally using the Python API and assigning a dedicated test engineer who tracks all changes made to the units. The third method is preferred in this report. It reduces time efficiency compared to traditional methods but increases design accuracy and mitigates the risk of using invalid units greatly.

For system testing, changes in value to determine how these changes propagate throughout the system can be done, but this introduces the risks of changing the system history and failing to undo changes made to test the system. Instead, it is suggested to copy the system for independent testing, such that the original system remains intact and thorough system testing can be done. The Firesat example from SMAD is copied this way to verify the DSE design recreation in this report.

For the third research question, it was found that Valispace allowed for improvements in the DSE project. The first improvement concerns the design concept trade-off, for which the criteria could be quantified more effectively when implemented in Valispace, allowing the concept selection to be verified at a later stage. The flexibility of Valispace can also allow suspension of concept design trade-offs until the designs are more certain. The sensitivity analyses could be done more efficiently and account for snowballing effects on design budgets. Lastly, an assumption made without quantified substantiation during the DSE could be verified quickly using Valispace.

While answering the research questions, it was found Valispace has some strengths and some weaknesses. These are listed below:

- Strength 1: Automatic change propagation. The direct connections between properties of the system components reduces time spent manually propagating changes and reduces the risk of failing to account for changes.
- Strength 2: Analysis generation allows budgets, values and datasets to be documented and communicated effectively.
- Strength 3: Design flexibility allows users to quickly analyse the impact of changes to better optimize the design. Furthermore, it allows assumptions made and concept trade-offs done beforehand to be verified or changed at a later stage in the design process.
- Strength 4: Requirement verification allows users to quickly identify which design choices still fulfill the requirements, and which do not.
- Weakness 1: Delayed change propagation, occasionally requiring a browsers refresh which poses a risk to documentation.
- Weakness 2: Limits to sensitivity analyses, where only one-to-one design property dependencies can be analyzed, while more complex dependencies require manual approaches.
- Weakness 3: Flaws and bugs. The margins feature contains an update bug which forms a risk, and is flawed in its implementation. Dataset interpolation and extrapolation is limited to linear and step-wise options, and more complex options such as polynomial interpolation requires manual implementation of the relation.

Overall, it is concluded that Valispace increases design accuracy and mitigates risks because of these strengths, and the weaknesses can be accounted for. While time efficiency is increased for some processes and reduced for some other processes, it is expected to increase overall due to the great reduction in design conflict risks and increase in sensitivity analysis efficiency.

Future work could focus on assessing the impact of using the ValiAssistant, which can automatically generate requirements. Furthermore, the project performance metrics could be quantified to analyse in-depth how Valispace will affect the design process. A generalized space system design template could be made to greatly reduce time spent creating the components in Valispace. Finally, the research can be extended to design phases beyond pre-phase A.

Contents

Preface i

Executive summary ii

Nomenclature vi

List of Figures ix

List of Tables x

1 Introduction 1

2 Background 3
2.1 Current state of space mission design process . . . 3
2.2 CubeSats . . . 4
2.3 Satellite design process during the DSE . . . 4
2.3.1 Requirement generation . . . 4
2.3.2 Concept generation . . . 6
2.3.3 Concept trade-off . . . 8
2.3.4 Payload design . . . 9
2.3.5 Subsystem design . . . 10
2.3.6 Subsystem integration and system topology . . . 12
2.3.7 Verification and validation . . . 12
2.3.8 Sensitivity analysis . . . 12
2.3.9 Documentation . . . 13
2.3.10 Design description . . . 13
2.4 Project performance metrics . . . 13
2.5 Valispace for space mission design . . . 14
2.6 Interview with Marte Medina León . . . 15

3 DSE recreation methodology 18
3.1 Before-and-after analysis case study set-up . . . 18
3.2 System design tree . . . 18
3.3 System design inputs . . . 19
3.4 Spacecraft characteristics estimation . . . 20
3.5 Astrodynamic characteristics . . . 21
3.5.1 Delta-V budget . . . 21
3.5.2 Geometric properties . . . 23
3.6 Subsystems . . . 24
3.6.1 Attitude Determination and Control System . . . 24
3.6.2 Command and Data Handling . . . 26
3.6.3 Power . . . 26
3.6.4 Propulsion . . . 26
3.6.5 Structure . . . 27
3.6.6 Thermal . . . 27
3.6.7 Telemetry, Tracking & Command . . . 28
3.7 COTS component selection . . . 29

4 Satellite design from Valispace implementation **30**
4.1 Tool description and limitations . . . 30
4.2 Design summary . . . 31
4.3 Verification process . . . 32
4.3.1 The delta-V budget . . . 33
4.3.2 Astrodynamic characteristics . . . 33
4.3.3 Subsystem design . . . 33

5 Experimentation methodology **35**
5.1 Requirements . . . 35
5.2 Margins . . . 36
5.3 Sensitivity analysis . . . 36
5.3.1 Analysis set-up . . . 36
5.3.2 Selected sensitivity analysis cases . . . 37
5.3.3 Valispace utilization . . . 37
5.4 Concept trade-off and payload design . . . 38
5.5 Necessity of taxi vehicle in DSE design . . . 40

6 Results **42**
6.1 Requirements . . . 42
6.2 Margins . . . 44
6.3 Sensitivity analysis . . . 46
6.3.1 Payload mass increase . . . 46
6.3.2 Different injection inclination . . . 46
6.3.3 Different thruster selection . . . 48
6.3.4 Structure subsystem mass increase . . . 49
6.3.5 Errors while performing sensitivity analyses . . . 49
6.4 Concept trade-off and Valispace . . . 50
6.4.1 Redesigns for other design concepts . . . 50
6.4.2 Concept trade-off procedure validity and necessity . . . 50
6.5 Taxi vehicle necessity in DSE . . . 51
6.6 Bugs and flaws in the software . . . 52

7 Discussion **54**
7.1 Impact of Valispace on project performance . . . 54
7.1.1 Mission design . . . 54
7.1.2 System design . . . 54
7.1.3 Verification . . . 55
7.1.4 Sensitivity analysis . . . 56
7.2 Strengths and weaknesses of the tool . . . 56
7.3 Verification of projects in Valispace . . . 57
7.3.1 Unit testing . . . 57
7.3.2 System testing . . . 58

8 Conclusion and recommendations **59**
8.1 Conclusion . . . 59
8.2 Recommendations . . . 60

References **63**

A Verification of Valispace implementation using DSE **64**

B Questions asked during the interview with Marte Medina León **67**

C Log of sensitivity analysis bug in Valispace **69**
C.1 Reproduction . . . 69

D Payload design **70**

Nomenclature

Abbreviations

Abbreviation	Definition
API	Application Programming Interface
ADCS	Attitude Determination & Control Subsystem
CDH	Command & Data Subsystem
COTS	Commercial Off-The-Shelf
DoD	Depth of Discharge
DOT	Design Option Tree
DSE	Design Synthesis Exercise
EPS	Electrical Power Subsystem
ESA	European Space Agency
LICCA	Laser Interferometer Cubesat Constellation Antenna
MoS	Margin of Safety
NASA	National Aeronautics and Space Administration
SMAD	Space Mission Analysis & Design
SNR	Signal-to-Noise Ratio

Symbols

Symbol	Definition	Unit (SI)
a	Acceleration	[m/s²]
a	Semi-major axis	[m]
A	Area	[m²]
B	Magnetic field strength	[A/m]
C_b	Ballistic coefficient	[-]
C_d	Drag coefficient	[-]
d	Distance	[m]
d	Degradation	[year^{-1}]
D	Diameter	[m]
e	Eccentricity	[-]
E	Young's modulus	[N/m²]
F	Force	[N]
G	Energy flux	[W/m²]
G	Gain	[-]
I	Moment of inertia	[kgm²]

Symbol	Definition	Unit (SI)
I_{sp}	Specific impulse	[s]
i	Inclination	[rad]
k	Spring constant	[N/m]
l	Linear dimension	[m]
L	Loss	[-]
m	Mass	[kg]
$\dot{m}$	Mass flow	[kg/s]
m	Magnetic moment	[Nm-Tesla]
n	Distance ratio	[-]
N	Storage size	[bit]
P	Power	[W]
P	Load	[N]
p	Pressure	[N/m²]
Q	Heat flow	[W/s]
R	Rate	[bit/s]
r	Radius	[m]
r	Ratio	[-]
T	Period	[s]
T	Temperature	[K]
t	Time	[s]
t	Thickness	[m]
V	Volume	[m³] *or* [U]
V	Velocity	[m/s]
w	Width	[m]
α	Angular acceleration	[rad/s²]
α	Absorptivity	[-]
δ	Angular radius	[rad]
ΔV	Delta-V	[m/s]
ϵ	Emissivity	[-]
η	Efficiency	[-]
η	Minimum elevation angle	[-]
θ_{min}	Minimum elevation angle	[rad]
Θ_{Earth}	Earth central angle	[rad]
λ	Wavelength	[m]
λ_{max}	Maximum Earth central angle	[rad]
ρ	Density	[kg/m³]
ρ	Angular radius of Earth	[rad]
σ	Stress	[N/m²]
τ	Torque	[Nm]
ω	Angular velocity	[rad/s]
ω	Natural frequency	[Hz]

Constant	Definition	Value	Unit (SI)
c	Speed of light	$2.998 \cdot 10^8$	[m/s]
g	Gravitational acceleration at Earth's surface	9.807	[m/s²]
G_{SC}	Solar constant energy flux	$1.373 \cdot 10^3$	[W/m²]
k_b	Boltzmann constant	$1.381 \cdot 10^{-23}$	[m²kgs^{-2}K^{-1}]
m_E	Magnetic dipole moment of Earth	$8.0 \cdot 10^{22}$	[J/T]
R_{Earth}	Earth radius	$6.378 \cdot 10^6$	[m]
μ	Gravitational parameter of Earth	$3.986 \cdot 10^5$	[km³/s²]
σ	Stefan-Boltzmann constant	$5.670 \cdot 10^{-8}$	[Wm^{-2}K^{-4}]

List of Figures

2.1 The space mission project life cycle from NASA. Adapted from https://ntrs.nasa.gov/ . . 3
2.2 A flowdown of the design option tree used in the DSE. For the constellation and subsystem, a more detailed flowdown is shown. 7
2.3 The DOTs used for the constellation concept selection during the DSE. Concepts marked red are considered infeasible. 7
2.4 The DSE concepts generated, which were to be traded off in a later phase. These concepts are intended for forming a laser interferometer for measuring gravitational waves. All laser path lengths are identical. 7

3.1 The system design tree as created in the tool. The tree levels are indicated using colors, with the arrows forming the flow of change propagation. 19
3.2 A graph visualizing the first-order system characteristics estimation process, for utilization in Valispace. 20
3.3 The system characteristics estimation inputs, as structured in the system design module in Valispace 21
3.4 An illustration of the Hohmann-transfer, which is the maneuver from any orbit to another orbit with the lowest delta-V requirement. 22

4.1 The mass budget of the designed satellite, from the analyses module in Valispace. . . . 31
4.2 The power consumption budget of the designed satellite, from the analyses module in Valispace. 31
4.3 A sketch of the satellite topology. Adapted from the DSE final report[7] 31

5.1 The sensitivity analysis flow for determining the impact of inputs on outputs. 38
5.2 The sensitivity analysis flow for determining the impact of component selection on outputs. 38
5.3 The sensitivity analysis flow for determining the impact of subsystem margins on outputs. 38
5.4 The design concepts considered during the DSE, with each unique satellite labeled. . . 39

6.1 The implementation of five of the EPS subsystem requirements in Valispace. 42
6.2 A closeout reference in Valispace, used to automatically verify a requirement. 43
6.3 A test of the EPS battery storage size during an eclipse in Valispace. The first test step result is used to verify requirement LICCA-SYS-Sub-EPS-5. 44
6.4 The properties of the output mass vali in Valispace. 45
6.5 The connections graph from the satellite mass vali. Notice how every dependent variable is visualized and how they can be inspected to determine their value and applied margin. 45
6.6 The graph resulting from the sensitivity analysis in Valispace, showing the dependency of the satellite mass on the payload mass. 46
6.7 The relation between the inclination angle of the injection orbit and the delta-V budget, from the sensitivity analysis feature in Valispace 47
6.8 The relation between the inclination angle of the injection orbit and the delta-V budget in case the injection orbit and mission orbit have identical apogees and perigees. 48
6.9 A visualization of the effect of changing the power consumption of the thruster. 49
6.10 The system mass budget in case the satellite is designed to perform the transfer maneuver from GTO to GEO by itself. 52
6.11 The properties of the density vali. Notice how that although the value and margins are zero, the worst case values are not. 53

D.1 The measurement principle of the payload. One of the laser beam paths is shown, with the numbers indicating the order of reflection. Adapted from the DSE final report[7]. . . 70

List of Tables

2.1 The performance and budget user requirements, adapted from LICCA baseline report, Table 8.1[13]. . . . 5
2.2 System requirement for the EPS subsystem, adapted from LICCA final report, Table 9.1[7]. 6
2.3 The high-level concept trade-off table from the DSE 'Small Satellites for Gravitational Waves Observation'[14]. Colors are used to show relative scores per criterion, with red for 0-0.25, yellow for 0.25-0.5, blue for 0.5-0.75 and green for 0.75-1. . . . 9
2.4 The detailed concept trade-off table from the DSE 'Small Satellites for Gravitational Waves Observation'[14]. Colors are used to show relative scores per criterion, with red for 0.25-1, yellow for 0.25-0.5, blue for 0.5-0.75 and green for 0.75-1. . . . 10
2.5 The minimum and maximum values for each of the system budgets per subsystem as determined by the sensitivity analysis from the DSE. . . . 13
2.6 Effects of the implementation of Valispace on the project performance parameters, from the interview with Marte Medina León. . . . 17

3.1 The inputs of the system design implementation in Valispace . . . 19

4.1 Summary of the components per subsystem as selected during the DSE and in Valispace. Adapted from the DSE final report[7]. . . . 32
4.2 Delta-V budget verification of the tool using the FireSat example from SMAD . . . 33
4.3 Astrodynamic characteristic verification of the tool using the FireSat example from SMAD 34
4.4 Verification of the performance-indicative parameters of the tool using the FireSat example from SMAD, or from an external tool for the TTNC subsystem. The number are taken from the indications sections in SMAD[16]. . . . 34

5.1 The payload design mass and power consumption and mission orbit altitude for each of the satellites in the considered design concepts . . . 40
5.2 The orbital characteristics of the GTO and GEO orbits, which form inputs in the Valispace design . . . 40

6.1 System requirement for the EPS subsystem, adapted from LICCA final report, Table 9.1[7]. 43
6.2 An overview of the satellite masses, power consumptions and delta-V budgets for each concept, estimated using the tool. . . . 50

A.1 The requirement verification matrix for the tool compared to the DSE design for the (sub)system budgets . . . 64
A.2 Delta-V budget verification of the tool compared to the DSE design. . . . 65
A.3 Verification of the component selection in the tool compared to the DSE for all subsystems. 65
A.4 Verification of the performance-indicative parameters in the tool compared to the DSE design for all subsystems and astrodynamic characteristics. . . . 66

D.1 The components of the payload. The transponders are required for all passive connections, and the telescopes are required for all active connections. Adapted from DSE final report[7]. . . . 70

1 Introduction

Over the past decades, the small satellite development industry has grown exponentially and it will continue to do so for the coming decade[1]. The need for commercialized space activities is greater than ever, and the industry norm is growing towards smaller satellites in larger numbers[2]. Space projects in the past have suffered from inconsistencies, inefficiencies and inaccuracies due to the lack of a standard for engineering tools[3]. While the space industry has set a standard for some engineering tools and processes with which to streamline the design and development of satellites, such as CDP4 COMET by RHEA Group[4], there is currently no industry standard for a collaborative design tool which addresses the unique challenges posed by modern small satellite development, such as a need for high project effectiveness and efficient verification and validation processes. This leads to the problem statement: "There is currently no widely adopted standard for a collaborative space system design tool that addresses the need for accurate, efficient, and repeatable space system design processes."

One such tool which may sufficiently address this problem is Valispace, which is a web-based collaborative engineering design tool. In Valispace, the user defines a number of requirements which the system has to fulfill, and can model the system as a tree of components which can branch to any desired level of detail. The system parameters, which Valispace refers to as Valis, may be used to determine the value of other system parameters, creating a chain of Valis through which changes are calculated and propagated. The user may then perform analyses, tests, and gather documentation regarding the design. While Valispace can be used for any engineering design, its use in space system design is limited.

A standard platform is desirable, especially among young engineers, because creativity in design framework selection leads to variance in designs, which makes them hard to reproduce, potentially leading to requiring redesigns. Valispace may have the potential to be such a platform, but its effectiveness has not been studied academically.

The main objective of this report is to determine the impact of the implementation of Valispace in the small satellite design process. The platform(s) used for satellite design may have an impact on the project performance, for example, how quickly and accurately information is exchanged between teams. The research questions are:

1. *In what ways does the integration of Valispace into the small satellite design process influence project performance?*
2. *To what extent can a satellite designed using Valispace be effectively verified?*
3. *How can Valispace be used to identify potential improvements to the project 'Small Satellites for Gravitational Waves Observation'?*

The first of these questions addresses the need for an academic study into the impact on project performance, and aims to determine the impact of implementing Valispace on time efficiency, design accuracy and risk mitigation. The second question concerns the usefulness of Valispace in reducing errors in model implementation and preventing incorrect selection of Commercial Off-The-Shelf (COTS) components, either through unit and system testing, or using sensitivity analyses. The third question ad-

dresses the potential Valispace has for identifying potential improvements to existing projects.

The selection of an engineering tool can have a large impact on the project effectiveness and the optimization of the satellite design. This research will assess the advantages and limitations of Valispace as a collaborative design tool to assist users with this selection, as well as the impact of its integration on young engineers in (pre-)phase A space mission design. Furthermore, it will offer insight into what can be considered best practices within the small satellite design process and how they can differ per mission, depending on the customer requirements, team size, budget, time window, and so forth.

To gain insight into the usage of Valispace, an interview was held. This is an interview with an engineering team which implements Valispace in their bachelor thesis, and forms a qualitative assessment of the usage of Valispace. This method aims to paint an initial picture of the common practices among young engineers and their experience with Valispace. The goal of this interview is to assess the pros and cons of Valispace compared to traditional methods.

To answer the research questions, a case study was done. The case study concerns a before-and-after analysis to qualitatively analyse the impact of the implementation of Valispace on a specific case[5][6]. The selected case is a bachelor thesis project at the aerospace engineering faculty of the TU Delft which is named the Design Synthesis Exercise (DSE), henceforth referred to as the DSE project. In this project, a team of eleven students designed a laser interferometer antenna which aims to measure gravitational waves using a constellation of three Cubesats in an equidistant geostationary orbit around Earth[7]. This case was selected because it shows how eleven young engineers performed a small satellite design project in a typical mission scenario. Furthermore, one of the engineers is the author of this thesis report, hence the author is very familiar with this project and the engineering practices selected during the project. The project experienced some design conflicts and limitations due to the selected engineering tools. The design is then recreated within Valispace, implementing identical design models and performing a verification procedure and sensitivity analysis to determine the impact of implementing Valispace with regards to project performance, and to determine whether design conflicts and limitations can be mitigated using Valispace. The design recreation will not include the financial budget or technical risk mitigation to limit the scope of this research. This analysis qualitatively determines the impact of implementing Valispace on design process time efficiency, design accuracy and risk mitigation.

This thesis will first sketch a background on the commonly applied small satellite design processes, how they were applied during the DSE, and the use of Valispace for satellite design. Then, in the methodology, it is shown how the DSE was recreated in Valispace for the before-and-after analysis case study and how Valispace is used for experimentation with this design. The satellite design from the DSE recreation tool in Valispace is then described, and the verification procedure is explained. The results of the before-and-after analysis case study and experimentation with the features of Valispace are then given, and these are discussed to determine the answers to the research questions. Finally, a conclusion to the thesis is given, and recommendations for further research are described.

2

Background

To determine the impact of Valispace on project performance, the satellite design process needs to be investigated to better understand the applicability of Valispace. Furthermore, to answer the research questions, definitions of the project performance and the verification process are given.

In this section, the results from the literature study performed will be given, starting with a background on the current state of space mission design and CubeSats, followed by a description of the design process during the DSE. Then, the performance metrics are defined and the features of Valispace are described. Finally, an interview held with Marte Medina León about the use of Valispace is summarized.

2.1 Current state of space mission design process

The space mission design process is categorized by the National Aeronautics and Space Administration[8] and the European Space Agency[9] into a number of different design phases. These design phases are referred to as phases A through F, along with a pre-mission phase referred to as pre-phase A. Pre-phase A and phase A form the preliminary design phases. Phases B and C further define the satellite to the level where it can be developed. Phase D is the development phase, phase E is the utilization of the satellite, and F is the disposal phase. These phases are visualized in Figure 2.1. The satellite design process is thus done in phases pre-A through C, but to limit the scope of this thesis, only phases pre-A and A are investigated as the DSE was limited to these phases. In pre-phase A, the preliminary requirements are established, costs are estimated, and approaches to verification and validation plans are made. In phase A, the requirements are more thoroughly defined, the cost estimations are updated, preliminary design solutions are established, interfaces are defined and integration plans are made. Furthermore, the requirements are more thoroughly defined. The impact of Valispace on the satellite design process on phases B and C may be extrapolated from the obtained results, but further research would be required for any quantitative measurements.

Currently, there exist a couple of platforms which are often used in the satellite design process. These platforms allow users to design satellites using complex computations, track the current state of the satellite and determine if the design complies with the requirements. Engineers may opt for platforms specifically made for commercial satellite design such as the SatCatalog Designer[10], or a flexible and complex sandbox platform such as COMET[4] which is currently used by the European Space Agency[11]. Valispace might be a viable alternative for a satellite design platform, which is

Figure 2.1: The space mission project life cycle from NASA. Adapted from https://ntrs.nasa.gov/

described as 'intuitive', 'easy-to-use' and 'powerful' by reviews online. It is a web-based application which has a free tutorial online, that supposedly teaches the tool within two hours, significantly less than other tools on the market.

2.2 CubeSats

CubeSats are a class of small satellites which have grown exponentially in popularity over the last two decades, with over 2300 deployments of satellites in this class, 396 of which in 2023[12]. CubeSats are often launched as secondary payloads on larger missions, reducing launcher costs. CubeSats are satellites which have standardized 'units', which are 10cm x 10cm x 11.35cm cubes. These dimensions are chosen such that the useful volume within a cube is 10cm x 10cm x 10cm. CubeSat structures are design to have multiples of these units, often 10cm x 10cm with a variable length, but larger structures exist to support CubeSats with a larger width or height as well. The volume of CubeSats is referred to using U such that 1U is equivalent to 1000cm³ of useful volume, or 1135cm³ of occupied space. The industry has created this standard such that COTS components are compliant with all CubeSat structures. Selection of COTS components is often preferable over designing custom components because they do not require design and testing costs, and their Technology Readiness Level (TRL) is already determined. The parameters which are used to select the COTS components are indicative of the required performance of the COTS components, and are thus referred to as **performance-indicative parameters**.

2.3 Satellite design process during the DSE

In this section the DSE project 'Small Satellites for Gravitational Waves Observation'[13][14][7] is described in detail. This DSE was a space mission project which was designed at pre-phase A level. It spanned 10 weeks in total, and was split up in five phases:

1. Project planning and definition (1 week), during which the project planning and schedule were made, the problem was explored and high-level requirements were generated.
2. Exploration (1 week), during which the stakeholder and system requirements were generated and preliminary design concepts and budgets were determined.
3. Preliminary design (3 weeks), during which the design concepts were generated and traded off to formulate a final design concept.
4. Final design (4 weeks), during which each of the subsystems were designed, along with an integration plan and topography, a verification and validation plan, a sensitivity analysis, a financial budget and a risk mitigation plan.
5. Close-up (1 week), during which the documentation was done and the deliverables were handed in.

The preliminary design, final design and close-up phases will be described in more detail below. Besides the requirement generation, the first two phases are not required to answer the research questions. Valispace is not a tool for planning or performing literature studies and its implementation would thus not impact these phases. The hypothetical mission was named 'Laser Interferometer Cubesat Constellation Antenna' (LICCA).

In the next sections, requirement generation is described in more detail, followed by the concept generation and trade-off, a description of the payload and subsystem design, the subsystem integration and topology, the verification and validation approach, the sensitivity analysis and documentation. These phases are described in more detail to give an indication of what the pre-phase A and phase A processes entail, and some of these will be recreated in Valispace. Finally, a design description is given.

2.3.1 Requirement generation

The requirement generation was done during the first two phases of the project. It started with identification of the user requirements, followed by generating a list of stakeholders and their requirements. A requirement discovery tree was then used to generate system requirements. Each of these processes are described in detail here.

User requirements
The user requirements were identified from a description of the system by the client. This description included system performance, safety & reliability, sustainability, cost and engineering budgets. Some of these requirements were identified as driving user requirements, which were identified as requirements which impacted the design choices to be made more than the average requirement. Two others were considered killer user requirements, which over-constrained the design space and had to be disregarded at least during the concept generation and trade-off phases.

The performance and budget requirements are given in Table 2.1. The other requirements are omitted from this report as Valispace was not used to identify safety, reliability, sustainability or cost. Deleted requirements are omitted as well.

Requirement ID	User Requirement	Type
LICCA-SYS-Perf-01	The system shall be able to detect gravitational waves in the 1-10 Hz band.	Driver
LICCA-SYS-Perf-03	The spacecraft shall be orbiting in Geostationary Orbit (GEO).	
LICCA-SYS-Perf-05	The system shall provide command uplink.	Killer
LICCA-SYS-Perf-06	The system shall provide a telemetry downlink to the ground station.	
LICCA-SYS-Perf-07	The system shall provide scientific data download to the ground station.	
LICCA-SYS-Perf-08	The mission shall be composed of at least 3 small satellites in formation.	
LICCA-SYS-Perf-09	The constellation of satellites shall be flying with a distance of 72 000 km from each other	Driver
LICCA-ST-Perf-01	The spacecraft shall carry a low power laser payload for interferometry.	Driver
LICCA-SYS-Bud-01	The spacecraft shall be compliant with the CubeSat form factor i.e. it should be made up of multiple cubic units each of 10 cm x 10 cm x 10 cm.	Driver
LICCA-SYS-Bud-02	The spacecraft shall have a size of≤12U.	Driver
LICCA-SYS-Bud-03	The total required electrical power shall not exceed 40W.	
LICCA-SYS-Bud-04	The spacecraft shall have a propulsion system able to provide the <TBD>ΔV needed for the transfer from Geostationary Transfer Orbit (GTO) to GEO.	Killer
LICCA-SYS-Bud-05	The required data rate shall not exceed 1 Mbps.	
LICCA-SYS-Bud-06	A Commercial Off-The-Shelf (COTS) option shall be identified during thedesign of each subsystem, and included in the trade-off.	Driver
LICCA-SYS-Bud-07	The spacecraft design shall be compatible with existing launchers andCubeSat deployers.	
LICCA-ST-Bud-01	The launch date shall be 2030.	Driver

Table 2.1: The performance and budget user requirements, adapted from LICCA baseline report, Table 8.1[13].

Stakeholder requirements
All other stakeholders in the mission, besides the mission team and the client, were identified. These included the space industry, universities and other scientific communities, regulatory bodies, companies (for transportation, launcher, ground station etc.), sponsors, media and the public. For each of these, a couple of requirements were generated pertaining to potential opportunities, risks, legal considerations and compliance with regulations. These are omitted from this report as only a few affect the system design choices, and those that do affect the design are treated identically to the user requirements.

System requirements
The system requirements were set up using a requirements discovery tree (LICCA baseline report, Figure 8.1[13]). It identifies the difference between performing the mission technically and performing it within constraints set by the user and stakeholders. The tree then further flows down to identify functional reliability requirements, operational requirements, payload functionality requirements, and some requirements pertaining to constraints set by cost, sustainability, legality, the mission launch date, system budgets, safety and reliability.

The tree is used to generate a list of 132 system requirements. Most of these will be omitted here. Only the Electrical Power System (EPS) system requirements are given in Table 2.2. These will be implemented in Valispace in this report. At the time of the requirement generation some values were yet to determined. These were determined during the detailed system design phase, hence why the requirements are adapted from the final report rather than the baseline report.

Requirement ID	User Requirement
LICCA-SYS-Sub-EPS-1	The total required electrical power shall not exceed 40W.
LICCA-SYS-Sub-EPS-2	The EPS shall be able to receive data from the CDH
LICCA-SYS-Sub-EPS-3	The EPS shall deliver 30.6 W during daytime.
LICCA-SYS-Sub-EPS-4	The EPS shall deliver 31.0 W during eclipse.
LICCA-SYS-Sub-EPS-5	The EPS shall be able to store 90 Wh of energy.
LICCA-SYS-Sub-EPS-6	The EPS shall provide the right amount of power to each subsystem.
LICCA-SYS-Sub-EPS-7	The EPS shall adjust the voltage for each subsystem.
LICCA-SYS-Sub-EPS-8	The EPS shall provide the correct current type to each subsystem.
LICCA-SYS-Sub-EPS-9	The EPS shall protect the system from power spikes.
LICCA-SYS-Sub-EPS-10	The EPS shall provide data about power regulation to the CDH subsystem.
LICCA-SYS-Sub-EPS-11	The power source of the EPS should have an efficiency of at least 31.8 %.
LICCA-SYS-Sub-EPS-12	The EPS shall have a TRL of 7 or higher.

Table 2.2: System requirement for the EPS subsystem, adapted from LICCA final report, Table 9.1[7].

2.3.2 Concept generation
The concept generation was done during brainstorm sessions, which lead to numerous different concepts for each of the satellite properties. These concepts were split in a number of **Design Option Trees** (DOT) which gave an inclusive overview of all design possibilities. All justifiably unfeasible concepts were scrapped, and the remainder were worked out in more detail to lead to the trade-off. An overview of the DOT is given in Figure 2.2. The full DOTs for constellation and orbits are shown Figure 2.3, where the infeasible concepts are marked in red.

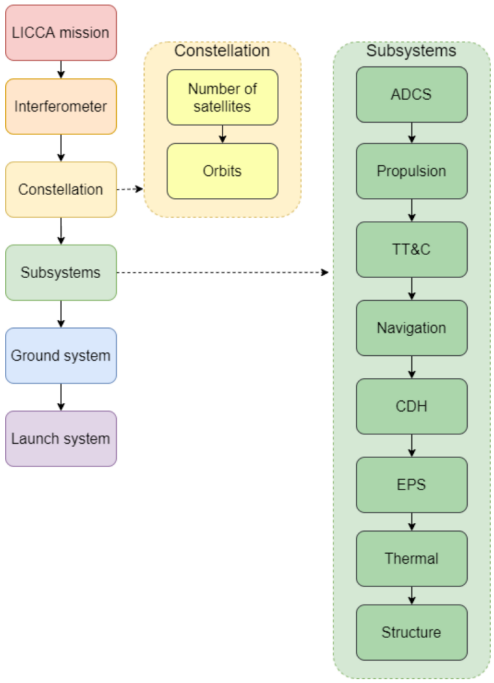

Figure 2.2: A flowdown of the design option tree used in the DSE. For the constellation and subsystem, a more detailed flowdown is shown.

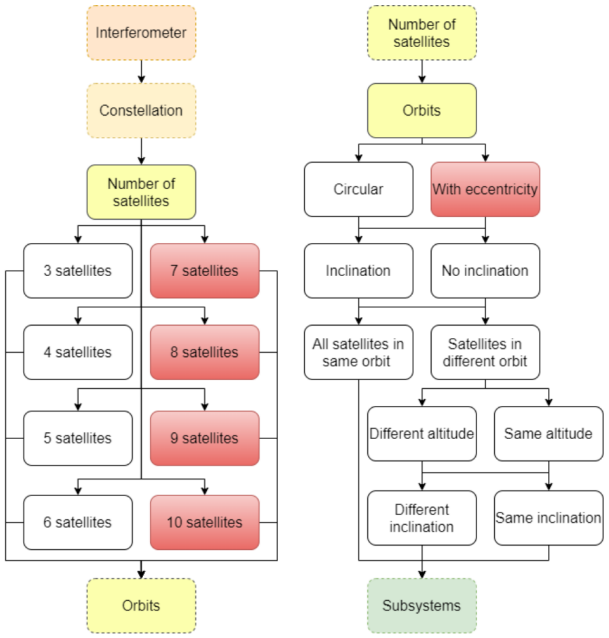

Figure 2.3: The DOTs used for the constellation concept selection during the DSE. Concepts marked red are considered infeasible.

The constellation DOTs were then used to generate eight design concepts to be traded off later, as described in Section 2.3.3. These eight concepts are described only by the number of satellites, the constellation they are in and how the lasers used to measure gravitational waves are configured. These eight concepts are given in Figure 2.4.

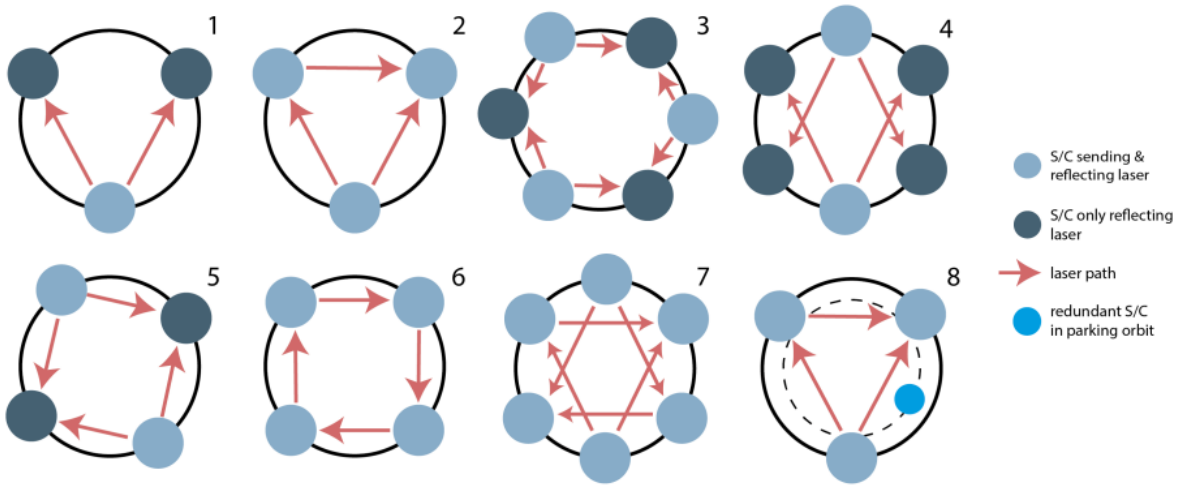

Figure 2.4: The DSE concepts generated, which were to be traded off in a later phase. These concepts are intended for forming a laser interferometer for measuring gravitational waves. All laser path lengths are identical.

Each of these concepts were compared to each other to determine if any can be scrapped before the trade-off takes place. Note that the laser path length must always be equal to $(R_e + h_{geo})\sqrt{3}$, where R_e is Earth's radius and h_{geo} is the altitude for a geostationary orbit. This is because the laser interferometer design is identical to that of the SAGE mission[15]. This means that any triangular interferometer set-up will be in a geostationary orbit, while any other shape will result in an altitude above a geostationary orbit. This means concepts 1, 2, 4, 7 and 8 are in geostationary orbits, and concepts

3, 5 and 6 are not.

Two concepts were scrapped before the trade-off. Concept 3 was scrapped as its altitude is rather high (66647 km), which would drastically increase the delta-V budget if accounting for the transfer maneuver from injection orbit to mission orbit. Concept 6 was scrapped because of the combination of a higher altitude (53258 km) which would increase the delta-V budget, and the extra generated dataset compared to concept 2 was deemed not valuable enough to justify adding an extra satellite to the constellation. Later in this report the choice to scrap these concepts will be evaluated using Valispace.

2.3.3 Concept trade-off

The trade-off was split in three: a high-level concept trade-off, a payload trade-off and a detailed concept trade-off. The separate trade-off done for the payload design is omitted here as it will not be used in this report, but the payload design itself is given in more detail in Section 2.3.4. The high-level and detailed trade-offs will be described here in more detail, as they will be evaluated using Valispace later to determine their effectiveness and necessity.

High-level trade-off

The high-level trade-off was done by taking a number of criteria, assigning a weight to each of these criteria and then scoring each of the design concepts on each of these criteria. The sum of the weighted scores per criterion was then used to eliminate three of the six concepts, with the remainder being selected for the detailed concept trade-off. The selected criteria, along with their relative scoring weight, are:

- Reliability (3)
- Cost (1)
- Delta-V (1)
- Sustainability (2)
- Quality of measurements (3)

Reliability was chosen in the context of redundancy, which in this context refers to the number of redundant active (laser sending and receiving) and passive (laser receiving) satellites. For example, concept 8 has a redundant active satellite, while concept 5 has a redundant active satellite but no redundancy in the two passive satellites. The cost was based on the number of satellites in the constellation, whether those are active/passive and where they are positioned. Delta-V, the required velocity differential to maneuver from the injection orbit to the mission orbit, was determined by the position only. The sustainability was determined by the number of satellites and how many are active. The quality of measurements was determined by the relative size of the dataset and whether it contains independent measurements. The high-level trade-off summary can be found in Table 2.3.

A sensitivity analysis was done to determine the impact of the weight selection on the concept selection, which would impact the top-scoring concept order, while the bottom 3 concepts were always in the same order. From the high-level trade-off, it was determined that concepts 1, 4 and 5 were scrapped, leaving concepts 2, 7 and 8 which were considered during the detailed trade-off.

Detailed trade-off

The detailed trade-off was done similarly to the high-level trade-off, but differed in that each of the criteria were analyzed more in-depth. The criteria with their relatively weight in this trade-off were changed to the following values:

- Risk (3)
- Cost (2)
- Delta-V (1)
- Sustainability (2)
- Quality of measurements (3)

For each of these criteria, a mix of qualitative and quantitative estimations were made to score the concepts. For risk, the likelihood of different failures taking place was compared to determine which

Concept	Reliability	Cost	ΔV	Sustainability	Quality of measurements
1	No redundant s/c *(yellow)*	GEO, 3 s/c *(green)*	GEO *(green)*	3 s/c, 1 active *(green)*	Smallest dataset *(red)*
2	No redundant s/c, redundant interferometer *(yellow)*	GEO, 3 active s/c *(green)*	GEO *(green)*	3 s/c, 3 active *(green)*	Semi-dependent dataset *(blue)*
4	Redundant s/c constellation *(blue)*	GEO, 6 s/c *(blue)*	GEO *(green)*	6 s/c, 2 active *(blue)*	Independent dataset *(blue)*
5	Redundant active s/c *(blue)*	Above GEO, 4 s/c *(blue)*	Above GEO *(red)*	4 s/c, 2 active, above GEO, multiple GS *(blue)*	Small dependent dataset *(yellow)*
7	Redundant s/c constellation *(green)*	GEO, 6 active s/c *(yellow)*	GEO *(green)*	6 s/c, 6 active *(yellow)*	Large independent dataset *(green)*
8	Redundant s/c *(green)*	GEO, 4 active s/c *(blue)*	GEO, one s/c parked *(blue)*	4 s/c, 4 active, parking orbit *(blue)*	Semi-dependent dataset *(blue)*

Table 2.3: The high-level concept trade-off table from the DSE 'Small Satellites for Gravitational Waves Observation'[14]. Colors are used to show relative scores per criterion, with red for 0-0.25, yellow for 0.25-0.5, blue for 0.5-0.75 and green for 0.75-1.

concept had the least risk of failure. For cost, a budget per satellite was estimated. For delta-V, the relative delta-V required to bring each satellite to their mission orbit was computed. For sustainability, the impact of mission phases and space debris was scored. For quality of measurements, the availability and data redundancy was taken into account. The results from this trade-off can be found in Table 2.4.

Another sensitivity analysis was done to determine the impact of different weights on the scoring results. It was determined that concept 7 was the best for the mission.

2.3.4 Payload design

Although the payload is generally considered a subsystem in itself, the goal of the DSE was not to design a payload capable of detecting gravitational waves. Rather, it was to design a satellite capable of supporting such a payload, and the payload was determined by doing a literature study on laser interferometers on other spacecraft missions, such as the SAGE (SagnAc interferometer for Gravitational wavE) and LISA (Laser Interferometer Space Antenna) projects.

For the selected concept, a list of required components was set up for measuring gravitational waves using the principle of laser interferometry. This included all components for the outgoing laser, laser transponders for reflection, and telescopes. This lead to a payload design with a mass of 5.75 kg and a power consumption of 5.114 W. A full list of components and a measurement principle description are given in Appendix D.

Other concepts, all of which are given in Figure 2.4, would have differently designed payloads to support the incoming and outgoing laser beams. These alternate designs are explored in more detail

Concept	Risk	Cost	ΔV	Sustainability	Quality of measurements
2	No redundant s/c, No redundant constellation *(yellow)*	High budget per s/c *(green)*	GEO *(green)*	Low space debris *(green)*	Semi-dependent dataset, discontinuities *(blue)*
7	Redundant constellation *(green)*	Low budget per s/c *(yellow)*	GEO *(green)*	High space debris *(yellow)*	Independent dataset, continuous measurement *(green)*
8	Redundant s/c *(blue)*	Medium budget per s/c *(blue)*	GEO, one s/c parked *(yellow)*	Medium space debris *(blue)*	Semi-dependent dataset, discontinuities *(blue)*

Table 2.4: The detailed concept trade-off table from the DSE 'Small Satellites for Gravitational Waves Observation'[14]. Colors are used to show relative scores per criterion, with red for 0.25-1, yellow for 0.25-0.5, blue for 0.5-0.75 and green for 0.75-1.

in Section 5.4. They include active satellites which are satellites that perform measurements and can both send and reflect lasers, and passive satellites which are satellites that only reflect incoming lasers and do not perform measurements.

2.3.5 Subsystem design

The subsystem design was done iteratively, where the team performed a subsystem design iteration using the most accurate available input parameters. For the first iteration, the system budgets were taken from the system budget estimation done during the exploration phase. The subsystem design iterations differ per subsystem, hence the processes are described here for each subsystem individually. Each of these design processes will be recreated in Valispace, as explained in Chapter 3. The subsystems are:

- Attitude Determination & Control Subsystem (ADCS), which determines the orientation of the satellite and makes changes to it using its control system.
- Command & Data Handling Subsystem (CDH), which processes commands, controls satellite operations and handles data.
- Telecommunications Subsystem, which receives commands from the ground station and transmits telemetry and payload data to the ground station.
- Electrical Power Subsystem (EPS), which is responsible for generating, storing, conditioning and distributing electrical power to the other subsystems.
- Temperature Control Subsystem, or Thermal Subsystem, which controls the temperature of the satellite components to maintain them within the acceptable ranges.
- Propulsion Subsystem, which provides thrust to alter or maintain the orbit of the satellite.
- Structure Subsystem, which provides the framework of the satellite and provides structural stability.

For the mechanical devices subsystem, no models were used as it is a generalized subsystem for all mechanical devices, which was accounted for by introducing design margins for all subsystems instead. Below is a description of each of subsystem design processes, simplified to the design principles used, which are marked in bold text.

Attitude Determination & Control Subsystem

For the ADCS, a number of sensors and actuators are used to determine and control the attitude of the satellite. A **momentum budget** was generated to determine if the subsystem can tackle momentum

originating from external disturbances and from slew manoeuvres, and how often the momentum must be dumped before reaction wheels become (over)saturated. This budget uses angular momentum as a model. Furthermore, a **pointing accuracy** and **pointing stability** was determined from the pointing accuracy of the sensors and the precision of the actuators, which were estimated by the manufacturer of the respective components. The selection of sensors and actuators depended on the mission characteristics, including the need for high-precision attitude determination and control, the environment and the pointing direction needed for the payload and telecommunications subsystems. ”Space Mission Analysis and Design”[16] was used for its procedure for estimating these parameters and selecting sensors and actuators, but the actual sizing of the ADCS subsystem depended greatly on the COTS options available, hence these models for parameter estimation were not used.

Command & Data Handling Subsystem
For the CDH, a central command & data handling subsystem aims to read sensors, respond to the satellite status and give instructions to other subsystems. For this subsystem, a **data rate budget** was generated by estimating the total bit rate coming from all sensors, and the bit rate required to communicate with all actuators. This data rate budget concerns the internal data rate and should not be confused with the link budget, which is treated in the telecommunications section. Furthermore, a **memory budget** was established by determining the size of all data to be stored, and how often its memory can be ’cleared’ by downlinking data to a ground station using the telecommunications subsystem. A **processing budget** was generated based on data processing requirements.

Estimations on data rates, memory budgets and processing budgets are taken from ”Space Mission Analysis and Design”[16] and ”Elements of Spacecraft Design”[17]. Finding better CDH sizing estimations proved difficult, but should be considered later in the tool development process.

Telecommunications Subsystem
The telecommunications subsystem concerns all communication between the satellite and a ground station. For the link between the satellite and a ground station, a **link budget** was generated. The link budget relates the transmitter power to the received signal power and Signal-to-Noise Ratio (SNR) using a number of parameters, such as free space loss, antenna gains, system losses using system noise temperatures and potentially more, depending on the level of detail required. The receiver SNR can then be used to estimate the potential bit rate of the connection.

A generic model for link budget analysis is given in ”Handbook of Satellite Applications”[18]. Some typical parameters for the telecommunications subsystem are also given in ”Space Mission Analysis and Design”[16]. A model for determining the transmission bandwidth and link budget for a Cubesat is provided in ”Link budget analysis for a proposed Cubesat Earth observation mission”[19]. The latter also gives loss estimates caused by rain attenuation and polarization effects.

Electrical Power Subsystem
The EPS uses the **power budget** to estimate the power consumption of the subsystems, such that an power income requirement could be defined. The power income was then used to design solar panels based on their area, efficiency and degradation. Furthermore, a **energy storage** estimation was done based on maximum eclipse time and the required energy to maintain system functionality for the eclipse duration, which in turn can be used to estimate battery size. The article ”Design and Management of Satellite Power Systems”[20] lists a number of design parameters and how to estimate these using a power budget and energy storage estimates. ”Space Mission Analysis and Design”[16] provides a model to estimate the power budget and energy storage.

Thermal Subsystem
For the thermal subsystem, a **heat balance** was used to determine the minimum and maximum temperature of the satellite under different conditions (e.g. orientation, eclipsed) using the outgoing heat from black-body radiation and incoming heat from satellite components, sunlight and other incoming radiation, and potentially heater components. The heat flow can then be used to determine the equilibrium temperatures. Satellite components were used to determine the allowable temperature range before component failure can occur. The heat balance was established by modeling the satellite as a number of nodes, assuming heat is absorbed and radiated from these nodes and flowing between the nodes. The heat flowing between nodes was done using time constants.

The Small Satellites Thermal Modeling Guide[21] gives an overview of how such a model is generated, and an indication of how accurate the model is for the chosen level of detail. It also includes tables of the absorbtivity of materials, and suggested temperature ranges of different subsystems. The article "Small Satellite Thermal Design, Test and Analysis"[22] gives an indication of the worst and best-case scenarios for simulation, including day/night times and albedo factors.

Propulsion Subsystem
For the propulsion subsystem a **Delta-V budget** was generated based on the astrodynamic characteristics of the mission, which was used to relate the propellant exhaust velocity to the propellant mass and system mass using the **Tsiolkovsky rocket equation**. The propellant volume and propellant storage requirements were used for tank sizing. The selection of the propulsion type and the propellant was done using the characteristics of the mission. "An Overview of Cube-Satellite Propulsion Technologies and Trends"[23] gives an overview of available CubeSat-compatible propulsion type solutions, along with some performance estimations. "Space Mission Analysis and Design"[16] lists applications for each of these types, along with their advantages and disadvantages.

Structure Subsystem
A structure COTS component of the desired size of 12U (30cm x 20cm x 20cm) was selected, which was analyzed to determine if it was sufficient for the mission. A **structural analysis** of the structure relates the system mass and structural topology to the maximum loads and acceleration experienced in-flight (typically experienced during launch). The maximum acceleration of the launcher was used to calculate the maximum internal stresses, which was compared to the yield load in longitudinal and lateral directions. Furthermore, a **vibrational analysis** was done to determine if the structure could experience vibrational resonance during launch, which is undesirable.

2.3.6 Subsystem integration and system topology
One of the client requirements was that the satellite was a CubeSat with dimensions equal to 20cm x 20cm x 30cm. Once it was determined that the sum of subsystem volumes was lower than the structure can fit, a few members dedicated to assigning the satellite faces to the different components which require them, such as solar array panels, thrusters (both propulsive and rotational) and the antennae.

2.3.7 Verification and validation
After models were set up to design the satellite, these models required verification and validation. Verification is defined as *"substantiating that the model is transformed from one form into another, as intended, with sufficient accuracy"*[24]. Validation is defined as *"substantiating that within its domain of applicability, the model behaves with satisfactory accuracy consistent with the study objectives"*[24]. The difference lies in that verification aims to compare the implemented model to the intended model, whereas validation aims to compare the implemented model to reality.

Verification of the design required verification of the used computation sheets and software. Unit testing of individual components was done to ensure the results were as expected, even if changes were made to these units[25]. System testing was done to ensure the components were integrated correctly and the system gave the expected results[26]. Verification of the product was done by creating a verification matrix, which contains the requirements for each of the subsystems and the verification method: analysis, test or inspection. The requirements in this matrix were set up during the mission planning phase for system requirements and detailed development phase for subsystem-specific requirements. The design of the satellite was compliant with each of these requirements on paper.

Validation of the product was done by establishing a validation matrix, giving the validation capabilities of the selected validation methods. The methods were selected using a mission fault tree. The selected validation methods were end-to-end information system testing, mission scenario tests, operation readiness tests, stress-testing and simulation.

2.3.8 Sensitivity analysis
The sensitivity analysis was done by identifying the largest uncertainties for each of the subsystems and determining their effect on the system budgets. These were used to determine the range of the

volume, power and mass estimations. An overview of these ranges is shown in Table 2.5.

Subsystem	*Minimum Power [W]*	*Maximum Power [W]*	*Minimum Mass [kg]*	*Maximum Mass [kg]*
Payload	4.63	5.61	4.93	6.38
ADCS	9.90	9.90	2.84	2.84
Propulsion	5.99	11.50	4.35	4.97
TTNC	5.50	5.50	0.53	0.53
CDH	0.55	3.30	0.01	0.29
EPS	0.80	0.80	1.15	1.41
Thermal	0.00	10.00	0.39	0.48
Structure	0.00	0.00	3.65	5.48
Total	**27.4**	**46.6**	**19.6**	**26.8**

Table 2.5: The minimum and maximum values for each of the system budgets per subsystem as determined by the sensitivity analysis from the DSE.

The worst and best case scenarios were analyzed, which gave the impact on the overall design. For some of the worst case scenarios it was determined that the effect would require a partial redesign of subsystems. For example, if the maximum power requirement of 46.6W is reached, the power subsystem would require a redesign. Some of the snowballing effects were accounted for as well, for example the maximum mass scenario would require additional propellant, which pushes the required volume over the 12U supported by the structure, which would require a larger-scale redesign. This effect was not accounted for during the DSE as the sensitivity analysis accounts only for uncertainties in individual subsystems and not their effect on the design as a whole. The impact of these limitations during the DSE analysis will be evaluated using Valispace in Section 6.3.

2.3.9 Documentation

Most of the documentation was done in Excel files, Python code, and figures generated in Draw.IO, and Python. This was later compiled in LaTeX/Overleaf. The Excel files were maintained separately for each mission aspect and subsystem, leading to some design conflicts during the detailed design integration phases, although this did not prove to be a challenge during the documentation phase as the values were 'frozen'.

2.3.10 Design description

The chosen design concept consists of two constellations of three satellites in an equatorial geostationary orbit. Each of these satellites has an identical design. They are 12 U Cubesats (2x2x3) weighing 22.3 kg, with a power consumption of 30.60 W. It uses a taxi vehicle for orbital injection and a propulsion mechanism for orbital maintenance, realignment, and disposal only. A full overview of the design, including budgets, components, and performance-indicative parameters, is given in Appendix A.

2.4 Project performance metrics

The research questions of this thesis refer to the impact of the integration of Valispace, which affects the design process in certain ways. This impact must be measured by some metrics to determine if the impact is not merely perceived as significant by the teams, but rather if it actually impacts the design process greatly enough to justify subjecting the engineers to the process of learning Valispace and still save resources and generate value overall.

To estimate the impact of the implementation of Valispace, the project performance metrics are defined. Selection of these metrics was done to compare and emphasize the differences in projects which do and do not implement Valispace, and they are hence relative metrics[27]. The performance metrics are defined to be independent, such that they can be analyzed independently, and are inspired by the selection of performance metrics from generalized projects[28].

1. Time effectiveness, which is a measure of the time spent per engineering process, accounting for delays introduced by the engineering framework and processes selected by the team.
2. Design accuracy, which is a qualitative measure of how well the design reflects the real-world performance and behavior of the satellite during each of the design iterations, as well as how well the design can be verified and analyzed to determine its accuracy.
3. Risk mitigation, which is a qualitative measure of how well risks are mitigated by the selection of the processes.

Each of these metrics will be assessed qualitatively in this research. Quantification of time effectiveness would require experimentation or surveying, which is outside the scope of this research. Design accuracy can be measured quantitatively, but specifically the impact of using Valispace on design accuracy could not be quantified analytically in this report. This is because differences in the design cannot be attributed to Valispace exclusively. Any change to the design in Valispace could be attributed to user choices, meaning it is not an objective measure of how Valispace affects design accuracy. Risk mitigation is too complex to quantify because each risk has its own unique impact, each of which would need to be quantified separately.

It should be noted that these metrics are generically not independent (e.g. design inaccuracies may introduce delays, which may increase risks), but they are to be analyzed/measured independently in this thesis.

2.5 Valispace for space mission design

Valispace is a web-based collaborative systems and requirements engineering platform. It is created by Louise Lindblad and Marco Witzmann for engineering applications, drawing inspiration from their experience with collaboration on space projects and the potential risks of not implementing an effective framework[29]. It has since been acquired by Altium.

The application allows the user to create requirements for their system design, and verify these requirements by creating a system design tree tree which defines the design parameters on any level of detail. The user can also create simulations to verify these requirements, or perform an analysis. Each of these features are referred to as modules within the software, and they are explained in more detail below[30]:

- **Requirements:** in the requirements module, the user can enter requirements which need to be fulfilled when designing the product. These requirements can be verified by a number of different means, each of which are connected to other modules in the application: rules (system design module), analysis (analyses module), test (test module), review and inspection. The requirements can be structured as a graph, where requirement verification can depend on sub-requirement verification. These requirements can be visualized and exported, and their status can be tracked.
- **System design:** in the system design module the product to be designed can be modeled as a graph, in which each tree component can be assigned a number of properties. The most simple property, a physical quantity with a value and unit, are referred to as 'valis'. These valis can have any unit, and the application automatically converts all units to a standard unit. Valis are calculated using a user-defined function, which may depend on other valis or constants defined within the application. Changes to these valis are then automatically propagated. Budgets for each vali are given within this module. The user can also define upper and lower margins for the valis, representing best-case and worst-case scenarios. The user can also design different system states using modelists, allowing the user to simulate how variables may be affected depending on the state of the system. The user can also add a dataset to a component, allowing the user to interpolate or extrapolate valis from a dataset rather than a known formula or value.
- **Analyses:** in the analyses module the user can analyse the product to be designed in a number of ways, either by directly displaying the data in a tabular structure or by allowing the user to visualize the data on documents. These documents can display text, tables, graphs and videos of the specifications of the product. Users can add discussions to these analyses to make conclusions about the state of the product.

- **Time sequences:** in the time sequences module a user can perform a simulation of their product in different system states using the modelists as defined in the system design module. The user can simulate the system state changing in time and compute variables integrated over an arbitrary time window. The user can also create new variables to track and create exit conditions.
- **Tests:** in the tests module the user can define tests to verify their product. Each such test can be assigned to a specific requirement from the requirements module to propagate the result of a test. Each test is defined in steps, each of which can be split up in sub-steps to clearly display the details of the test.
- **Python:** a Python-based Application Programming Interface (API) exists which allows the user to access and update Valispace objects from within Python, which may be useful for circumventing limitations to the features of Valispace if required.

Each of these modules are connected in their own ways, and work together asynchronously to allowing the user to work on a product simultaneously by propagating changes to the product whenever the server is ready to process the new instructions.

Besides these modules, a number of features are added to Valispace to simplify and document the design process, and improve collaboration by localizing communication to the relevant regions. These include:

- **Discussions:** the user can start discussions on any Valispace item (such as a vali, time sequence or requirement) and mention other users to resolve design conflicts or issues.
- **Tags:** the user can add tags to components and requirements to indicate a certain quality of those components and requirements, such as design version, the state of implementation in Valispace or the people working on them.
- **History:** all changes to the product are tracked in Valispace. The user can view a list of each change throughout the history of the design process, including the author and the source of the change. A source might be a change in the function computing the value or a change to a value used within the function.
- **Subscriptions:** The user can subscribe to Valispace elements (requirements, components, tests and time sequences) to be notified of changes made to these elements.
- **Tasks:** the user can create tasks that serve as inputs or outputs of Valispace elements, or may relate to the element itself in another way. These tasks have a state (to-do, started, completed) and may be scheduled with a start date, end date and due date.
- **Permissions:** Valispace has a permission system where each element in a workspace can have their own user permissions, with all permissions granted by default. Permission options include 'read-only', 'read&write' or 'manage', the latter of which means the user can also change the permissions of other users. 'Superusers' are given all rights, including the right to grant and revoke superuser status to other users.
- **Exporting:** the user can export data and visualizations to their computer or to other applications.

2.6 Interview with Marte Medina León

A meeting was held with a member of DSE group 24 of spring 2023, Marte Medina León. The purpose of this meeting was to exchange experiences with Valispace. The DSE group used other applications for the purposes of communication and tracking the design process, and Valispace was simultaneously maintained by León to learn to use the application and determine its usefulness. An overview of the questions asked during the interview, along with the reasoning for asking these questions, can be found in Appendix B. León addressed a number of advantages and disadvantages of utilizing Valispace to design their mission:

Advantage 1: acceptable learning curve

León stated that Valispace was not intuitive at first. Valispace offers its own tutorial, which took approximately 2 hours to do and gave an overview of the capabilities of Valispace. Léon himself was dedicated to maintaining the Valispace workspace, and was hence the only member to perform the tutorial. Beyond the tutorial, becoming used to Valispace required more hands-on experience with the

application. However, after a day of use Léon became accustomed to using Valispace, and the rest of the team members could learn to use Valispace directly from León without too much hassle.

Advantage 2: connected system properties
León mentioned that the automatic propagation of changes throughout components allowed them to quickly check how small changes to the properties of components affected the system, and whether the system fulfilled the requirements. Furthermore, the connections between the system reduced communication time and allow them to track changes to valis effectively.

Advantage 3: scripting and automation
León had utilized the scripting capabilities within the Valispace UI to set the requirement fulfillment status of a parent requirement depending on the status of its children. The automation feature within Valispace allowed León to automatically run this script whenever a requirement fulfillment status was changed. Furthermore, one of the bugs present within Valispace, the automatic generation of a new 'settings.py' file within the Valispace script repository, could be resolved by adding a script which removed excess copies of this file whenever another script was run. Valispace is thus flexible in the sense that its limitations can be worked around using the scripting and automation features.

Disadvantage 1: delays in change propagation
León stated that it would sometimes take Valispace up to a minute to propagate all changes to the system. Given the contex of their workspace only being used for a 10-week pre-phase A project, these delays may be much larger if the workspace is used for a detailed design during design phase B. Besides change propagation delays, the workspace would still load acceptably quickly, taking 5 seconds at most to load the components or requirements.

Disadvantage 2: bugs in software
León stated that a number of bugs were present within Valispace, requiring additional resources to prevent these bugs from interfering with the design process. Bug 1 is the aforementioned 'settings.py' file generation bug, with posed no major threat to the design process but still required an additional script to prevent excess file generation. Bug 2 concerned the Valispace API, preventing a module aiming to connect Valispace to Satsearch, a database for satellite components, from functioning. This was determined to be a fault on the end on Valispace.

Disadvantage 3: flaws in software
A flaw of Valispace was that there is no setting which would automatically mark a parent requirement as fulfilled once all of its children are fulfilled, which is in practice often part of what is required for a parent requirement to be fulfilled. It should hence be noted that, while such flaws can be worked around, more can be found and these may require additional scripting to resolve, if they can be resolved at all. The need for scripting to work around such flaws requires additional time and may introduce risks.

Each of these six findings can be described in terms of the three project performance parameters, although only qualitatively, as the impact of the implementation of Valispace is from the perspective of one user and cannot be measured. Furthermore, a single interview would be a sample size '1' in the context of a survey, hence conclusions drawn can not be generalized to all projects. The perceived effects on the advantages and disadvantages on the project performance parameters (compared to *not* implementing Valispace) are given in Table 2.6. These observations lead to believe Valispace has both significant upsides and downsides, and the way these may balance out cannot be determined from one interview. If the discussed flaws and bugs are resolved in later versions of the software, only the delayed change propagation seems to remain as a downside.

Aspect	Time effectiveness	Design accuracy	Risk mitigation
Adv. 1	No effect	No effect	Less risk, as there is enough space for all required resources
Adv. 2	Less time, as changes do not need manual propagation	No effect	Less risk, as propagation is not dependent upon people
Adv. 3	No effect	Better design due to large flexibility of software	No effect
Disadv. 1	More time due to requirement to learn software	No effect	No effect
Disadv. 2	More time as bugs require workarounds	No effect	More risk if bugs are not caught
Disadv. 3	No effect	Worse design if flaws are not accounted for	More risk if flaws are not accounted for

Table 2.6: Effects of the implementation of Valispace on the project performance parameters, from the interview with Marte Medina León.

3

DSE recreation methodology

In this chapter, the methodology for performing the research will be described. The DSE project is recreated in Valispace to perform a qualitative analysis of the impact of utilization of Valispace in a satellite design project on the project performance.

The impact of utilizing Valispace needs to be researched. For this purpose, a before-and-after analysis is done, in which the DSE project 'Small Satellites for Gravitational Waves Observation' is recreated within Valispace, and verified to make sure the recreation is accurate. The engineering processes are then recreated to simulate usage of the application in the engineering framework and to allow experimentation with the features of Valispace. For each of the engineering processes, the impact of project performance can then be measured or analyzed (depending on the measurability of the metrics). The recreation of the DSE project within Valispace will henceforth be referred to as 'the tool'.

The tool creation in Valispace was done first by estimating the system characteristics, then by estimating the astrodynamic characteristics and finally designing each of the subsystems in the system design module.

3.1 Before-and-after analysis case study set-up

The implementation of Valispace in a space mission design project is a case study. A case study is here defined as "an intensive study about a unit, which is aimed to generalize over several units"[5]. The case considered here is the implementation of Valispace in the system design of the DSE project, which is aimed to be generalized over all pre-phase A space system designs.

In this before-and-after analysis the before-case is the DSE without Valispace implementation, and the after-case is with Valispace implementation. This is done to identify the differences between space system design processes with and without Valispace. The before-case does not require further set-up, as it is already done and verified by the supervisors of the project. The after-case still requires implementation in Valispace and will require verification with regards to both the DSE to guarantee an identical design, and another space mission to verify correct implementation of design models.

The before-and-after analysis case is a qualitative analysis. The project performance parameters are difficult to quantify and statistics are not readily available concerning Valispace usage and its impact on project performance. The insider's perspective of the researcher, also a participant during the DSE project, allows for a qualitative assessment of the impact of Valispace usage[6].

3.2 System design tree

The system design tree in the system design module, which forms the basis of the tool, is visualized in Figure 3.1. The colors indicate the levels of the tree, while the arrows illustrate the dependency of components on other components. Component A pointing towards component B means that component B is dependent upon properties of component A. It should be noted that that not all connections are visualized here as they can be rather complicated.

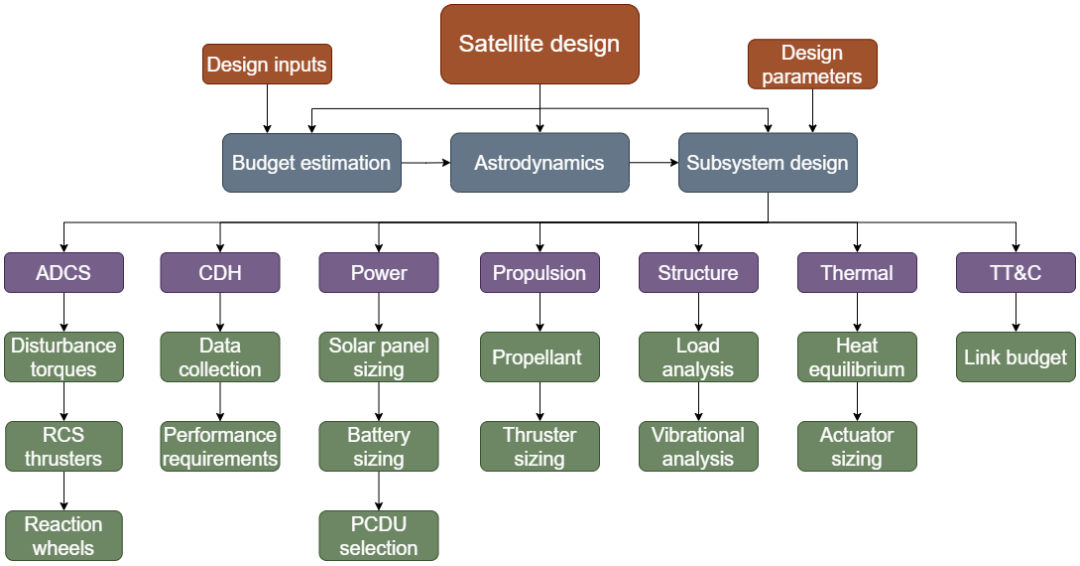

Figure 3.1: The system design tree as created in the tool. The tree levels are indicated using colors, with the arrows forming the flow of change propagation.

Input	Value [unit]
Mission lifetime	2 [year]
Payload mass	5.75 [kg]
Payload power consumption	5.114 [W]
Payload data generation rate	7680 [bit/s]
Injection semi-major axis	42077 [km]
Injection eccentricity	0 [-]
Injection inclination	0 [deg]
Mission orbit semi-major axis	42157 [km]
Mission orbit eccentricity	0 [-]
Mission orbit inclination	0 [deg]
Pointing accuracy	0.13178 [deg]

Table 3.1: The inputs of the system design implementation in Valispace

3.3 System design inputs

In space system design projects, the system design is dependent upon the specification of the missions. Hence, the mission design is partially done beforehand. From the mission design phase done during the DSE, some specifications were selected which were identified as drivers for the system design. These form the inputs of the system design and will be treated as such in Valispace by creating a separate component in the system design tree named 'Inputs'. These inputs include the following parameters:

- The lifetime of the spacecraft.
- The mass, power consumption and data generation rate of the payload.
- The semi-major axis, eccentricity and inclination angle of the launcher injection orbit and mission orbit.
- The required pointing accuracy of the satellite.

Each of these inputs has varying degrees of impact on the final design, where the lifetime mostly relates to the gradual degradation of components, the payload is used for initial characteristics estimations, the orbital elements determine the bulk of the delta-V budget and astrodynamic characteristics, and the pointing accuracy is relevant for the ADCS subsystem design. An overview of each of the design

inputs for the DSE case is given in Table 3.1. It should be noted that the payload is further expanded upon for an analysis of the concept trade-off procedure in Section 5.4, but since it will be verified to yield the same mass, power, and data generation rate in the DSE case, the increased detail of the payload design will not lead to new inputs for the system design and is thus not included here. Some of the design inputs were selected specifically to allow for a first-order system characteristics estimation, which is described in the next section.

3.4 Spacecraft characteristics estimation

After implementing the design inputs, the first-order system characteristics estimation was done using the customer requirements and payload estimations. The inputs used for these characteristics are the system design inputs mentioned in Section 3.3. Models were taken from Space Mission Analysis and Design[16] (SMAD), which include the model for first-order system mass and power budget estimations. The system characteristics estimation flow is given in Figure 3.2. The ground station and launcher selection was done at a later stage in the DSE in compliance with the satellite design, but in the system characteristics estimation step of the tool creation, these must be selected in advance and can be changed during later experimentation.

Figure 3.2: A graph visualizing the first-order system characteristics estimation process, for utilization in Valispace.

The payload is treated as a subsystem in that it has interfaces with many of the other subsystems, but its design is done in advance to make sure the other subsystems are sized effectively. Any changes to the payload design in a later stage would create a snowball effect on the system characteristics.

As can be observed in Figure 3.2, the four inputs form the basis of the system characteristics estimation. Other drivers for the first system characteristics estimation, such as customer requirements and sustainability requirements, are considered to be accounted for in one of these four inputs. For example, the customer requirement which states that the satellite volume must be at most 12U (Cubesat units) is accounted for in the launcher selection and payload design.

The first system characteristics estimation is done using SMAD models, which form simple relations between these inputs and the characteristics. These relations are given here:

$$R_{data,spacecraft} = R_{data,payload} + 3620 bit/s \tag{3.1}$$

$$P_{spacecraft,out} = 6.67 \cdot P_{payload,in} \tag{3.2}$$

$$m_{spacecraft,dry} = 3.3 \cdot m_{payload} \tag{3.3}$$

$$m_{spacecraft,wet} = m_{spacecraft,dry} \cdot e^{\frac{\Delta V}{gI_{sp}}} \tag{3.4}$$

$$V_{spacecraft} = \frac{m_{spacecraft,wet}}{1619.6 \frac{kg}{m^3}} \tag{3.5}$$

With R denoting the data generation rate, P for power, m for mass and V for volume. I_{sp} is the specific impulse of the propulsion system, g the gravitational acceleration on Earth's surface and ΔV is the delta-V budget of the mission. The numbers come from empirical relationships, some of which are not directly taken from SMAD but from more recent models[31]. For example, the system density estimation given in SMAD is more applicable to larger satellites, whereas the density of 1619.6 kg/m^3 is an estimation applicable to CubeSats specifically. The additional 3620 bit/s in Equation 3.1 follows from the sensor selection for housekeeping data, taken from Table 8.3 in the DSE final report[7]. The propellant mass follows from the Tsiolkovsky rocket equation given in Equation 3.4, which shows how the relation between the dry and wet mass of the satellite depends on the Delta-V required for the mission.

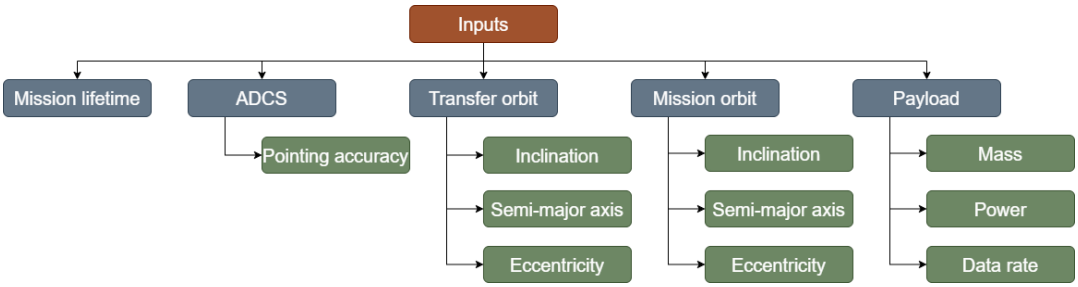

Figure 3.3: The system characteristics estimation inputs, as structured in the system design module in Valispace

The system characteristics estimation inputs were given a separate tree in the Valispace system design module. This tree is given in Figure 3.3. As can be seen, the number of inputs is very limited. The other design drivers are instead taken from 'typical values' for various parameters within the system or subsystem design, which will be referred to as the **design parameters**. These design parameters are defined within their relevant sections within the system design module, but are not considered inputs for the system design. The resulting characteristics from these inputs are also given space within their relevant (sub)systems, but it should be noted that these are the initial estimations and not the results of the system design. Hence they are named accordingly, for example, the first system mass estimation is named 'SystemMassEstimation' in the structure subsystem section, as to not be confused with the 'Mass' in the satellite system section. With the first characteristics established the astrodynamic characteristics can be determined as described in the next section.

3.5 Astrodynamic characteristics

With the inputs and first-order characteristics estimations done, some of the astrodynamic characteristics are determined. The astrodynamic characteristics of the mission were estimated and computed in a separate section in the system design module. The main goals of the astrodynamic characteristics estimation are to find the delta-V budget, the minimum and maximum orbital radius, the velocity, the orbital period, the maximum eclipse period and the maximum time in view of the satellite. The inclusion of the astrodynamic characteristics in the tool is described in detail below.

3.5.1 Delta-V budget

The delta-V budget is an important mission parameter, stemming from required orbital maneuvers and maintenance and used to determine the propellant mass required. The estimation of the delta-V budget is done by determining the minimum delta-V required to perform a maneuver from the injection orbit to the mission orbit, to perform orbital maintenance in the form of drag compensation and realignment maneuvers, and to perform a maneuver from the mission orbit to the end-of-life orbit. Realignment maneuvers are required for the DSE mission as the constellation requires equidistant satellites. The maneuvers are assumed to be Hohmann-transfers, which are the orbital maneuvers with the least required delta-V. A Hohmann transfer is illustrated in Figure 3.4.

For such a transfer, two impulsive thrusts are required. It should be noted that neither the initial orbit nor the final orbit are necessarily circular.

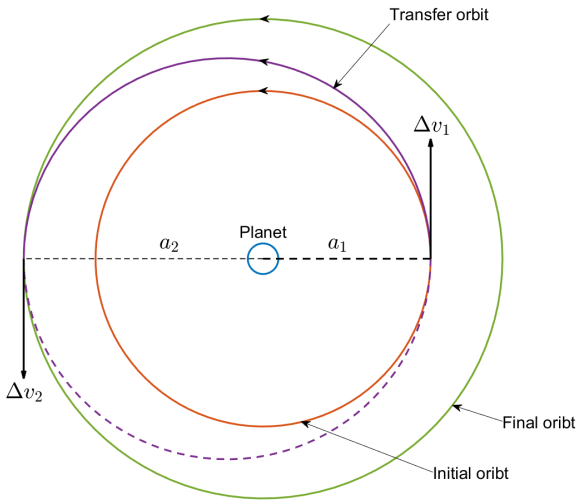

Figure 3.4: An illustration of the Hohmann-transfer, which is the maneuver from any orbit to another orbit with the lowest delta-V requirement.

The apogee and perigee of the injection orbit, mission orbit and end-of-life orbit are computed using Keplerian orbital elements. These are then used to compute the delta-V required for each of the impulsive shots. The computation is split up in four delta-V calculations denoted delta-V 1 through 4:

- The delta-V required to go from the initial orbit to the circular initial orbit.
- The delta-V required to go from the circular initial orbit to the transfer orbit.
- The delta-V required to go from the transfer orbit to the circular final orbit.
- The delta-V required to go from the circular final orbit to the final orbit

It should be noted that delta-V 1 and delta-V 4 may be negative, which implies that negative delta-V is required to go from one orbit to another. Fundamentally, delta-V cannot be negative, yet delta-V 2 and delta-V 3 may be lowered by these values. The following equations show how these delta-Vs are computed, and how they are summed. Subscripts 'i' and 'f' refer to the initial and final orbits of the Hohmann transfer, as shown in Figure 3.4.

$$\Delta V_1 = \sqrt{\frac{\mu}{r_{i,perigee}}} \cdot \left(\sqrt{\frac{2 \cdot r_{i,apogee}}{r_{i,perigee} + r_{i,apogee}}} - 1 \right) \tag{3.6}$$

$$\Delta V_2 = \sqrt{\frac{\mu}{r_{i,perigee}}} \cdot \left(\sqrt{\frac{2 \cdot r_{f,apogee}}{r_{i,perigee} + r_{f,apogee}}} - 1 \right) \tag{3.7}$$

$$\Delta V_3 = \sqrt{\frac{\mu}{r_{f,apogee}}} \cdot \left(1 - \sqrt{\frac{2 \cdot r_{i,perigee}}{r_{f,apogee} + r_{i,perigee}}} \right) \tag{3.8}$$

$$\Delta V_4 = \sqrt{\frac{\mu}{r_{f,apogee}}} \cdot \left(1 - \sqrt{\frac{2 \cdot r_{f,perigee}}{r_{f,perigee} + r_{f,apogee}}} \right) \tag{3.9}$$

Here, μ is the gravitational parameter of Earth and r refers to the distance to the center of Earth. For equation Equation 3.6 it can be seen that on the right-hand side, if the injection orbit is circular, then the apogee and the perigee of the injection orbits are equal and delta-V 1 becomes zero. For Equation 3.9, a similar conclusion can be drawn if the mission orbit is circular. The total delta-V can then be calculated. The inclination adjustment is accounted for during the second impulsive shot using the cosine law as shown in Equation 3.12. The subscripts 't' and 'f' refer to the transfer and final orbits shown in Figure 3.4.

$$\Delta V_{impulsive,1} = |\Delta V_2 - \Delta V_1| \tag{3.10}$$

$$V_{apogee,t} = V_{apogee,f} - \Delta V_3 + \Delta V_4 \tag{3.11}$$

$$\Delta V_{impulsive,2} = \sqrt{V_{apogee,t}^2 + V_{apogee,f}^2 - 2 \cdot V_{apogee,t} \cdot V_{apogee,f}^2 \cdot cos(i_{injection} - i_{mission})} \tag{3.12}$$

$$\Delta V_{transfer,total} = \Delta V_{impulsive,1} + \Delta V_{impulsive,2} \tag{3.13}$$

These equations can be used for both transfers, from injection to mission and mission to end-of-life. Besides the transfer delta-V, the budget must also account for maintenance maneuvers and drag compensation. This is done by estimating the drag experienced by the satellite and assuming the satellite must continuously compensate for this drag using its delta-V budget.

Space Mission Analysis and Design (SMAD) gives an estimate for the delta-V required for drag compensation at a given altitude in the form of a graph (Fig. 7-8, SMAD[16]). This graph gives the relation between the delta-V-to-ballistic-coefficient ratio per year and the altitude of the satellite. Implementation of this graph in Valispace is done by creating a dataset and entering the estimated ratio at many different altitudes. Since the graph ends at near-zero, extrapolation is set as equal to the last known value, which is estimated to be zero at an altitude of 2000km. For interpolation, linear interpolation is selected since it is better than step-wise, but any polynomial of exponential option would fit better. An estimation for the ballistic coefficient is given in Equation 3.14, where m is the mass of the satellite and A_f the frontal area.

$$C_b = m/A_f \tag{3.14}$$

At the altitude in the given scenario, with a geostationary mission orbit, the drag compensation delta-V is negligible. At an altitude of 1000 km the compensation delta-V at maximum solar activity is in the order of 10^{-1} m/s, which is further halved every 70 km. Therefore at the mission altitude of 35786 km this number is negligible compared to the other delta-V components.

For orbital maintenance SMAD gives an estimate of $73m/s$ per year.

Finally, a delta-V budget estimation is made, shown in Equation 3.15.

$$\Delta V = \Delta V_{transfer1} + \Delta V_{transfer,2} + \Delta V_{drag} + \Delta V_{maintenance} \tag{3.15}$$

3.5.2 Geometric properties

From the mission orbit, a number of geometric properties are estimated which are relevant for subsystem design. First, the minimum and maximum radii of the mission orbit must be estimated, as well as the orbital period and average velocity as shown below.

$$r_{min} = a \cdot (1 - e) \tag{3.16}$$

$$r_{max} = a \cdot (1 + e) \tag{3.17}$$

$$T = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{3.18}$$

$$V_{avg} = \frac{2\pi a}{T} \tag{3.19}$$

Where a is the semi-major axis of the orbit, e is the eccentricity and μ is the gravitational parameter of Earth. The minimum angular radius of Earth, relative from the satellite, is given in Equation 3.20, followed by the maximum eclipse period in Equation 3.21.

$$\delta_{min} = asin\left(\frac{R_{earth}}{r_{min}}\right) \tag{3.20}$$

$$T_{eclipse,max} = \frac{\delta_{min} T}{\pi} \tag{3.21}$$

Where R_{Earth} is Earth's radius. The maximum time in view can be calculated using the earth central angle Θ_{Earth}, and assuming a value for the minimum elevation angle θ_{min} for communication with ground stations.

$$\Theta_{Earth} = \frac{\pi}{2} - \theta_{min} - asin\left(cos\left(\theta_{min}\right) sin\left(\delta_{min}\right)\right) \tag{3.22}$$

$$T_{view,max} = \frac{\Theta_{Earth} T}{\pi} \tag{3.23}$$

Lastly, an estimation for the air density at the altitude of the satellite is made using SMAD's graph relating these parameters, which is included in Valispace as a dataset. Valispace can interpolate using stepwise interpolation and linear interpolation, the latter of which is used here. Furthermore, Valispace can extrapolate by continuing the last interpolation step, or by giving the same value as the last available point. While these interpolation and extrapolation options work as intended, none of them are accurate for the logarithmic relation between air density and satellite altitude. However, this poses no challenge here as the air density can be assumed zero for all related purposes for altitudes above $900km$. The delta-V budget and geometric properties are then used for subsystem design, as explained in the next section.

3.6 Subsystems

In this section, the inclusion of each of the subsystems in the satellite design in Valispace is described. Note that the payload subsystem is not included here, as its design is considered an input of the subsystem design process and will itself not be designed here. Most subsystems will require the selection of COTS components for which some of the specifications follow from the design in Valispace. The COTS component selection procedure in Valispace is described in Section 3.7. It should be noted that the subsystem specifications do not necessarily follow directly from these design processes, as they depend on the specifications of the selected COTS components.

3.6.1 Attitude Determination and Control System

The ADCS subsystem design process is done in a number of steps. First, the sensor selection is done. Then, the highest disturbance torque must be calculated, after which the Reaction Control System (RCS) thruster component and propellant mass can be determined, and finally the reaction wheels can be sized. Each of these processes are done in separate tree structures within Valispace, with their results visible in the ADCS block.

The sensor selection is done according to the process described in SMAD, where the main driver for the sensor selection is the pointing accuracy requirement, which may come from the payload or other subsystems. This table (Table 11-8, SMAD[16]) is included in Valispace by adding datasets for each of the four potential sensors (horizon sensors, sun sensors, star sensors and gyros) and plotting the relation between pointing accuracy and sensor selection. Stepwise interpolation and last value extrapolation in Valispace allows a reasonable sensor selection for any accuracy requirement.

For the disturbance torque, the largest of the possible disturbance torques for satellites is calculated. These torques are given below, with estimations for each taken from SMAD.

1. Gravity gradient torque (τ_1), a torque caused by the gravitational field of Earth, which introduces a torque if the satellite is not symmetrical.
2. Solar radiation torque (τ_2), a torque caused by the asymmetric pressure of sunlight on the body of the satellite.
3. Magnetic torque (τ_3), a torque caused by the magnetic dipole of the satellite and the magnetic field of Earth.
4. Aerodynamic torque (τ_4), a torque caused by the drag force on the satellite body.

Each of these torques are calculated using values from the characteristics estimation for body dimensions such as length and area, astrodynamic characteristics and estimated values from other sources.

$$\tau_1 = \frac{3\mu}{2r_p^3} \cdot \Delta I \cdot sin(2\delta) \tag{3.24}$$

In this equation, μ is Earth's gravitational parameter, r_p is the perigee radius, ΔI is the difference in moment of inertia between the axis with the highest and the axis with the lowest moment of inertia in the spacecraft, and δ is the maximum deviation of the z-axis from the local vertical.

$$\tau_2 = p_s \cdot d_p \tag{3.25}$$

Here, p_s denotes the solar pressure and d_p is the maximum distance between the center of pressure and the center of mass, which is estimated as a percentage of the satellite linear dimension, as shown in Equation 3.26.

$$d_p = n(\%) \cdot l_{sat} = n(\%) \cdot \rho_{sat} \cdot m_{sat} \tag{3.26}$$

The satellite mass m_{sat} is estimated in the first characteristics estimation, and the satellite density ρ_{sat} and center of pressure distance percentage n are taken empirically.

$$\tau_3 = m \cdot B = m \cdot (1 + sin(i)) \cdot \frac{m_{Earth}}{r_p^3} \tag{3.27}$$

Here, m is the magnetic moment of the satellite, estimated using a linear relation between itself and the system mass, B is the magnetic field strength of Earth near the satellite, i is the orbital inclination, m_{Earth} is Earth's magnetic moment and r_p is the perigee radius.

$$\tau_4 = \frac{1}{2}\rho V^2 c_D d A_{f,max} \tag{3.28}$$

Here, ρ is the air density at the perigee altitude of the satellite, V is the satellite velocity at perigee, c_D is the drag coefficient of the satellite, d is the maximum distance between the center of pressure and the center of mass, and $A_{f,max}$ is the maximum frontal surface area, which is calculated using the satellite volume estimation as shown in Equation 3.29. This equation assumes a cubic satellite structure. For the 2x2x3 U structure used in the DSE, the maximum frontal surface area is 0.072 m³.

$$A_{f,max} = \sqrt{3} \cdot V_{satellite}^{\frac{2}{3}} \tag{3.29}$$

Valispace computes each of these torques, and is then capable of selecting the maximum value using the 'max()' function. Only the largest torque is selected as the disturbance torque, which forms the driver for the RCS thruster and tank design.

The RCS propellant mass is estimated by assuming the maximum disturbance torque is continuously applied to the satellite, and the satellite continuously counters this torque using an RCS thruster. The constant force the RCS thruster must deliver is calculated using Equation 3.30.

$$F = \frac{\tau_{max}}{r} = 2\frac{\tau_{max}}{l} \tag{3.30}$$

The arm length r is assumed to be equal to half the linear satellite dimension l. Then, this constant force can be used to calculate the propellant mass flow.

$$\dot{m} = \frac{F}{I_{sp} \cdot g} \tag{3.31}$$

Where I_{sp} is the specific impulse of the RCS propellant, and g is the gravitational acceleration at Earth's surface. This mass flow is multiplied by the lifetime of the satellite to compute the total RCS propellant mass.

The reaction wheel sizing is done using a slew maneuver duration requirement which is the main driver for the reaction wheel torque. This relation is given in Equation 3.32.

$$\tau = I \cdot \alpha = \frac{m_{sat} l_{sat}^2}{6} \cdot \frac{4\pi}{t_{slew}^2} \tag{3.32}$$

Where m_{sat} and l_{sat} denote the mass and linear dimension of the satellite respectively, and t_{slew} is the duration of a 180-degree slew maneuver. The resulting torque is used to compute the power consumption and mass of the reaction wheels empirically.

The total mass of the ADCS subsystem is estimated as the sum of the sensors, propellant, thruster and reaction wheels. For the thruster mass, an empirical relation is used between the thruster mass and minimum thrust force. For the power consumption, the sensors are taken into account as well as the reaction wheels, while the thruster power is considered negligible due to its infrequent use in practice, which is periodic momentum dumping when the reaction wheels become oversaturated.

3.6.2 Command and Data Handling

The CDH subsystem design recreation is rather simple. The data rate which the CDH system must handle is determined in Equation 3.1. The minimum storage size is determined from the data rate and orbital period, as shown in Equation 3.33 (Section 11.3.2, SMAD[16]).

$$N = MoS \cdot R_{data} \cdot T \tag{3.33}$$

Here, N is the minimum storage size, MoS is a reasonable margin of safety, R_{data} is the CDH receiving data rate and T is the orbital period.

The first CDH mass and power consumption are estimated using empirical relations, after which an iteration is performed where a COTS CDH system is selected using as criteria a processing rate capable of processing data at the given data rate, and a storage size larger than the minimum computed storage size.

3.6.3 Power

The power subsystem concerns the design of a power source, power storage, and power distribution. The power required distribution is known from the first characteristics estimation, from which the power which must be generated can be determined, as shown here.

$$P_{out} = \eta_{PDU} \cdot P_{in} \tag{3.34}$$

Where η_{PDU} is the Power Distribution Unit efficiency. Next, the solar panel sizing is done by first determining the End-of-Life (EoL) solar panel efficiency, using Equation 3.35

$$\eta_{EoL} = \eta_{initial} \cdot (1 - d)^{t_{life}} \tag{3.35}$$

Where η is the solar panel efficiency, d is the yearly solar panel degradation rate and t_{life} is the mission lifetime. The solar power which must be received is then calculated using the power generation requirement and the EoL solar panel efficiency, and combined with the solar flux near Earth to estimate the solar array area.

$$A_{array} = \frac{P_{in}}{\eta_{EoL} G_{SC}} \tag{3.36}$$

From the solar array area, the solar array mass can be estimated using an estimation of the mass per area, which is a design parameter taken from an empirical relation.

The battery sizing is done using the maximum eclipse period as calculated in the astrodynamic characteristics. For the given distributed power, the battery must store enough energy to power the satellite during the duration of an eclipse.

$$E_{storage} = P_{out} \frac{T_{eclipse}}{\eta_{battery} \cdot DoD} \tag{3.37}$$

Where $E_{storage}$ is the battery storage size, $T_{eclipse}$ is the eclipse period, $\eta_{battery}$ is the battery efficiency and DoD is the Depth of Discharge.

An initial mass estimation follows from estimations for the solar panel area density, battery mass density and the PDU mass. After component selection the masses are known more precisely, and a margin is used for cabling.

3.6.4 Propulsion

The propulsion subsystem is designed by estimating propellant mass and volume, and selecting a viable COTS propellant tank and thruster. The propellant mass is calculated using the mission-characteristic delta-V, as shown in Equation 3.38.

$$m_p = \left(e^{\frac{\Delta V}{I_{sp}g}} - 1\right) m_{dry} \tag{3.38}$$

Where m_p is the propellant mass, ΔV is the delta-V, I_{sp} is the specific impulse of the propellant, g is the gravitational constant of Earth and m_{dry} is the dry mass of the spacecraft. The specific impulse of the propellant is a design parameter, for which it is assumed a chemical bi-propellant is selected. This assumption can be changed during design iterations.

From the propellant mass, the propellant volume can be calculated by using the density of the selected propellant. The mass of the propellant is added to the selected COTS components for the thruster and propellant tank. The power consumption follows from the thruster specifications.

3.6.5 Structure

The structure subsystem concerns supporting the other subsystems physically using a skeleton. This skeleton can be designed manually, but for CubeSats many COTS options are available. While these COTS components are tested for typical CubeSat usage, manual verification of these components reduces risk of structural failure.

Maximal structural loads typically occur during launch, where the launcher's manufacturer specifies the maximum axial and lateral acceleration and vibrational frequencies experienced by the satellite during launch. The structure must at least be strong enough to withstand loads due to these accelerations, and be rigid enough such that the natural frequencies are at least as high as the highest occurring vibrational frequency within the launcher.

The maximum stress that will occur in any direction can be calculated using Equation 3.39.

$$\sigma = \frac{P}{A} = \frac{m_{sat} \cdot a_{launch}}{A} \tag{3.39}$$

Where P is the load, A is the smallest cross-sectional area of the structure orthogonal to the acceleration, m_{sat} is the mass of the satellite and a_{launch} is the acceleration experienced during launch.

The buckling load of a column can be calculated using Equation 3.40, to determine whether or not the satellite will buckle due to compression from lateral acceleration experienced during launch.

$$P_{cr} = \frac{\pi^2 EI}{l_e^2} \tag{3.40}$$

Where EI is the flexural stiffness, and l_e^2 is the effective length of the column, which for this case is two times the length of the satellite. The flexural stiffness is a property of the selected material of the structure. The natural frequencies of the satellite can be determined by computing the moments of inertia and spring constants of the satellite structure in all directions as shown in Equation 3.41.

$$\omega_{n,x} = \sqrt{\frac{k}{m_{sat}}} = \sqrt{\frac{3EI_{xx}}{l_e^3 m_{sat}}} = \sqrt{\frac{3E\Sigma w \cdot t^3}{12 l_e^3 m_{sat}}} \tag{3.41}$$

Where $\omega_{n,x}$ is the natural frequency in the x-direction, k is the spring constant, E is Young's modulus, I_{xx} is the moment of inertia about the x-axis, w is the width of each cross-sectional beam and t the thickness, assuming a thin-walled structure. The natural frequencies should all be at least as high as the highest occuring frequency in the launcher.

3.6.6 Thermal

The thermal subsystem concerns creating a heat balance. At all times, the satellite temperature must remain within a range for which an initial upper and lower bound are set, values which are taken from literature.

The thermal subsystem design is done by calculating the lowest temperature in the cold case, and the highest temperature in the warm case. The cold case assumes minimal incoming solar flux (which happens during an eclipse), a low Earth albedo and the minimum frontal surface area exposed to Earth's infrared flux. The warm case assumes the highest frontal surface area for the incoming solar flux and Earth's infrared flux and a high Earth albedo. The incoming heat is then calculated according to Equation 3.42.

$$Q_{in} = \left(\left(T_{Earth,eff}^4 k_B + G_{SC} A_G \right) \cdot \frac{R_{Earth}^2}{r^2} + G_{SC} \right) \cdot A_f \cdot \alpha + Q_{internal} \tag{3.42}$$

Where $T_{Earth,eff}$ is the effective temperature of Earth, k_B is the Boltzmann constant, G_{SC} is the solar flux near Earth, A_G is the geometric albedo of Earth, R_{Earth} is the radius of Earth, r_p is the radius of the satellite, A_f is the frontal surface area of the satellite which depends on the satellite orientation with relation to the sun vector, $alpha$ is the absorptivity of the satellite and $Q_{internal}$ is the heat generated by the satellite itself.

The incoming heat is set equal to the outgoing heat to calculate the equilibrium temperature of the satellite. It is assumed that the satellite cools down via infrared emission only, for which the equation is rewritten such that it can be solved for the equilibrium temperature as shown in Equation 3.43.

$$T_{eq} = \left(\frac{Q_{in}}{k_B A \epsilon} \right)^{1/4} \tag{3.43}$$

Where A is the total surface area of the satellite and ϵ is the emissivity of the satellite exterior.

If the temperature is not within the acceptable range for both the warm case and the cold case, the heat must be generated or dispersed using heating or radiating components respectively to bring the temperature to within the acceptable range. Equation 3.43 is rewritten to compute the heat differential required to bring the equilibrium temperature to the acceptable range, and the heat differential computed is used to select COTS heaters and radiators.

3.6.7 Telemetry, Tracking & Command

The TTNC subsystem requires the generation of a link budget, which allows the computation of the required transmission power for a certain Signal-to-Noise Ratio (SNR). Values for noise temperatures, antenna efficiencies, attenuation loss and line losses are taken from literature, while the receiver diameter and uplink and downlink frequencies form design parameters.

The receiver antenna gain can then be calculated using Equation 3.44, which combined with the free space path loss from Equation 3.45 and the losses and efficiencies can be combined to calculate the required transmission power as shown in Equation 3.46.

$$G_R = 6 \left(\frac{D_r}{\lambda} \right)^2 \tag{3.44}$$

Where G_R is the receiver antenna gain, D_r is the receiver antenna diameter and λ is the wavelength of the signal.

$$L_{FSP} = \left(\frac{\lambda}{4\pi(r - R_{Earth})} \right)^2 \tag{3.45}$$

Where L_{FSP} is the free space path loss, r is the radius of the satellite and R_{Earth} is Earth's radius.

$$P_t = \frac{SNR \cdot k_B \cdot T_N \cdot R_{data}}{L_{attenuation} \cdot L_{t,line} \cdot G_T \cdot L_{FSP} \cdot G_R \cdot L_{r,line}} \tag{3.46}$$

Where P_t is the transmission power, SNR is the signal-to-noise ratio, k_B is the Boltzmann constant, T_N is the effective noise temperature, R_{data} is the data transmission rate, $L_{attenuation}$ is attenuation loss, $L_{t,line}$ is the transmitter line loss, G_T is the transmitter antenna gain and $L_{r,line}$ is the receiver line loss. The transmission power requirement can be used alongside the frequency band selection, which has to match a band supported by the ground station.

An initial mass estimation for this system follows from SMAD, as a percentage of the initial full system mass estimation. The DSE approach selected COTS antennae first and verified their usage using the relations above. Hence, these components will be selected identically and requirements are used to verify their use.

3.7 COTS component selection

After determining the requirements of each of the subsystems during the subsystem design, components must be selected to fulfill each of these requirements. Commercial Off-The-Shelf components are preferred because they are readily designed and tested. To select COTS components during a design process, an engineering team must determine some sort of protocol. Such a protocol could be manually selecting components to fulfill the requirements, writing software to query the most efficient component and implementing it automatically, or anywhere in between. In Valispace, some protocol needs to be determined as well, which leads to the question of how well Valispace is suited for COTS component selection.

Component selection within Valispace can be challenging. The challenge comes forth from the fact that the user cannot fully design custom components, hence the design choices and performance requirements affect the availability of COTS components.

Ideally, Valispace would select the components itself, which is possible but not practical. The user can write a script in Valispace which connects Valispace to an external COTS component database. Such a script can indeed select a component and update the relevant component parameters by querying the database for the 'optimal' COTS component which fulfills all requirements (where the user must define what makes such a component optimal, perhaps the least mass, lowest price, availability, TRL or a combination of these parameters). While automation does exist in Valispace which would allow the user to re-run the script whenever the performance requirements are updated, but this would create a loop where the design parameters and performance parameters continuously update each other.

A more simple version of a script would allow the user to find a fitting component by creating an ordered list of all COTS components which fulfill the requirements and having the user select the component. The component selection can be linked to all relevant design parameters.

In practice, manual entry of the design parameters is simpler. Searching for COTS components manually such that the performance requirements are fulfilled in external databases is a relatively simple process, while scripting requires additional verification.

4

Satellite design from Valispace implementation

As described in Chapter 3, the implementation of Valispace has been used to recreate the satellite designed during the DSE, which yields results that require verification. During the DSE, the book Space Mission Analysis and Design[16] was used during the detailed design phase, and its models can be reused to verify the design as they should yield identical results. In this section the tool created will be described and its limitations will be discussed, the DSE satellite designed in Valispace will be summarized and the verification procedure will be shown.

4.1 Tool description and limitations

The tool was created to support the DSE design, but is created flexibly such that it is possible to apply the tool to other, similar space missions where the payload is modeled as a black box. The tool is limited by the following assumptions:

- The system is an unmanned small satellite in the range of 1-1000kg, which is put into its injection orbit by a selected launcher. From this injection orbit it performs a Hohmann maneuver to enter its mission orbit.
- The satellite orbits Earth during its operation, and the moon and other celestial bodies do not affect the satellite.
- The mission orbit has a small eccentricity, such that the semi-major axis is a good average for computation of effects such as aerodynamic drag and magnetic torque.
- Earth is assumed to be a point mass to determine the orbital trajectory.
- The satellite uses reaction wheels and RCS thrusters for attitude control.
- The satellite uses deployable solar panels for energy generation, a battery for storage and a PCDU for power conditioning and distribution.
- The satellite uses one type of propellant for all propulsive maneuvers.
- The satellite uses radiators for cooling and heaters for heating if necessary. The necessity of these is determined automatically by the tool.
- The payload is fully defined by its pointing accuracy requirement, mass and power consumption.

The sensor selection for the ADCS subsystem is made flexible by assuming the quantity of each of the sensor types depends on the pointing accuracy requirement. Otherwise, the component types are limited to the ones listed above. Components which are not named are only defined by their mass and power consumption, and other performance-indicative parameters, but are otherwise flexible. For example, the user may select a COTS antenna for any bandwidth, as long as its properties are known.

4.2 Design summary

The design is summarized using the analysis module of the tool. This allows easily visualization of the system budgets. The selected COTS components are tracked using the description fields on valis in the system design module instead. The resulting design is a satellite with volume of 0.0117m³ (approximately 12U), a dry mass of 18.81kg and a propellant mass of 3.38kg. The total power budget is 40.62W. The ADCS subsystem comprises two RCS thrusters, four reaction wheels, one gyro, six sun sensors, two star sensors and one horizon sensor. It uses a CDH component with a memory size of 1.22GB, six solar panels with 0.04m² area and a battery which can store 92.2Wh. It uses a bipropellant with a thruster with a thrust force of 0.22N at end-of-life. It uses active heating for thermal control while eclipsed, and no active cooling. It communicates using an s-band and a Ultra-High Frequency (UHF) antenna for both the uplink and downlink. For all design choices, the chosen options are identical to those chosen during the DSE. The summaries for the mass and power consumption system budgets and the selected components can be found in Figure 4.1, Figure 4.2 and Table 4.1, respectively.

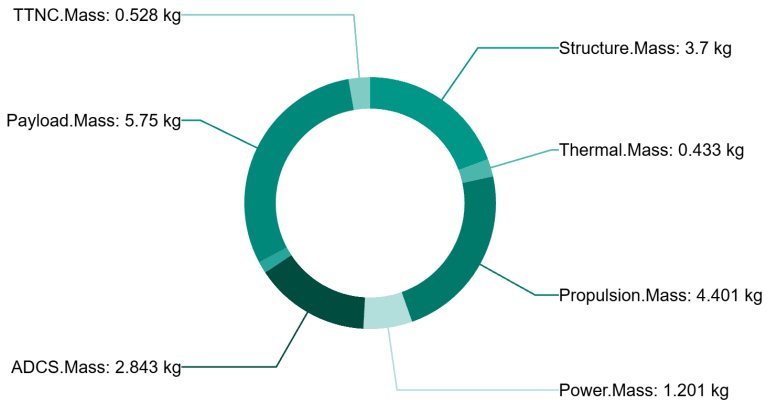

Figure 4.1: The mass budget of the designed satellite, from the analyses module in Valispace.

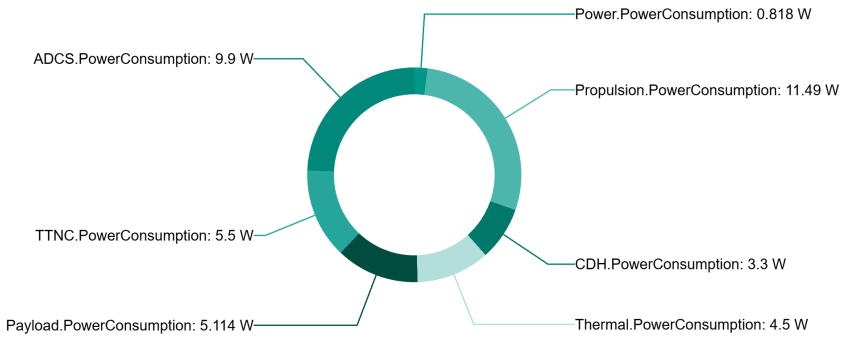

Figure 4.2: The power consumption budget of the designed satellite, from the analyses module in Valispace.

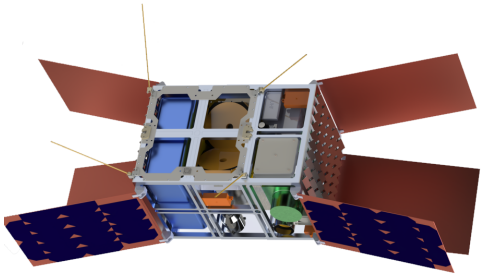

Figure 4.3: A sketch of the satellite topology. Adapted from the DSE final report[7]

.

Section	Component	Model
ADCS	6 Sun sensors	BiSon64-ET, LENS R&D[32]
	2 Star sensors	ST200, Hyperion Technologies B.V[33]
	1 Earth sensor	CubeSense N, CubeSense N[34]
	1 Gyro	STIM 300, Sensonor[35]
	1 Reaction wheels	RWP100, Blue Canyon Technologies Inc.[36]
	3 Reaction wheels	RWP050, Blue Canyon Technologies Inc.[37]
	2 RCS thrusters	X14029003-1X, VACCO[38]
CDH	1 OBC	OBC-P3, Space Inventor[39]
Power	96 Solar cells	QJ Solar Cell 4G32C, AZUR SPACE[40]
	1 PCDU	Starbuck-NANO Plus, AAC Clyde Space[41]
	1 Battery	ICP-20, O.C.E. Technology[42]
Propulsion	1 Thruster	EPSS C1, NanoAvionics[43]
	Propellant	LMP-103S, ECAPS[44]
	1 Propellant tank	Custom Ti-6Al-4V tank
Structure	1 Skeleton structure	12U aluminum 6082-T6 structure, ISIS[45]
	6 Solar panels	Custom AlBeMet sheets and mechanism
	2 RCS thruster supports	Custom support and mechanism
	5 radiation shielding panels	Custom panels on outside skeleton
Thermal	1 Heater	STOCK Flexible Heater, Zoppas Industries[46]
	Louvers	Form Factor Thermal Control Louvers, NASA[47]
	Coating	Aluminised kapton foil (2mm), goodfellow[48]
TTNC	1 UHF antenna	UHF antenna, Nanoavionics[49]
	1 UHF Transceiver	UHF transceiver, Nanoavionics[50]
	1 S-band antenna	S-Band patch antenna, GOMspace[51]
	2 S-band transceivers	S-Band transceiver, Satlab[52]

Table 4.1: Summary of the components per subsystem as selected during the DSE and in Valispace. Adapted from the DSE final report[7].

The mission consists of six of these satellites, equidistant in a geostationary orbit. The system measures gravitational waves in the range of 1-10 Hz. Because of the redundancy in these systems, they can perform realignment maneuvers without interrupting measurements, leading to an availability of 100%. Their astrodynamic characteristics and performance-indicative parameters can be found in Appendix A.

The resulting design has no topology description, as Valispace is not practical for this purpose (i.e. it does not support 3D modelling). A topology of the satellite from the DSE is given in Figure 4.3. It should be noted that the linear dimension computation, which is done to estimate other subsystem parameters, assumes a cubic satellite, which is not realistic for a volume of 12U where an engineer would prefer a COTS CubeSat structure compliant with the CubeSat standard, for example with dimensions 20 cm x 20 cm x 30 cm.

The subsystem component selection was done using requirements, which came forth from parameters calculated within Valispace, which are specific per subsystem. These will be discussed alongside the verification done using SMAD in Section 4.3.

4.3 Verification process

The verification process consists of two parts. First, the satellite design from the Valispace implementation must be identical to the one designed during the DSE. Second, the models used within Valispace

were tested using the source of the models, SMAD, and some available computational tools for more complex computations.

The verification with relation to the DSE design should yield identical results. This verification process formed a checklist of parameter values, COTS components, and system budgets, which are documented in Appendix A.

Verification using SMAD was done using the calculation examples given throughout the book. SMAD shows the design process of a mission referred to as FireSat. One way of verifying the tool would be to change the inputs to each of the subsystems and determine if the performance-indicative parameters are identical to the examples given. Alternatively, the whole system can be copied and renamed to FireSat before changing these values, such that the original system remains intact. Valispace can copy the system design tree or parts of it while still using the same external dependencies such as constants and inputs. Any internal dependencies are kept internal within the copy, such that the copy does not depend on the original system at all. The copy was renamed 'FireSat', and unit and system testing could be done without risks to the original system.

A verification criterion was set up to verify each parameter in the checklists given below. For each of these, the tested parameter is deemed acceptable if it is within $\pm 2\%$ of the value in SMAD.

4.3.1 The delta-V budget

SMAD section 7.3 concerns the delta-V budget and its dependent parameters. Although the DSE omits altitude maintenance due to the high orbital altitude, this cannot be done for the FireSat example as its mission altitude is only 700 km. Furthermore, its injection orbit is a circular orbit with an altitude of 150 km. The delta-V budget verification is shown in Table 4.2. It includes estimations for the ballistic coefficient and maximum density, as these form inputs for computing the altitude maintenance delta-V.

Maneuver	Value [unit]	OK
Transfer 1st burn	156 [m/s]	✓
Transfer 2nd burn	153 [m/s]	✓
Ballistic coefficient	25 [kg/m²]	✓
Maximum density	2.73e-13 [kg/m³]	✓
Altitude maintenance	19 [m/s]	✓
EoL maneuver	198 [m/s]	✓
Total delta-V	526 [m/s]	✓

Table 4.2: Delta-V budget verification of the tool using the FireSat example from SMAD

It should be noted that the FireSat example assumes that margins are not introduced here, but rather in the propellant budget. ESA guidelines suggest margins should be introduced to the delta-V budget, but for the verification process, they will not be introduced.

4.3.2 Astrodynamic characteristics

Starting at page 977 of SMAD is a section named *"Explanation of Earth Satellite Parameters"*. Here, the computation methods for the used astrodynamic characteristics are given. These equations are solved for the FireSat example, and yield identical results. The astrodynamic characteristic verification is shown in Table 4.3.

4.3.3 Subsystem design

In section 11 of SMAD, the subsystem design procedure for each subsystem is given, along with FireSat examples. Here, each of the performance-indicative parameters of the subsystems will be verified using the FireSat examples. The component selection procedure is not included here because it is does not verify the models used, and identical performance-indicative parameters should yield nearly identical components depending on the chosen trade-off procedure. The results from the verification of the FireSat examples in Valispace are given in Table 4.4. It should be noted that the TTNC subsystem design during the DSE was not done using the SMAD models, and instead an external tool[53] for link budget calculations was used for the FireSat example.

Characteristic	Value [unit]	OK
Minimum radius	7071 [km]	✓
Maximum radius	7071 [km]	✓
Orbital period	5917 [s]	✓
Average orbital velocity	7508 [m/s]	✓
Minimum angular radius of Earth	1.122 [rad]	✓
Maximum eclipse period	2114 [s]	✓
Earth central angle	0.3048 [rad]	✓
Maximum time in view	574.1 [s]	✓

Table 4.3: Astrodynamic characteristic verification of the tool using the FireSat example from SMAD

Section	Parameter	Value [unit]	OK	Section
ADCS	Gravity-gradient torque	1.8e-6 [Nm]	✓	11.1
	Solar radiation torque	6.6e-6 [Nm]	✓	
	Magnetic torque	4.5e-5 [Nm]	✓	
	Aerodynamic torque	3.4e-6 [Nm]	✓	
	avg. RCS propellant mass flow	1.54e-8 [kg/s]	✓	
	min. RCS propellant mass	2.43 [kg]	✓	
	min. reaction wheel torque	0.52 [Nm]	✓	
CDH	Data rate	1.5e+8 [bit/s]	✓	11.3
	Data per orbit	8.876e+11 [bit]	✓	
	Min. storage size	110.95 [GB]	✓	
Power	Power generation	110 [W]	✓	11.4
	Solar cell efficiency at EoL	82.6 [%]	✓	
	Solar array size	1.9 [m²]	✓	
	Battery size	357 [Wh]	✓	
Propulsion	Propellant mass	34 [kg]	✓	10.6
	Propellant volume	37.78 [dm³]	✓	
Structure	Occupied volume	1.170 [m³]	✓	11.6
	Max. frontal area	1.923 [m²]	✓	
	Min. frontal area	1.110 [m²]	✓	
	Moment of inertia	17.09[kgm²]	✓	
Thermal	Heat cold case	391 [W]	✓	11.5
	Heat warm case	852 [W]	✓	
	Min. passive equilibrium temp.	-30.77 [° C]	✓	
	Max. passive equilibrium temp.	21.32 [° C]	✓	
	Min. controllable equilibrium temp.	10 [° C]	✓	
	Max. controllable equilibrium temp.	21.32 [° C]	✓	
TTNC	S-band receiver power	17.5 [W]	✓	11.2
	S-band transmitter power	40 [W]	✓	

Table 4.4: Verification of the performance-indicative parameters of the tool using the FireSat example from SMAD, or from an external tool for the TTNC subsystem. The number are taken from the indications sections in SMAD[16].

5

Experimentation methodology

In Chapter 3, the methodology for using Valispace to recreate the DSE design was described. Now that the design has been recreated successfully, some experimentation can be done using features in Valispace. First, the usage of the margins feature is explained, followed by experimentation with the sensitivity analysis feature. Finally, the tool is used to trade off generated design concepts to determine if this process can be improved upon using Valispace.

5.1 Requirements

The system requirements can be added to Valispace to track their status throughout the system design process. The twelve EPS-specific requirements from Table 2.2 were added to the Valispace workspace. Valispace supports a number of different verification methods, including analysis, inspection, review, rules, and test.

Analysis, inspection and review cannot be automated, and require the user to manually change the verification status of a requirement whenever it changes. Many of the stakeholder and system requirements require manual verification. Out of the twelve EPS requirements, seven require manual verification. Manual tracking of verification status is expected to be as effective in Valispace as in any other documentation tool.

The potential strength of Valispace lies in the rules and test verification methods. Using these the verification status can automatically be updated depending on the connected rules and tests. The 'rules' used here are boolean comparisons between a system property and a requirement vali. If an upper limit is set on a system property, the boolean statement would state that the system property must be smaller than the requirement vali, while the inverse is true for a lower limit. The five requirements of the EPS subsystem for which verification can be automated using rules are listed here:

- *LICCA-SYS-Sub-EPS-1: The total required electrical power shall not exceed 40W*. This requirement is connected to the power consumption of the system.
- *LICCA-SYS-Sub-EPS-3 The EPS shall deliver 30.60 W during daytime*. This requirement is connected to the power output of the EPS subsystem.
- *LICCA-SYS-Sub-EPS-3 The EPS shall deliver 31.0 W during eclipse*. This requirement is connected to the power output of the EPS subsystem.
- *LICCA-SYS-Sub-EPS-5 The EPS shall be able to store 90 Wh of energy*. This requirement is connected to the battery storage size of the EPS subsystem.
- *LICCA-SYS-Sub-EPS-11: The power source of the EPS should have an efficiency of at least 31.8%*. This requirement is connected to the efficiency of the selected solar cell component at end-of-life (i.e. accounting for degradation).

Tests in Valispace are simulations of the system over a period of time, where the user can change the state of the system at any point in the simulation. One test case will be created to evaluate Valispace in this regard: the depletion of the battery while eclipsed. For this test case it is assumed the battery depletes at a rate equal to the power consumption of the system, and does so for the duration of the

maximum eclipse period of the satellite. At the end of the test, the battery charge remaining must be at least zero.

5.2 Margins

During any satellite design process, margins are introduced in various aspects of the design to account for uncertainties in the earlier phases. While margins can be introduced as separate parameters in any computation tool, Valispace supports margins directly on valis. The effectiveness of these margins will be investigated by applying them on the DSE case and analyzing the impact. During the DSE, margins were applied using the ESA margin philosophy for science assessment studies[54]. This philosophy introduces the following margins on components:

- 5% margin on the mass and power consumption of COTS components
- 10% margin on the mass and power consumption of COTS components which require modification
- 20% margin on the mass and power consumption of custom components

Furthermore, ESA specifies some specific margins to be applied[54]:

- 20% margin on the dry mass at launch
- 20% margin on the power consumption requirement
- 2% margin on the propellant
- 5% margin on the delta-V of precisely calculated maneuvers
- 100% margin on the delta-V of orbital maintenance
- 100% margin on the delta-V of attitude maneuvers and momentum dumping maneuvers

Margins are typically applied as separate variables which are taken into account when computing the next dependent variable. In Valispace, margins can be applied to valis as percentages, which do not affect the computed values directly. The margins can be entered separately for the upper bound and the lower bound. These margins are then added to the computed margins, which are computed from the margins of the valis upon which the vali is dependent. Valispace separately displays the upper and lower bound of the vali, referred to as the 'worst case' values.

Valispace propagates margins proportionally. That is to say, if there is a vali A and vali B and these are summed to yield vali C, the following margin for C would be computed:

$$margin_C = \frac{value_A \cdot margin_A + value_B \cdot margin_B}{value_A + value_B} \tag{5.1}$$

The process for subtraction is done similarly. In the case two valis are multiplied or divided to compute a third vali, the margins of the third vali are calculated by multiplying the margins of the two valis.

Margins will be implemented for the design identically to how they were implemented during the DSE. Implementation of and experimentation with these margins will show the usefulness of how margins work in Valispace. Results from this experimentation are given in Section 6.2.

5.3 Sensitivity analysis

Sensitivity analyses are done after a design is created to determine the impact of changes inputs to the design on the outputs. These changes could be changing the mission requirements, making different assumptions, changing component selection and selecting different models. A sensitivity analysis has been performed using the DSE design to determine the effect of utilizing Valispace for sensitivity analyses on project performance.

5.3.1 Analysis set-up

The design of the satellite in Valispace is done using a couple of **inputs**, which are formed by the payload, the required pointing accuracy, the injection orbit and the mission orbit. The payload is represented as a grey-mass model which is defined only by its mass, power consumption rate and data generation rate. Furthermore, the launcher injection orbit and the mission orbit must be set in terms of

Keplerian orbital elements. Besides the inputs, a number of parameters are assumed for the computation of the satellite design. These parameters are referred to as the **design parameters**, which are set to be identical to those of the DSE mission, and may be altered to determine their impact.

During the DSE project, a sensitivity analysis was done in the final report. In this analysis, the best and worst case scenarios were described for the payload and each of the subsystems, along with the impact of these cases on the final design. A similar sensitivity analysis can be done here by assuming the same worst and best case scenario values for each of these subsystems, and letting Valispace determine the impact of these scenarios.

The sensitivity analysis cases done here make use of the sensitivity analysis feature in Valispace. The user first selects a vali (A) for inspection. Then, the user selects another vali (B) which is a dependency of vali A. Valispace automatically determines all valis required to directly or indirectly compute vali A, and allows the user to select one of them. The user then selects a range for vali B, and the resulting values of vali A are calculated for values throughout the range of vali B. Valispace automatically selects an appropriate step size for values within the range to limit the computation time. Valispace then generates a graph which displays the range of values for vali B on the x-axis and the resulting values for vali A on the y-axis. This feature only works for one-to-one relations between valis, and other relations require more creativity to analyze.

5.3.2 Selected sensitivity analysis cases

To gain insight into the sensitivity analysis process while utilizing Valispace, four cases will be studied. These four cases are:

1. A payload mass increase of +10.9%, the worst case scenario identified during the DSE. It is expected to cause a snowball effect on the system mass.
2. An increase in the injection inclination angle from 0° to 3°, for an injection orbit with a perigee lowered by 10% and an apogee increased by 10%. It is expected to increase the delta-V budget.
3. Thruster selection, with a mass increase from 3.7 to 3.84 kg, and a decreased power consumption, from 11.49 W to 10 W. It is expected than the increased mass will increase the propulsion subsystem mass with some snowball effect, and the decreased power consumption should similarly decrease the subsystem power consumption.
4. Structure subsystem mass increase of +48.1%, from 3.7 kg to 5.48 kg, the worst case scenario identified during the DSE. It is expected to cause a snowball effect on the system mass.

In each of these cases, their impact on the total satellite mass will be measured. These four cases are described in the final report of the DSE project as worst-case scenarios, and come forth from uncertainties in the payload components and the delta-V budget. They are selected because they are fundamentally different: the payload mass is an input to the system on which the whole system will depend, thus the mass increase will cause a snowball effect. The inclination angle uncertainty is a risk to the delta-V budget with an uncertain impact, which may require an analysis. The thruster selection affects design parameters which come from the COTS component selection process, showing the impact of COTS component selection. The structure subsystem mass is an intermediate result which is treated as a margin. Because of these fundamental differences between these cases, the sensitivity analysis process requires a different approach for each of these in Valispace. These different approaches are described in the next section.

5.3.3 Valispace utilization

For the first and second cases the sensitivity analysis feature of Valispace can be used. This feature measures the impact of one vali on another vali and is hence only useful for one-to-one sensitivity analyses. It shows the hypothetical impact to a vali of changing another vali but will not affect any vali or requirement verification status. The user may select any of the Valis in their system design and select another vali on which it is dependent. The dependency selection menu is generated automatically, allowing the user to gain insight into what affects a vali and what does not. Valispace then generates a graph and a relation between the two parameters. This sensitivity analysis flow is visualized in Figure 5.1.

For the third case, a different method is used. The thruster selection affects more than one parameter, hence this does not form a one-to-one relation and Valispace's integrated sensitivity analysis

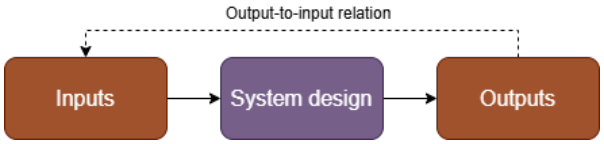

Figure 5.1: The sensitivity analysis flow for determining the impact of inputs on outputs.

feature cannot be used. Instead, the user may simply alter the affected values and determine the impact, although that would require manually changing the values many times to visualize the impact of thruster selection on the total satellite mass. Instead, each of the affected design parameter valis (here thruster mass and power consumption) can be used to find the formulaic relation between that parameter and the total satellite mass. For small changes, the slopes of these relations can be used to visualize the impact of multiple parameters changing. The slopes can be calculated at the extremes of this visualization to determine their accuracy. This sensitivity analysis flow is visualized in Figure 5.2.

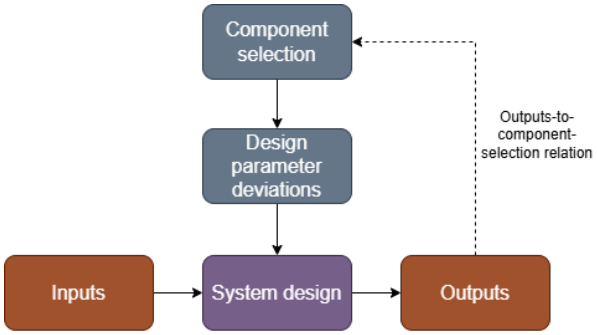

Figure 5.2: The sensitivity analysis flow for determining the impact of component selection on outputs.

Lastly, the fourth case assumes a larger margin for the structure subsystem, which is an intermediate result when computing the satellite mass (hence neither an input nor a design parameter). For this case, it can be assumed that the mass increase will snowball over design iterations, which would take a large time to perform manually. Instead, in Valispace, the user may account for the structure subsystem mass increase by proportionally increasing the initial mass estimation. This is done by adding a new parameter, the 'structure subsystem mass deviation'. The sensitivity analysis is then done by measuring the impact of this parameter on the satellite mass. This sensitivity analysis flow is visualized in Figure 5.3.

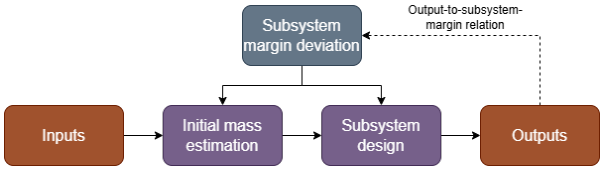

Figure 5.3: The sensitivity analysis flow for determining the impact of subsystem margins on outputs.

5.4 Concept trade-off and payload design

As discussed in Section 2.3.4, the payload design was done during the mission design phase, and the implementation and verification of the system design in Valispace are done using the mass and power consumption of the payload subsystem known in advance as inputs. In this section, the possibility of changing the selected concept from the concept trade-off phase, as discussed in Section 2.3.2, is investigated. This requires a payload redesign and changes to the constellation. The methods used to estimate these are given below.

Each of the satellites in the 8 different design concepts supports one or two laser interferometers, either actively by using two lasers in their payload, passively by using transponders to reflect incoming lasers, or both. The design of the payload is determined by how many lasers are supported actively and/or passively. The number of outgoing lasers is referred to as the number of active connections, and the number of incoming lasers is referred to as the number of passive connections. The active satellite payload set-up, which always supports two outgoing lasers, consists of two piezoelectric stages, a laser, three fiber-optic couplers, three diodes, and a phase modulator. The passive payload set-up only requires one transponder laser per incoming laser. For reference, a full list of components is given in Appendix D. All satellites require telescopes, where each telescope can support up to one active and one passive connection. Margins, as described in Section 5.2, are accounted for in these set-ups.

Each unique satellite is given a label to differentiate them for each concept. These labels are shown in Figure 5.4. The payload was then redesigned for each of the satellites depending on the number of active and passive satellites required. Not all concepts have all satellites in a geostationary orbit, but rather all concepts have an equal laser path length, such that the distance between the satellites which are connected is always $(R_e + h_{geo})\sqrt{3}$, where R_e is Earth's radius and h_{geo} is the altitude for a geostationary orbit. This means concepts 3, 5 and 6 are not in a geostationary orbit, and the rest are. Table 5.1 shows the difference in payload mass and power consumption and mission orbit altitude for each of the satellites in the design concepts considered during the concept generation and trade-off phases. As expected, the payload design and altitude for satellite 7 are identical to the DSE payload design. The other satellites vary significantly in mass, power, and mission altitude.

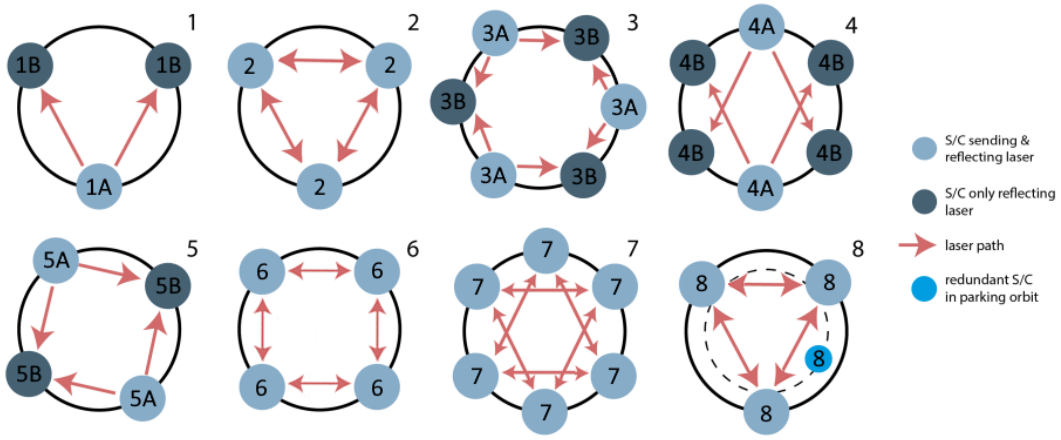

Figure 5.4: The design concepts considered during the DSE, with each unique satellite labeled.

The impact of these changes can be tested in Valispace using the sensitivity analysis tool, but the goal of finding these changes is not merely to determine how well the sensitivity analysis tool in Valispace works. The payload designs can also be used to compare the resulting satellite designs and determine how well the concept trade-off phase was performed, and if potential improvements to the DSE can be identified. The following criteria were used to do the trade-offs:

- Reliability (and risk)
- Cost
- Delta-V
- Sustainability
- Quality of measurements

Risk is used as a criterion instead of reliability for the detailed trade-off. Possible risks include satellite and payload failure, transportation damage and assembly failure, the latter two of which are not considered in the Valispace implementation. The risk of payload and satellite failure is directly linked to satellite reliability. Hence, reliability will be used for the entire trade-off procedure, and risk is omitted.

While the reliability of the satellites, the sustainability and the quality of measurements are not quantified during the DSE project, the cost and delta-V are. Hence, they can be re-evaluated for the other

Satellite	Sent lasers	Reflected lasers	Mass [kg]	Power [W]	Mission altitude [km]
1A	2	0	5.30	5.00	35786
1B	0	1	1.42	2.56	35786
2	2	2	5.75	5.11	35786
3A	2	0	5.30	4.97	66647
3B	0	2	2.84	2.60	66647
4A	2	0	5.30	4.99	35786
4B	0	1	1.4	2.56	35786
5A	2	0	5.30	4.97	53248
5B	0	2	2.84	2.60	53248
6	2	2	5.75	5.11	53248
7	2	2	5.75	5.11	35786
8	2	2	5.75	5.11	35786

Table 5.1: The payload design mass and power consumption and mission orbit altitude for each of the satellites in the considered design concepts

concepts to do a quantitative comparison, rather than just a qualitative one.

If comparing satellites using payloads and mission orbit altitudes from different concepts proves especially easy, the strength of Valispace could prove that the design concept trade-off phase could perhaps be omitted entirely and Valispace could instead be used retroactively to determine the best concept.

5.5 Necessity of taxi vehicle in DSE design

One of the recommendations of the DSE mentions the possibility to redesign the satellite such that it can perform the maneuver from GTO to GEO itself, rather than the selected option of using a taxi vehicle for this purpose. It is mentioned that this is unfeasible for a 12 U CubeSat, but perhaps a 16 U CubeSat could perform such a task.

The nominal satellite design has an occupied volume of 10.9 U including propellant, therefore an additional 5.1 U of volume would be available for additional propellant if no other design changes were implemented. However, the increased propellant tank mass and structure subsystem mass would cause a snowball effect to the overall design, hence this redesign is more complex than simply adding more propellant to the design. For this reason it was omitted from the DSE project.

In Valispace such a redesign is easier to perform. The GTO and GEO orbit parameters are direct inputs to the system and are known from the midterm report. They are given in Table 5.2. The GEO orbit forms the mission orbit, hence no changes are made to these inputs, but the GTO orbit changes will propagate to the delta-V budget, increasing the propellant mass. Furthermore, a new structure has to be selected depending on the volume estimation, which will further cause a snowball effect on the mass.

Characteristic	GTO	GEO
Perigee altitude	250 km	35786 km
Apogee altitude	35786 km	35786 km
Inclination angle	6 deg	0 deg

Table 5.2: The orbital characteristics of the GTO and GEO orbits, which form inputs in the Valispace design

Lastly, should the volume budget exceed the 16 U, a different propellant could be selected. A

propellant with a higher mass density and/or a higher specific impulse will reduce the total occupied volume of the system. The sensitivity analysis feature will be used as it could prove useful here. The dependency graph from the sensitivity analysis will be used to relate volume to either the mass density or the specific impulse and determine the required value to reduce the volume to 16 U.

6 Results

The results come from recreating the design, as described in Chapter 3, and experimenting with features within Valispace, as described in Chapter 5. In this chapter, these results are described per feature that was experimented with. First, the results from the implementation of requirements will be discussed, followed by the use of margins and by the sensitivity analysis cases. Then, the concept trade-off is described, as well as the necessity of a taxi vehicle. Finally, some bugs and flaws of Valispace are described.

6.1 Requirements

The requirements of the EPS subsystem were added to the Valispace workspace. These requirements are listed in Table 6.1, including the selected verification method(s). The implementation of the first five of these are shown in Figure 6.1. The text column gives a description of the requirement, and Valispace automatically determines which valis can be generated based off of these descriptions. Notice, for example, how the value of 40.00 W in LICCA-SYS-Sub-EPS-1 is highlighted in blue. The verification status gives a quick overview of whether the requirement is verified, which green meaning all verification methods pass, yellow means they pass partially, and red means none pass.

Identifier	Text	Verification Status	Verification Methods
LICCA-SYS-Sub-EPS-01	The total required electrical power shall not exceed 40.00 W	0 / 1	Rules
LICCA-SYS-Sub-EPS-02	The EPS shall be able to receive data from the CDH.	1 / 1	Inspection
LICCA-SYS-Sub-EPS-03	The EPS shall deliver 30.60 W during daytime.	1 / 1	Rules
LICCA-SYS-Sub-EPS-04	The EPS shall deliver 31.00 W during eclipse.	1 / 1	Rules
LICCA-SYS-Sub-EPS-05	The EPS shall be able to store 90.00 Wh of energy.	1 / 2	Rules Test

Figure 6.1: The implementation of five of the EPS subsystem requirements in Valispace.

As can be seen in Table 2.2, the selected methods include inspection, rules and test. Analysis and review were not selected, as they are functionally similar to inspection in that they require manual

Requirement ID	User Requirement	Verification method(s)
LICCA-SYS-Sub-EPS-1	The total required electrical power shall not exceed 40W.	Rules
LICCA-SYS-Sub-EPS-2	The EPS shall be able to receive data from the CDH	Inspection
LICCA-SYS-Sub-EPS-3	The EPS shall deliver 30.6 W during daytime.	Rules
LICCA-SYS-Sub-EPS-4	The EPS shall deliver 31.0 W during eclipse.	Rules
LICCA-SYS-Sub-EPS-5	The EPS shall be able to store 90 Wh of energy.	Rules, test
LICCA-SYS-Sub-EPS-6	The EPS shall provide the right amount of power to each subsystem.	Inspection
LICCA-SYS-Sub-EPS-7	The EPS shall adjust the voltage for each subsystem.	Inspection
LICCA-SYS-Sub-EPS-8	The EPS shall provide the correct current type to each subsystem.	Inspection
LICCA-SYS-Sub-EPS-9	The EPS shall protect the system from power spikes.	Inspection
LICCA-SYS-Sub-EPS-10	The EPS shall provide data about power regulation to the CDH subsystem.	Inspection
LICCA-SYS-Sub-EPS-11	The power source of the EPS should have an efficiency of at least 31.8 %.	Rules
LICCA-SYS-Sub-EPS-12	The EPS shall have a TRL of 7 or higher.	Inspection

Table 6.1: System requirement for the EPS subsystem, adapted from LICCA final report, Table 9.1[7].

verification. Inspection is selected for all requirement which do not have a value that can be compared to or inferred from the system design tree. These require manual verification, done by simply marking the requirement as verified. The analysis method also requires the user to attach a file, but this file is not read by Valispace and serves only for user reference.

For requirements which include a value which can be compared to a vali in the system design tree, rules can be selected. Rules make use of a closeout reference, where the user can add one or more rules. One such rule is visualized in Figure 6.2. Here, LICCA-SYS-Sub-EPS-5 is verified automatically by comparing the generated verification vali 'energy_storage_capacity' to the system design tree vali 'BatterySize'. The values for these valis are added below in blue, such that the user can quickly infer the outcome before committing to creating the rule.

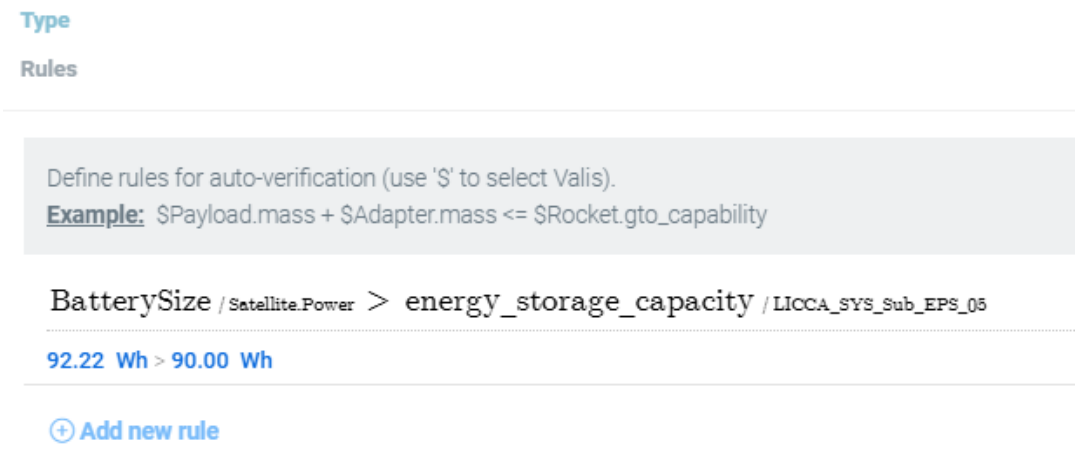

Figure 6.2: A closeout reference in Valispace, used to automatically verify a requirement.

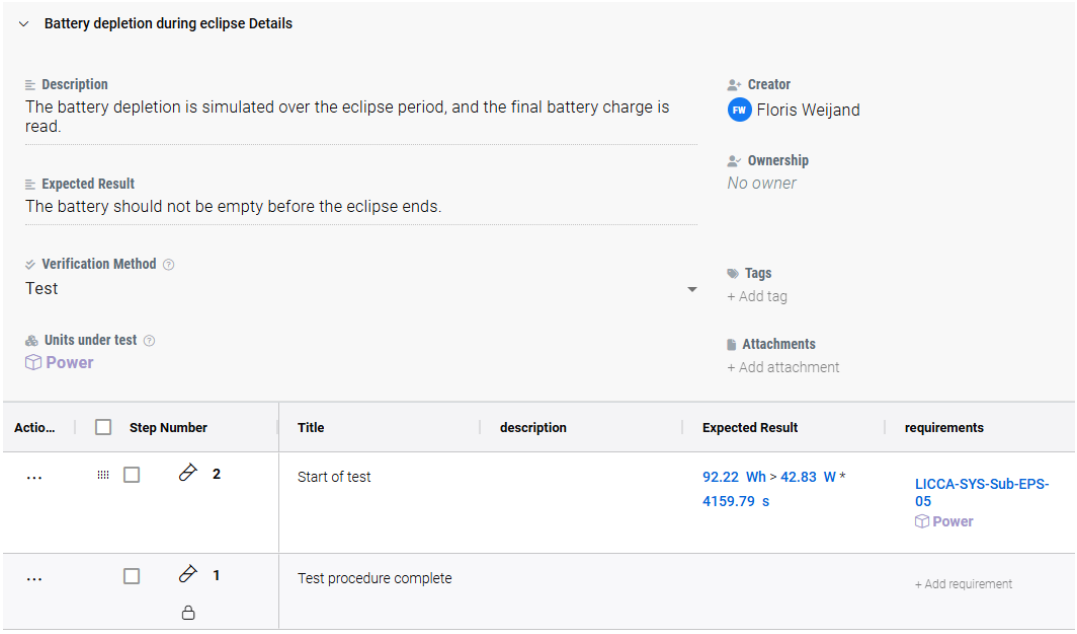

Figure 6.3: A test of the EPS battery storage size during an eclipse in Valispace. The first test step result is used to verify requirement LICCA-SYS-Sub-EPS-5.

Lastly, tests can be used for verification, done here only for LICCA-SYS-Sub-EPS-5. A test was generated to determine whether the battery fully depletes or not during an eclipse. This is done by simulating the battery depletion for the duration of an eclipse and comparing it to the battery storage size. The battery is assumed to be fully charged initially. The test runs for a number of steps, including a completion step. Each step can be linked to any number of requirements, and if the actual result matches the expected result, the test step will be successful. The implementation in Valispace of this test is shown in Figure 6.3. The verification status is only updated whenever the corresponding tests are ran.

Requirements in Valispace prove highly effective for quickly determining whether requirements are no longer verified while making changes to the system. Both the rules and tests quickly reflect if the changes are acceptable. However, creating these rules and tests requires some additional time during the mission and system design phases, as the requirement tree must be connected to the system design tree where possible. Otherwise, the user may incorrectly assume a requirement remains fulfilled while making changes, when in actuality the status is not updated.

An overview of the requirements is given on the dashboard of Valispace, and shows which requirements had recent activity. It also gives a pie graph with the statuses of the requirements. This allows users to quickly determine which requirements still require verification, and which ones are no longer verified after changes are made.

6.2 Margins

The margins have been added in Valispace identically to how they were added during the DSE, as described in Section 5.2. This does not affect the way that the outputs are displayed directly. Instead, Valispace displays the valis with margins as their worst case values. This effect can be seen in Figure 6.4. Notice how the mass itself is given as 19.14 kg. During the DSE, the value including the margin was assumed to be true while designing the system[7]. This may lead to confusion when documenting values or copying them for use elsewhere, as even the analyses module does not account for these margins and instead displays the satellite mass as 19.14 kg. In sensitivity analyses, these margins can also not be tested nor accounted for. This forms a risk of failing to identify key dependencies during a sensitivity analysis, as sensitivity analyses in the earlier stages of the design will not account for the larger impact by dependent valis with a larger margin.

Margins can still prove useful, as Valispace does automatically compute how these margins are

propagated. The margins are also displayed within the connections graph for the system, allowing the user to see where the margins come from and what their values are per dependent vali. In Figure 6.5 the connections graph from the satellite mass vali is shown. It can be inspected to allow the user to determine where the margins are defined and what the values of the dependent valis are. The graph can be extended further clicking the blue circles to show further dependencies. Furthermore, the margins feature in Valispace removes the need to define margins separately and multiply them after the design, reducing the risk of forgetting to take these margins into account during computation and allowing margins to be omitted during sensitivity analyses. Thus, the addition of margins in Valispace may save time by not having to define margins manually, but introduces the risks of failing to document values correctly and identify key dependencies.

Lastly, a flaw was found in the software, where a worst case value would not update if the value was set to zero. This would lead to the propagation of the worst case value and margins would be determined using this worst case value, despite that the worst case should actually be zero. To circumvent this, an infinitesimally small value (e.g. 1e-100) can be added to any function which may otherwise yield zero, to properly update the worst case scenario. Finding the source of a worst case value which does not update requires checking each dependency manually, and overall this process reduces design accuracy and increases risks and time spent resolving it.

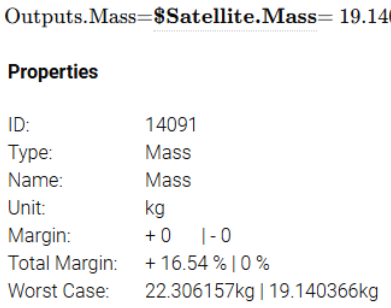

Figure 6.4: The properties of the output mass vali in Valispace.

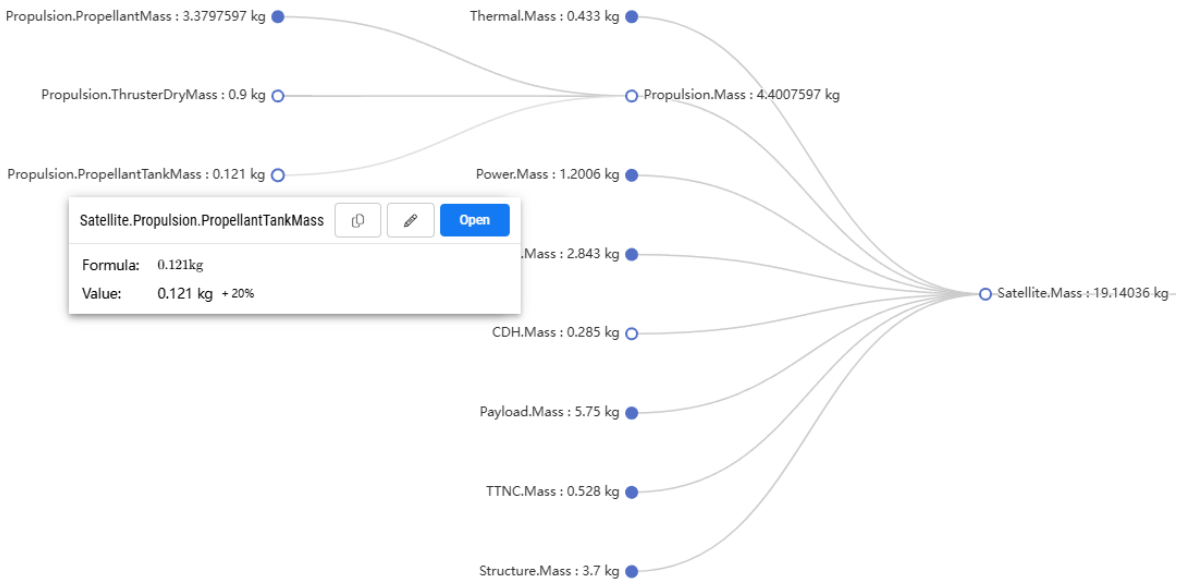

Figure 6.5: The connections graph from the satellite mass vali. Notice how every dependent variable is visualized and how they can be inspected to determine their value and applied margin.

6.3 Sensitivity analysis

Four sensitivity analysis cases were performed using the design in Valispace to determine the usefulness of Valispace for performing sensitivity analyses and to improve the design. These four cases are as follows:

1. A payload mass increase of +10.9%, from 5.75 kg to 6.38 kg
2. A different injection inclination angle of 3 ° instead of 0 °, for an uncertain satellite injection orbit.
3. A different thruster selection, with a mass increase from 3.7 kg to 3.84 kg, and decreased power consumption, from 11.49 W to 10 W.
4. A structure subsystem mass increase of +48.1%, from 3.7 kg to 5.48 kg

In each of these cases, the effect on the total mass of the satellite is measured and compared to the estimated effect on the DSE design. These four cases are treated fundamentally differently in Valispace, and are hence treated in different sections.

6.3.1 Payload mass increase

The first sensitivity analysis case concerns an increase in the payload mass. In this case a sensitivity analysis case is performed in Valispace by selecting the satellite mass vali, then selecting the payload mass as the dependent vali, and then selecting a range for the payload mass which covers the increased payload mass case and the nominal case. This yields the graph shown in Figure 6.6.

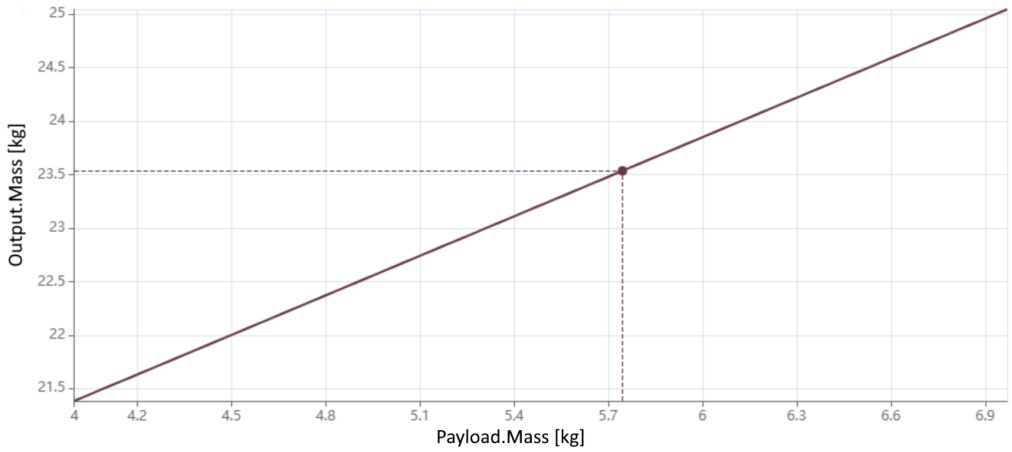

Figure 6.6: The graph resulting from the sensitivity analysis in Valispace, showing the dependency of the satellite mass on the payload mass.

It can be concluded that the relation is functionally linear, with an average slope over the selected domain of approximately 1.2 [-]. For a payload increase of 10.9% to 6.38 kg, the satellite mass will increase to 24.3 kg. The resulting linear graph and slope value are plausible, given that the satellite is designed to support the payload first and foremost, and snowballing effects are to be expected when increasing payload mass. Accompanied with the sensitivity analysis graph is a formula which indicates the relation between the selected vali and dependency. In this case, the formula has a rather large number of components because of the large number of masses which are summed to compute the satellite mass, but differentiation of this formula gives an indication of the local gradient of the sensitivity analysis, which is approximately 1.2, matching the computed average slope. For more complex relations, these graphs are not consistently reliable and Valispace can time out and fail to compute a formula. However, rerunning the computation often solves these complications. An exception where these complications cannot be solved is described in Section 6.3.5.

6.3.2 Different injection inclination

During the DSE case, it was determined that a taxi vehicle should be used to bring the satellite to its mission orbit, as it was estimated the delta-V required to go from the launcher injection orbit to the mission orbit would increase the system mass far over the budget. However, a risk that was not taken into account during the DSE design process is taxi vehicle deployment inaccuracy, where a maneuver

is required for the satellite to correct its orbit. To create such a case, it is assumed a taxi vehicle may lead to an uncertainty in the inclination angle, the effect of which is studied in this case. Furthermore, the inclination angle change is done in the DSE mission design during the Hohmann maneuver at the second burn. Hence, it is assumed the perigee and apogee altitude have an uncertainty of 10 % of the mission orbit at the geostationary orbit altitude of 35786 km, such that the perigee is at 32207 km and the apogee is at 39364 km. The impact of perigee and apogee uncertainty may also be studied using a sensitivity analysis, but is not done here.

The injection inclination angle is accounted for optimally in the Hohmann transfer by assuming an impulsive burn is done to change it to the mission orbit inclination during the second burn maneuver[55]. This way, the second burn and inclination correction burn are performed simultaneously, and the cosine law can be used to determine the delta-V required for this maneuver[56]. The impact of the injection inclination angle on the delta-V budget is shown in Figure 6.7. Note that the mission orbit has a inclination angle of 0 °, hence why the delta-V budget is the lowest at no injection inclination.

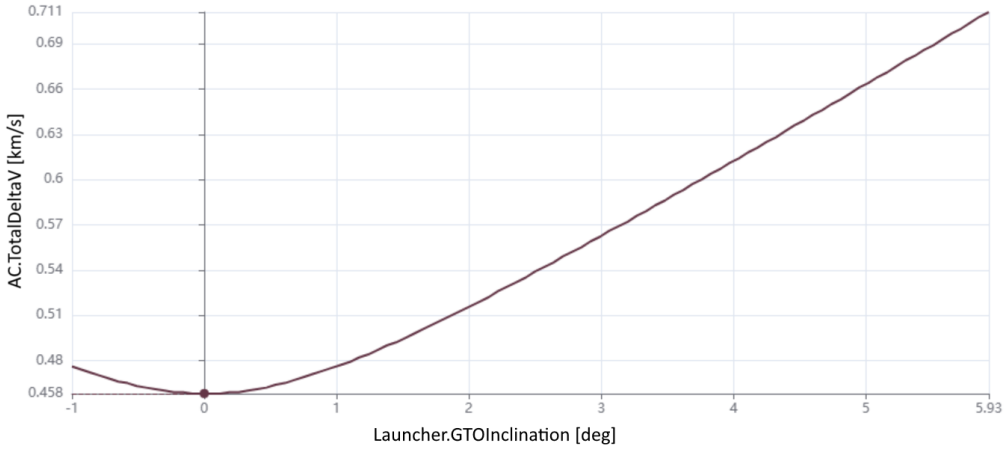

Figure 6.7: The relation between the inclination angle of the injection orbit and the delta-V budget, from the sensitivity analysis feature in Valispace

It can be observed that the relation is non-linear, with a minimum near zero and a parabola-like graph. Valispace includes a formula showing the relation between the valis, in this case as given in Equation 6.1.

$$AC.TotalDeltaV = (((((63.175m/s + ((sqrt(18491115m^2/s^2 - ((18486364m^2/s^2 \cdot (cos(X - 0.0deg))) :: m^2/s^2))) :: m/s)) :: m/s) + 0.0m/s) + 146m/s) + 180.0m/s) :: km/s \quad (6.1)$$

Here, X is the dependent vali used for the sensitivity analysis, in this case the inclination in degrees. The cosine function in also implemented with respect to degrees. The double colons are used to denote unit conversions, where every value is converted to m/s until the end, where the delta-V budget is converted to the selected display unit km/s. The other numbers come from other valis.

The cosine rule used to account for inclination change is in the middle of this equation. It would suggest that the relation between inclination and the delta-V budget always takes the shape of a negative cosine. However, this does not explain the behavior of the graph in the case the injection orbit and mission orbit have identical apogees and perigees, which is shown in Figure 6.8. Here, it seems as though the relation between the delta-V and the absolute value of the inclination angle is linear.

The linearity can be explained using the cosine law and a Taylor series expansion. The cosine law is given in Equation 6.2.

$$\Delta V_2 = \sqrt{V_2^2 + V_{mission}^2 - 2V_2 V_{mission} cos(i_{mission} - i_{injection})} \quad (6.2)$$

Where ΔV_2 is the velocity differential during the second burn, V_2 is the velocity before the second burn, $V_{mission}$ is the mission orbital velocity and thus the velocity after the second burn, $i_{injection}$ is the

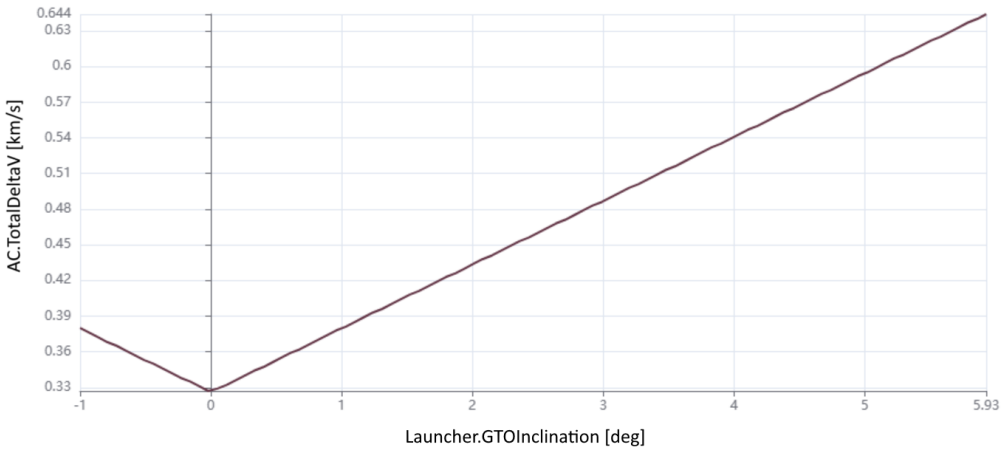

Figure 6.8: The relation between the inclination angle of the injection orbit and the delta-V budget in case the injection orbit and mission orbit have identical apogees and perigees.

inclination from the taxi injection orbit and $i_{mission}$ is the mission orbit inclination. If V_2 and $V_{mission}$ are approximately equal, denoted V, and a Taylor expansion is used for the cosine, then the equation becomes:

$$\Delta V_2 = \sqrt{2V^2 - 2V^2 cos(i_{mission} - i_{injection})} = V\sqrt{2 - 2cos(i_{mission} - i_{injection})}$$
$$= V\sqrt{2 - 2 + (i_{mission} - i_{injection})^2} = V|i_{mission} - i_{injection}| \quad (6.3)$$

This proves that the relation between the absolute value of the inclination angle difference and the delta-V budget is approximately linear, with a derivative equal to the velocity at the point where the maneuver is performed. However, this cannot be determined from the formula given by Valispace. Hence, the sensitivity analysis can be effective for identifying complex relations between valis, but cannot be used to explain them.

6.3.3 Different thruster selection

For the third case of the sensitivity analysis, a different thruster was selected. Instead of the EPSS C1, as selected during the DSE project, the 1N HPGP was selected. The properties of the thruster are part of the propulsion subsystem components in the design parameters. The thruster is modeled using two valis: the dry mass of the thruster and the maximum power consumption during use. The mass is increased from 3.7 kg to 3.84 kg, and the power consumption is decreased from 11.49 W to 10 W. It should be noted that the thruster selection may also impact the effective specific impulse, as the thruster may change the exhaust velocity of the propellant. In this case, however, no such impact was found between the two thrusters considered.

The sensitivity analysis module in Valispace can be used here to determine the gradient between changes to each dependent vali and the tested vali. In this case, the dependent variable are the thruster properties and the tested vali is the propulsion subsystem mass. From the sensitivity analyses in Valispace, the gradient can be taken from the formula given with the graph that indicates the relation between the output and the dependency. If the relation is linear, the gradient can easily be determined from the function. For example, if one were to apply a flat margin of +15% to the mass of the thruster subsystem and then did a sensitivity analysis on the propulsion subsystem mass with as dependency the mass before adding the margin, the formula would read ”Propulsion.Mass = 1.15 * X”, where X refers to the chosen dependent vali. For more complex relations the gradient can be computed by differentiation of this formula, or the graph can be used to compute the local gradient. For the sensitivity of the final mass compared to the dry thruster mass, the gradient is locally approximately 1.14, and shows nearly linear behavior. For the sensitivity of the power budget compared to the thruster power consumption, this relation is approximately 0.98. These gradients give insight into the snowball effect of

these parameters: although an increasing thruster mass exponentially increases the system mass, the thruster power consumption does not have such a snowball effect. In fact, the gradient of 0.98 being less than a linear relation (which would be a gradient of 1) suggests a reverse snowball effect occurs here, on account of the excess heat reducing the thermal heating required by the thermal subsystem. This effect is visualized in Figure 6.9. Notice the inverted (1:-1) relation between internal heat generation and thermal heating power consumption, which explains the inverted snowball effect of increasing the thruster heat generation.

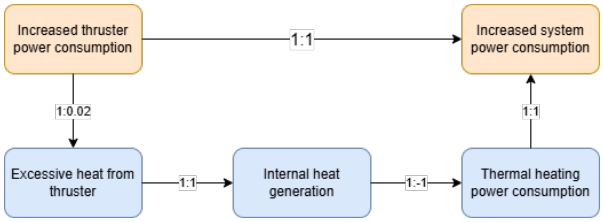

Figure 6.9: A visualization of the effect of changing the power consumption of the thruster.

6.3.4 Structure subsystem mass increase

For the fourth sensitivity analysis case, the effect of the uncertainty of the mass of the structure subsystem is determined by increasing the margin. The margin is largest when introduced in the first design iteration, and increasing the result of the last iteration would not adequately represent the way the system mass could snowball if different margins were selected during the first iteration.

The system component in Valispace cannot display the output of multiple design iterations, nor perform iterations by itself as this would form a circular dependency. However, design iterations can be implemented manually by creating a separate system component for each of the iterations and forwarding the results of the previous to the preliminary estimations of the next iteration. This is effectively done here for one iteration: the initial budget estimation is the first iteration, and the system design the second.

The structure subsystem margin is introduced here by proportionally increasing the structure subsystem mass estimation during the first budget estimation. This is done by adding an extra vali, the *structure mass deviation* ratio, with a value of 1.481. This is multiplied by the estimated mass fraction as given by SMAD, which assigns 20% of the dry mass to the structure subsystem for the first budget estimation. This increases the first system dry mass estimation. The increased budget is then manually subtracted from the structure design, and added again after the mass is estimated as shown below. This way the structure subsystem design remains identical, but the increased mass is accounted for in the margin.

$$r_{sys,dev} = r_{struct,dev} \cdot 0.2 \tag{6.4}$$

$$m_{sys,est} = r_{system-to-payload} \cdot m_{payload} \cdot (1 + r_{sys,dev}) \tag{6.5}$$

$$m_{struct,est} = 0.2 \cdot m_{sys,est} \cdot (1 - r_{sys,dev}) \tag{6.6}$$

$$m_{struct} = soc \cdot (1 + r_{struct,dev} - r_{sys,dev}) \tag{6.7}$$

Where m denotes mass, r denotes ratio, est refers to an estimation before computing the result, sys refers to the system and $struct$ refers to the structure. Finally a regular sensitivity analysis can be done in Valispace to determine the impact of changing the margin of the structure subsystem on the satellite mass. This is omitted here as regular sensitivity analyses were done in the first and second case.

6.3.5 Errors while performing sensitivity analyses

More sensitivity analysis cases were considered, such as identifying the relation between the pointing accuracy system requirement and the ADCS subsystem mass, which is not continuous because the sensor selection would change due to the sensor quantities being defined in a dataset in a vali. However, any attempt to analyse the impact of a dataset would yield the error *"Neither Quantity object nor its magnitude (1.0630000000000002) has attribute 'formula' error: AttributeError"* and no graph or formula would be given. Further documentation of this error is given in Appendix C.

6.4 Concept trade-off and Valispace

In Section 2.3.4 the design of the payload was discussed to determine how the tool could be used to generate the satellites in all other design concepts. The ability to change some input values and quickly generate a new satellite design allows each of these concepts to be worked out further. In this section, the redesigned satellites for the payload in each concept is given, and then the results are compared to the trade-off procedure to determine how well Valispace can assist with this procedure.

6.4.1 Redesigns for other design concepts

An overview of the satellites per design concept is given in Table 6.2. It should be noted, as is with the DSE case, that no transfer is required from the injection orbit to the mission orbit and this is instead done using a taxi vehicle.

Concept	Satellite	Qty.	Mass [kg]	Power [W]	Delta-V [m/s]
1	1A	1	22.3	39.9	326
	1B	2	11.1	31.0	326
2	2	3	23.5	40.6	326
3	3A	3	20.3	40.0	146
	3B	3	14.0	33.6	146
4	4A	2	22.3	40.0	326
	4B	4	11.1	31.0	326
5	5A	2	20.3	40.0	146
	5B	2	13.9	33.6	146
6	2	4	21.3	40.7	146
7	2	6	23.5	40.6	326
8	2	4	23.5	40.6	326

Table 6.2: An overview of the satellite masses, power consumptions and delta-V budgets for each concept, estimated using the tool.

It can be observed that many satellites have an equal delta-V budget. The only exceptions are higher altitudes satellites, where no end-of-life maneuver is required as per the sustainability requirements. Delta-V due to drag compensation are neglected in all cases because of the high altitudes in all cases; at only 1000 km altitude, the air density is less than $10^{-14} kg/m^3$, hence it can be neglected beyond that altitude[16]. In concept 8, one satellite is parked in a parking orbit, but this orbit is never specified, hence the impact on its delta-V budget cannot be determined. If it is already in a geostationary orbit, only a phase-corrective orbital maneuver would be required. Altitude also affects the design in the thermal, power and TTNC subsystems, increasing the mass budget. Component reselection was not done here as the satellite still fulfilled the requirements, hence no impact is seen on, for example, the power required for the TTNC subsystem despite the altitude impacting the link budget.

6.4.2 Concept trade-off procedure validity and necessity

In this section, each of the criteria will be discussed to determine if they can fully be quantified using Valispace, and to determine if the trade-off procedure can be omitted entirely when all trade-off criteria can be quantified for all concepts using Valispace, and then determining that one design concept is objectively superior to the rest. Below is a list of the criteria and a summary regarding their quantification in Valispace with more details below.

- Delta-V: the delta-V was fully quantified in Valispace
- Cost: the cost was not implemented because of research limitations, but can be fully quantified
- Reliability: the reliability budget can be set up in Valispace, but the DSE case only considered redundancy in the constellations which leads to a relative failure probability, which is not enough to quantify reliability

- Sustainability: can not be quantified
- Quality of measurements: requires much more knowledge on payload, and hence could not be quantified

The delta-V budgets could be quantified for each of the satellite concepts. Interestingly, during the design concept trade-off, it was stated that concepts 3 and 6 were scrapped because of the impact of the increased mission altitude on the delta-V budget, but here the opposite seems true. Concept 5 was later scrapped as well due to its poor score for the delta-V criterion. An argument could still be made for these concepts scoring poorly because the taxi service utilized would require additional delta-V, increasing the cost of the service but this should then be accounted for in the cost criterion, as it does not affect the delta-V budget of the satellite.

The cost was not implemented in Valispace. If this were done identically to the DSE, the cost for each of the satellites can be estimated by changing the inputs and reselecting the COTS components for each satellite concept. This requires more resources than how the cost criterion was handled during the trade-off procedure, where it was done simply by accounting for the number of active and passive satellites in the constellation, but allows much better quantification of the cost.

Reliability can be accounted for with a reliability budget estimation, but reliability is usually traded off with cost. For the trade-off these were considered independent variables, where the reliability depends on redundancy within the constellation. Although a reliability budget would allow an estimation of the probability of system failure for each concept, the quantified probability cannot directly be used to determine which concept is superior.

The sustainability can still not be quantified. Sustainability is a legal and moral consideration, while any quantification of sustainability procedures would only conclude that it requires additional resources at no pay-off to the mission.

Quality of measurements, which in this case refers to the number of datasets obtained from separate laser interferometers and the independence of such datasets, can still not be quantified. Quantification would require more knowledge on how effective one dataset is and how much value is added by adding more, as well as determining the impact of dataset independence. All of this fell outside of the scope of the DSE mission. For these reasons, the quality of measurements cannot be quantified here.

Hence, it is determined that the trade-off procedure cannot be omitted entirely. However, an implementation in Valispace which supports all concepts can be used to perform a sensitivity analysis with regards to the design concept and perhaps change concepts after pre-phase A, or such an implementation can be used to suspend the trade-off phase until after pre-phase A, when more information regarding the concept can be quantified and the trade-off is thus more accurate.

6.5 Taxi vehicle necessity in DSE

Filling in the adjusted GTO orbit inputs yields an increase in the total delta-V compliant with the values computed during the DSE. Furthermore, another CubeSat skeleton was selected to support the increased mass and volume budgets. The impact on the design affects the mass and volume drastically. The updated mass budget can be seen in Figure 6.10. It can be observed that the propulsion subsystem mass now forms the majority of the system mass. The propellant mass is now 27.9 kg, whereas the system dry mass is 15.8 kg. The total mass is 43.7 kg, almost double the nominal value 22.3 kg. The volume has increased to 31.2 U, far over the nominal volume limit requirement of 12 U and the updated volume limit requirement of 16 U.

The mass density of the selected propellant is 1.26 kg/m³. If another propellant were to be selected to reduce the system volume to 16 U without changing the specific impulse, that propellant would need a mass density of at least 6.81 kg/m³, an unrealistic number for any existing chemical propellant. If only the specific impulse would be changed to reduce the volume to 16 U, the specific impulse would need to be increased from 204.62 s to 488.3 s. This specific impulse is unachievable by liquid propellants and even cryogenic propellants. Hall-effect thrusters were also a design consideration during the DSE, and research into the feasibility of such a thruster was also a recommendation proposed by the DSE report.

It is thus concluded that the assessment that the satellite could not reasonably perform the maneuver itself was correct. The additional research performed here shows that such design considerations can

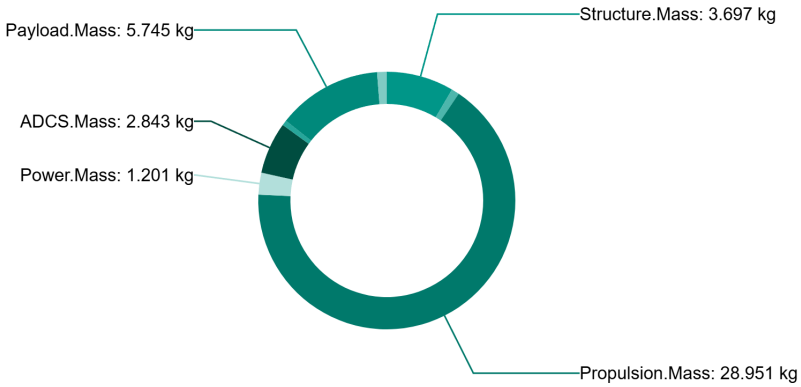

Figure 6.10: The system mass budget in case the satellite is designed to perform the transfer maneuver from GTO to GEO by itself.

quickly be analyzed using Valispace, such that the design choice regarding using a taxi vehicle can be made and supported by research without using only preliminary estimations.

6.6 Bugs and flaws in the software

From the interview with León, it was determined some bugs and flaws are present in Valispace. In this section, each of these will be discussed to see if they still exist and what their impact is.

Settings file generation bug

One of the mentioned bugs concerned the automatic generation of a new 'settings.py' file whenever a script was executed, which caused a large number of these to be created because the script execution was automated. During this research the bug did not seem to exist anymore, nor did it cause much impact at the time as the file was rather small and duplicates are easy to remove in bulk periodically.

Valispace API bug

A method of the API (post, including header and payload) in Valispace was needed to connect to an external API, in this case that of Satsearch to query for COTS components, however this did not function and would result in an internal server error (500). This bug had a large impact as it reduced the automatization potential of the software, but this bug cannot be reproduced anymore as the API method was found to function properly.

Margin update for zero values bug

A bug was discovered while implementing margins using the built-in margin feature in Valispace, where a worst-case value would not update if the value of the corresponding vali was set to zero. This results in incorrect margins propagating throughout the design, leading to inaccurate designs. This potentially has a very large impact on the design. This bug was found while changing the mission altitude, which affected the air density. In Figure 6.11 the effect of this bug can be seen. While the value and margins for this vali are zero, the worst case values remain identical to the last non-zero values. These worst-case values are propagated throughout the drag-compensation delta-V calculations, leading to a worst-case total delta-V budget of 2.2 km/s, despite the actual budget being only 0.33 km/s.

Automatic update of requirement parent flaw

A flaw was discovered in the software where, if all children of a requirement were fulfilled, the parent requirement cannot automatically be marked as fulfilled. In practice, requirements are always structured such that a parent requirement *is* fulfilled if all of its children are. Valispace has since updated the requirements module to automatically mark parent requirements as fulfilled, hence this flaw no longer has any impact.

Margins visibility flaw

Margins on valis and their corresponding worst-case values are only visible if the vali is inspected. For example, if a user wants to add a 20 % margin on a system mass of 10 kg, they would effectively design

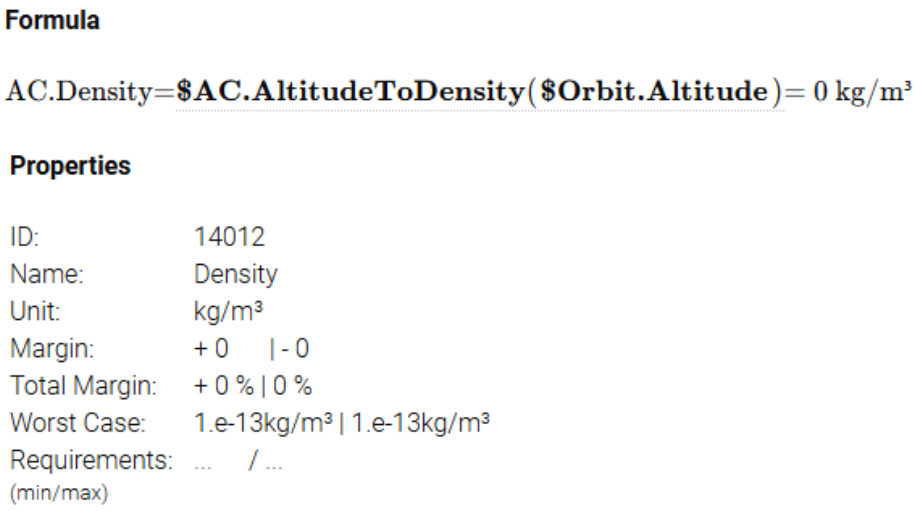

Figure 6.11: The properties of the density vali. Notice how that although the value and margins are zero, the worst case values are not.

a system with a mass of 12 kg. However, Valispace would still suggest the system mass is 10 kg, with the worst case value of 12 kg as one of its attributes if inspected. If the user is aware of this, this can be worked around, but this may lead to incorrect values being documented if they are copied from the properties of components directly. This, combined with the margin update bug and the fact that a sensitivity analysis cannot be performed on a margin, leads to the conclusion that margins are more effective if implemented as separate valis. That way, these bugs and flaws are not present, although margins do take up more visual space in the components module this way.

7 Discussion

The interview and usage of Valispace for the recreation of the DSE project gave plenty of insight into the impact on project performance and the process of verification while using Valispace. The insights follow from the tool recreation and verification as described in Chapter 4, as well as the experimentation results described in Chapter 6. In this chapter, these insights will be discussed.

7.1 Impact of Valispace on project performance

The project performance is a combination of the time efficiency, design accuracy and risk mitigation during a project. Usage of Valispace in practice determined that for each of the processes during a satellite design project the effect Valispace has on these project performance parameters is different. For this reason, the discussion is split up between the design, verification, and sensitivity analysis processes. The other project phases, such as the concept generation, trade-off, integration, and documentation are not included because they are not feasible to automate and hence the strengths of Valispace cannot effectively be utilized. The impact on project performance is only analyzed qualitatively, and further research could be performed on quantification of the impact by performing surveys or conducting experiments.

7.1.1 Mission design

During the mission design phase the system requirements can be added to Valispace (Section 6.1). The centralization of the requirements reduced the risk of overlooking requirements or failing to account for requirement which are no longer verified due to recent changes. The rules and test verification methods allow users to quickly determine the feasibility of changes to the design. The test verification method is especially flexible and provides good traceability for requirements which are complex to describe numerically. The rules method is great for quick automatization of a requirement, as it compares its valis to the system design tree directly, which is easy to implement. Manual requirement verification is also supported, allowing full inclusion of all generated requirements within Valispace.

Automation of requirement verification still requires some additional time compared to manual verification methods. This reduces time efficiency during the mission planning phase and early system design phase. Hence, the time efficiency during the mission design phase will be lowered. Valispace also has a ValiAssistant, an AI which is capable of generating, improving and breaking down requirements. Perhaps the ValiAssistant could increase time efficiency during this phase, but this requires further research.

7.1.2 System design

The design phase requires setting up the design tree in the system design module (Section 2.5, Section 3.2). Setting up an effective design tree requires time to fully understand the design process, and restructuring the tree introduces the risk of incorrect vali dependencies or values (Section 3.3). Young engineers tend to have an understanding of the structures used in satellite design, but working with Valispace requires more flexibility to be able to process design iterations and track/document the results.

Creating a single tree for the satellite created issues when computing the preliminary budget estimations(Section 3.4). For example, this leads to the first system mass estimation, but this will not be the mass of the satellite, merely a first estimation. Fortunately, Valispace does allow moving individual valis or entire tree branches throughout the component tree, and accurately updates the dependencies to yield the same results, as tested in Section 4.3.

Valispace also requires additional time when setting up an effective method to swap components during the design process (Section 3.7, Section 6.3.3). Although a user may update component values individually, this requires manually copying values from one application to another, which introduces additional risks (Section 6.6). On the other hand, programming in the Python API to be able to swap components using commands requires a lot more time and the availability of a component database. Components can also be set up using datasets, which has proven to be an effective way introducing a couple of components without requiring additional programming, but still requires additional time to set up and introduces risks in automatic margin propagation (Section 4.1, Section 6.6).

Valispace is limited in its ability to make complex computations (Section 6.6). Again, the Python API can be used to mitigate this problem, but this requires additional time for development and verification. Valispace does support simple interpolation and extrapolation of data points, which has proven to be effective for complex but predictable relations such as density versus altitude (Section 6.3.2), but since Valispace only supports stepwise and linear interpolation and extrapolation, this would not suffice for unpredictable relations and unknown outputs of the dataset(Section 3.5.2).

Valispace does excel in its ability to track valis and their dependencies. The inclusion of tags, subscriptions, histories, comments, descriptions, references, dependencies (used in/used by) and marking of impacts allows the user to understand the position of each vali and its status very well (Section 6.2), which is especially nice for guaranteeing an accurate design and reduces the risk of incorrectly copying values, as they are all maintained within one application.

While generating the design tree, the user can start connecting valis to requirements for automated verification (Section 6.1). This requires additional time to do, but reduces the risk of failing to account for requirements, as any requirement which requires rules or tests can be marked as such, and any requirements which lack a rule or test will stand out.

The introduction of margins did not prove effective (Section 6.2), because they are unpredictable and have a bug. Margins are taken into account wherever they are introduced, and they flow through all dependent valis to keep track of the total margin. Worst case values are computed automatically, and these are checked for in the requirements. However, for some valis the margins are added automatically. This problem arises especially when interpolating values in datasets, for which values may not actually be uncertain. Furthermore, a bug exists where margins are not updated for valis with a value of zero, instead retaining the margin as it was before the value became zero. This creates the risk of inaccurate margins and may lead to a design where the margins are larger than needed (Section 6.6).

The time efficiency is thus significantly lower during the system design phase. The structuring of the design tree and connecting valis requires much more time than using computation sheets would. The swapping of COTS components either requires additional time or introduces the risk of inserting conflicting values. Margins create risk and are not clearly displayed, hence adding separate valis to represent margins seems preferable. For further research, a template could be considered which further generalizes the commercial spacecraft design in Valispace. This would increase time efficiency during the system design phase by reducing time spent structuring the tree, instead only altering the tree where necessary for the specific mission.

7.1.3 Verification

The conventional verification process is done using unit tests and system tests (Section 4.3, Appendix A). During the DSE project, the system tests were limited to single subsystems as the software was developed per subsystem, and not for the system as a whole (Section 2.3.7).

Unit tests are not easily implemented in Valispace, as the functions which form the units are saved per vali and cannot be called independently (Section 4.3). Any independent implementation of unit tests (i.e. copying the function and testing it elsewhere) leads to the risk of someone modifying the function later without accounting for the corresponding unit test. Although the history feature of Valispace does easily allow the user to determine whether or not unit tests are still valid by tracking the most recent

date where the unit test function was correctly copied, these are all workarounds.

System tests do not have the problem of independency as even without Valispace, systems can usually not be called independently. The DSE project did not need to isolate subsystems for system testing as the software was developed separately for each subsystem (Section 4.3). System tests can be performed manually by changing values and determining that the outcomes are correct, but this system test would need to be run manually whenever changes are made which requires a lot of additional time.

Verification of a product in Valispace thus requires a different strategy than products created using code. These unique strategies will be discussed further in **??**. To conclude on the effect of implementing Valispace on the project performance during the verification process for now, it needs to be stated that it depends largely on the selected strategy.

7.1.4 Sensitivity analysis

The sensitivity analysis tool proved to be a very effective feature in Valispace, where the graph visualizing the dependency on another vali allows the user to perform system analyses for optimizing parameters, selecting components and checking for the impact of the selected margins. The accompanying formula did not prove useful for analyzing the system (Section 6.3.2). Compared to traditional methods, the main advantage of Valispace is the completeness of the connections between valis in the workspace (Section 6.3.1). Whether the user prefers to only use the sensitivity analysis tool in some of the ways described in Section 6.3, or they prefer to simply change values and determine the results of the change propagation (Section 6.3.3), the user can very quickly determine at large scale how their system would change. Performing sensitivity analyses in Valispace thus greatly increases time efficiency, allows the user to better optimize their design by visualizing the impact of changes and reduces the risk of sensitivity analyses failing to account for all impacts they may have. However, using the sensitivity analysis tool for certain sensitivity analysis cases requires some creativity (Section 6.3.3, Section 6.3.4), as changing values directly leaves traces in the history and may unnecessarily notify subscribed users, and introduces the risks of these values not being changed back correctly afterwards.

Furthermore, the concept trade-off procedure done during the DSE was found to be partially inaccurate, and it was found that a flexible implementation of Valispace such that all design concepts are supported by the implementation allows the user to better quantify trade-off criteria and suspend or omit the concept trade-off phase entirely depending on the selection of criteria which are quantifiable, increasing design accuracy and increasing time efficiency by omitting redundant phases and accounting for them during the sensitivity analysis (Section 6.4).

7.2 Strengths and weaknesses of the tool

From the research, a couple of strengths and weaknesses of the tool were identified. Here, they will be discussed and their impact will be assessed. These strengths follow from the case study, which is here performed only for a pre-phase A design. Further research into the strengths and weaknesses of Valispace for later design stages is recommended.

Strength: Automatic change propagation

The direct connections between valis and the way changes to the values and margins are propagated automatically greatly reduce time spent updating and communicating these manually, and mitigate the risk of users assuming or communicating incorrect values (Section 2.6, Section 4.3). It allows the user to quickly determine the impact of inputs and connections, either by changing values and observing their impact or by using the sensitivity analysis tool (Section 6.3). It also allows for quickly verifying assumptions that were made in advance (Section 6.5).

Strength: Analysis generation

Valispace can be used to quickly document and communicate budgets, values, and datasets within Valispace using the analysis module (Section 4.2), for which the values are taken directly from the system design module such that it is quickly reproduced for other system inputs, such as another design concept (Section 6.4.2) or a system verification case (Section 4.3).

Strength: Design flexibility
Datasets can be used and interpolated/extrapolated for complex relations such as air density at high altitudes, or discontinuous relations such as sensor selection for different input parameters (Section 6.3.2). Less complex relations can be entered as a formula. This allows the user to determine the impact of large or fundamental design changes quickly (Section 6.3.3). This proves especially useful during the sensitivity analysis process as other design concepts from the concept generation phase can be analyzed if they are accounted for in the tool design (Section 6.4.2).

Strength: Requirement verification
Often during the DSE process, it was found out only after a couple of hours or occasionally even days that there was a design choice made which did not comply with the requirements set up beforehand (Section 2.3.7). Valispace would quickly show that a requirement is no longer fulfilled once changes are propagated (Section 6.1), informing the requirements engineer automatically that they should alert the engineer responsible for the change (Section 6.4.1). This allows for quickly checking the feasibility of design choices (Section 6.5).

Weakness: slow change propagation
Valispace can be inconsistent with its ability to propagate changes (Section 2.6). Occasionally, changes were only visually propagated after refreshing the browser or inspecting the vali which should be updated. This may lead to incorrect copying of values while documenting or using external tools, although it should not lead to an incorrect design within Valispace itself as values should not be copied within Valispace, but rather connected, having their changes propagated eventually.

Weakness: limits to sensitivity analyses
Currently, only one-to-one sensitivity analyses can be done using the sensitivity analyses feature (Section 6.3). Many of the proposed design changes during the sensitivity analyses process would impact two or more inputs and/or outputs of the system. It would be valuable to have the ability to plot multiple inputs or outputs within generated graphs, or to generate formulas to show the impact of all selected dependent valis. Alternatively, multiple graphs could be generated such that the impact could at least be observed at once, rather than having to switch between inputs and outputs often.

Weakness: flaws and bug in margins
The dataset interpolation and extrapolation options are limited to linear and step-wise, and may require manual regression before implementation in Valispace if polynomial, exponential, and other options are desirable (Section 3.5.1). The margin feature has some flaws and a bug present, which forms a great risk to the obtained design (Section 6.2). Although this can be worked around simply by not using it, it still poses a risk because it *could* be used by users not aware of these flaws.

7.3 Verification of projects in Valispace
For both the unit testing and system testing, it seems that traditional methods are not applicable. In this section a couple of Valispace-specific strategies are discussed to determine how well Valispace allows verification of a product, as well as their effects on project performance.

7.3.1 Unit testing
The testing of a unit is done to prove that the unit works as intended. Typically, this requires that many different cases are often ran through the unit, each of which are checked for the correct predetermined output. Checking all units in the system should ideally be quick, repeatable and show clear results.

One unit testing strategy would be to copy the functions directly to an independent 'testing' component multiple times and marking their most recent copying date. These valis containing the functions can then be checked for correctness via requirements or by asserting their value equal to zero by subtracting the expected value, showing clearly whenever a unit test fails. This method requires a lot more time to copy all valis, write unit test assertions and periodically checking if all copied units still correctly represent their base unit. Hence this is not time efficient and risky compared to traditional methods. It is done once afterwards in Section 4.3 using SMAD models for this case, but only because the system

could be completed before unit testing had to be done, which is usually not the case.

A second unit testing method would be scripting in Python. Importing a vali and its dependencies using the Python API is relatively simple, and changing these dependencies and asserting that the vali's value matches the expected value is rather simple. However, this requires repeated API calls and the performance would be rather poor when done at a large scale. Furthermore, each unit test run would introduce changes in the history of the dependencies of the vali for each test case, which is undesirable as it makes actual changes made to these dependencies hard to identify and would repeatedly trigger subscriptions. This method is thus considered risky, but quite efficient as no additional components are required and the code is simple to set up.

A third unit testing strategy could be maintaining a unit testing document for each vali which requires unit testing, including all required test cases. A test engineer could tag each vali which has passed unit testing at any point in time and tag the vali as tested. The test engineer could then subscribe to all tested valis to be notified whenever changes are made to them, while other users can see that a vali is tested and no longer requires changes to their function. If changes are made still, the test engineer can rerun the test cases and perhaps update them. This method requires a dedicated test engineer (which is not unusual in practice) and a team that is aware of the unit testing strategy, but this may save the team the effort of writing tests using software. Furthermore, the fact that the test engineer is notified of changes also means that the test engineer may determine that the test cases do not (still) decently cover all use cases of the unit. This strategy thus requires additional time and introduces the risk of the team not using the strategy effectively, but increases the design accuracy and reduces risk by notifying the test engineer of changes and required updates, allowing them to act quickly and accurately.

7.3.2 System testing

In system testing, it is assumed the larger units are correct, and the main purpose is to test correct integration of the units. For this purpose, the system to be tested should be isolated to allow testing using non-nominal cases. Reproduction of the system elsewhere (e.g. in Python) partially defeats the purpose of the system test as the integration is done manually again. The user may cross-reference with Valispace for system testing, but this more accurately represents verification by inspection than by system testing.

Instead the system can be isolated by copying the system entirely, including all inputs to the system as done in Section 4.3. Since no valis in the original system are dependent upon the copied system, the copied system can be unit tested and system tested freely. The user may then alter values in the inputs (whether they are part of the system or not) and check the results for all test cases manually, removing the copied systems afterwards and marking the system as verified. The user may also copy the system multiple times, once for each test case, and connect each test case to a single requirement which represents the status of the system tests. However, excessive copying of components may lead to workspace performance reduction. In conclusion, system isolation in Valispace can be done with good time efficiency, and can either be checked manually or automatically, introducing either the risk of a user not marking a system as verified correctly or the risk of the reduced workspace performance. In neither case is the design efficiency affected.

8

Conclusion and recommendations

A before-and-after analysis of the Valispace implementation was done to assess the impact of utilizing Valispace during the satellite design process on project performance and system verification. In this chapter, this research is concluded, and recommendations for further research are given.

8.1 Conclusion

The usefulness of Valispace for satellite design was assessed by performing an interview, performing a before-and-after case analysis study, and experimenting with features in the software. The three research questions were answered as given below.

1. *In what ways does the integration of Valispace into the small satellite design process influence project performance?*
2. *To what extent can a satellite designed using Valispace be effectively verified?*
3. *How can Valispace be used to identify potential improvements to the project 'Small Satellites for Gravitational Waves Observation'?*

For the first question, project performance parameters were set up to qualitatively describe the impact of implementing Valispace, which were defined as the time effectiveness of the engineering processes, the design accuracy, and risk mitigation.

It is concluded that, for the mission and system design processes, the time efficiency is lowered because of the required structuring of the requirements tree and the system design tree, and connecting the requirements to valis and the valis to each other (Section 7.1.1, Section 7.1.2). The design accuracy and risk mitigation are increased because of this inter-connectivity and flexibility, and the subscription feature increases awareness of unexpected changes to the design.

The verification process requires some additional time, but reduces risk and increases design accuracy due to the inter-connectivity of the system (Section 7.1.3).

The sensitivity analysis feature greatly increases time efficiency as the impact of any change can quickly be assessed due to the automatic change propagation (Section 7.1.4). Furthermore, the sensitivity analysis feature proved to be an excellent way to gain insight into one-on-one vali relations. Time can be saved on design concept trade-offs because the flexibility in the software can support all design concepts, which increases design accuracy by either performing a more accurate trade-off or being able to select the optimal concept directly, although implementing this flexibility requires additional time. It greatly mitigates the risk of failing to account for unexpected side effects of changes in the design, as Valispace accounts for these automatically.

For the second question, it is concluded that the verification process requires a unique strategy for unit testing compared to typical computation sheets and programming software (Section 7.3.1). Unit testing any model once is simple, but repeated unit testing requires an active test engineer to be notified of any model changes or usage of the Python API. Both of these require additional time to implement, more than conventional unit tests would, but mitigate risk equally effectively. System testing can be done regularly, although system copying allows quick verification using another case (Section 7.1.2).

For the third question, it is found that Valispace allows for improvements in the design concept trade-off (Section 6.4.1), the sensitivity analyses (Section 6.3) and the verification of an assumption made about the necessity of a taxi vehicle for an orbital maneuver (Section 6.5).

8.2 Recommendations

For further research, it is recommended to consider using the ValiAssistant for automatic requirement generation. It may prove useful in reducing time spent exploring requirements, but could pose a risk when assessing the necessity of each requirement.

Further research should be done into quantifying the project performance metrics. Time efficiency should be quantified by performing surveys and experiments with teams using Valispace. Setting up strategies for quantifying design accuracy and risk mitigation may be more challenging, but is useful to prove the added value of implementing Valispace.

Another recommendation for the future is to create a template for space system design in Valispace which is generalized for many space missions, such that Valispace can be used with components readily defined. The project in Valispace created for recreation of the DSE design has many assumptions, and any reduction of these would increase the flexibility of the project, forming a better template for space missions.

Finally, further research can be done into the application of Valispace for other phases of the design process. This thesis describes only implementing Valispace for phase A and pre-phase A, while it could also prove useful for further design phases.

References

[1] World Economic Forum. *Space is booming. Here's how to embrace the $1.8 trillion opportunity*. 2024. URL: https://www.weforum.org/agenda/2024/04/space-economy-technology-invest-rocket-opportunity/ (visited on 10/29/2024).

[2] Fortune Business Insights. *Small Satellite Market Size, Share & COVID-19 Impact Analysis, By Component, By Application, By Weight, By End-use, and Regional Forecast, 2023-2030*. 2024. URL: https://www.fortunebusinessinsights.com/industry-reports/small-satellite-market-101917 (visited on 10/29/2024).

[3] Harry Jones. "Explaining Space Project Failures". In: (2008).

[4] RHEA Group. *Concurrent Design*. 2024. URL: https://www.rheagroup.com/services-solutions/system-engineering/concurrent-design/.

[5] Roberta Heale and Alison Twycross. "What is a case study?" In: *Evidence Based Nursing* 21.1 (Nov. 2017), pp. 7–8. DOI: 10.1136/eb-2017-102845.

[6] D.R. Hancock, B. Algozzine, and J.H. Lim. *Doing Case Study Research: A Practical Guide for Beginning Researchers*. Teachers College Press, 2021. ISBN: 9780807779828. URL: https://books.google.nl/books?id=G3FEEAAAQBAJ.

[7] J. Achterberg et al. *Small Satellites for Gravitational Waves Observation, final report*. TU Delft, 2020.

[8] NASA. *SEH 3.0 NASA Program/Project Life Cycle*. 2019. URL: https://www.nasa.gov/seh/3-project-life-cycle.

[9] ESA. *How a mission is chosen*. 2020. URL: https://www.esa.int/Science_Exploration/Space_Science/How_a_mission_is_chosen.

[10] SatCatalog. *A better way to design satellites.* 2024. URL: https://www.satcatalog.com/designer/.

[11] ESA. *Comet upgrade for ESA's mission design centre*. 2022. URL: https://technology.esa.int/news/comet-upgrade-for-esas-mission-design-centre.

[12] Erik Kulu. *CubeSats & Nanosatellites - 2024 Statistics, Forecast and Reliability*. 2024. DOI: IAC-24.B4.6A.13. URL: https://www.nanosats.eu/assets/CubeSats-Nanosatellites-2024_ErikKulu_IAC2024.pdf (visited on 10/18/2024).

[13] J. Achterberg et al. *Small Satellites for Gravitational Waves Observation, baseline report draft*. TU Delft, 2020.

[14] J. Achterberg et al. *Small Satellites for Gravitational Waves Observation, midterm report*. TU Delft, 2020.

[15] S. Lacour et al. "SAGE: using CubeSats for Gravitational Wave Detection". In: (2018).

[16] James R. Wertz and Wiley J. Larson. *Space Mission Analysis and Design*. Microcosm Press, Springer, 1999.

[17] Charles D. Brown. *Elements of spacecraft design*. American Institute of Aeronautics and Astronautics, Inc., 2002.

[18] Arthur Norman Guest. "Handbook of Satellite Applications". In: New York, NY: Springer New York, 2013, pp. 293–323. ISBN: 978-1-4419-7671-0. DOI: 10.1007/978-1-4419-7671-0_69. URL: https://doi.org/10.1007/978-1-4419-7671-0_69.

[19] Dominik Barbarić, Josip Vukovic, and Dubravko Babic. "Link budget analysis for a proposed Cubesat Earth observation mission". In: (May 2018), pp. 0133–0138. DOI: 10.23919/MIPRO.2018.8400026.

[20] Eugene Kim Jinkyu Lee and Kang G. Shin. “Design and Management of Satellite Power Systems”. In: *2013 IEEE 34th Real-Time Systems Symposium* (Dec. 2013). DOI: 10.1109/RTSS.2013.18.

[21] Isaac Foster. “Small Satellite Thermal Modeling Guide”. In: *Defense Technical Information Center* (July 2022). URL: https://apps.dtic.mil/sti/pdfs/AD1170386.pdf.

[22] Millan Diaz-Aguado et al. “Small satellite thermal design, test, and analysis”. In: *Proceedings of SPIE - The International Society for Optical Engineering* (May 2006). DOI: 10.1117/12.666177.

[23] Akshay Reddy Tummala and Atri Dutta. “An Overview of Cube-Satellite Propulsion Technologies and Trends”. In: (2017). DOI: 10.3390/aerospace4040058.

[24] J Banks. *Handbook of Simulation - Principles, Methodology, Advances, Applications, and Practice*. John Wiley & Sons, Inc., 1998.

[25] Hong Zhu, Patrick A. V. Hall, and John H. R. May. “Software unit test coverage and adequacy”. In: *ACM Comput. Surv.* 29.4 (Dec. 1997), pp. 366–427. ISSN: 0360-0300. DOI: 10.1145/267580.267590. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/267580.267590.

[26] Boris Beizer. *Software system testing and quality assurance*. USA: Van Nostrand Reinhold Co., 1984. ISBN: 0442213069.

[27] D. Gransberg and M. Villarreal Buitrago. *Construction Project Performance Metrics*. AACE International Transactions, 2002.

[28] Awad S. Hanna. “Benchmark Performance Metrics for Integrated Project Delivery”. In: *Journal of Construction Engineering and Management* 142.9 (2016), p. 04016040. DOI: 10.1061/(ASCE)CO.1943-7862.0001151. eprint: https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29CO.1943-7862.0001151. URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29CO.1943-7862.0001151.

[29] Zach Peterson, Louise Lindblad, and Marco Witzmann. *Revolutionary System Design: How Valispace is Changing Engineering*. Interview on youtube. Mar. 2024.

[30] Valispace. *Valispace Help Desk*. 2024. URL: https://docs.valispace.com/vhd/.

[31] D.L. Oltrogge and K. Leveque. “An Evaluation of CubeSat Orbital Decay”. In: (2011).

[32] Lens R&D. *Bison64-ET FM*. URL: https://lens-rnd.com/products/-bison64-et-fm.

[33] Hyperion Technologies B.V. *ST200*. URL: https://www.aac-clyde.space/what-we-do/space-products-components/adcs/st200.

[34] Cubespace_ADCS. *Cubesense earth*. URL: https://www.cubespace.co.za/products/cubesense-earth/.

[35] Sensonor. *STIM 300*. URL: https://novatel.com/products/gnss-inertial-navigation-systems/imus/stim300.

[36] Blue Canyon Technologies Inc. *RWP100*. URL: https://www.satcatalog.com/component/rwp100/.

[37] Blue Canyon Technologies Inc. *RWP50*. URL: https://www.satcatalog.com/component/rwp50/.

[38] VACCO. *X14029003-1X*. URL: https://www.vacco.com/images/uploads/pdfs/MiPS%5C_standard%5C_0714.pdf.

[39] Space Inventor. *OBC-P3*. URL: https://www.satcatalog.com/component/obc-p3/.

[40] AZUR SPACE. *QJ Solar Cell 4G32C*. URL: https://satsearch.co/products/azur-space-qj-solar-cell-4g32c-advanced.

[41] AAC Clyde Space. *Starbuck-NANO Plus*. URL: https://www.aac-clyde.space/what-we-do/space-products-components/pcdu/cubesat-eps-starbuck-nano.

[42] O.C.E. Technology. *ICP-20*. URL: https://satsearch.co/products/oce-technology-icp20-lithium-ion-battery.

[43] Ahmed Nosseir, Angelo Cervone, and Angelo Pasini. “Modular Impulsive Green Monopropellant Propulsion System (MIMPS-G): For CubeSats in LEO and to the Moon”. In: *Aerospace* 8 (June 2021), p. 169. DOI: 10.3390/aerospace8060169.

[44] ECAPS. *LMP-103S*. URL: https://www.ecaps.se/rocket-fuel-1.

[45] ISIS. *12U aluminum 6082-T6 structure*. URL: https://catalog.orbitaltransports.com/isis-cubesat-structures/.

[46] Zoppas. *STOCK Flexible Heater*. URL: https://zoppasindustries.com/wp-content/uploads/2023/06/Catalogo-Flexible-Heaters-for-Space-STOCK.pdf.

[47] NASA. *GSC-TOPS-40*. URL: https://technology.nasa.gov/patent/GSC-TOPS-40.

[48] goodfellow. *DuPont Kapton HN PI Metallized Film with 30nm Aluminum*. URL: https://www.goodfellow.com/eu/kapton-hn-pi-metallised-film-with-30-nm-aluminium-sizes-group.

[49] NanoAvionics. *CubeSat UHF Antenna System*. URL: https://nanoavionics.com/cubesat-components/cubesat-uhf-antenna/.

[50] NanoAvionics. *CubeSat UHF Digital Radio Transceiver satCOM UHF*. URL: https://nanoavionics.com/cubesat-components/cubesat-uhf-digital-radio-transceiver-satcom-uhf/.

[51] GOMspace. *S-band patch antenna for high speed communication*. URL: https://gomspace.com/shop/subsystems/communication-systems/nanocom-ant2000.aspx.

[52] Satlab. *SRS-3 Full-duplex S-Band transceiver*. URL: https://www.satlab.com/products/srs-3/.

[53] Pasternack. *Link budget calculator*. 2024. URL: https://www.pasternack.com/t-calculator-link-budget.aspx.

[54] SRE-PA & D-TEC staff. “Margin philosophy for science assessment studies”. In: (2012).

[55] Robert A. Braeunig. *Orbital Mechanics*. URL: http://www.braeunig.us/space/orbmech.htm#maneuver.

[56] Neil Bibby and Doug French. “Pythagoras extended: A geometric approach to the cosine rule”. In: *The Mathematical Gazette* 72.461 (1988), pp. 184–188. DOI: 10.2307/3618247.

A

Verification of Valispace implementation using DSE

A part of the verification procedure of the tool which was implemented in Valispace is to make sure it yields results identical to the satellite design from the DSE. In this appendix, the tool design is compared to the DSE design in the following three aspects:

1. The subsystem budgets, given in Table A.1
2. The delta-V budget, given in Table A.2
3. The selected COTS components, given in Table A.3
4. The performance-indicative parameters, given in Table A.4

In Table A.1, the power consumption is given during daytime. It is assumed that the propulsion subsystem is inactive while eclipsed, such that its power can be allocated to the thermal subsystem instead. In Table A.2 the transfer is omitted as it is assumed to be performed by a taxi vehicle.

The performance-indicative parameters here refer to subsystem-specific parameters which give an indication of the requirements for the COTS component selection process. It should be noted that these only concern computed parameters. Any design choices made during the DSE are copied literally and do not require verification. The performance-indicative parameters are selected to be the results of the model implementation, as any incorrect implementation of intermediate values would result in invalid model results.

(Sub)system	**Target mass [kg]**	**OK**	**Power consumption [W]**	**OK**
ADCS	2.843	✓	9.9	✓
CDH	0.285	✓	3.3	✓
Payload	5.75	✓	5.114	✓
Power	1.2	✓	0.818	✓
Propulsion	4.4	✓	11.49	✓
Structure	3.7	✓	0	✓
Thermal	0.433	✓	0	✓
TTNC	0.528	✓	5.5	✓
Margin	*11.21%*	✓	*7.45%*	✓
Total	22.3	✓	40.6	✓

Table A.1: The requirement verification matrix for the tool compared to the DSE design for the (sub)system budgets

Maneuver	Delta-V [km/s]	OK
Transfer	0	✓
Apogee deviation	0.003	✓
Phase shift	0.102	✓
Realignment	0.225	✓
Orbit maintenance	0.027	✓
Drag compensation	0.000	✓
End-of-life	0.011	✓
Total	0.33	✓

Table A.2: Delta-V budget verification of the tool compared to the DSE design.

Section	Component	Model	OK
ADCS	6 Sun sensors	BiSon64-ET, LENS R&D	✓
	2 Star sensors	ST200, Hyperion Technologies B.V	✓
	1 Earth sensor	CubeSense N, CubeSense N	✓
	1 Gyro	STIM 300, Sensonor	✓
	1 Reaction wheels	RWP100, Blue Canyon Technologies Inc.	✓
	3 Reaction wheels	RWP050, Blue Canyon Technologies Inc.	✓
	2 Rotational controls thrusters	X14029003-1X, VACCO	✓
CDH	1 OBC	OBC-P3, Space Inventor	✓
Power	96 Solar cells	QJ Solar Cell 4G32C	✓
	1 PCDU	Starbuck NANO Plus PCDU	✓
	1 Battery	Li-Ion ICP-20	✓
Propulsion	1 Thruster	EPSS C1 thruster	✓
	Propellant	LMP-103S	✓
	1 Propellant tank	Custom Ti-6Al-4V tank	✓
Structure	1 Skeleton structure	12U aluminum 6082-T6 structure, ISIS	✓
	6 Solar panels	Custom AlBeMet sheets and HRM	✓
	2 RCS thruster supports	Custom support and HRM	✓
	5 radiation shielding panels	Custom panels on outside skeleton	✓
Thermal	1 Heater	Zoppas Industries STOCK Flexible Heater	✓
	Louvers	Form Factor Thermal Control Louvers, NASA	✓
	Coating	Aluminised kapton foil (2mm)	✓
TTNC	1 UHF antenna	Nanoavionics UHF antenna	✓
	1 UHF Transceiver	Nanoavionics UHF transceiver	✓
	1 S-band antenna	GOMSpace S-Band patch antenna	✓
	2 S-band transceivers	Satlab S-Band transceiver	✓

Table A.3: Verification of the component selection in the tool compared to the DSE for all subsystems.

Section	Parameter	Value [unit]	OK
Astrodynamic Characteristics	Orbital radius	42157 [km]	✓
	Velocity	3074 [m/s]	✓
	Delta-V requirement	329 [m/s]	✓
	Maximum eclipse period	4160 [s]	✓
	Period	86142 [s]	✓
ADCS	Maximum torque	1.1836e-8 [Nm]	✓
	avg. RCS propellant mass flow	1.7e-10 [kg/s]	✓
	min. RCS propellant mass	10.73 [g]	✓
	Reaction wheel torque	0.00513 [Nm]	✓
CDH	Data rate	11300 [bit/s]	✓
	Data per orbit	9.73e+8 [bit]	✓
	Min. storage size	1217 [MB]	✓
Power	Power generation	32.12 [W]	✓
	Solar cell efficiency at EoL	31.37 [%]	✓
	Solar array size	0.0749 [m²]	✓
	# of solar cells	96 [-]	✓
	Battery size	92.2 [Wh]	✓
Propulsion	Propellant mass	3.38 [kg]	✓
	Propellant volume	2.682 [dm³]	✓
	Tank mass	0.121 [kg]	✓
	Tank outer volume	3.21 [dm³]	✓
Structure	Occupied volume	11.716 [dm³]	✓
	Max. frontal area	0.08935 [m²]	✓
	Min. frontal area	0.05158 [m²]	✓
	Moment of inertia	0.16313 [kgm²]	✓
Thermal	Heat cold case	14.22 [W]	✓
	Heat warm case	25.4 [W]	✓
	Min. passive equilibrium temp.	-45.2 [° C]	✓
	Max. passive equilibrium temp.	-9.6 [° C]	✓
	Min. controllable equilibrium temp.	3.7 [° C]	✓
	Max. controllable equilibrium temp.	32.8 [° C]	✓
	Max. power consumption	11 [W]	✓
TTNC	Link budget remainder UHF, up	0.38 [dB]	✓
	Link budget remainder S-band, up	2.48 [dB]	✓
	Link budget remainder UHF, down	2.24[dB]	✓
	Link budget remainder S-band, down	21.17 [dB]	✓

Table A.4: Verification of the performance-indicative parameters in the tool compared to the DSE design for all subsystems and astrodynamic characteristics.

B

Questions asked during the interview with Marte Medina León

An interview was held with Marte Medina León, member of a team of young engineers, to determine their experience with Valispace. They had utilized Valispace as their main design platform during their Design Synthesis Exercise (DSE). The interview was analyzed to better understand the role of Valispace, to determine the impact of Valispace on project performance, and to understand the advantages, disadvantages, and limitations of the utilization of Valispace.

The DSE-project is a mission design project in which ten engineering students have ten weeks to hand in a pre-phase A report detailing the system design to a reasonable level of detail. The first five weeks concern planning, conceptual design and setting up requirements. Only the final five weeks are used to determine the key design parameters. The system design, which is the phase where Valispace can be utilized, is thus done in approximately 2000 man-hours. The interview will provide insight into the impact of utilizing Valispace for this context specifically.

A couple of questions were prepared in advance, each with a specific goal. These questions are listed here:

- *"Could you give a description of the need statement and goal of your space mission?"* This question was asked to gain insight into how well the space mission can be compared to other space missions. A space mission with an uncommon goal may change the context in which Valispace is used.
- *"Has anyone in your team used Valispace before?"* This question was asked to determine if the learning curve of Valispace for this space mission may have been less steep, reducing the impact on project performance. Perhaps, if a user had used Valispace previously, the utilization here was only considered advantageous due to the reduced learning time.
- *"How many man-hours did it take your team to learn Valispace?"* Knowing that the system design phase of the project is done in approximately 2000 man-hours, the estimated Valispace tutorial time of 2 hours would allocate 0.1% of human resources per person to this task, with the utilization perhaps requiring more. This question aims to better understand the learning time of Valispace.
- *"What are the advantages of utilizing Valispace?"* This question was asked rather early, to gain insight into how the user perceives the advantages of Valispace without leading them on. This question was asked to achieve one of the main goals of this interview, gain insight into the advantages of Valispace.
- *"What are the disadvantages of utilizing Valispace?"* Similar to the previous question, this question tried to find an unbiased perception of Valispace.
- *"How have you included the requirements of your mission in Valispace?"* Valispace has a number of modules, some of which can be used independently. The technical design is stored in the system design module, but the requirements may be omitted and instead checked manually from

the technical design. The requirements may also only be included at a higher level, with more detailed requirements either checked manually or following logically from constants or design models in the system design module. This question was asked to determine to what extent the requirements module was used, and how well this module is perceived.

- *"Does the tracking of the design properties of the space system take more or less time in Valispace compared to traditional methods?"* Besides the learning time of Valispace, using Valispace to track the design may require more time than other methods, such as maintaining Excel sheets, using programming code, or documenting the design. The aim of the questions was to gain insight into the time lost or gained when utilizing Valispace.
- *"What are the limitations of Valispace?"* This question was asked to gain insight into the perceived limitations of Valispace, both in the sense that features may be missing which are expected or would be nice, or perhaps there are flaws or bugs in the software. Such limitations can only be found by users of the software.
- *"Why do you think other teams do not implement Valispace?"* It is known the other teams which do the DSE simultaneously do not implement Valispace. Perhaps this is because the utilization is not deemed valuable, or other teams are unaware of the software. The person interviewed may have more insight into why the utilization of Valispace is rare during the DSE.
- *"Does the utilization of Valispace save your team time?"*, *"Does the utilization of Valispace lead you to a more accurate design?"*, *"Does the utilization of Valispace mitigate or introduce any risks in the design process?"* Finally, the user was asked these three questions to gain insight directly into how Valispace may affect the project performance metrics. No answers came directly from these questions though, as León indicated the usage of Valispace had a too complex impact on these parameters.

As the interview progressed, it was expected that certain tangents in the answers may lead to more questions which are subject-specific. These questions could not be prepared in advance. Nevertheless, the tangents in these answers may further suggest how Valispace is perceived by the interviewee, and extra time was taken into account for these extra questions to make sure all prepared questions were asked.

C

Log of sensitivity analysis bug in Valispace

During testing in Valispace using its sensitivity analysis feature, a bug was found. This bug seems to affect all sensitivity analyses where a dataset is involved in the dependencies.

C.1 Reproduction

Reproduction of this bug requires three valis. These valis can have any dimension, and in this case are selected to be dimensionless.

- A dependent vali 'A' with an arbitrary value (e.g. 1).
- A dataset vali 'B' with two available datapoints (e.g. 0,0, 1,1), stepwise interpolation and 'Same value as last available point' extrapolation
- A vali 'C' for which the value is computed as a function of the dataset vali at the dependent vali value, such that C = B(A).

Then, navigate to vali C sensitivity analysis and select vali A as the dependency, then click apply. This shows a pop-up error with the message *"Neither Quantity object nor its magnitude (1.0630000000000002) has attribute 'formula' error: AttributeError"*. The developer console gives the errors:

- *https://o566664.ingest.us.sentry.io/api/4508042664673280/envelope/?sentry_key=910087cc9d9a9571e841b91e8a6f0efd&sentry_version=7&sentry_client=sentry.javascript.angular-ivy%2F7.119.1 net::ERR_BLOCKED_BY_CLIENT*
- *TypeError: Cannot read properties of null (reading '0')*
- *TypeError: Cannot read properties of undefined (reading 'toString')*

D

Payload design

The payload designed during the DSE was largely inspired by the SAGE project[15]. It is a payload designed to function as a Sagnac-type interferometer to measure gravitational waves. The measurement principle is illustrated in Figure D.1, where one laser beam path is shown. The other laser beams travels in the other direction first. It functions by emitting two laser beams in two different directions(1), which are then reflected back to the payload(2). Each laser beam is then reflected from the payload to the other direction(3), and reflected back again(4). The lasers are then superimposed(5). This way, both laser beams have traveled the same path. If a difference in phase is measured, that difference would be caused by a gravitational wave. The components were selected identically to the SAGE mission, and the list of components is given in Table D.1.

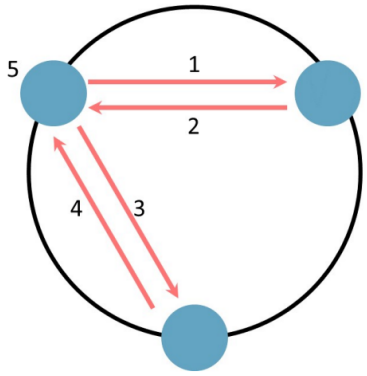

Figure D.1: The measurement principle of the payload. One of the laser beam paths is shown, with the numbers indicating the order of reflection. Adapted from the DSE final report[7].

Components	Mass [kg]	Power [W]
Main system		
Piezo stages	1.50	0.00
Laser	0.15	0.04
Couplers	0.17	0.00
Diodes	0.06	1.80
Phase modulator	0.44	0.20
Clock	0.10	2.50
Margins	0.48 (20%)	0.45 (10%)
Total main system	2.90	4.994
Transponders		
Laser	0.30	0.08
Margins	0.15 (50%)	0.04 (50%)
Total transponders	0.45	0.12
Telescope		
Primary mirrors	1.52	0.00
Secondary mirror	0.13	0.00
Margins	0.74 (31%)	0.00
Total telescope	2.39	0.00
Total	**5.75**	**5.114**

Table D.1: The components of the payload. The transponders are required for all passive connections, and the telescopes are required for all active connections. Adapted from DSE final report[7].