MODELING AND SIMULATION OF PERMANENT ON-ORBIT SERVICING INFRASTRUCTURES DEDICATED TO MODULARIZED EARTH-ORBITING PLATFORMS

BY

TRISTAN SARTON DU JONCHAY

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Advisor:

Assistant Professor Koki Ho

Abstract

This research aims to quantify the responsiveness and cost-effectiveness of permanent on-orbit servicing (OOS) infrastructures providing services to multiple serviceable platforms in coplanar medium Earth orbit (MEO) and/or geostationary orbit (GEO). The customer satellites are assumed to be made of elementary units (EUs). EUs are small standardized structural units capable of aggregating with each other and gathering the key functions of a typical satellite within the size of a 6U cubesat. Two OOS infrastructures are modeled in this research. The first one, called "*Without depot*" (WoD), includes a launch vehicle and a robotic servicer. The second infrastructure, called "*With Depot*" (WD), includes a launch vehicle, a robotic servicer and an orbital depot of EUs. This research is divided in two parts. The first part quickly developed a Simulink-based event-driven simulation framework to compare the responsiveness of WoD and WD, and provide some insight into their respective cost-effectiveness. The metrics used to quantify responsiveness for this first study are the service completion rate and the average waiting time before an EU is replaced over a 10-year period of operation. It is shown that WD is more responsive than WoD but is also likely to be more expensive to run. Based on this observation, the second part of this research developed a Python-based event-driven simulation framework capable of capturing a much larger trade space of the WD infrastructure than the Simulink framework does. The Python framework considers more accurate models and includes much more OOS design features, such as the number of servicers, more efficient service dispatch strategies and new space trajectories. For this second study, responsiveness is measured via the average working state of the population of customer satellites, which captures how well the satellites work based on three different failure severities and the number of failures. Cost-effectiveness is measured thanks to the average mass sent to orbit per year required to efficiently run the OOS infrastructures. It is first shown that there exist designs based on propellant optimal trajectories yielding similar levels of responsiveness as designs using Lambert-trajectory-based propellant-time-traded trajectories but at much lower costs. The second conclusion is that finding responsive and cost-effective OOS designs is not intuitive. This has to be done through an exhaustive exploration of the trade space of OOS, given the high number of design variables. This research si believed to be a critical milestone in the design of a responsive integrated space infrastructure dedicated to the development and prosperity of a new GEO/MEO economy.

Acknowledgments

My stay at the University of Illinois at Urbana-Champaign (UIUC) has been so far the best experience in my academic and personal life. Studying on such a big and multicultural campus grew me both as a person and a global citizen. It broadened my mind through enlightened discussions with people from different backgrounds, cultures and specializations.

On the academic side, I am so grateful for the high-level education I have been provided for the last two years by UIUC's Aerospace Engineering Department. In addition to being experts in their respective field of study, professors I met were also passionate about space engineering, which definitely drove me even more towards my goal of becoming a space explorer as an engineer and astronaut. Doing research as a Masters student taught me so much. This helped me gain self-confidence by finding out that I could solve problems I would have never suspected being able to solve. Finally, it taught me to be more rigorous and critical of my own work, two qualities I consider crucial to become an engineer.

I am so grateful to the people who gave me this opportunity and supported me throughout this wonderful experience. I would first like to thank the Institut Superieur de l'Aeronautique et de l'Espace (ISAE), Toulouse, which has been the starting point of those two years of my life. Studying at UIUC would have never been possible without the partnerships that ISAE is building with prestigious universities all around the world. I strongly encourage them to continue on this path and give more motivated students the same chance that I had to study abroad.

My research experience would have never been what it was without the guidance of Prof. Koki Ho, my thesis advisor. I thank him so much for giving me the opportunity to work on the exciting topic of on-orbit servicing, a field which is currently gaining momentum in the industry. Through my interactions with him, I have realized how pleasant it was to have an adviser who fully trusts your work while at the same always testing your rigor. In particular, he taught me to always appreciate the big picture of any project I am involved in by stepping back from the details. This definitely helped me a lot when it was time to publish my first engineering paper. My experience at UIUC would have

never been so rich without working with Koki.

While at UIUC, I became friends with many international and American people with whom I shared incredible experiences. I thank them for giving me insights into their respective cultures and countries, and learn more about our world. I am also grateful to them and my friends in France for supporting me in stressful periods and wishing me the best of luck to follow my ambitious dreams.

Last but not least, I would like to thank the members of my family who encouraged me, throughout my academic life, to follow my dreams and never give up. More particularly, I would like to thank my father for having indirectly taught me to be determined, hard-worker and creative to succeed. Finally, I don't want to miss this opportunity to thank my mother for showing us that life is worth living despite one's sickness.

Table of Contents

List of Tables . . . . . **vii**

List of Figures . . . . . **viii**

List of Abbreviations . . . . . **x**

List of Symbols . . . . . **xi**

Chapter 1 Introduction **1**
1.1 Background and Motivation 1
1.2 Problem statement 2
1.3 Thesis Outline 3

Chapter 2 Literature Review **4**
2.1 On-Orbit Servicing as a Disrupter of the Space Industry 4
2.2 Satellite Serviceability 6
2.3 On-Orbit Servicing Concepts of Operations 7

Chapter 3 Methodology and Modeling **9**
3.1 Queueing Theory-based Even-Driven Simulations applied to On-Orbit Servicing 9
3.2 Elementary Units Modeling 15
3.3 Customer Satellites Modeling 16
3.4 On-Orbit Servicing Infrastructures Modeling 17
3.4.1 On-Orbit Servicing Infrastructures Overview and Decision Making 18
3.4.2 Launch Vehicle Model 20
3.4.3 Servicer Model 21
3.4.4 Depot Model 36
3.4.5 Service Dispatch Strategies 37

Chapter 4 Trade-Off between "Without Depot" and "With Depot" Infrastructures **41**
4.1 Simulation Settings and Metrics 41
4.1.1 Simulation Settings 41
4.1.2 Metrics 42
4.2 Simulation Results and Analysis 42
4.2.1 Responsiveness of the servicing infrastructures 43
4.2.2 Insight into the cost of the servicing infrastructures 46
4.2.3 Sizing of the orbital depot 48

Chapter 5 Trade Space Exploration of the "With Depot" Infrastructure 51
5.1 Simulation Settings and Metrics 51
5.2 Simulation Settings 51
5.3 Metrics 52
5.4 Propellant Optimal Trajectories 52
5.4.1 Scenarios with one servicer 53
5.4.2 Scenarios with three servicers 56
5.4.3 Scenarios with five servicers 59
5.5 Propellant-Time-Traded Trajectories 62
5.5.1 Scenarios with one servicer 63
5.5.2 Scenarios with three servicers 66
5.5.3 Scenarios with five servicers 69
5.6 Comparisons and Discussions 72
5.6.1 Comparison between propellant optimal and propellant-time-traded trajectories 72
5.6.2 Trade-off discussion based on propellant optimal trajectories 74

Chapter 6 Conclusion 76

Appendix A Satellite Subsystems Failure Data 79

References 80

List of Tables

3.1 Failure rates of each type of EU 16

4.1 Parameters common to both servicing infrastructures 41
4.2 Parameters specific to the “Without Depot” infrastructure 42
4.3 Parameters specific to the “With Depot” infrastructure 42

5.1 Parameters specific to the “With Depot” infrastructure 52
5.2 Characteristics and performance of the designs A, B, C, D introduced in figure 5.13 73
5.3 Performance of designs A, B, C, D using the other trajectory paradigm 73
5.4 Performance of designs A, B, C, D using the other trajectory paradigm 74

A.1 Failure rates of each type of EU 79

List of Figures

3.1 State-based schematic of the $M/M/1/6/6/FIFO$ queueing model . . . 12
3.2 Model of a finite source of platforms integrated in an $M/M/1/6/6/FIFO$ queueing model for the validation of Simulink as a viable simulation platform . . . 13
3.3 Mean response time W of the servicing system under Simulink . . . 14
3.4 Mean response time W of the servicing system under Python . . . 14
3.5 Interaction of the EUs of one of the platforms considered in this research . . . 17
3.6 Overview of the WoD and WD infrastructures . . . 18
3.7 Concept of operation and decision making process of the WoD infrastructure (PF means "Platform") . 19
3.8 Concept of operation and decision making process of the WD infrastructure (PF means "Platform") . 20
3.9 An example of servicing operation and the delta Vs defined for each travel leg . . . 24
3.10 Notations for the sizing of the propellant tank . . . 25
3.11 Calculation of ΔV^c_{ref} . . . 27
3.12 Example of a phasing maneuver-based rendezvous . . . 31
3.13 Variation table of t^p_{travel} studied as a function of the travel angle γ . . . 34
3.14 Plot used to find the range of admissible travel times . . . 35
3.15 SR service dispatch strategy . . . 38
3.16 First step of the AWSM strategy . . . 39
3.17 Second step of the AWSM strategy . . . 39

4.1 Pareto fronts of the servicing infrastructures . . . 44
4.2 Service completion rate as a function of the parking orbit . . . 45
4.3 Average waiting time before replacement as a function of the parking orbit altitude . . . 45
4.4 Propellant consumed by the servicer per service for both WoD and WD when there are as many GEO satellites as MEO satellites, and when there are more GEO satellites than MEO satellites . . . 47
4.5 Propellant consumed by the servicer per service for both WoD and WD when there are as many GEO satellites as MEO satellites, and when there are more GEO satellites than MEO satellites . . . 48
4.6 Sensitivity of the service completion rate with the depot capacity for a parking orbit of the servicer at 27,000 km . . . 49
4.7 Sensitivity of the average waiting time before EU replacement with the depot capacity for a parking orbit of the servicer at 27,000 km . . . 50

5.1 Data clouds for one servicer and propellant optimal trajectories . . . 53
5.2 Data analysis for one servicer, propellant optimal trajectories and AWSM strategy . . . 54
5.3 Data clouds for three servicers and propellant optimal trajectories . . . 57
5.4 Data analysis for three servicers, propellant optimal trajectories and AWSM strategy . . . 58
5.5 Data clouds for five servicers and propellant optimal trajectories . . . 60
5.6 Data analysis for five servicers, propellant optimal trajectories and AWSM strategy . . . 61
5.7 Data clouds for one servicer and propellant-time-traded trajectories . . . 63
5.8 Data analysis for one servicer, propellant-time-traded trajectories and AWSM strategy . . . 65
5.9 Data clouds for three servicers and propellant-time-traded trajectories . . . 67
5.10 Data analysis for three servicers, propellant-time-traded trajectories and AWSM strategy . . . 68

5.11 Data clouds for five servicers and propellant-time-traded trajectories 70
5.12 Data analysis for three servicers, propellant-time-traded trajectories and AWSM strategy 71
5.13 Pareto fronts obtained with propellant optimal and propellant-time-traded trajectories 72
5.14 Average working state versus mass sent to orbit per year (propellant optimal trajectories) 75

List of Abbreviations

AWSM	Average working state maximization service dispatch strategy
DARPA	Defense Advanced Research Projects Agency
EU	Elementary Unit
FIFO	First-In-First-Out
GEO	Geostationary Orbit
HST	Hubble Space Telescope
JAXA	Japan Aerospace Exploration Agency
MEO	Medium Earth Orbit
NASA	National Aeronautics and Space Administration
OOS	On-Orbit Servicing
OR	Operations Research
QT	Queueing Theory
SR	Servicer rotation service dispatch strategy
WD	With Depot
WoD	Without Depot

List of Symbols

ΔV	Delta V
g_0	Sea-level gravitational acceleration
I_{sp}	Specific impulse
μ	Earth's gravitational parameter
r_f	Radius of final orbit
r_i	Radius of initial orbit

Chapter 1

Introduction

1.1 Background and Motivation

Today, humankind is at the dawn of a new era of space exploration driven by a dramatic development in space technology. Together, the public and private actors of the space sector are investing a vast amount of resources to promote military, scientific and communication missions as well as pushing further the frontiers of robotic and human space exploration. Robotic on-orbit servicing (OOS) will help us towards these goals by significantly simplifying the overwhelming logistics currently required for space missions. From on-orbit manufacturing and assembly to upgrade and repair, through maintenance (preventative and corrective) and replenishment of consumables, OOS is expected to be a game-changing alternative to the current replacement scheme of Earth-orbiting platforms.

In the history of space exploration, multiple OOS missions have been performed or proposed for repair or upgrade of existing spacecraft. The classical on-orbit servicing example is the repair and upgrade missions of the Hubble Space Telescope (HST) by the crew of the Space Shuttle. Although these missions were remarkable and much cheaper than HST itself, they remained very expensive and, more importantly, incurred high levels of risk for the astronauts. More recently, robotic systems have been considered to be more cost-effective than human servicing missions. Several programs such as JAXA's Manipulator Flight Demonstration program [1], NASA's Robotic Refueling Missions [2] and DARPA's Phoenix program [3] have developed and demonstrated potential robotic hardware for the purpose of OOS, consequently increasing the technology readiness level of those technologies significantly. In parallel to hardware development, these agencies are trying to define the frame and the actors of OOS for the development of the future space commerce.

Developing a servicing infrastructure based permanently in orbit to provide responsive services will cost time and money. Consequently, to test and explore the potentials of robotic technologies in the early days of OOS, the first investigated paradigm is to build and launch a robotic spacecraft to replenish the customer satellites in propellant when several satellites are in need. The advantage in this business case is that the flight path of the spacecraft may

be optimized before launch to provide the best value to the replenishment operation. However, services cannot be provided in a timely manner, and the spacecraft is usually discarded in a graveyard orbit when the mission is over. Reusability is a concept that rocket manufacturers are investigating as a viable way to cut launch costs dramatically. This concept, however, may also be beneficial for OOS in the future by considering infrastructures permanently based in orbit and providing various types of on-demand and almost immediate services to defective customer platforms.

In addition, OOS is all the more likely to change the paradigm of space commerce and logistics that the space platforms are designed to be serviceable. Serviceability can be significantly improved thanks to the "plug-and-play" concept. More specifically, several projects today deal with designing and manufacturing small standardized structural units gathering the key functions of a typical satellite within the size of a 6U cubesat and capable of aggregating with each other. These elementary units (EUs) have the potential to change the current paradigm of satellite development by decoupling and simplifying the design processes of the bus and the payloads.

Motivated by this background, this research develops models and simulations of servicing infrastructures permanently based in orbit to provide responsive repair services to modularized Earth-orbiting platforms. The simulations can be used to find the most appropriate design and sizing of the OOS infrastructure to provide services to a given population of satellites at a lower cost and higher responsiveness.

1.2 Problem statement

This research assesses the responsiveness of two different designs of OOS infrastructures by simulating one Simulink-based model and one Python-based model over the equivalent of ten years of operation. In order to yield realistic results, the launch and platform failures are taken into account in those models [4] [5] [6]. The first infrastructure considered in this research includes a launch vehicle and a robotic spacecraft, called *servicer*, used to provide the spare EUs to the customer platforms. The second infrastructure includes a launch vehicle, one or more servicers and an orbital depot of spare EUs. Because of the presence or not of the orbital depot in those servicing infrastructures, they are referred to as "Without Depot" (WoD) and "With Depot" (WD) respectively.

The mathematical framework of queueing theory (QT) is used to model the management of the platform failures by the servicing infrastructures and validate the event-driven simulation frameworks implemented in Simulink and Python. Queuing theory is generally applied in urban operations research. One application of spatial queue is to find

the location of a fire station providing the most responsive answers to fire emergencies randomly occurring throughout a city [7]. By analogy to the previous example, the queuing theory-based simulation frameworks are used in this research to find for both OOS infrastructure designs (WoD and WD) the best sizing to provide the most responsive answers to platform failures. The uniqueness of the approach carried out in this research is to use the orbital transfer instead of the ground transportation for dispatches, which adds complexity in the problem. Our analysis simulates over time the relative motion of several spacecraft on different orbits and quantifies the responsiveness and cost-effectiveness of different sizings of both OOS infrastructure designs.

This research is divided in two parts. For the first part, a Simulink-based event-driven simulation framework was quickly developed to perform a trade-off between the WoD and WD designs mainly based on responsiveness considerations. This was necessary to confirm the intuition that WD is more responsive than WoD. For the second part, a more detailed Python-based event-driven simulation framework was implemented to account for a larger trade space of the WD infrastructure and perform more thorough analysis of its responsiveness and cost-effectiveness.

Although both simulation frameworks are flexible enough to simulate the failures and orbital characteristics of a very large number of satellites orbiting in the equatorial plane of the Earth, the scenarios simulated in this work considered only six satellites in geostationary orbit (GEO) and medium-Earth-orbit (MEO).

1.3 Thesis Outline

The remaining of this thesis is organized as follows: Chapter 2 provides a literature review of the research. Chapter 3 introduces QT, a thorough description of both infrastructure designs, and the models used in the Simulink and Python-based simulation frameworks. Chapter 4 presents the settings and results of the simulations run to perform the trade-off between WoD and WD based mainly on responsiveness considerations. Chapter 5 focuses on the trade space exploration of the WD infrastructure design using the Python framework. Finally, chapter 6 concludes the thesis with key contributions and recommendations for future work.

Chapter 2

Literature Review

2.1 On-Orbit Servicing as a Disrupter of the Space Industry

Since Sputnik first orbited the Earth in 1957, the satellite industry has considerably grown and, along with dramatic technology development, has improved humans' lives significantly. In particular, GEO satellites have played an important role in the emergence of highly efficient telecommunication systems which our societies heavily rely on. Satellite services generated in 2014 a revenue of $122.9B which was about 38 % of the revenues generated by the entire satellite industry [8]. A GEO telecommunication satellite for instance, depending on the number of on-board transponders, generates in average $40 to $50 millions per year which add up to $750 millions over its 15-year estimated lifespan [9]. But in order to survive the harsh space environment and be resilient to the variations of the space market for 10 to 15 years, space engineers have been designing heavier and more complex commercial platforms over time. Consequently, today's satellites usually incorporate many redundancies, large propellant tanks and sophisticated technologies driving up their mass, cost and development time. This ends up in large insurance premiums limiting initiatives for disruptive but risky architecture innovations that could make satellites more flexible with respect to uncertainty. Therefore, unlike terrestrial vehicles, space systems cannot adapt to the evolution of the market due to their rigid designs, and their lifetime cannot be easily extended because regular maintenance is not included in their lifecycle. If satellites were physically accessed in orbit to receive upgrade and repair among other operations, this would free the satellite designers of the current stringent design constraints, leading to simpler designs more flexible with respect to the uncertainties inherent to the space market and space environment.

On-orbit servicing precisely intends to be the disruptive industry that will provide a way to physically access, upgrade, and modify a spacecraft. In other words, OOS intends to provide flexibility to space systems. Saleh defines the "flexibility of a design as the property of a system that allows it to respond to changes in its intial objectives and requirements - both in terms of capabilities and attributes - occurring after the system has been fielded, i.e., is in operation, in a timely and cost-effective way" [10]. This definition gives three main objectives for the design of an OOS infrastructure. Its first goal is to provide services in line with the way the engineers have defined the flexibility

of the design of the serviced spacecraft. For instance, there would not be any value in designing an OOS infrastructure which cannot provide refueling services to easily-refueled spacecraft. The literature lists various types of servicing missions ranging from on-orbit refueling and re-boost to subsystem deployment and repair [6] [11] [12]. The second objective of an OOS infrastructure is to provide affordable services that customers may consider as an alternative to launching a brand new satellite to orbit. Hastings, Putbrese and La Tour shows that the willingness of a customer to pay for on-orbit services depends on the balance between the spacecraft serviceability and the cost of the servicing operations. In particular, they show that OOS will need to be offered at the lowest cost possible and that the owners of platforms designed to be efficiently serviced are shown to be likely willing to pay up to a moderate cost [13]. Finally, given that today's satellites generate high levels of revenues a year, a too long waiting time before the actual service is provided would incur significant cashflow losses. Consequently, an OOS infrastructure must be able to provide services in a timely manner.

On-orbit servicing is believed to be a game-changing alternative to the current paradigm of building and launching a new spacecraft to replace a defective one. However, initiating the momentum the industry needs to develop the technology required to implement a reliable OOS infrastructure costs time and money. Seventy years have run since space designers first imagined OOS, but governmental agencies have had only a few opportunities to test in space the robotic technologies they had developed [1] [2] [3]. Today, a few private companies, such as Orbital ATK and MDA, have started developing life-extension servicers [14] [15]. Life extension through refueling and re-boost is critical but those mission types are not the only ones. In fact, the trade space of the servicing options is large. Therefore, it is necessary to assess every single candidate servicing mission in terms of the three objectives previously outlined before considering developing the appropriate technologies. Technology development to perform particular servicing missions is likely to be expensive and long. Consequently, as recommended in the literature, early OOS projects should be run on behalf of an agency such as NASA or be public-private ventures where the public agency provides guidance in the development and operation of the servicing infrastructure by private firms [13].

The development of an OOS infrastructure and a shift in the current satellite design paradigm cannot be performed independently. OOS can provide benefits to space platforms as long as their design is modified to take advantage of most of the provided services.

2.2 Satellite Serviceability

DARPA's Phoenix program partnered with private companies to investigate new disruptive ways to break apart the cost/mass/performance equation having driven the traditional design of satellites for decades. No matter the mass of the space system which is considered, from cubesats to large GEO communication platforms, the internal configuration of the typical subsystems is identical. Consequently, the mass/cost behavior is the same whether we consider a small or large satellite. In particular, the cost increases linearly with the mass or equivalently, the development budget of a satellite will determine its final mass [16] [17]. Unfortunately, performance usually follows the same trend as the cost and mass, explaining why large systems are in general more efficient than smaller ones.

Phoenix's objective was to find a way to decrease the slope of the cost/mass linear relationship while keeping the level of performance of a traditional satellite. In order to do so, Phoenix's designers drew their inspiration from biological systems whose complex behaviors are the results of a perfect symbiosis between identical or almost identical cells. They ended up imagining a new concept of elementary units (EUs), called *satlets*, capable of aggregating with each other to combine their limited resources and create flexible platforms as efficient as traditionally-designed space systems. According to Phoenix's engineers, this new design paradigm would leverage standardization and economies of scales by producing high volumes of identical systems and improve manufacturing and assembly processes, thereby decreasing the costs dramatically. In addition, it would improve the flexibility of the space platforms with respect to the uncertainties of the market and the space environment by allowing for easy changes in requirements over time and easy replacement of EUs.

The research done under the Phoenix program led to various cellularization paradigms extending from *heterogeneous* EUs to *homogeneous* EUs [18] [19]. Each heterogeneous EU includes one single function of a typical satellite (propulsion, attitude sensing, communication, etc.) . On the other hand, homogeneous EUs encompass all the functions of a space platform, like cubesats with the difference that EUs can be assembled with each other and cubesats cannot. The literature recommends the use of at least one central type of EU performing computation and data processing, and an actuation type of EU in the form of either reaction wheels or thrusters and local computing resources [18]. However, some actors such as Novawurks have decided to focus their development efforts toward the homogeneous paradigm to take advantage of the economies of scales [19].

The satlet paradigm has the potential to make the design of future satellites simpler and more flexible. But it could also drive down the complexity of future OOS infrastructures thanks to the standardized interfaces between the EUs and between the OOS infrastructure and the EUs.

2.3 On-Orbit Servicing Concepts of Operations

Before the satellite industry starts shifting the current design paradigm towards a more flexible alternative, the first tasks of OOS are likely to be observation, re-boosting, repositioning and de-orbiting, which a priori do not require a great deal of serviceability from the customer platform. The industry has already initiated the design of servicers capable of extending the life of satellites thanks to re-boosting, an operation requiring the spacecraft to dock with the customer satellite and perform an impulsive maneuver in this tandem configuration [14] [15].

The literature has been interested in exploring the utilities of various servicing options, including repair and refueling, based on the expected risk and revenues associated with such operations, as well as the uncertainties inherent to the end-user market of the customer satellites and inherent to the space environment [9] [20]. Other works explore the sequencing of accurate trajectories for the purpose of servicing several satellites in a row [9] [21]. These studies assume explicitly or implicitly different kinds of OOS concept of operations, corresponding to different stages of technology development. The next paragraphs introduce two different concept of operations which were explicitly identified or implicitly assumed in the past literature.

The first OOS paradigm consists in building and launching a new servicer whenever there are needs for services from one or more customer satellites [9]. Before engaging the resources to send a servicer to orbit, decision makers choose the space platforms to be operated on and the optimal service sequencing based on the risk and revenues of the services that are needed. Once in orbit, the servicer realizes its mission as predefined before launch, and is finally discarded in a graveyard orbit when its propellant level is too low. The main advantage of this concept is that the flight path may be optimized to maximize the revenues of the OOS operator at the lowest risk. This is because there is no servicer directly in orbit to provide on-demand and immediate services. However, this also means that this paradigm will not really be sustainable because servicers are not reusable, eventually increasing the number of debris in orbit. This paradigm is likely to be the business case chosen for the early days of OOS due its rather simple concept of operations.

Later in the future, when autonomous robotics technologies have been proven in space, and engineers have shifted the current satellite design towards a more flexible and serviceable one, servicing systems other than servicers will be sent to orbit to enable the sustainability of OOS. This will be achieved through the use of orbital depots to store the spares and propellant needed to provide the repair services and replenish the propellant tanks of the servicers [20]. At least one propellant depot frequently replenished from Earth would significantly improve the OOS sustainability. This would allow one or more servicers to provide continuous observation, re-boosting, repositioning and de-orbiting services. Finally, the past literature believes that a permanent and sustainable OOS infrastructure around Earth would

bring new perspectives to the space commerce and facilitate human space exploration [22].

Chapter 3

Methodology and Modeling

This chapter will introduce the methodology followed in this research as well as the models considered in the Simulink-based and Python-based simulation frameworks. Section 3.1 describes the methodology employed in this work. Section 3.2 details how the EUs are modeled in both simulation frameworks. Section 3.3 builds on the material presented in the preceding section to model the customer space platforms. Finally, section 3.4 introduces in detail the OOS infrastructures considered in this thesis for each simulation framework.

3.1 Queueing Theory-based Even-Driven Simulations applied to On-Orbit Servicing

This section introduces Operations Research (OR) and Queueing Theory, and how they are used in this research as rigorous tools for the analysis of OOS infrastructures.

Operations research is defined by the Institute of Operations Research and the Management Sciences as "the application of scientific and mathematical methods to the study and analysis of problems involving complex systems" [23]. Other said, OR develops tools to help make better decisions in various applications such as network optimization, supply chain management, transportation or scheduling among others. This field of research encompasses a wide range of problem-solving techniques and methods applied in the pursuit of improved decision-making and efficiency, such as mathematical optimization, stochastic-process models, simulation and QT. Thanks to the use of these techniques, OR has the potential to accurately model and simulate the processes driving the supply chain and service demand management likely to be involved in future OOS infrastructures. Of particular interest in this research is the use of QT.

QT is a mathematical framework used to describe the stochastic service of randomly arriving customers in a servicing system in order to evaluate its performance. Discrete event-driven processes such as the arrival process in a queue are modeled by this theory which provides convenient analytical results about, for instance, the average waiting

time or number of entities in the queue [24] [25]. However, when the processes become too complex to be analyzed analytically, simulations capable of modeling discrete events such as the random failures of a satellite are required.

One application of QT is to find, given the spatial probability distribution of fire occurrences in a city, the most responsive design of a fire station in terms of its position in the city and the number of fire engines. In the context of this urban example, the arrivals are the fire occurrences throughout the city. In OOS, the arrivals correspond to the failures or demands for service from the EUs, whereas the service times correspond to the times the servicer spends to service the EUs. If the servicer does not have the spares in its own stock, the time to wait until the next launch (in WoD architecture) or the time to procure spares from the depot (in WD architecture) is also included in the service time. In order to introduce QT with its own terminology, the next paragraphs refer to the servicing units as the "servers". The queueing models are classified with Kendall's notation:

$$A/B/m/K/n/D \tag{3.1}$$

where:

A customer arrival process;

B distribution function of the service time;

m number of servers;

K maximum capacity of the queue;

n population size or number of sources of customer (the customer platforms in this research);

D service discipline.

In the simplest queueing models, A and B are modeled by memoryless probability distribution functions (denoted by M), when the customer arrival follows a Poisson process and the service time follows an exponential distribution. For such processes, this probability distribution function enables the derivation of very useful analytical results for simple queueing models. However, when the probability distribution of A and/or B is not memoryless, the models are not so easily solved and event-driven simulations may be of interest.

A typical service discipline for the Kendall's D-notation is the *First-In-First-Out* scheme (FIFO), used in the Simulink-based simulation framework to model how the platform failures are managed by the servicing infrastructures. More specifically, with the FIFO scheme, customers are served according to their arrival time: the server processes the customer which has waited the longest time in the queue. In the Python-based simulation framework, more sophisticated service disciplines are modeled to increase the responsiveness. They will be discussed in section

3.4.

The scenarios we consider are different from the simple queueing model for the following reasons, which make the application of analytical results to our problem difficult. First, the considered servicing infrastructures comprise several servicing systems (launch vehicle, robotic servicer, orbital depot) dependent on each other. Consequently, they cannot simply be seen as one server as in QT terminology. Furthermore, although the failures of the platforms can be modeled as a Poisson process, the servicing time depends on the relative position between the servicer and the failed platforms, consequently making it dependent on the timing of the failures. Finally, the size of the population of EUs of the platforms to be serviced is finite. Those scenario features limit the use of analytical results.

To simulate and compare the servicing infrastructures, two different simulation frameworks are developed. The first simulation framework uses the SimEvents toolbox of Simulink. Simulink is a very intuitive tool based on blocks interacting with each other via wires. The models are simulated with a continuous-time step-size-varying solver. The SimEvents toolbox also offers various blocks of interest to model queueing problems, such as queue blocks with or without priority and entity generators based on a time or event basis, with the entities being for instance the random failures of a platform. Those blocks can be used to model the EU-based platforms as well as the servicing infrastructures. However, although learning and using Simulink is fast and intuitive, the framework developed with SimEvents in not flexible enough in terms of the design of the OOS infrastructure. Consequently, it was used only to confirm the intuition that WoD is not as much responsive as WD and is likely to cost more. The second simulation framework was developed with the goal of capturing a much larger trade space of the WD infrastructure than the one offered by the Simulink framework. It has been developed in Python with the Simpy module dedicated to event-driven simulations.

Thanks to random number generators, event-driven simulations can in theory reproduce the results of QT. In order to validate SimEvents and Simpy as appropriate and accurate simulation platforms, an $M/M/1/6/6/FIFO$ queueing model was implemented and simulated to verify that the outputs were matching the theoretical results. The following paragraphs presents the case study for the validation of SimEvents and Simpy as viable simulation platforms.

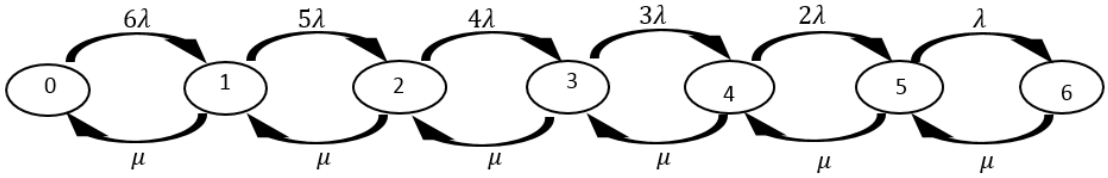

Figure 3.1: State-based schematic of the $M/M/1/6/6/FIFO$ queueing model

Figure 3.1 introduces a convenient schema used to derive the formulas of interest. The parameter λ is the arrival rate of one customer in the servicing system. The parameter μ is the rate at which the server accomplishes one service. For a detailed demonstration of those derivations and an explanation of Little's laws used to determine the average performances of a servicing system, the reader should refer to [24] [25].

In figure 3.1, the ovals represent the states of the servicing system corresponding to the number of customers waiting for or receiving a service. The transitions between those states as well as their rates are represented by the arrows and the parameters above or below. The steady-state probability of being in state $k = 0..n$, where $n = 6$ here, follows the law:

$$P_k = \frac{\frac{n!}{(n-k)!}\rho^k}{\sum_{i=0}^{n} \frac{n!}{(n-i)!}\rho^i} \tag{3.2}$$

where ρ is the traffic intensity equal to the ratio λ/μ.

Once this state probability is known, Little's laws give simple analytical results for the characteristic average quantities of the queueing model. The mean number of customers in the system and in the queue are respectively

$$L = \sum_{i=0}^{n} iP_i \tag{3.3}$$

$$L_q = \sum_{i=1}^{n} (i-1)P_i \tag{3.4}$$

The waiting time in the queue and the mean response time respectively follow

$$W_q = \frac{1}{\mu}\left(\frac{L}{(n-L)}\frac{\mu}{\lambda} - 1\right) \tag{3.5}$$

$$W = W_q + \frac{1}{\mu} \tag{3.6}$$

This queueing model with $n = 6$ can be interpreted as an on-orbit servicing infrastructure providing services to 6 platforms. The failures of each individual platform would follow a Poisson process and their interval will have a mean time set arbitrarily to 36 months ($\lambda = 1/36$ failure/month). Modeling the service time by the infrastructure with an exponential distribution is more delicate. However, since the service time includes the launch vehicle bringing the spares from Earth's surface to orbit, and since it can be assumed that the interval between two launches is much larger than the time for the robotic servicer to provide the services, the parameter can be assumed to be equal to the launch rate. Modeling the launch windows as a Poisson process can be questionable, but for the purpose of this example, we will make this assumption. For an arbitrary time of 6 months between two launches ($\mu = 1/6$ launch/month), the values of the average quantities of interest are $L = 1.59$, $L_q = 0.855$, $W = 12.97$ months, and $W_q = 6.97$ months. The $M/M/1/6/6/FIFO$ queueing model was implemented under both simulation platforms and simulated for a sufficiently long time so as to reach steady state and be able to compare the numerical results with the analytical results.

The SimEvent toolbox does not include built-in blocks to model finite sources of customers. To model the fact that the customers come back to their source once they have been served, the basic blocks of Simulink and a feedback loop are used as illustrated in figure 3.2.

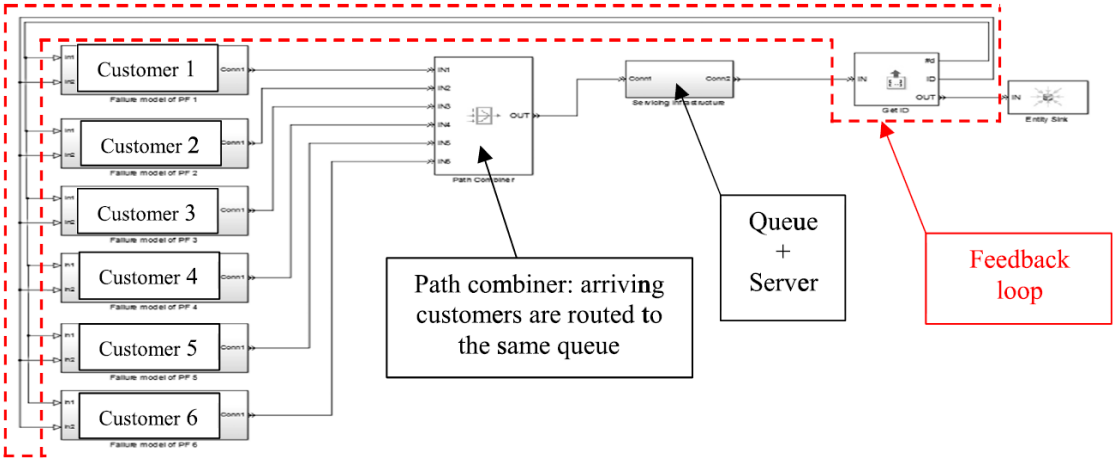

Figure 3.2: Model of a finite source of platforms integrated in an $M/M/1/6/6/FIFO$ queueing model for the validation of Simulink as a viable simulation platform

Simpy module in Python offers many convenient functions to model a finite source of customers and as many servers as desired. The $M/M/1/6/6/FIFO$ queueing model was also modeled and simulated over a sufficiently long time to reach steady state and compare with the analytical results. The average waiting time before service of the customers is plotted over time in figure 3.3 for the Simulink-based framework and in figure 3.4 for the Python-based framework.

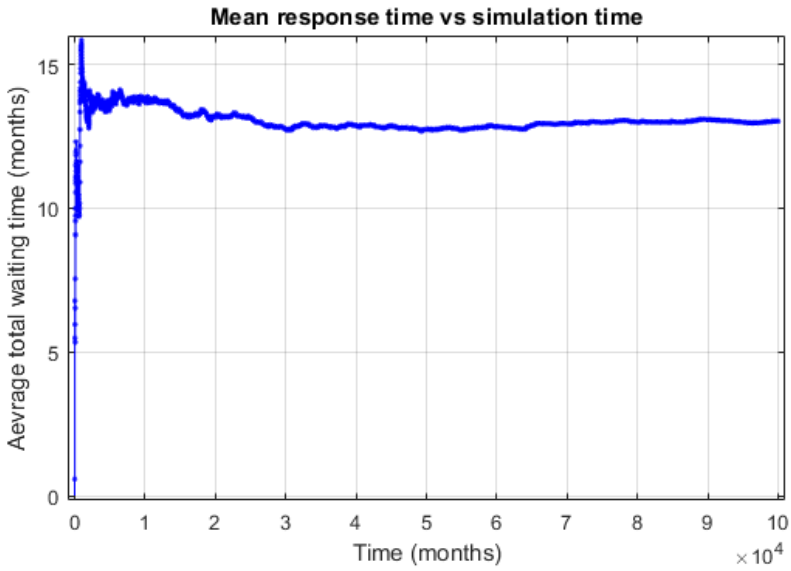

Figure 3.3: Mean response time W of the servicing system under Simulink

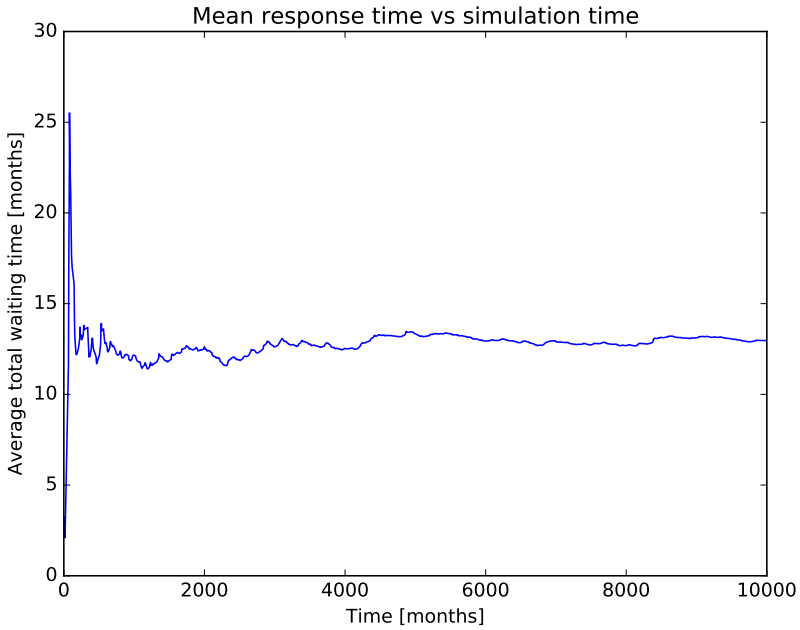

Figure 3.4: Mean response time W of the servicing system under Python

After 10000 months of simulation time, the mean response times are 13.03 months and 12.97 months with the Simulink and Python models respectively, which are very close to the theoretical 12.97 months value. Therefore, those simulation platforms are validated as viable to simulate OOS infrastructures.

3.2 Elementary Units Modeling

As stated in the introduction, EUs have the potential to greatly simplify the design of future Earth-orbiting satellites destined to be serviced, by giving them an intrinsic plug-and-play property. This subsection describes the different types of EU as well as the models of their failure probability.

Reference [6] identifies 8 main categories for the breakdown of a satellite: the propulsion system, the attitude control system, the electric power subsystem, the thermal power subsystem, the structural subsystem (comprising materials, structural integrity, structural protection, etc.), the deployment mechanisms, the Control and Data Handling subsystem (CDH) and the RF communication subsystem. Regarding those functions, 4 types of EUs are considered. The first type of EU (EU1) is defined as the mother unit of the bus of an EU-based platform dedicated to central attitude sensing and computation, CDH, RF communications. The second type of EU (EU2) is defined as an actuation EU capable of both propulsion and attitude control. The third type (EU3) is defined as the power plant of the platform comprising the power generation and distribution subsystems. It is assumed that the solar panels of EU3 are directly on its surface and that, consequently, there is no solar array deployment system. The last type (EU4) is defined to deal with thermal management.

Separating the functions of a platform bus in such a way is convenient to benefit from the failure data found in the literature. Reference [5] proposes a rigorous analysis and classification of the statistical data drawn from the SpaceTrack database. Scorecards are defined for each subsystem of a satellite with various interesting data. More specifically, we use the average times before fatal, major and minor failures as defined in reference [5] so as to deduce the failure rate of each EU type. The probability of failure of a system can be modeled with a good fidelity by a Poisson process as long as the next failure is independent of the previous ones. Thus, this paper models the failure time of each EU type as a Poisson process with a parameter equal to the average failure rate deduced from reference [5]. The data of interest from reference [5] can be found in table A.1 in Appendix.

Table 3.1 summarizes the average times before a failure occurs as well as the systems identified in reference [5] and associated with the relevant EUs. The average times may seem low, but that is because this data is deduced from all types of failures (minor, major, fatal) defined in reference [5] and from a population of large, complex satellites. The Simulink-based simulation framework assumes that one failure of any type induces the complete failure of the EU. Other said, all failures are assumed to be fatal. However, the Python-based simulation framework makes the distinction between these failure types in the analysis of the responsiveness of the WD infrastructure.

Table 3.1: Failure rates of each type of EU

EU type	*Functions*	*Systems studied and classified in reference* [5]	*Average time before failure* [months][1]
1	Central computation, attitude sensing, communication, data handling and processing	- Beam/Antenna operation/Deployment - Payload instrument/Amplifier/On-board data/Computer/Transponder - Telemetry, tracking and command	1.38
2	Station keeping, attitude control	- Gyro/Sensor/Reaction wheel - Thruster/Propellant	4.76
3	Power generation and distribution	- Battery/Cell - Electrical distribution	2.02
4	Thermal management and mechanisms	- Thermal management and other mechanisms	1.58

[1] Calculated from a probability property stating that the sum of independent random variables modeled by exponential probability distribution functions can itself be modeled by an exponential distribution function with its parameter equal to the sum of the parameters of the summed random variables.

3.3 Customer Satellites Modeling

For the purpose of the simulations, the platforms are attributed an ID, an orbit (MEO or GEO), an initial angular position and a set of EUs of each type. The EUs of a specific platform are also given an ID so as to identify them individually in the platform architecture.

For simplicity, the platforms are assumed to be identical. A platform design in terms of EU type and number is determined as follows. First, it is assumed that the EU3s and EU4s work in symbiosis. The EU3 provides power to the EU4 which in returns provides thermal management to the former. Secondly, each EU1 and EU2 is assumed to be coupled with one EU3 and one EU4. Finally, it is assumed that the payloads of a platform need at least one EU3 and one EU4. More specifically in this paper, each platform is assumed to have only one EU1, two EU2s and a payload which requires only one EU3 and one EU4. According to the above assumptions, this platform design leads consequently to a bus made of one EU1, two EU2s, four EU3s and four EU4s. Figure 3.5 shows how the EUs of a platform interact.

In the Python-based simulation framework, the average working state is modeled as an attribute characteristic of each customer satellite. Its value is calculated over time whenever a failure occurs. If there are N_{fatal}, N_{major} and N_{minor} failures of fatal, major and minor failures recorded for a given satellite, its working state W is calculated as

$$W = (w_{fatal})^{N_{fatal}} \times (w_{major})^{N_{major}} \times (w_{minor})^{N_{minor}} \tag{3.7}$$

where $w_{fatal} < w_{major} < w_{minor}$ are coefficients between 0 and 1 attributed to each type of failure. The average working

state is the working state averaged over time.

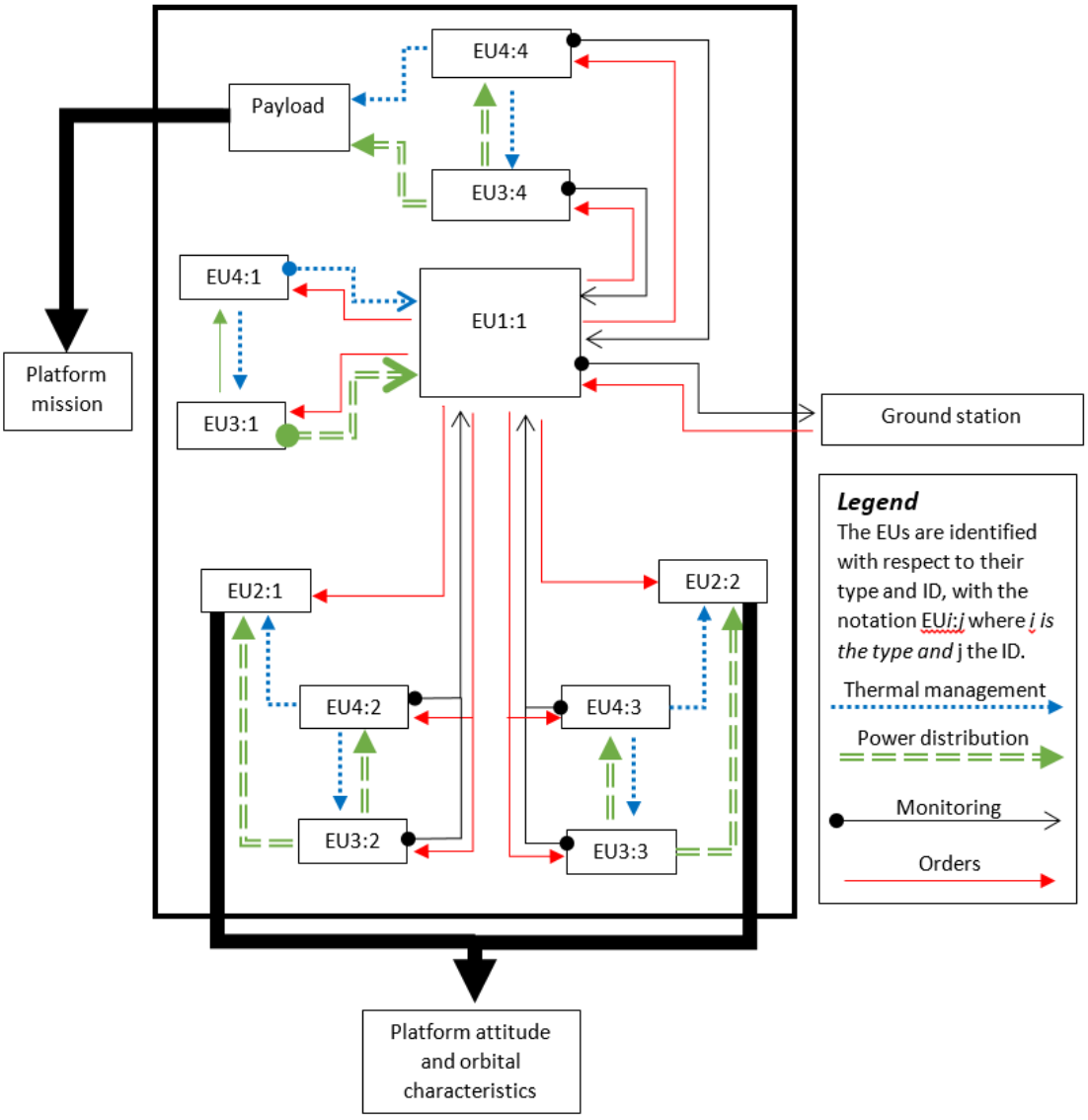

Figure 3.5: Interaction of the EUs of one of the platforms considered in this research

3.4 On-Orbit Servicing Infrastructures Modeling

This section describes how the WoD and WD infrastructures are modeled. An overview of the concept of operations and the decision making process is given for each infrastructure in subsection 3.4.1. The models of the launch vehicle, the servicers and the orbital depot are introduced in subsections 3.4.2, 3.4.3, 3.4.4 respectively. Subsection 3.4.5 introduces the service dispatch strategies implemented in Simulink and Python to distribute the demanded EUs

among the servicers.

3.4.1 On-Orbit Servicing Infrastructures Overview and Decision Making

The overview of the concept of operations of the WoD and WD infrastructures is given in figure 3.6. Note that although the Python-based framework can model and simulate several servicers, only one servicer is illustrated in 3.6 for simplicity.

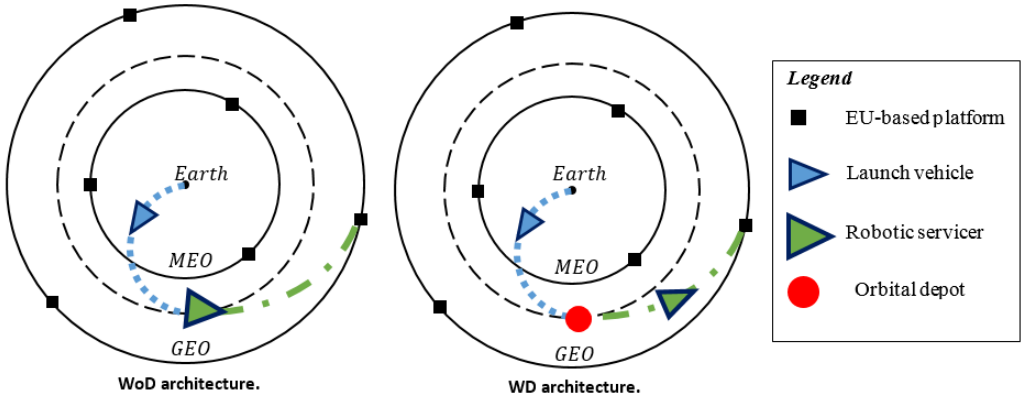

Figure 3.6: Overview of the WoD and WD infrastructures

The decision making processes driving the infrastructures are given in figure 3.7 for WoD and 3.8 for WD. The dash-lined blocks are driven by stochastic processes due to the random failures of a platform and the random launch failures of the launch vehicle. In figure 3.7, "r_i" designates the radius of the initial orbit, and "r_f" the radius of the final orbit. "$r_i = r_f$" trajectories are either phasing maneuvers or trajectories found solving the Lambert equation for identical initial and final orbits. "$r_i \neq r_f$" trajectories are either Hohmann transfers or trajectories found solving the Lambert equation for different initial and final orbits. The way decisions are made with respect to the launch vehicle and the servicers is explained in the following subsection of this section. The next paragraph gives a quick description of the concept of operations of both infrastructures.

The WoD infrastructure comprises two servicing components: the robotic servicer and the launch vehicle. The servicer grabs the spare EUs sent from Earth to its parking orbit and departs to service the failed platforms in the order defined by the chosen service dispatch strategy. The launch vehicle sends the spare EUs from Earth to the exact location of the servicer on its parking orbit on a regular basis when there is demand. In addition to the servicing systems of the WoD infrastructure, the WD infrastructure includes an orbital depot storing the spare EUs to provide

more responsive services. Spare EUs are sent to orbit in the launch vehicle whenever there remains a limited number of EUs in the depot.

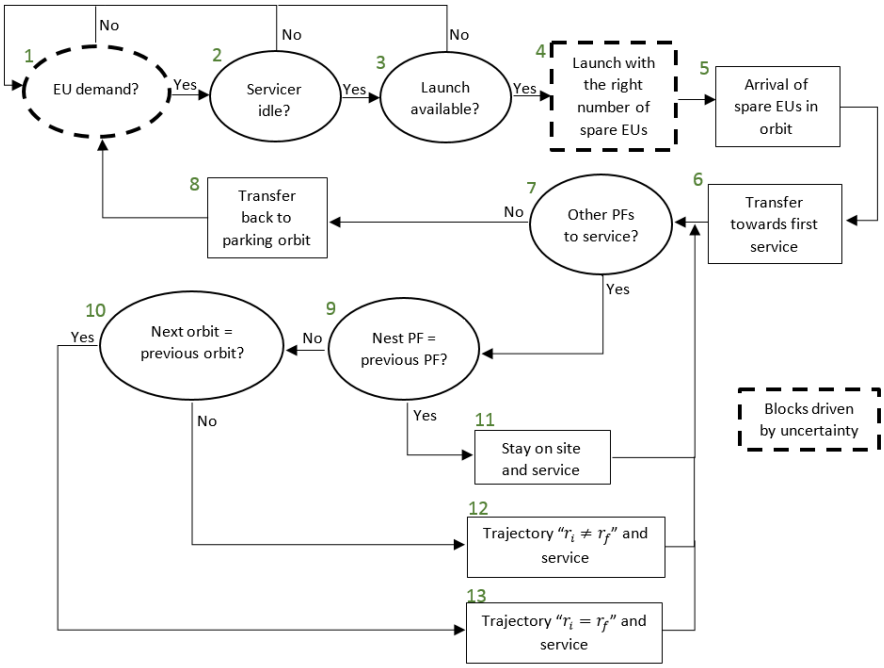

Figure 3.7: Concept of operation and decision making process of the WoD infrastructure (PF means "Platform")

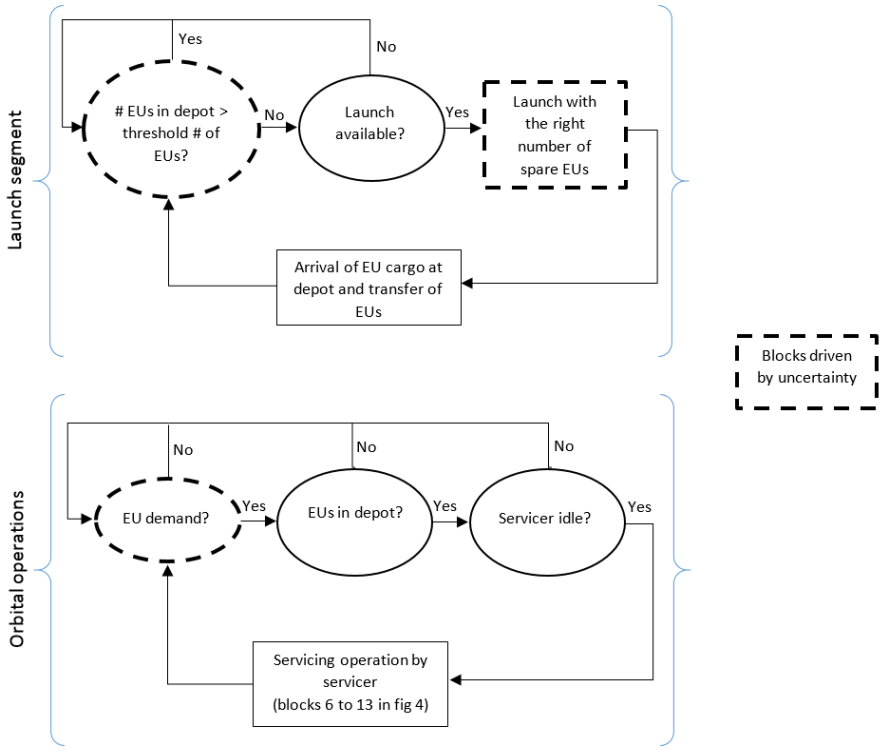

Figure 3.8: Concept of operation and decision making process of the WD infrastructure (PF means "Platform")

3.4.2 Launch Vehicle Model

First, we describe how blocks 3, 4 and 5 of figure 3.7 and the "Launch segment" block of figure 3.8 are modeled. Several assumptions are made to model the launch vehicle. First, the rockets are assumed to be launched on a deterministic basis every X months with a certain launch failure rate deduced from historical data [26]. Also, the rockets are assumed to send the cargo of EUs at the exact location of the servicer in its parking orbit in such a way that the servicer just needs to grab the EUs before servicing the platforms. The last assumption indicates that the servicers are operated at a minimum propellant consumption for long-term operation.

The four main parameters of the launch vehicle's model are (i) its launch frequency, (ii) its predicted success rate, (iii) its altitude at the end of the ascent phase and (iv) the ascent time to reach this altitude. For a specific launch vehicle (Falcon 9, Delta 2), the number of launches, successful or not, and parameters (i) and (ii) can be found in reference [26] at any year. Parameters (iii) and (iv) can be found in the user's guides of the launchers of interest [27].

Regarding the WoD infrastructure, blocks 3, 4 and 5 of figure 3.7 sums up the important phases related with the

model of the launch vehicle. First, the rocket is launched with the right number of EUs to be brought to the servicer. Then, its travel to the servicer's position is modeled. Finally, the arrival of the cargo of EU at the servicer's position triggers the next servicing operation, defined as the set of next services to be performed before the servicer comes back to its parking orbit. Block 4 of figure 3.7 models the process of launching the demanded number of EUs. The launches are planned on a regular basis, so there will be launches even though there is no demand of EUs. This is because we assume that the rockets are also delivering other payloads than EUs. If the demand of EUs exceeds the maximum capacity of the rocket, then only a number of spare EUs equal to this rocket capacity is launched. The remaining EUs are sent on the next launches. If the demand of EUs is less than the maximum capacity of the rocket then only a number of EUs equal to the number of demanded EUs is sent to orbit. Finally, block 5 of figure 3.7 models the trajectory of the launch vehicle. Only two phases of the trajectory are taken into account to simplify the model. The ascent phase is not strictly speaking modeled. Parameter (iv) is used to account for the duration of this phase. The geostationary transfer orbit between the altitude at the end of the ascent phase and the altitude of the parking orbit of the servicer is modeled as a Hohmann transfer.

The model of the launch vehicle in the case of the WD infrastructure is the same as for WoD except about two main points. First, EUs are sent from Earth when the orbital depot is almost empty, and not when a service is needed and the servicers are idle. Second, the arrival of the new cargo of EU to the orbital depot does not necessarily trigger servicing operations because there might be no service to provide or all servicers might already be providing services.

3.4.3 Servicer Model

The robotic servicers are the most important component of an OOS infrastructure. Their trajectories must be accurately modeled to yield realistic data about the responsiveness of the OOS infrastructures. The first part of this section presents the assumptions used in both simulation frameworks regarding the servicers. The second part defines the servicing time. The third part describes how the propellant tanks of the servicers are sized. The fourth part describes how the trajectories are traded against each other. The last part introduces the models of the trajectories considered in this research.

Assumptions

Several assumptions are made about the servicer in the Simulink-based simulation framework. A great accuracy in the servicer's modeling with this simulation framework was not necessary because we just wanted to check our intuition that the WD infrastructure was much more responsive than the WoD infrastructure. It is first assumed that loading the spare EUs into the servicer is instantaneous. Second, it is assumed that the servicer is refueled before

each servicing operation so as to avoid to keep track of the remaining propellant from one servicing operation to the other. Third, the mass of the EUs is not taken into account for the calculation of the propellant consumption over time. Finally, the additional delta V resulting from the rendezvous and docking maneuvers is not taken into account.

The objective of the Python-based simulation framework, as compared to Simulink, is to capture a large trade space of the WD infrastructure. Consequently, the servicers are modeled much more accurately. There still are two assumptions. The first one is that servicers are refueled before each servicing operation, assuming there is enough propellant in the orbital depot at any time a servicer is about to leave it to provide services. The second one is that the delta V to perform rendezvous and docking maneuvers is negligible compared to the delta Vs characteristic of the orbital transfers.

Servicing Time

The servicing time for a servicing operation providing n services is calculated over time by both simulation frameworks as follows

$$t_s = \sum_{i=1}^{n} \left(t^i_{wait} + t^i_{travel} + t^i_{replace} \right) + t^{end}_{wait} + t^{end}_{travel} \tag{3.8}$$

where

t^i_{wait} waiting time of the servicer in its current orbit before an opportunity occurs to travel to the next platform associated with the i^{th} service of the servicing operation. It is set to 0 when the next service is provided to the same platform as the previous service's or when the next trajectory can be performed without waiting time;

t^i_{travel} travel time of the servicer from its previous orbit to the orbit on which the i^{th} service must be provided. It is set to 0 when the next service is provided to the same platform as the previous service;

$t^i_{replace}$ time taken by the servicer to perform the i^{th} EU replacement. It is assumed to be independent of the service being provided and of the type of the EU being replaced;

t^{end}_{wait} waiting time of the servicer in its current orbit before an opportunity occurs to travel back to its parking orbit after the last service is accomplished;

t^{end}_{travel} travel time of the servicer from its previous orbit to its parking orbit.

The way the travel and waiting times are calculated is presented in the last part of this section.

Propellant Tank Sizing

The analysis presented in this subsection has been implemented in the Python-based simulation framework to accurately model the propellant consumption of the servicers. The user of the framework specifies the actual propel-

lant capacity $M_{p,spe}$ that he/she wants for the design of the servicers. This value is then compared to the propellant mass $M_{p,req}$ required for the servicers to perform the most expensive servicing operations. If $M_{p,spe} \geq M_{p,req}$, the design of the servicers specified by the user of the framework is used to run the simulations without inconsistencies. If $M_{p,spe} < M_{p,req}$, the simulations cannot be run due to a possible lack of propellant for the most expensive servicing operations. The objective of the remaining analysis is precisely to compute $M_{p,req}$.

Both simulation frameworks model two types of trajectories. The first type, or "$r_i = r_f$" type, corresponds to trajectories between two different angular slots on a same circular orbit. The second type, or " $r_i \neq r_f$" type, corresponds to trajectories between two different circular orbits. Each type of trajectory includes two different subtypes of trajectories. The "$r_i = r_f$" type includes phasing maneuvers and Lambert trajectories. The "$r_i \neq r_f$" type includes Hohmann transfers and Lambert trajectories without coasting period. The reader can refer to the last part of this section for the models of those trajectories.

Of the four subtypes of trajectories, only Hohmann transfers offer a delta V independent of the phasing angle between the servicer and its next target, either a customer satellite or the orbital depot. For all other trajectories, the delta V is computed as a function of the initial angle between the servicer and its target.

For each type of trajectory, a "reference" trajectory is chosen to calculate the size of the propellant tank allowing the servicer to respond to any demand no matter what the phasing angle is. This is done thanks to the calculation of upper bounds of the delta V. The chosen reference trajectories are the Hohmann transfer and the phasing maneuver since they are easier to calculate than Lambert trajectories.

Eight different types of delta V must be defined to properly calculate the appropriate size of the servicers' tanks:

ΔV the actual delta V used during a given operation (i.e. between departure from and arrival to the orbital depot). It is calculated during the simulation for the trade-off between the different subtypes of trajectories. This delta V is different for each servicing operation;

ΔV_{req} minimum capacity of the reservoir of delta V defining the minimum size of the propellant tank required to perform services in the worst case scenarios. This delta V is constant during the simulation and used to calculate the minimum required size of the propellant tank in terms of propellant mass;

ΔV_{res} reservoir delta V specified by the user of the framework. This delta V must be higher than ΔV_{req} in order for the servicers to be able to provide services in the worst case scenarios. Similarly, the user of the framework may directly specify the size of the propellant tank in terms of propellant mass, from which is then calculated ΔV_{res};

ΔV_{extra} extra delta V specified by the user of the framework to enable more flexibility in the trade-off between the different subtypes of trajectories for a given travel leg. This delta V is constant during the simulation;

ΔV_{ref} the delta V that the servicer would consume for a given servicing operation if it was following the reference trajectories only. As for ΔV, it is calculated during the simulation for the trade-off between the different types of trajectories. This delta V is different for each servicing operation like ΔV;

ΔV_{ref}^{max} upper bound of the reference delta V used to size the propellant tank of the servicer. It is constant during the simulation;

ΔV_{ref}^{c} minimum delta V the servicer must carry in its delta V reservoir to perform the given operation without missing propellant. It is calculated using the reference trajectories. It varies across the servicing operations;

ΔV^{c} actual delta V carried in terms of propellant mass by the servicer for a given operation. It is different for each servicing operation.

For a given servicing operation, the delta Vs of interest are ΔV, ΔV_{ref}, ΔV_{ref}^{c} and ΔV^{c}. The last three are related with the following inequality relationship:

$$\Delta V^{c} \geq \Delta V_{ref}^{c} \geq \Delta V_{ref} \tag{3.9}$$

However, ΔV is not necessarily greater than ΔV_{ref}. For instance, there are cases where Lambert trajectories yield lower delta Vs than with phasing maneuvers.

A servicing operation (or mission) is defined as the sequence of travel legs and on site-replacements performed from the moment the servicer leaves the orbital depot to the moment it comes back to the orbital depot. Assuming a servicing operation includes n travel legs, each travel leg and the corresponding actual and reference delta Vs are given an index from 1 to n. The total number of travel legs is equal to $n_{sat}+1$ where n_{sat} is the number of satellites that the servicer is going to visit for a given servicing operation. Figure 3.9 gives an example of servicing operation and the delta Vs for each travel leg. The servicer first visits *sat1* located in an orbit of radius r_B. It then travels to an orbit of radius r_C to visit *sat4* and so on until it docks to the depot again.

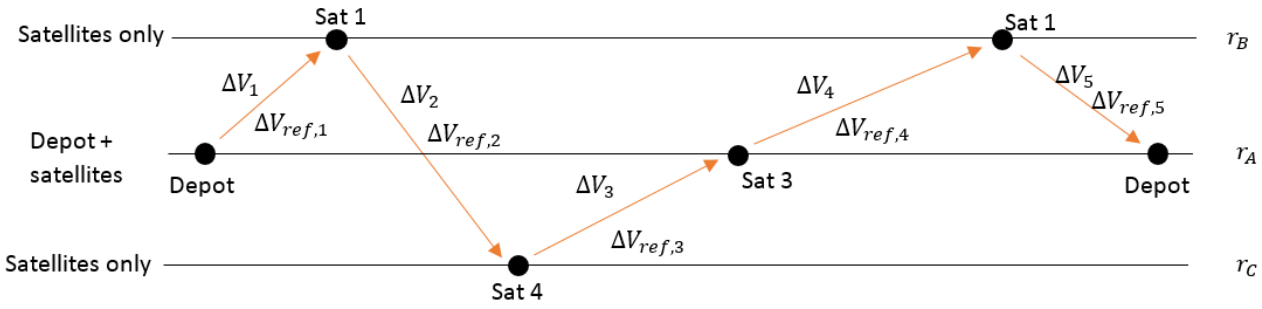

Figure 3.9: An example of servicing operation and the delta Vs defined for each travel leg

From figure 3.9 and the introduced notations, the actual delta V consumed during a servicing operation is

$$\Delta V = \sum_{i=1}^{n} \Delta V_i \tag{3.10}$$

The delta V the servicer would consume if it was following reference trajectories only is

$$\Delta V_{ref} = \sum_{i=1}^{n} \Delta V_{ref,i} \tag{3.11}$$

The rocket equation is used to relate the consumed propellant mass with the actual delta V consumed during the servicing operation:

$$\Delta V = g_0 I_{sp} \ln\left(\frac{M_{dry} + M_{EUs} + M_p}{M_{dry} + M_{EUs}}\right) \iff M_p = \left(M_{dry} + M_{EUs}\right)\left[\exp\left(\frac{\Delta V}{g_0 I_{sp}}\right) - 1\right] \tag{3.12}$$

where g_0 is the sea-level gravitational acceleration, I_{sp} is the specific impulse, M_{dry} is the dry mass of the servicer, M_{EUs} is the mass of the EUs in its own depot for the service operation, and M_p is the consumed propellant mass.

The propellant tank is sized so that the servicer is never limited by a lack of propellant during a servicing operation. In order to do so, we must calculate an upper bound of delta V defined earlier as ΔV_{ref}^{max} to compute ΔV_{req}. In order to do so, every combination of trajectory between the existing orbits is considered and its delta V calculated. Figure 3.10 introduces the notations. In figure 3.10, the notation ΔV_{ref}^{ij} stands for the delta V from orbit i to orbit j along a reference trajectory.

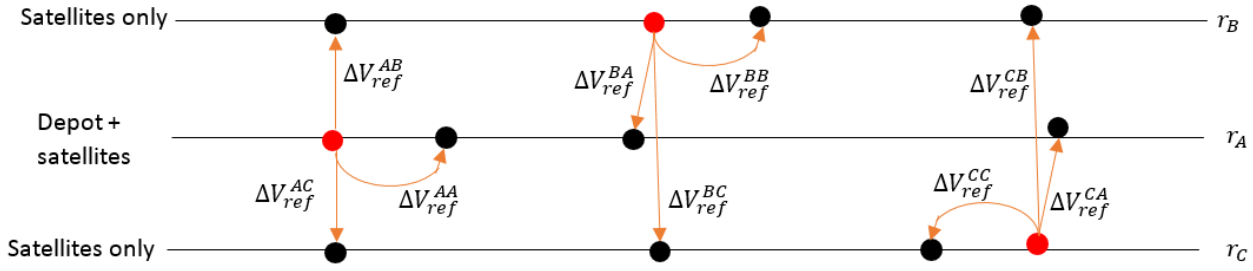

Figure 3.10: Notations for the sizing of the propellant tank

The only way to make ΔV_{ref}^{max} a constant throughout the simulation while allowing the servicer to provide EUs along any possible path is to follow the two following first steps:

1. $\forall (i,j) \in \{A,B,C\}^2$, ΔV_{ref}^{ij} must be maximized with respect to the phasing angle $\Delta\theta$ between the servicer and its target. The resulting delta V is written $\max_{\Delta\theta}\left(\Delta V_{ref}^{ij}\right)$;
2. The maximum value of the delta Vs determined in step 1 is chosen as $\max_{(i,j)}\left[\max_{\Delta\theta}\left(\Delta V_{ref}^{ij}\right)\right]$.

It must be noted that the expression of the maximum delta V given in step 2 is valid for any i, j in $\{A,B,C\}$ if there are satellites on the same orbit as the orbital depot's. If not, the expression is valid for any i, j in $\{A,B,C\}$ less the depot's orbit. This is because a similar analysis is performed separately for travel legs from and to the on-orbit depot. The maximum delta Vs required to leave and come back to the orbital depot are respectively $\max_j \left[\max_{\Delta\theta}\left(\Delta V_{ref}^{Aj}\right)\right]$ and $\max_i \left[\max_{\Delta\theta}\left(\Delta V_{ref}^{iA}\right)\right]$, where A is the name given to the orbit of the depot.

The final step to compute ΔV_{ref}^{max} is to add all those delta Vs by taking into account the worst case scenario. This scenario would be for the servicer to fly to a different satellite than the previous one for each travel leg along a trajectory consuming the delta Vs defined above. And since the maximum number of satellites that a servicer can visit is the maximum number of EUs n_{EU}^{max} that it can carry, we can write

$$\Delta V_{ref}^{max} = \max_j \left[\max_{\Delta\theta}\left(\Delta V_{ref}^{Aj}\right)\right] + (n_{EU}^{max}-1)\max_{(i,j)}\left[\max_{\Delta\theta}\left(\Delta V_{ref}^{ij}\right)\right] + \max_i \left[\max_{\Delta\theta}\left(\Delta V_{ref}^{iA}\right)\right] \tag{3.13}$$

The minimum required capacity of the reservoir of delta V is thereby

$$\Delta V_{req} = \Delta V_{ref}^{max} + \Delta V_{extra} \tag{3.14}$$

where, again, ΔV_{extra} is the extra delta V specified by the user of the framework to enable more flexibility in the trade-off between the different subtypes of trajectories for a given travel leg.

The minimum required propellant mass that the servicer must carry to be able to perform all servicing operation is then:

$$M_{p,req} = \left(M_{dry} + n_{EU}^{max} \times \max_i\left(M_{EUi}\right)\right)\left[\exp\left(\frac{\Delta V_{req}}{g_0 I_{sp}}\right) - 1\right] \tag{3.15}$$

where M_{EUi} is the mass of an EU of type i.

Once the minimum required propellant mass is known, it can be compared to the propellant mass $M_{p,spe}$ specified by the servicer. If $M_{p,spe} \geq M_{p,req}$, the design of the servicers specified by the user of the framework is used to run the simulations without inconsistencies. If $M_{p,spe} < M_{p,req}$, the simulations cannot be run due to a possible lack of propellant for the most expensive servicing operations.

Trajectory Decision Making

This section explains how ΔV^c_{ref}, ΔV_{ref}, ΔV^c and ΔV are computed and used to perform the trade-off between reference trajectories (Hohmann transfers and phasing maneuvers) and non-reference trajectories (Lambert trajectories) for each travel leg. More specifically, ΔV_{ref} and ΔV are used for the trade-off by comparing their values with ΔV^c_{ref} and ΔV^c. Consequently, ΔV^c_{ref} and ΔV^c must be calculated before the actual servicing operation starts.

To compute ΔV^c_{ref}, we first get the path of the servicer defined by its backlog of services and then maximize with respect to the initial angle $\Delta\theta$ between the servicer and its target the delta Vs along the appropriate reference trajectories. As an example, if the servicers backlog is (sat1, sat4, sat 3, sat 1) then we can assign the indices of the order in which the satellites are being served as depicted in figure 3.11.

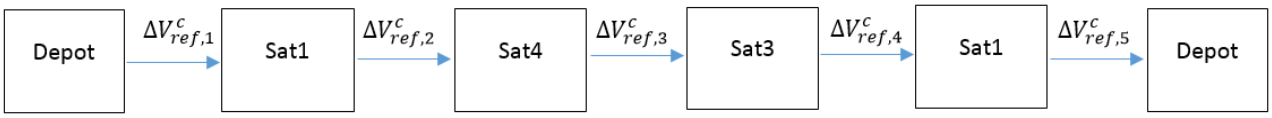

Figure 3.11: Calculation of ΔV^c_{ref}

where $\forall i \in [1..n]$ ($n = 5$ here), $\Delta V^c_{ref,i} = \max_{\Delta\theta}\left(\Delta V_{ref,i}\right)$ is the delta V maximized with respect to the phasing angle $\Delta\theta$ between the servicer and its target, along the reference trajectory corresponding to the travel leg i for the particular flight path defined by the servicer's backlog. So the final expression is given by

$$\Delta V^c_{ref} = \sum_{i=1}^{n} \max_{\Delta\theta}\left(\Delta V_{ref,i}\right) \tag{3.16}$$

Finally, ΔV^c must be chosen larger than ΔV^c_{ref} in the limit of the amount of propellant left in the orbital depot. However, the supply chain of the servicers propellant to the depot is not modeled in the current framework so it is assumed there is always propellant left in the depot with the amount that we need. In particular, the servicers are always fully refueled, which means that with this modeling:

$$\Delta V^c = \Delta V_{res} \geq \Delta V_{req} \geq \Delta V^c_{ref} \tag{3.17}$$

Note that if the propellant supply chain was modeled accurately, we would have

$$\Delta V_{res} \geq \Delta V^c \geq \Delta V_{req} \geq \Delta V^c_{ref} \tag{3.18}$$

Based on the notations introduced above, the trajectory decision making follows two steps. The first step, illustrated by blocks 6 to 13 in figure 3.7, is the decision between a "$r_i = r_f$"-type of trajectory and a "$r_i \neq r_f$"-type of

trajectory. This is simply done by comparing the radii of the initial and final orbits. Once the trajectory-type has been selected, the second step of the decision making consists in choosing between the reference or non-reference trajectories. This is less straightforward because it involves trade-off consideration based on travel time and delta V. After doing the work for the first two travel legs, one notices a recurrent scheme for any travel leg j. The trade-off between the trajectory subtypes for travel leg j is presented below.

At the end of the trade-off for travel leg j, $\Delta V_{ref,j}$ and ΔV_j are calculated and the trajectory subtype is chosen. It is recalled that ΔV_j is the actual delta V consumed by the servicer along travel leg j and $\Delta V_{ref,j}$ is the delta V the servicer would consume if it was flying along a reference trajectory. The following balance of delta V is fundamental to perform the trade-off leading to the choice of trajectory.

$$\Big(\Big(\big((\Delta V^c - \Delta V_1) - \Delta V_2\big) - \Delta V_3\Big)...\Big) - \Delta V_j \geq \Big(\Big(\big(\big(\Delta V^c_{ref} - \Delta V_{ref,1}\big) - \Delta V_{ref,2}\big) - \Delta V_{ref,3}\Big)...\Big) - \Delta V_{ref,j} \tag{3.19}$$

where the left hand side is the remaining delta V in the reservoir at the end of the j^{th} travel leg while the right hand side is the remaining delta V in the reservoir if the servicer had flown along the appropriate reference trajectory. Note that in this case, the reference delta V for the given travel leg (i.e. $\Delta V_{ref,j}$ here) is not maximized with respect to the angle $\Delta\theta$ between the servicer and its target because we now compute the actual delta V with the actual relative positions of the satellites at the time travel leg j starts.

Equation 3.19 may be rewritten as

$$\big(\Delta V_{ref,j} - \Delta V_j\big) + \big(\Delta V_{ref,j-1} - \Delta V_{j-1}\big) + \big(\Delta V_{ref,j-2} - \Delta V_{j-2}\big) + ... + \big(\Delta V_{ref,1} - \Delta V_1\big) \geq \Delta V^c_{ref} - \Delta V^c \tag{3.20}$$

We then define the algebraic remaining delta V, $\Delta V_{remaining,i} = \Delta V_{ref,i} - \Delta V_i$, that travel leg i leaves for the subsequent travel legs as compared to $\Delta V_{ref,i}$. We also define the initial available delta V, $\Delta V_{remaining} = \Delta V^c_{ref} - \Delta V^c$, for the entire servicing operation. $\Delta V_{remaining}$ is therefore a constant over the given servicing operation. Equation 3.20 may therefore be rewritten as

$$\Delta V_{remaining,j} + \sum_{i=0}^{j-1} \Delta V_{remaining,i} \geq \Delta V_{remaining} \tag{3.21}$$

where, in the sum, $\Delta V_{remaining,0} = \Delta V_0 = \Delta V_{ref,0} = 0$ for convenience. Note that in equation 3.21 the sum and right hand side are known. Equation 3.21 gives a lower bound for $\Delta V_{remaining,j}$. A natural upper bound for $\Delta V_{remaining,j}$ is obtained when the actual delta V, $\Delta V_{remaining,j}$, is equal to 0, i.e. the upper bound is $\Delta V_{ref,j}$. At this stage of the decision making, ΔV_j and $\Delta V_{ref,j}$ are still unknown. In oder to choose the most appropriate trajectory, the actual and

reference delta Vs are calculated for each trajectory subtype. The obtained values are then used to make the final choice of trajectory, with the decision making process indicated below.

From what has been introduced above, a trajectory is chosen if its actual and reference delta Vs are such that

$$\Delta V_{remaining,j} = \Delta V_{ref,j} - \Delta V_j \in \left[\Delta V_{remaining} - \sum_{i=0}^{j-1} \Delta V_{remaining,i} \ , \ \Delta V_{ref,j} \right] \tag{3.22}$$

where it can be shown that $\Delta V_{remaining} - \sum_{i=0}^{j-1} \Delta V_{remaining,i} \geq 0$. The trade-off to choose the right trajectory is driven by the following cases:

- If $\Delta V_{remaining,j} < \Delta V_{remaining} - \sum_{i=0}^{j-1} \Delta V_{remaining,i}$, the servicer consumes all the remaining delta V. The corresponding trajectory cannot be chosen;
- If $\Delta V_{remaining,j} \in \left[\Delta V_{remaining} - \sum_{i=0}^{j-1} \Delta V_{remaining,i} \ , \ 0 \right[$, $\Delta V_j > \Delta V_{ref,j}$. Therefore, the corresponding trajectory is chosen only if the travel time Δt_j along travel leg j for the tested trajectory is lower than the travel time $\Delta t_{ref,j}$ along the reference trajectory multiplied by a coefficient k defined between 0 and 1, i.e. $\Delta t_j < k \times \Delta t_{ref,j}$. In the Python-based simulation framework, k is set to 1;
- If $\left[0 \ , \ \Delta V_{ref,j} \right]$, $\Delta V_j \geq \Delta V_{ref,j}$. Since we want to improve the responsiveness of the OOS infrastructure, we must choose the trajectory corresponding to ΔV_j only if $\Delta t_j < \Delta t_{ref,j}$.

This trajectory trade-off analysis ends this subsection dedicated to the trajectory decision making. The next subsection introduces the model of the different trajectories considered in this research.

Trajectory Modeling

The Simulink-based simulation framework only models and simulates Hohmann transfer-based and phasing maneuver-based rendezvous between the servicer and the customer satellites based on their relative angle and respective orbit radii. In addition, the Python-based simulation framework models Lambert trajectory-based rendezvous to increase the responsiveness of the WD infrastructure.

Hohmann transfer-based rendezvous

Hohmann transfer-based rendezvous are trajectories between two positions on two different circular orbits. They are two-impulse optimal-optimal trajectories but, depending on how close is the initial orbit from the final one, they might require a long waiting time before the servicer can perform the initial impulse. This waiting time is highly

dependent on the initial relative angle between the servicer and its target.

The travel time of a Hohmann transfer is simply given by

$$t_{travel} = \pi \sqrt{\frac{a_H^3}{\mu}} \tag{3.23}$$

where μ is Earth's standard gravitational parameter, $a_H = (r_1 + r_2)/2$ is the semi-major axis of the Hohmann ellipse, and r_1 and r_2 are the radii of the initial and final orbits respectively.

The waiting time before the next impulse opportunity is given by

$$t_{wait} = \frac{\left(\pi - n_2 \pi \sqrt{\frac{a_H^3}{\mu}}\right) - \Delta\theta_0}{n_2 - n_1} + N \times T_{syn} \tag{3.24}$$

where n_1 and n_2 are the mean motions on the initial and final orbits respectively, $\Delta\theta_0$ is the initial relative angle of the target with respect to the servicer, $T_{syn} = 2\pi/|n_2 - n_1|$ is the synodic period between the initial and final orbits, and N is a positive integer found so as to make the waiting time positive. The presence of $n_2 - n_1$ in the denominator of the waiting time explains why it gets larger as the initial and final orbits get closer from each other.

Phasing maneuver-based rendezvous

Phasing maneuver-based rendezvous are two-impulse trajectories between two different positions on the same circular orbit. Unlike Hohmann transfers, there is no waiting time and the travel time depends on the initial relative angle between the target and the servicer. To prevent the servicer from entering a certain sphere around Earth, an altitude constraint is considered to calculate the characteristics of the phasing maneuver. The semi-major axis of the phasing orbit is calculated by first minimizing the travel time t_{travel} and then minimizing the delta V. Figure 3.12 illustrates the trajectory and introduces the notation to describe its model. r is the radius of the target's orbit, r_{crit} is the radius of the "forbidden zone" where the servicer cannot fly, $\Delta\theta_0$ is the initial relative angle of the target with respect to the servicer and α is the phasing angle flown by the target when the servicer is on its phasing orbit. "1" and "2" stand for the servicer and target respectively. "0" and "f" stand for the initial and final times respectively.

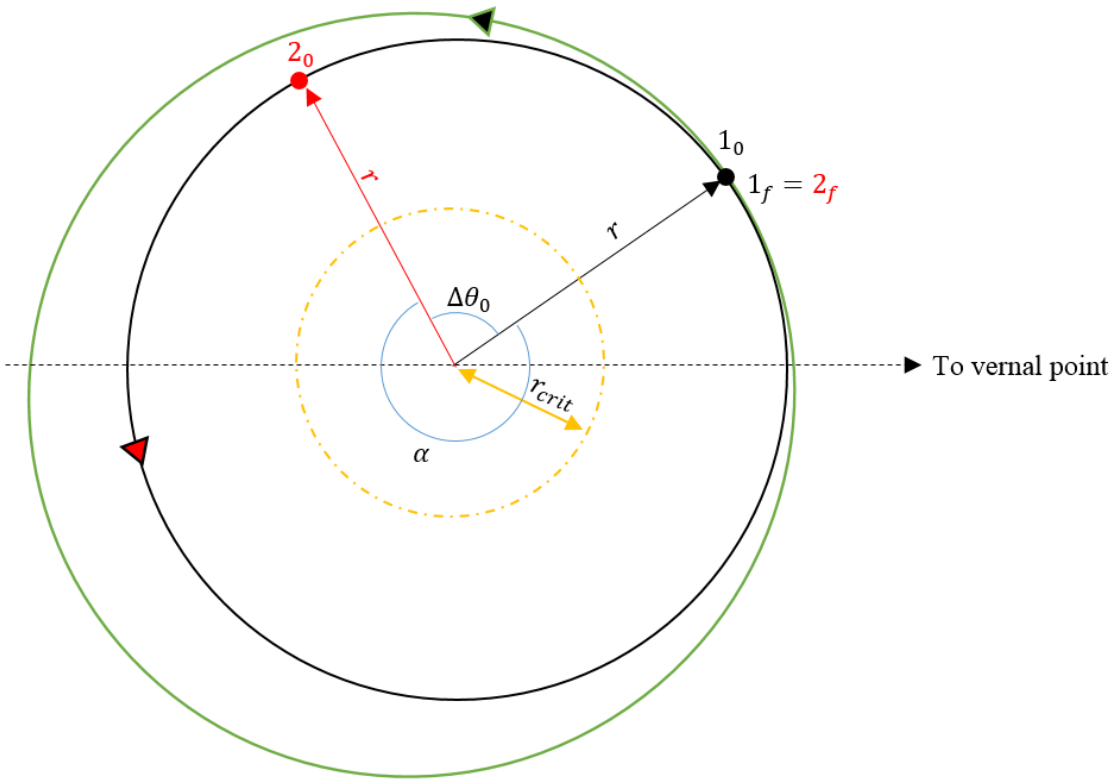

Figure 3.12: Example of a phasing maneuver-based rendezvous

The angle α must first be calculated. If $\Delta\theta_0 > 0$, we define $\alpha = 2\pi - \Delta\theta_0$. If $\Delta\theta_0 \leq 0$, we define $\alpha = -\Delta\theta_0$. In both cases, α is defined in $[0\ ,\ 2\pi[$. In order to define the travel time of the servicer and target on their respective orbit, we define two integers $k_1 > 0$ and $k_2 \geq 0$ corresponding to the number of complete orbits they fly before rendezvousing. From the point of view of the servicer, the travel time is

$$t_{travel} = k_1 T_1 = 2\pi k_1 \sqrt{\frac{a^3}{\mu}} \tag{3.25}$$

where T_1 and a are the period and semi-major axis of the phasing orbit respectively. From the point of view of the target, the travel time is

$$t_{travel} = \frac{\alpha}{2\pi} T + k_2 T = \left[\frac{\alpha + 2\pi k_2}{2\pi}\right] T = (\alpha + 2\pi k_2) \sqrt{\frac{r^3}{\mu}} \tag{3.26}$$

where T is the period of the orbit of the target.

Equaling equations 3.25 and 3.26 gives an expression for a as a function of k_1 and k_2:

$$a = \left(\frac{\alpha + 2\pi k_2}{2\pi k_1} \right)^{\frac{2}{3}} r \tag{3.27}$$

The altitude constraint can be written

$$a \geq \frac{r + r_{crit}}{2} \tag{3.28}$$

Solving for k_1 and k_2 is a two-step process. One can see from equation 3.26 that the travel time can be expressed as a function of only one unknown, k_2, from the perspective of the target. So this equation is used to minimize the travel time with respect to k_2. For each tested value of k_2, values for k_1 are tested using equations 3.27 and 3.28. If the pair (k_1, k_2) satisfies the altitude constraint, it is kept as candidate for solution. At the end of this first step, a unique value for k_2 is determined, thereby determining the travel time. However, there can be several k_1 values for a same k_2 value. Consequently, the second step of solving this problem consists in finding the candidate (k_1, k_2) pair, and *in extenso* the k_1 value, that minimizes the delta V on a phasing maneuver, given by

$$\Delta V = 2 \left| \sqrt{\frac{\mu}{r}} - \sqrt{\mu \left(\frac{2}{r} - \frac{1}{a} \right)} \right| \tag{3.29}$$

At the end of this two-step process, k_1 and k_2 are well defined and the characteristics of the phasing orbit can be calculated.

Lambert trajectory-based rendezvous

In orbital mechanics, Lambert's problem is concerned with the determination of an orbit from two fixed positions and the time of flight. Under the influence of a central gravitational force, the equation relating the travel time with the initial and final radii r_1 and r_2 and the travel angle γ is given by

$$t_{travel} = \sqrt{\frac{a^3}{\mu}} \left[(\alpha - \beta) - (\sin(\alpha) - \sin(\beta)) \right] \tag{3.30}$$

where a is the semi-major axis of the transfer orbit.

The angles α and β are given by

$$\sin\left(\frac{\alpha}{2}\right) = \pm \sqrt{\frac{s}{2a}} \tag{3.31}$$

$$\sin\left(\frac{\beta}{2}\right) = \pm\sqrt{\frac{s-c}{2a}} \tag{3.32}$$

The parameters s and c are given by

$$s = \frac{r_1 + r_2 + c}{2} \tag{3.33}$$

$$c = \sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos(\gamma)} \tag{3.34}$$

This problem is a Two-Point-Boundary-Value-Problem easily solved when the travel angle γ does not vary with time, which is the case for a transfer. However, for a rendezvous with a target on a circular orbit of radius r_2, the problem is not so easy to solve. Indeed in this case, the travel angle is related with the travel time by

$$\gamma = n_2 \times t_{travel} + \Delta\theta_0 \quad \in \quad]0\ ,\ 2\pi[\tag{3.35}$$

where r_1, r_2, and the initial relative angle $\Delta\theta_0$ are fixed. One can deduce from equation 3.35 the range of the travel time to be $[-\Delta\theta_0/n_2\ ,\ T_2 - \Delta\theta_0/n_2]$, where n_2 and T_2 are the mean motion and period of the orbit of the target.

However, solving for the travel time is not enough because it must satisfy the following dynamics constraint:

$$t_{travel} > t^p_{travel} > 0 \tag{3.36}$$

where t^p_{travel} is the travel time on a parabolic trajectory. Because of this time constraint, it is fundamental to understand how varies t^p_{travel} with the travel angle γ or, in the rendezvous case, with the travel time t_{travel}. Lambert's problem gives the travel time along a parabolic trajectory as a function of the initial and final radii and the travel angle thanks to the equation

$$t^p_{travel} = \frac{1}{3}\sqrt{\frac{2}{\mu}}\left[s^{\frac{3}{2}} - \text{sign}\left(\sin\left(\gamma\right)\right)\left(s-c\right)^{\frac{3}{2}}\right] \tag{3.37}$$

No matter the initial and final radii, t^p_{travel} studied as a function of the travel angle, or as a function of the travel time for the rendezvous case, always yields the same variations. Figure 3.13 gives the principal characteristics of this function.

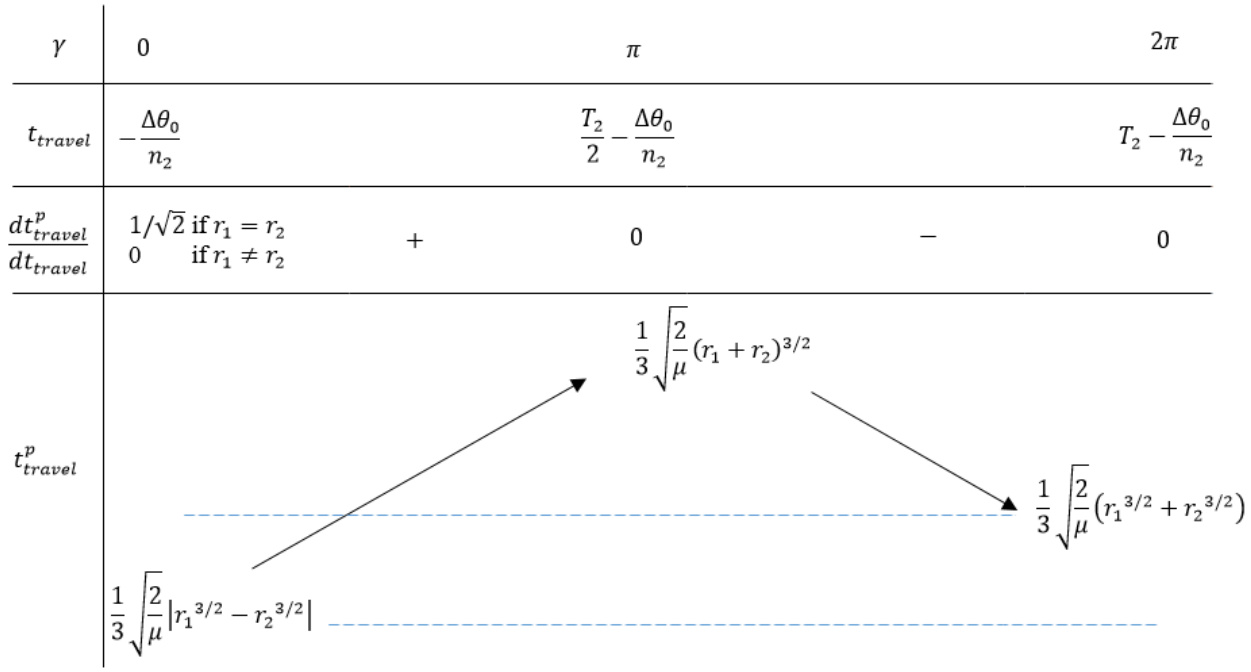

Figure 3.13: Variation table of t^p_{travel} studied as a function of the travel angle γ

The Python-based simulation framework also allows the user to specify an upper bound if the travel time must not be greater than a certain value t^{max}_{travel}. Once those lower-bound and upper-bound constraints are defined, the travel time along a parabolic trajectory and the $y = t_{travel}$ line are plotted on the same figure. Figure 3.14 gives a general overview of the plot which is used to calculate the range of admissible travel times. It is a frozen shot at given initial and final radii and given $\Delta\theta_0$. The modification of r_1 and r_2 will impact the shape of the curve of t^p_{travel}, while the modification of $\Delta\theta_0$ will translate the curve along the horizontal axis and always in the orange box depicted in figure 3.14. This box corresponds to the admissible travel times only based on equation 3.35, i.e. without the lower-bound and upper-bound constraints introduced above. This box translates along the $y = t_{travel}$ line as $\Delta\theta_0$ varies. This observation leads to the fact that there may be values of r_1, r_2 and $\Delta\theta_0$ for which there is no solution because of $-\Delta\theta_0$ being higher than t^{max}_{travel} or $T_2 - \Delta\theta_0/n_2$ being lower than t^p_{travel}.

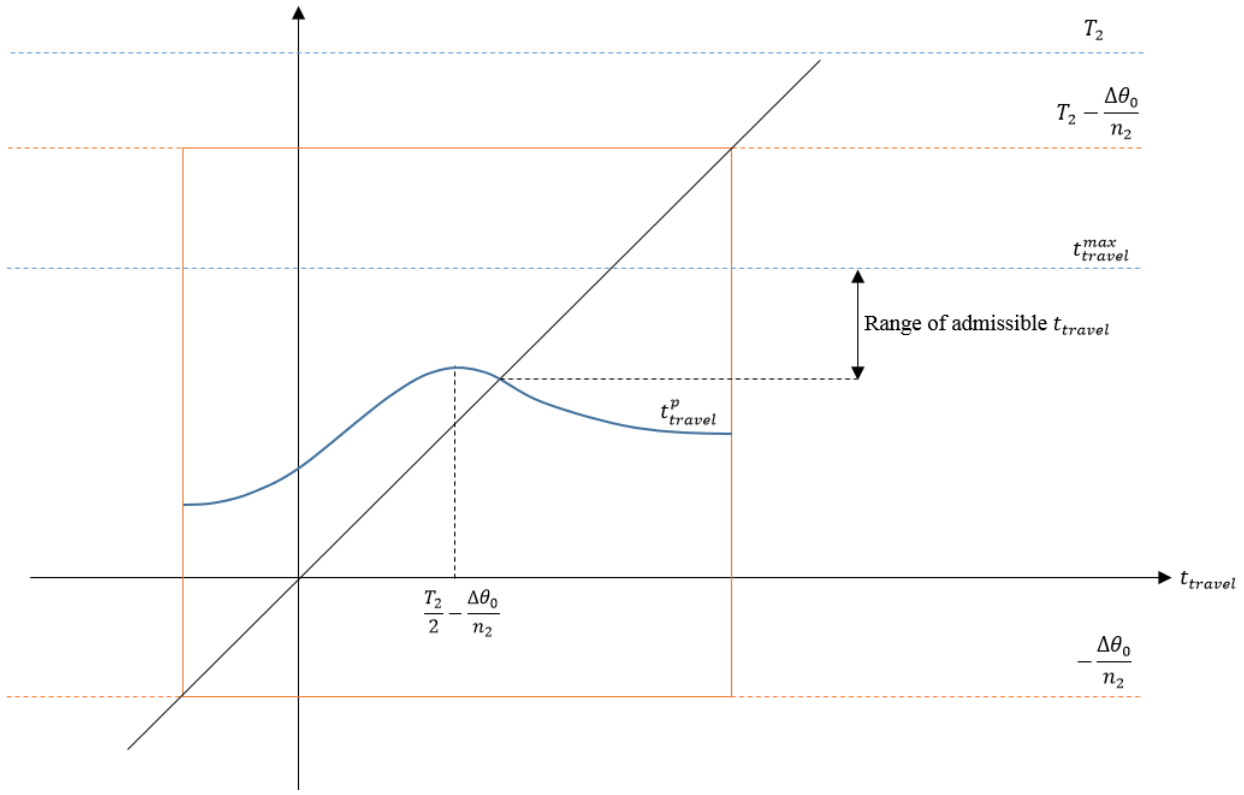

Figure 3.14: Plot used to find the range of admissible travel times

In order to find an appropriate Lambert trajectory, the range of admissible travel times must be calculated by computing its lower and upper bounds, t_{travel}^{low} and t_{travel}^{up} respectively. Depending on the value of $\Delta\theta_0$, or other said, depending on the relative position of the t_{travel}^{p} curve and the $y = t_{travel}$ line, t_{travel}^{low} and t_{travel}^{up} are calculated differently. In order to find the range of admissible travel times, five characteristic initial relative angles must be introduced. The first three, noted $\Delta\theta_0^0$, $\Delta\theta_0^\pi$ and $\Delta\theta_0^{2\pi}$, are defined by the intersection of the t_{travel}^{p} curve and the $y = t_{travel}$ line when the transfer angle γ is equal to 0, π and 2π. The fourth one, noted $\Delta\theta_0^{max,2\pi}$, is defined when the upper right apex of the orange box depicted in figure 3.14 is on the $y = t_{travel}^{max}$ horizontal line. The fifth one, noted $\Delta\theta_0^{max,0}$, is defined when lower left apex of the orange cube depicted in figure 3.14 is on the $y = t_{travel}^{max}$ horizontal line. Mathematically, this gives the following values for those characteristic angles.

$\Delta\theta_0^0$ found by solving $0 - \frac{\Delta\theta_0^0}{n_2} = t_{travel}^{p}(r_1, r_2, \gamma = 0)$, yielding $\Delta\theta_0^0 = -\frac{\sqrt{2}}{3}\left|1 - \left(\frac{r_1}{r_2}\right)^{\frac{3}{2}}\right|$;

$\Delta\theta_0^\pi$ found by solving $\frac{T_2}{2} - \frac{\Delta\theta_0^\pi}{n_2} = t_{travel}^{p}(r_1, r_2, \gamma = \pi)$, yielding $\Delta\theta_0^\pi = \pi\left[1 - \frac{\sqrt{2}}{3\pi}\left(1 + \frac{r_1}{r_2}\right)^{\frac{3}{2}}\right]$;

$\Delta\theta_0^{2\pi}$ found by solving $T_2 - \frac{\Delta\theta_0^{2\pi}}{n_2} = t_{travel}^{p}(r_1, r_2, \gamma = 2\pi)$, yielding $\Delta\theta_0^{2\pi} = \pi\left[2 - \frac{\sqrt{2}}{3\pi}\left(1 - \left(\frac{r_1}{r_2}\right)^{\frac{3}{2}}\right)\right]$;

$\Delta\theta_0^{max,2\pi}$ found by solving $T_2 - \frac{\Delta\theta_0^{max,2\pi}}{n_2} = t_{travel}^{max}$, yielding $\Delta\theta_0^{max,2\pi} = n_2\left(T_2 - t_{travel}^{max}\right)$;

$\Delta\theta_0^{max,0}$ found by solving $0 - \frac{\Delta\theta_0^{max,0}}{n_2} = t_{travel}^{max}$, yielding $\Delta\theta_0^{max,0} = -n_2 t_{travel}^{max}$.

It can be shown that

$$\Delta\theta_0^{max,0} < \Delta\theta_0^0 < \Delta\theta_0^\pi < \Delta\theta_0^{2\pi} \tag{3.38}$$

$$\Delta\theta_0^{max,2\pi} \in \left]\Delta\theta_0^{max,0}\ ,\ \Delta\theta_0^{2\pi}\right[\tag{3.39}$$

The following cases identifies whether the range of admissible travel times is empty or not depending on the value of $\Delta\theta_0$ with respect to the characteristic angles:

- If $\Delta\theta_0 \geq \Delta\theta_0^{2\pi}$: the range of admissible travel times is empty due to the lower-bound constraint "$t_{travel} > t_{travel}^p$";

- If $\Delta\theta_0^{max,0} < \Delta\theta_0 < \Delta\theta_0^{2\pi}$: t_{travel}^{low} and t_{travel}^{up} exist;

- If $\Delta\theta_0 \leq \Delta\theta_0^{max,0}$: the range of admissible travel times is empty due to the upper-bound constraint "$t_{travel} < t_{travel}^{max}$".

In the case where t_{travel}^{low} and t_{travel}^{up} exist, the following cases are used to calculate t_{travel}^{low}:

- If $\Delta\theta_0^\pi < \Delta\theta_0 < \Delta\theta_0^{2\pi}$: t_{travel}^{low} is found by solving $t_{travel} = t_{travel}^p\left(r_1, r_2, n_2 t_{travel} + \Delta\theta_0\right))$ with $t_{travel}^p = \frac{1}{3}\sqrt{\frac{2}{\mu}}\left[s^{\frac{3}{2}} + (s-c)^{\frac{3}{2}}\right]$;

- If $\Delta\theta_0 = \Delta\theta_0^\pi$: $t_{travel}^{low} = t_{travel}^p(r1, r2, \gamma = \pi)$;

- If $\Delta\theta_0^0 < \Delta\theta_0 < \Delta\theta_0^\pi$: t_{travel}^{low} is found by solving $t_{travel} = t_{travel}^p\left(r_1, r_2, n_2 t_{travel} + \Delta\theta_0\right))$ with $t_{travel}^p = \frac{1}{3}\sqrt{\frac{2}{\mu}}\left[s^{\frac{3}{2}} - (s-c)^{\frac{3}{2}}\right]$;

- If $\Delta\theta_0^{max,0} < \Delta\theta_0 \leq \Delta\theta_0^0$: $t_{travel}^{low} = -\frac{\Delta\theta_0}{n_2}$.

The following cases are used to calculate t_{travel}^{up}:

- If $\Delta\theta_0^{max,2\pi} < \Delta\theta_0 < \Delta\theta_0^{2\pi}$: $t_{travel}^{up} = T_2 - \frac{\Delta\theta_0}{n_2}$;

- If $\Delta\theta_0^{max,0} < \Delta\theta_0 \leq \Delta\theta_0^{max,2\pi}$: $t_{travel}^{up} = t_{travel}^{max}$.

Once t_{travel}^{low} and t_{travel}^{up} are found, an optimization routine is run over the range of admissible travel times to find the trajectory with the minimum delta V.

3.4.4 Depot Model

In addition to the servicing systems of the WoD infrastructure, the WD infrastructure includes an orbital depot storing the spare EUs to provide more responsive services. Spare EUs are sent to orbit in the launch vehicle whenever there remains a limited number of EUs in the depot. Unlike WoD, WD infrastructure decouples the launch and orbital operations as depicted in figure 3.8. WoD responds to demands in a sequential manner whereas WD parallelizes the launch and orbital operations thanks to the presence of the orbital depot. Intuitively, this parallelization will enable the WD to provide more responsive services than WoD.

The presence of the orbital depot requires explaining how the loops *Launch segment* and *Orbital operations* of figure 3.8 interact. The sizing parameter of the orbital depot is the maximum number of EUs of each type it can store.

Another important parameter in the decision process of sending new spare EUs from Earth is the threshold number of remaining EUs in the orbital depot below which new spare EUs must be sent during the next launch to refill the orbital depot. When the total number of EUs in the orbital depot becomes lower than the threshold number of EUs, the order to send spare EUs to orbit at the next launch is recorded. Between the moments when this order is recorded and the next launch occurs, the required number of EUs to completely refill the orbital depot is being updated. This is because more failures can occur between the time a launch order is recorded and the actual launch time.

The last characteristic of the orbital depot is the initial number of EUs of each type at the beginning of the simulation. If the orbital depot is fully replenished at the beginning of the simulation, the servicers will be able to provide services as soon as the first failures occur.

3.4.5 Service Dispatch Strategies

Distributing the EUs efficiently among the available servicers can greatly improve the responsiveness of an on-orbit servicing infrastructure. Although several dispatch strategies were implemented, only two were used to yield the results presented in the next two chapters. The first strategy, called "Servicers rotation" (SR), behaves as FIFO when there is only one servicer in the OOS infrastructure. The second one, called "Average working state maximization" (AWSM) distributes the EUs among the servicers so as to decrease the number of travel legs during one servicing operation and improve the average working state of the satellites.

Servicers Rotation

When there is only one servicer, the dispatch of EUs in the depot of the servicer is straightforward since this strategy behaves as a FIFO strategy. In this case, the EUs are put in the servicer in the order of the failures of the EUs they are aimed to replace. This strategy with one servicer is implemented in the Simulink-based simulation framework.

When there are more than one servicers, the dispatch of EUs is still done following the order of the failure occurrences recorded in a "failure backlog". The EU intended to replace the first failed EU recorded in the backlog is put in the first available servicer. The EU intended to replace the second failure EU recorded in the backlog is put in the second available servicer and so on by rotating the available servicers. In addition, the EUs are ordered per satellite in the own backlog of the servicers so as to minimize the travel legs. Figure 3.15 illustrate the basics of this strategy. This strategy with several servicers is implemented in the Python-based simulation framework.

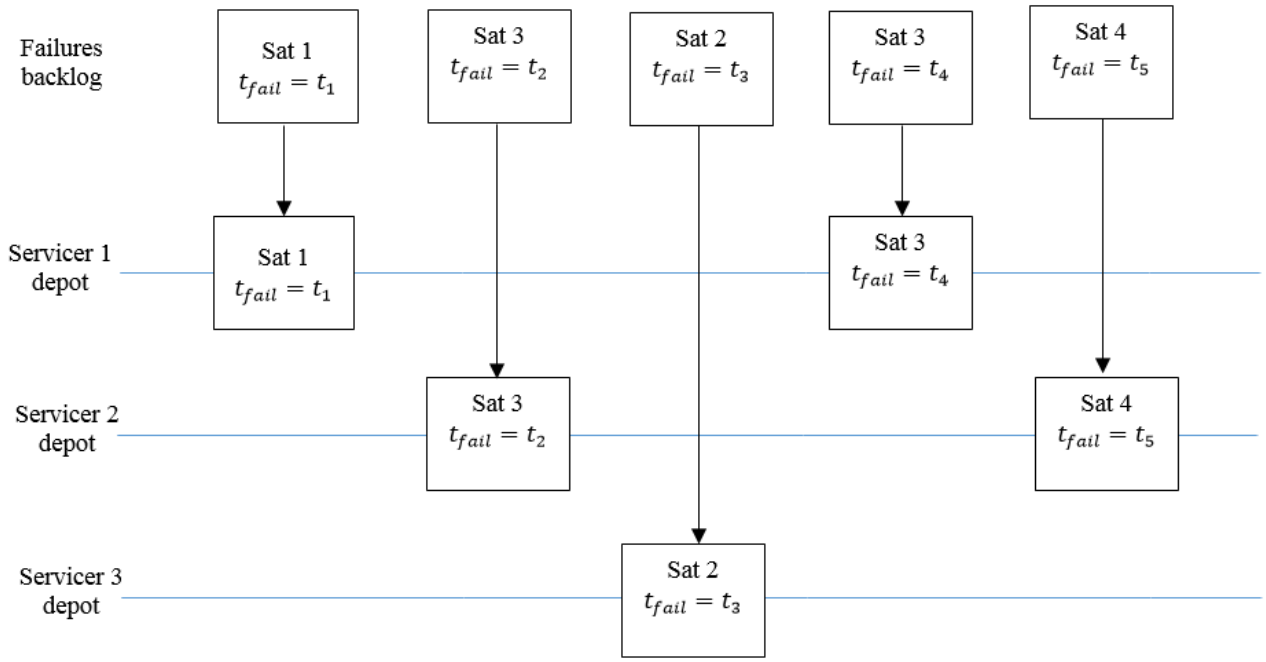

Figure 3.15: SR service dispatch strategy

Average Working State Maximization

While the SR strategy takes each failure in the order of their occurence timing, the AWSM strategy aims to maximize the average working state of the customer satellites by replacing the EUs in the order of priority defined by the type of failure. Fatal failures must be serviced first, then the major failures and finally the minor failures. This strategy also tries to completely fill in the depot of the available servicers before they start their respective servicing operation. This strategy realizes the dispatch in two steps.

The first step, illustrated in figure 3.16, breaks down the original failures backlog into as many backlogs as customer satellites in need for repair. The failures in the backlog associated with each satellite are then sorted with respect to the failure types. The satellite backlogs are then sorted in increasing order of their average working states. At the end of this first step, each satellite backlog is attributed an available servicer, the depot of which is filled in with EUs dedicated to the same satellite in the limit of its capacity. As illustrated in figure 3.16, there can remain “free spots” in the servicers at the end of this first dispatching phase.

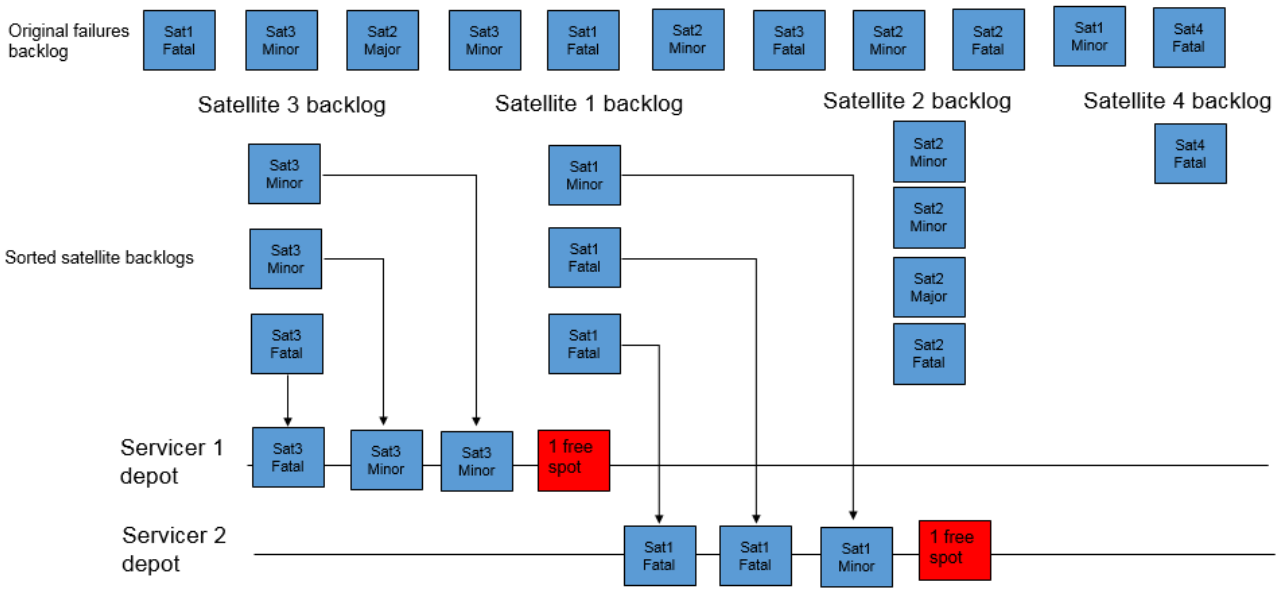

Figure 3.16: First step of the AWSM strategy

The second step of the AWSM strategy consists in filling in the free spots of the available servicers with the service demands which were not dispatched during the first step. In order to do so, the remaining service demands in the satellite backlogs defined in the first step are merged and sorted with respect to the failure types. This is illustrated in figure 3.17.

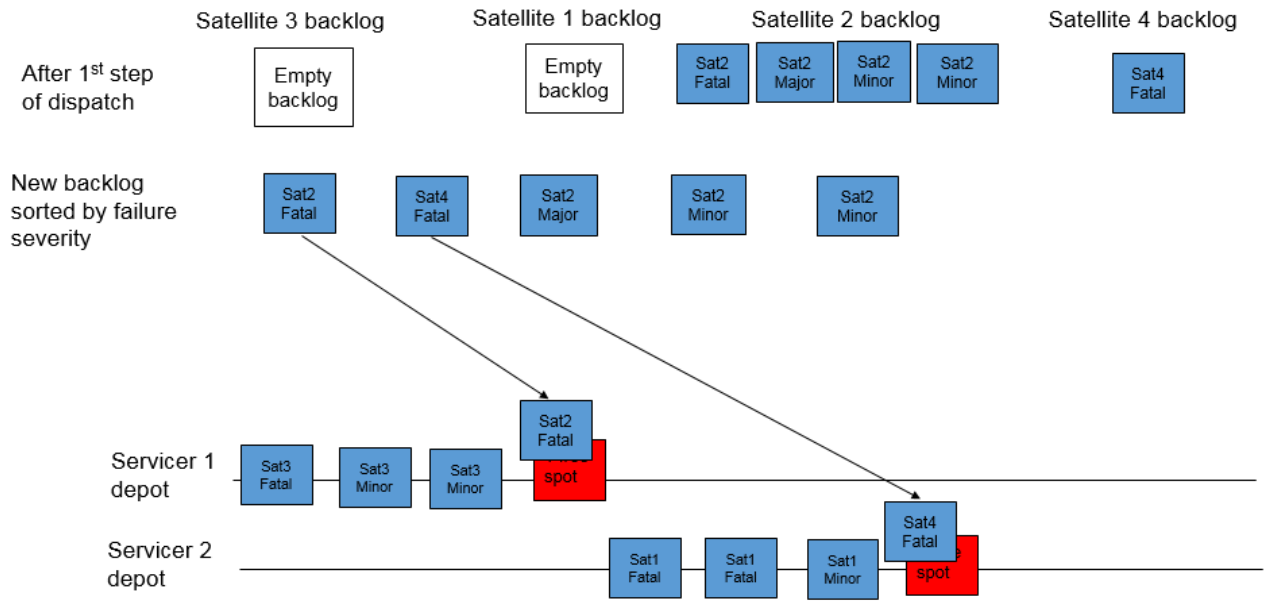

Figure 3.17: Second step of the AWSM strategy

The next two next chapters will present the case studies considered with the Simulink-based and Python-based

simulation frameworks as well as the results of the simulations.

Chapter 4

Trade-Off between "Without Depot" and "With Depot" Infrastructures

This chapter provides insight into the responsiveness of both OSS infrastructures in order to compare them. Cost-effectiveness is also assessed although this is not the primary objective of this trade-off analysis. Section 4.1 describes settings of the simulations and metrics used to perform the trade-off analysis. Section 4.2 presents the results from the simulations and conclude about the trade-off analysis.

4.1 Simulation Settings and Metrics

4.1.1 Simulation Settings

The models associated with each servicing infrastructure includes six identical platforms as defined in the previous subsection, three of them being in MEO at an altitude of 20,000 km and the other three being in GEO at an altitude of 35,786 km. The MEO platforms are equally distributed on their orbit with the initial positions 60deg, 180deg and 300deg. The GEO platforms are also equally distributed on their orbit with the initial positions 90deg, 210deg and 330deg. Table 4.1 gives the parameters considered for both servicing infrastructures. Table 4.2 and table 4.3 give the settings specific to WoD and WD respectively. The choice of the parking orbit of the servicer is considered as part of the trade space and will be analyzed in the next section.

Table 4.1: Parameters common to both servicing infrastructures

Parameter name	*Value*
Simulation time	10 years
Replacement time for one EU	4 hours
Initial angular position of the servicer	0 deg
I_{sp} of the servicer	300 s
Total wet mass of the servicer without EU mass	5000 kg
Maximum capacity EU capacity of the servicer	4
Launch frequency (Falcon 9)	Every 617 hours (26 days)
Launch success probability (Falcon 9)	0.88
Ascent time of launch vehicle (Falcon 9)	1584 s
Altitude at the end of the ascent phase (Falcon 9)	180 km

Table 4.2: Parameters specific to the "Without Depot" infrastructure

Parameter name	*Value*
Maximum EU capacity of the launch vehicle	4 EUs

Table 4.3: Parameters specific to the "With Depot" infrastructure

Parameter name	*Value*
Maximum EU capacity of the launch vehicle	40 EUs
Capacity of the orbital depot for each type of EU	10 EUs
Threshold number of EUs for each type of EU	5 EUs
Initial number of EUs of each type oin the orbital depot	10 EUs

4.1.2 Metrics

The metrics chosen to measure the responsiveness of both servicing infrastructures are their service completion rate and average waiting time before one failed EU is replaced. The first metric is defined as the ratio of the number of services to the number of failures, both of which are assessed at the end of the simulations equivalent to 10 years of servicing operations. The second metric is the average time over 10 years of operation that a failed EU has to wait before being replaced. These metrics must be used concurrently to yield any conclusion. A high service completion rate but a long waiting time does not result in a good system, nor does a short waiting time but a low service completion rate.

The metrics chosen to quantify the cost of the servicing infrastructures are the average amount of propellant consumed by the servicer per service and the utilization ratio of the servicer. Again, the major contribution of the trade-off analysis provided in this section is due to the comparison of the responsiveness between the WoD and WD infrastructures. However, the cost metrics gives some insight into their cost-effectiveness.

4.2 Simulation Results and Analysis

The results are collected by running the simulations for both servicing infrastructures with different altitudes of the servicer's parking orbit over a 10-year period. In subsection 4.2.1, the servicing infrastructures are compared in terms of responsiveness. In subsection 4.2.2, their cost is compared and discussed. Finally, in subsection 4.2.3, a sensitivity analysis with respect to the capacity of the orbital depot in the WD infrastructure is realized.

4.2.1 Responsiveness of the servicing infrastructures

A first general comparison of the servicing infrastructures is conducted by plotting the Pareto fronts between the two metrics defined previously. Figure 4.1 quantifies the benefits of the WD infrastructure with respect to the WoD. It is shown that the WD infrastructure is more responsive than WoD in terms of both service completion rate and servicing time.

As representative examples, two extreme sets of parking orbit altitudes are used to compare the WoD and WD infrastructures. The first set has the altitudes which yield the highest service completion rate for each servicing infrastructure: 21,100 km for WoD and 27,000 km for WD, whereas the second has the altitudes which yield the lowest average waiting time before replacement: 22,300 km for WoD and 25,000 km for WD.

Figure 4.1 summarizes the results. Both sets show a similar trend proving the benefits of the orbital depot. First, for both sets of altitudes, it is shown that about 99% of the failed EUs are replaced by the WD infrastructure whereas it is about 90% for WoD. Therefore, the service completion rate of WD is approximately 10% greater than that of WoD. In addition, for both sets of altitudes, it is shown that the WD infrastructure replaces a failed EU in about one month on average whereas it is more than one year for WoD. Therefore, the WD infrastructure provides services in about 12 times less time than WoD does. Those results can be explained by the presence of the orbital depot in the WD infrastructure which plays the role of a buffer against the relatively low frequency of the launch vehicle with respect to the service frequency of the servicer.

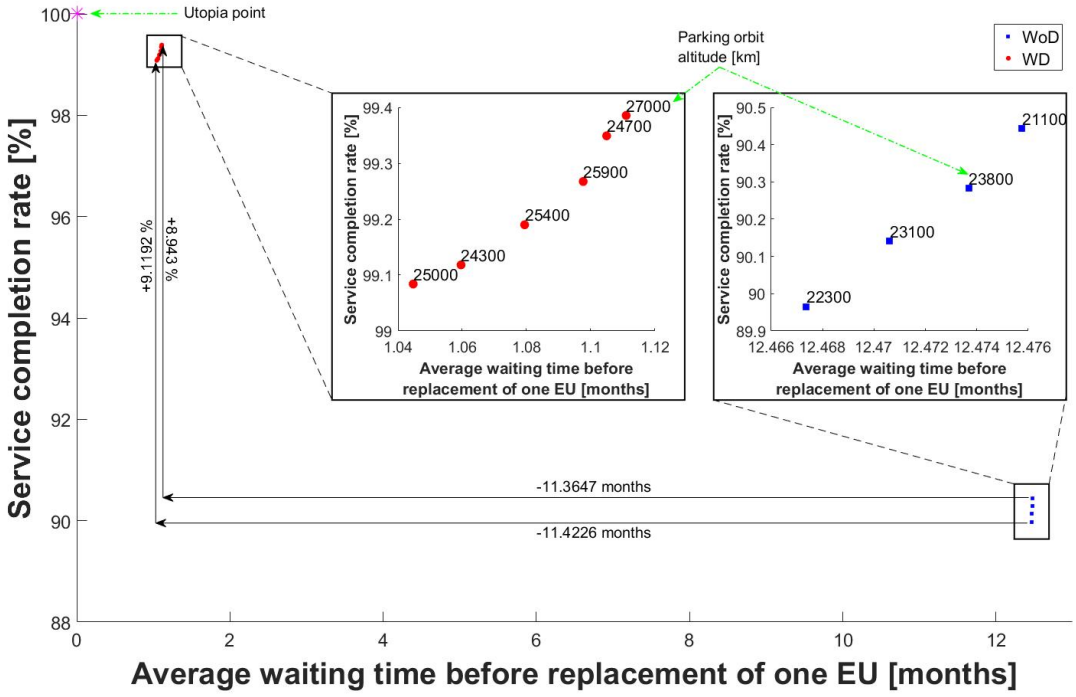

Figure 4.1: Pareto fronts of the servicing infrastructures

Figures 4.2 and 4.3 show each performance metric against the parking orbit altitude. It can be seen that both performance metrics are near optimal when the altitude is between MEO and GEO (23,00030,000 km). This is due to the assumption of using Hohmann transfer, leading to a long waiting time when the parking orbit is close to the target orbit. In addition, we can also see that the range of the altitudes yielding a near optimal performance is sufficiently large for both WoD and WD. Therefore, we do not need to be concerned by the lack of robustness against the variation of the altitude of the parking orbit.

There is an important point to be noted regarding the number of failures with WoD and WD. In the set of altitude 21,100 km for WoD and 27,000 km for WD, for example, WoD completed 511 services out of 565 failures, whereas WD completed 2426 services out of 2441 failures. We can see that WoD has less failures than WD, although the same failure rates are used. This is because it is assumed that once a failure happens to an EU, that EU stops operating. Since it takes longer for the failed EUs in WoD to be replaced than in WD on average, the EUs are operated less frequently in WoD than in WD during 10 years of operation, resulting in less failures in WD than in WoD.

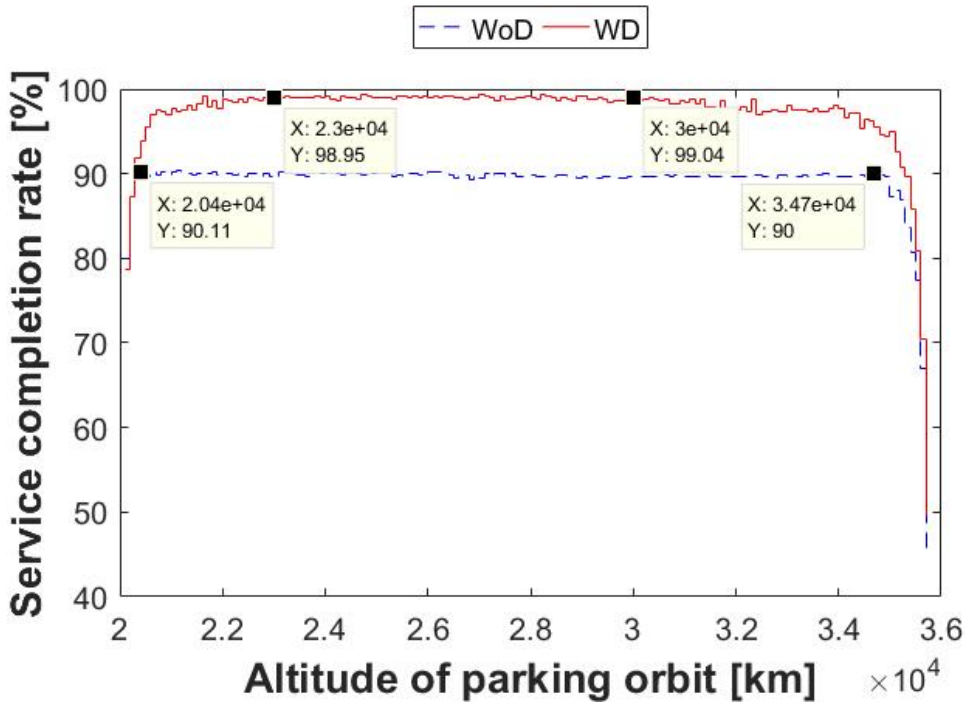

Figure 4.2: Service completion rate as a function of the parking orbit

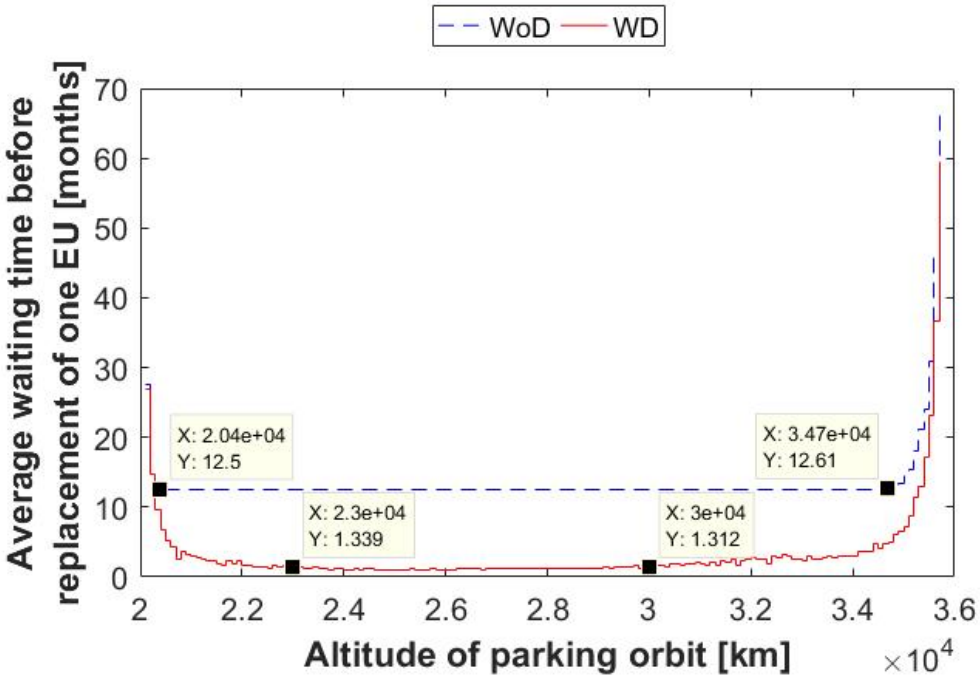

Figure 4.3: Average waiting time before replacement as a function of the parking orbit altitude

In conclusion, it is clear from those results that the WD infrastructure is much more responsive than the WoD infrastructure. However, the operation cost of these infrastructures are likely to be different. The next section provides some insight into it.

4.2.2 Insight into the cost of the servicing infrastructures

The cost of the servicing infrastructures is compared with respect to the amount of propellant the servicer consumes per service and the utilization ratio of the servicer.

First, we evaluate the amount of propellant the servicer consumes per service for each servicing infrastructure. Under the assumptions stated in subsection 3.4.3, the servicer of the WoD infrastructure consumes 392.8 metric tons of propellant for 511 accomplished services, whereas the servicer involved in the WD infrastructure consumes 1866 metric tons of propellant for 2426 accomplished services. Therefore, they consume on average respectively 770.2 kg and 769.5 kg per service. The similarity in these values may be surprising for two reasons: 1) the parking orbit altitudes are different in WoD and WD infrastructures; 2) the orbital depot is present in WD but not in WoD.

First of all, the parking orbit does not significantly affect the propellant consumption per service because there are the same number of satellites in MEO and GEO. In order to understand the impact of the ratio of number of satellites in MEO and GEO in the propellant consumption in WoD and WD, figure 4.4 plots the propellant consumed per service as a function of the parking orbit altitude when there are as many GEO satellites as MEO satellites (3 GEO satellites and 3 MEO satellites), and when there are 5 times more GEO satellites than MEO satellites (5 GEO satellites and 1 MEO satellite). It can be noticed two clearly delimited trends from figure 4.4. When there are as many GEO satellites as MEO satellites, the servicer consumes approximately the same amount of propellant per service no matter the orbit or the OOS infrastructure which is considered. However, when there are more GEO satellites than MEO satellites, the propellant consumption decreases, at the same rate for WoD and WD, as the parking orbit of the servicer gets closer to GEO.

In addition, the existence of depot (i.e., WoD or WD) does not significantly affect propellant consumption per service for any parking orbit altitude due to the high frequency of failures. In both WoD and WD, the servicer departs the parking orbit for one or more services and comes back after they are completed. When the failures occur rarely, WoD and WD can show different results in propellant consumption because the former is likely to service more failures in one service operation due to its long spare procurement process (i.e., launch from the ground). This would

lead to the difference in the propellant consumption in WoD and WD. However, when the failures occur sufficiently frequently, the servicer would almost always load the EUs to its full capacity (4 EUs) for servicing operations in both WoD and WD. This leads to a similar propellant consumption per service for these two cases, as observed in our results.

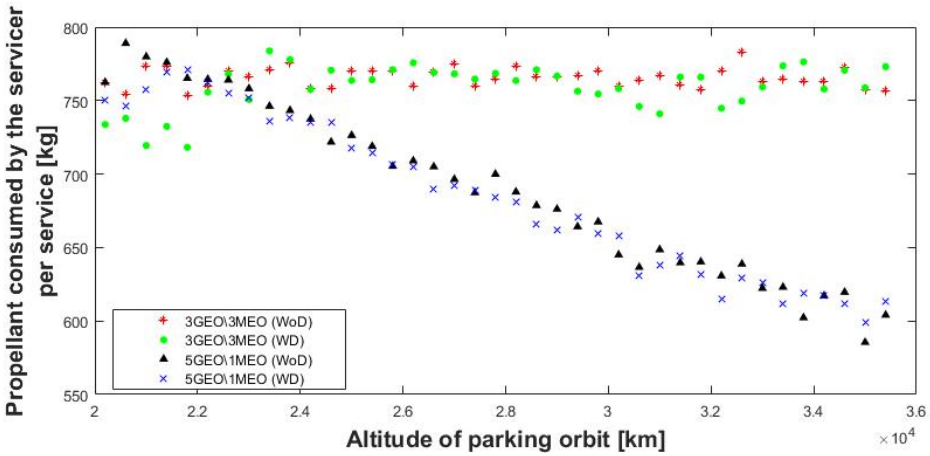

Figure 4.4: Propellant consumed by the servicer per service for both WoD and WD when there are as many GEO satellites as MEO satellites, and when there are more GEO satellites than MEO satellites

The utilization ratio of the servicer also gives a good indication on how much the servicer is used during the 10-year operations. Figure 4.5 shows that the WD infrastructure uses the most of the servicer. Indeed, at the end of 10 years of operation, figure 4.5 demonstrates that the servicer is used 97% of the time in WD whereas it is used only 29% of the time in WoD. WD consequently uses the servicer 70% more frequently than WoD. This is explained by the presence of the orbital depot in WD, which allows for more frequent services than WoD. Although this high utilization ratio of the servicer in WD indicates frequent services, it could also incur more cost to maintain the servicer itself. In fact, although the probability of failure of the servicer during its operations is not modeled in this project, the servicer is more likely to fail or be subject to technical problems when it is used more frequently. This would certainly entail additional maintenance or replacement cost from the point of view of the servicing company. Modeling and quantifying this additional cost is left as future work. In conclusion, the WD infrastructure will be more expensive than WoD because of a higher utilization ratio of the servicer which may require more frequent maintenance or replacement. The next subsection focuses on the sizing of the orbital depot in the WD infrastructure.

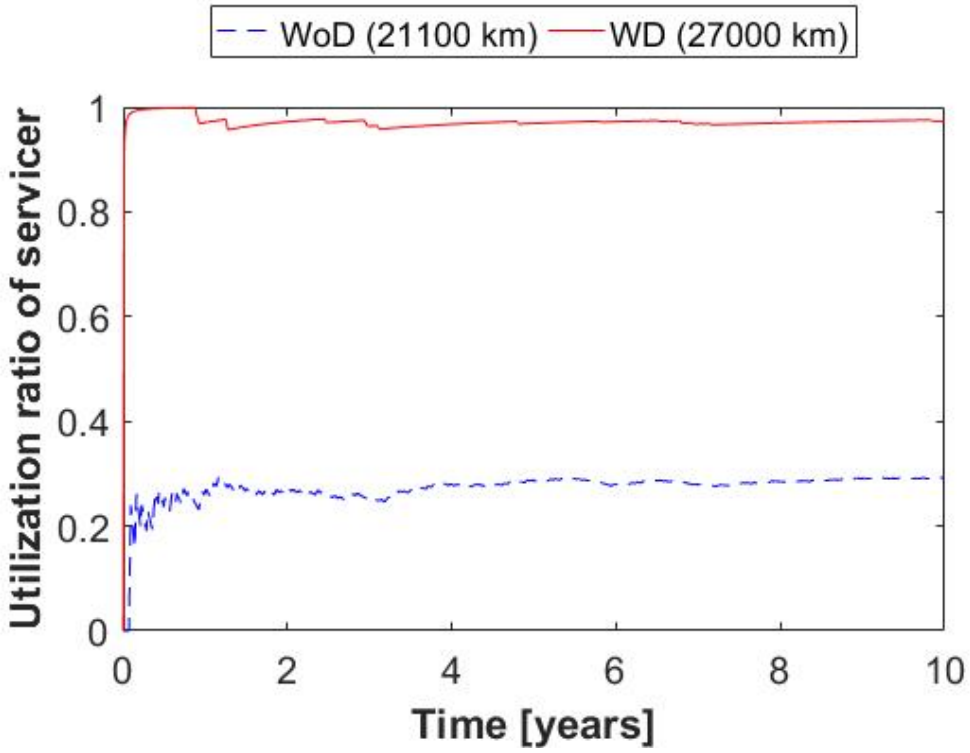

Figure 4.5: Propellant consumed by the servicer per service for both WoD and WD when there are as many GEO satellites as MEO satellites, and when there are more GEO satellites than MEO satellites

4.2.3 Sizing of the orbital depot

From the results presented in subsection 4.2.2, the WD infrastructure is definitely more responsive than WoD. Building on this fact, it is interesting to see how sensitive the responsiveness of the WD infrastructure for a parking orbit at 27,000 km is to the capacity of the orbital depot. For instance, finding the minimum capacity of the depot such that higher capacities will not significantly improve the responsiveness of the infrastructure can be of interest to design a cost-effective orbital depot. In order to give a general trend about this sensitivity, the model of the WD infrastructure is simulated for a depot capacity varying from 8 to 80 EUs by steps of 8 EUs, with the threshold number of EUs being half the depot capacity. For each capacity of orbital depot, the service completion rate and average waiting time are recorded.

Figures 4.6 and 4.7 plot the responsiveness metrics as a function of the depot capacity. It can be observed that the WD infrastructure with a depot capacity of 32 EUs is significantly more responsive than with a depot capacity of 8 EUs because it is about 5 times faster to provide services and its service completion rate is about 3.6% larger. However, beyond a capacity of 32 EUs, responsiveness is not significantly improved. This is because, for the given

launch frequency and capacity of the servicer, there remain EUs in the depot when new spares are launched to orbit for a capacity of 32 EUs and beyond. This shows that having significantly more EUs than 32 EUs is not necessary to obtain a good responsiveness; in fact, in these cases the variation in responsiveness would be dominated by the randomness in the simulations.

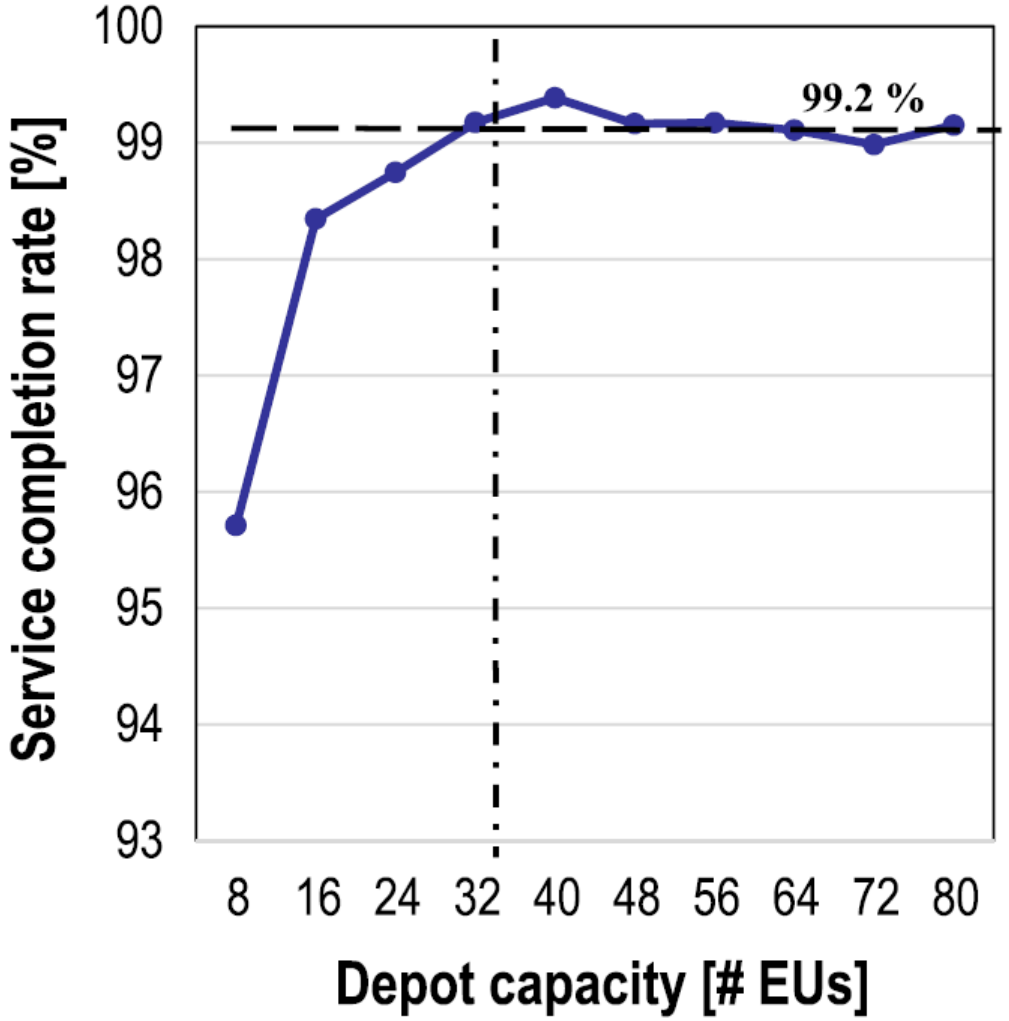

Figure 4.6: Sensitivity of the service completion rate with the depot capacity for a parking orbit of the servicer at 27,000 km

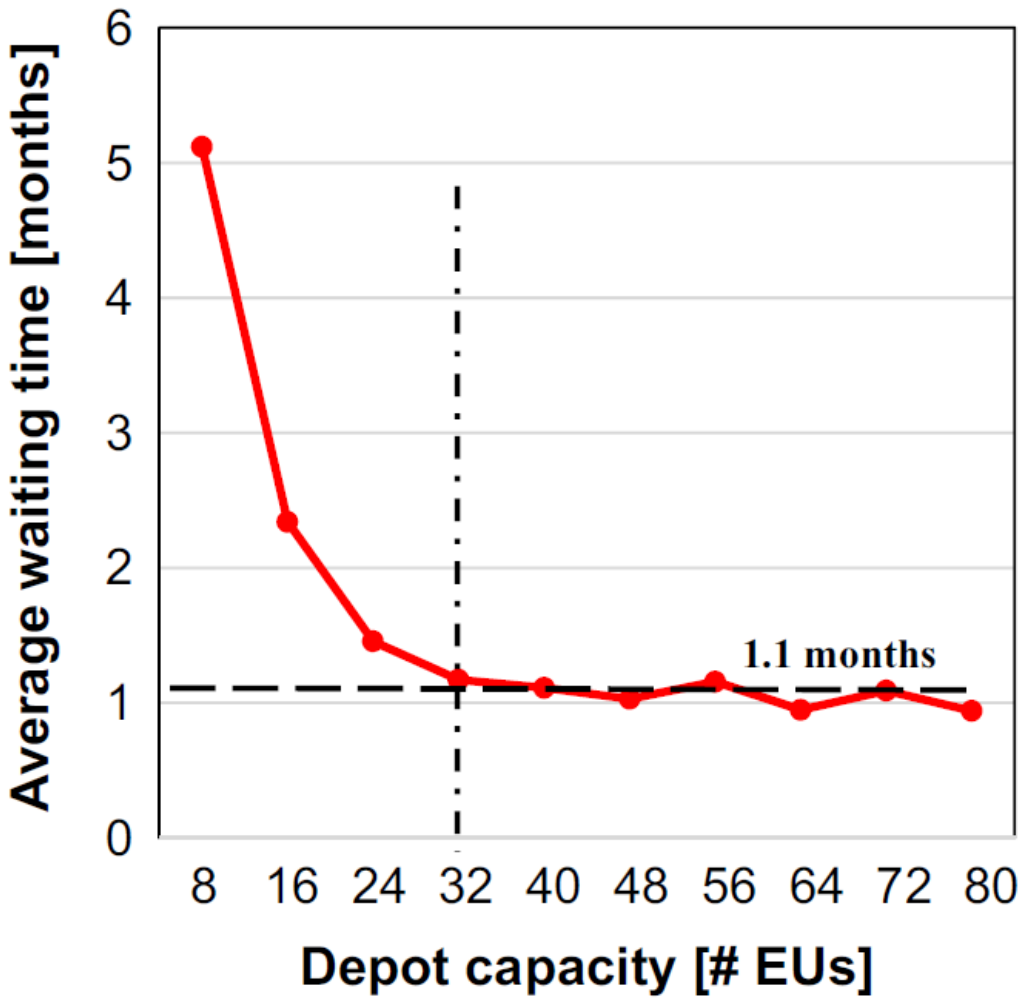

Figure 4.7: Sensitivity of the average waiting time before EU replacement with the depot capacity for a parking orbit of the servicer at 27,000 km

In conclusion, the sensitivity analysis of the responsiveness of the WD infrastructure with respect to the capacity of the orbital depot shows that the larger the depot is, the better the responsiveness, although there is a depot capacity beyond which the responsiveness of the infrastructure is not significantly improved.

Chapter 5

Trade Space Exploration of the "With Depot" Infrastructure

This chapter explores the trade space of the WD infrastructure by assessing the responsiveness and cost-effectiveness of a multitude of designs. Section 5.1 gives the settings of the simulations and the metrics used to perform the responsiveness and cost-effectiveness analysis. Section 5.4 presents the results obtained for the designs using propellant optimal trajectories, based on Hohmann transfers and phasing maneuvers only. Section 5.5 presents the results obtained for the designs using propellant-time-traded trajectories, based on combinations of Hohmann transfers, phasing maneuvers and Lambert trajectories. Finally, section 5.6 compares the scenarios with propellant optimal and propellant-time-traded trajectories before showing that finding a design maximizing responsiveness while limiting the cost is not trivial due to the large number of design variables.

5.1 Simulation Settings and Metrics

Given the very large trade space offered by the Python-based simulation framework, it was necessary to select the design variables most likely to have impact on the responsiveness and cost-effectiveness of the OOS infrastructure. Subsection 5.2 defines the chosen design variables as well as their ranges. Subsection 5.3 defines the metrics used to quantify the responsiveness and cost effectiveness.

5.2 Simulation Settings

The design variables chosen for the simulations as well as their ranges are presented in table 5.1. Note that the servicer propellant capacity corresponds to the propellant mass $M_{p,spe}$ specified by the user of the framework. This mass is compared to the minimum propellant mass required for the servicers to perform the most expensive servicing operations as explained in section 3.4.3. When this mass is lower than the computed minimum propellant $M_{p,req}$, the design is considered to be infeasible.

Table 5.1: Parameters specific to the "With Depot" infrastructure

Design Variables	*Values*
# servicers	1, 3, 5
Servicer propellant capacity [kg]	2000, 5000
Servicer EU capacity	1, 2, 3, 4
Depot EU capacity for each type of EU	5, 10, 15, 20
Initial number of EU of each type in depot	5, 10, 15, 20
Depot threshold number for each type of EU	5, 10, 15, 20
Service dispatch strategy	SR, AWSM
Depot altitude [km]	30000, 35000, 35786
Trajectory type	Propellant optimal, Propellant-time-traded

The simulations were run for the equivalent of 10 years of operation. Six GEO customer satellites were modeled and simulated.

5.3 Metrics

In order to measure the responsiveness of the OOS infrastructures, the working state of each customer satellite is averaged over time. The last value of the time-averaged working state of each satellite is then used to compute the arithmetic average over the number of platforms. This allows to have one single metric characteristic of the whole population of satellites. This average working state is measured in percentage.

Cost effectiveness is measured as the average mass sent to orbit per year, in kg/year. This mass includes the mass of the orbital depot, the mass of the servicers, and the propellant mass consumed by the servicers.

5.4 Propellant Optimal Trajectories

The OOS infrastructure designs investigated in this section were simulated by using propellant optimal rendezvous trajectories only, i.e. Hohmann transfers when the initial and final orbits are different, and phasing maneuvers when they are identical. The first, second and third subsections deal with the analysis of the data obtained for one servicer, three servicers and five servicers respectively.

5.4.1 Scenarios with one servicer

All the data points for the case with only one servicer are plotted on figure 5.1. Clearly, the AWSM service dispatch strategy increases the responsiveness of the OOS infrastructure while limiting the cost with respect to SR. As an example, lets choose the following design: a propellant tank capacity of 5000 kg, a capacity of the servicer equal to 2, a depot located in GEO and with a capacity of 15 for each EU type. In this case, using the AWSM service dispatch strategy yields an average working state of the satellites equal to 23.9% for a mass sent to orbit per year equal to 234716 kg. In the SR case, average working state is 6.3% only for a mass sent to orbit per year of 231334 kg. This observation can be generalized to almost all OOS design. Since we are interested in the most responsive OOS infrastructures, we will keep only the AWSM data for the remaining analysis of the scenarios with one servicer and propellant optimal trajectories.

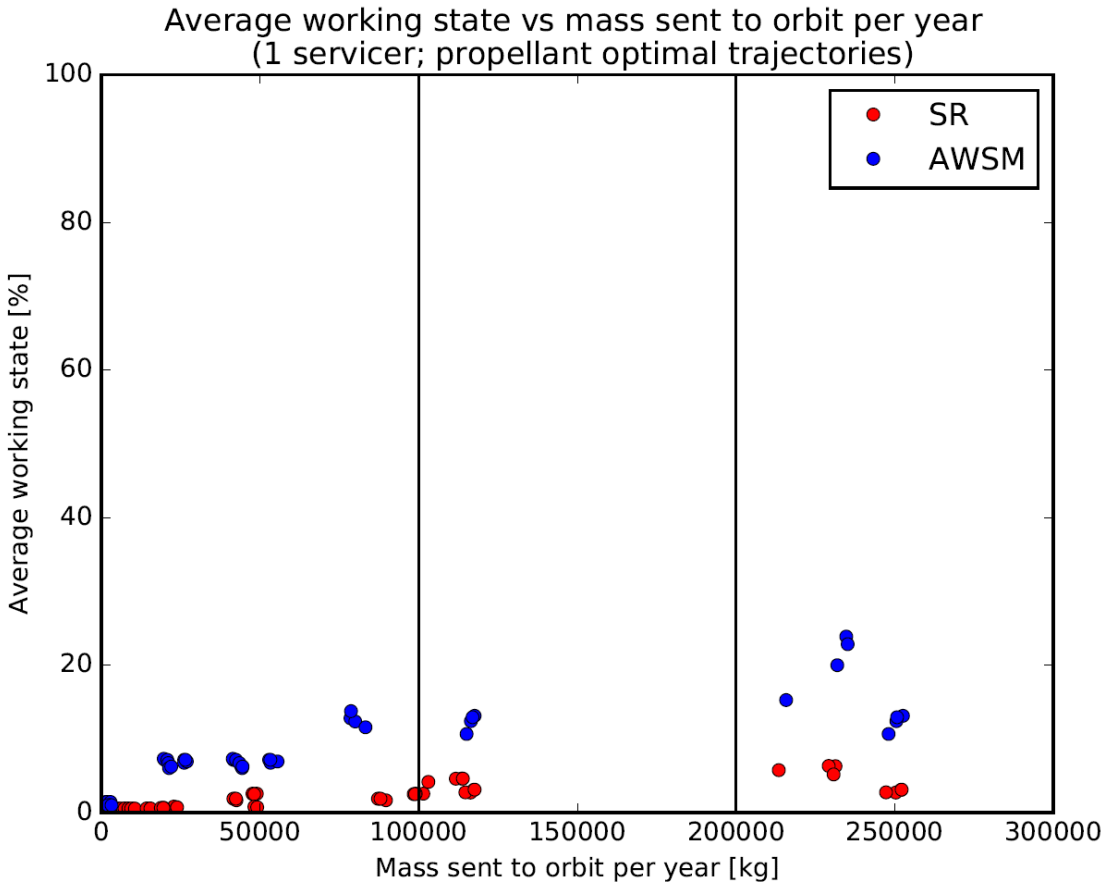

Figure 5.1: Data clouds for one servicer and propellant optimal trajectories

Figure 5.2 gives an idea of the trend which can be observed per design group. A design group is here defined with a 3x1vector. The first element of this vector is the propellant tank capacity of the servicer. The second one is the EU

capacity of the servicer. The third one is the altitude of the orbital depot. It must be noted that the scales on figure 5.2 are not consistent with that on figure 5.1.

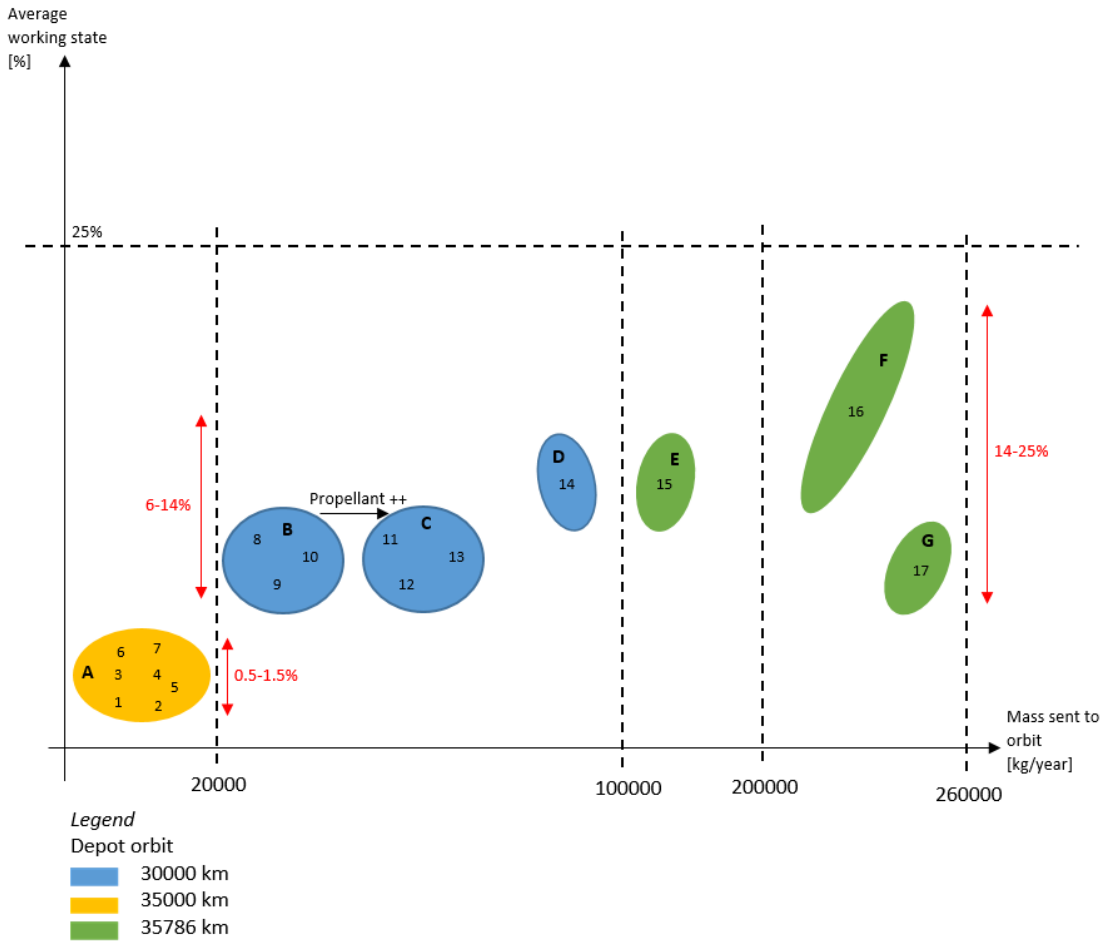

Figure 5.2: Data analysis for one servicer, propellant optimal trajectories and AWSM strategy

In figure 5.2, the indices correspond to the following design groups:

1 : (2000, 1, 35000)

2 : (5000, 1, 35000)

3 : (2000, 2, 35000)

4 : (5000, 2, 35000)

5 : (5000, 4, 35000)

6 : (2000, 3, 35000)

7 : (5000, 3, 35000)

8 : (2000, 1, 30000)

9 : (2000, 2, 30000)

10 : (2000, 3, 30000)

11 : (5000, 1, 30000)

12 : (5000, 2, 30000)

13 : (5000, 3, 30000)

14 : (5000, 4, 30000)

15 : (2000, 1, 35786)

16 : (5000, 2, 35786)

17 : (5000, 1, 35786)

Note that certain design groups, such as (2000, 3, 35786) and (5000, 4, 35786), are not represented because the amount of propellant specified is not sufficient to simulate the worst case servicing operations.

For the following analysis, the colored regions on figure 5.2 will be called bubble followed by their indices (A, B, C, D, E, F, G). The longer and more tilted with respect to the horizontal axis a bubble is, the more sensitive the responsiveness is to the capacity of the depot.

We first observe from figure 5.2 that there are 3 clearly delimited zones. The first one, for a cost varying between 0 and 20000 kg/year, corresponds to a depot at an altitude of 35000 km. The second one, for a cost varying between 20000 and 100000 kg, corresponds to a depot at an altitude of 30000 km. The third zone is associated with a depot in GEO. This observation can be explained by the fact that when the depot is not in GEO, the servicer has to wait a certain time on its orbit before the phasing angle between its targets position and its own position is appropriate for travel. From the physics of Hohmann transfers, the waiting time is longer when the servicer and its target are on very close circular orbits. This is why the OOS infrastructure is more responsive but also more expensive when the depot is at an altitude of 30000 km than when it is at an altitude of 35000 km. When the depot is in GEO, the servicer does not have to wait to travel which increases the responsiveness and increases the cost further.

The second overall observation is that except for bubble F/design group 16, the trend of all the other design groups is that, as the capacity of the depot increases, the cost increases without impoving the responsiveness. Indeed, for those design groups, the service frequency is so low that even when the capacity of the depot is equal to 5 for all EU types, the depot never becomes empty before the next replenishment from Earth. Therefore in this case, increasing the depot has no effect on the responsiveness. Note that the frequency of the services can be increased by increasing the capacity of the servicers depending on the altitude of the depot. For example, design group 14 is for a servicer with an EU capacity equal to 4. In this case, the responsiveness is sensitive to an increase in depot capacity. However, this

is not true for lower capacities for a depot at 30000 km or 35000 km because even an EU capacity equal to 3 cannot make the service frequency large enough to show the impact of the depot capacity on the responsiveness.

The third observation that we can make is that, as expected, increasing the propellant tank capacity of the servicer does not yield a better responsiveness. This is illustrated by bubble C being the translated version of bubble B towards the right along the horizontal axis. With propellant optimal trajectories, increasing the propellant tank size allows for a larger sizing of the capacity of the servicer. We indeed noticed that the design group with a servicer welcoming up to 4 EUs and a depot located at either 30000 or 35000 km were not simulated in the case where the propellant tank can contain only 2000 kg, because the servicer would not be large enough to perform the most expensive servicing operations.

The last interesting observation is that between bubbles F/design group 16 and G/design group 17, the only variable that varies is the capacity of the servicer. Design group 16 specifies a servicer with a 2-EU capacity whereas design group 17 specifies a servicer with a 1-EU capacity. It is interesting to observe that by decreasing the capacity of the servicer, we both decrease the responsiveness and increase the cost. This is explained by the way the AWSM dispatch strategy was designed. In the AWSM strategy, a servicer is more likely to provide several EUs to one satellite, thereby saving time and propellant expenditure. Consequently, a servicer with a capacity higher than 1 EU will generally provide more responsive and less expensive services than a servicer designed to provide 1 EU at a time to the customer platforms.

The next subsection will present a similar analysis with 3 servicers and propellant optimal trajectories.

5.4.2 Scenarios with three servicers

All the data points for the case with three servicers are plotted on figure 5.3. For the data points corresponding to the most responsive OOS designs, there is a slight difference between using the SR strategy and the AWSM strategy. But overall, AWSM remains the most responsive and least expensive. Since we are interested in the most responsive OOS infrastructure, we will keep only the AWSM data for the remaining analysis of the scenarios with three servicers and propellant optimal trajectories.

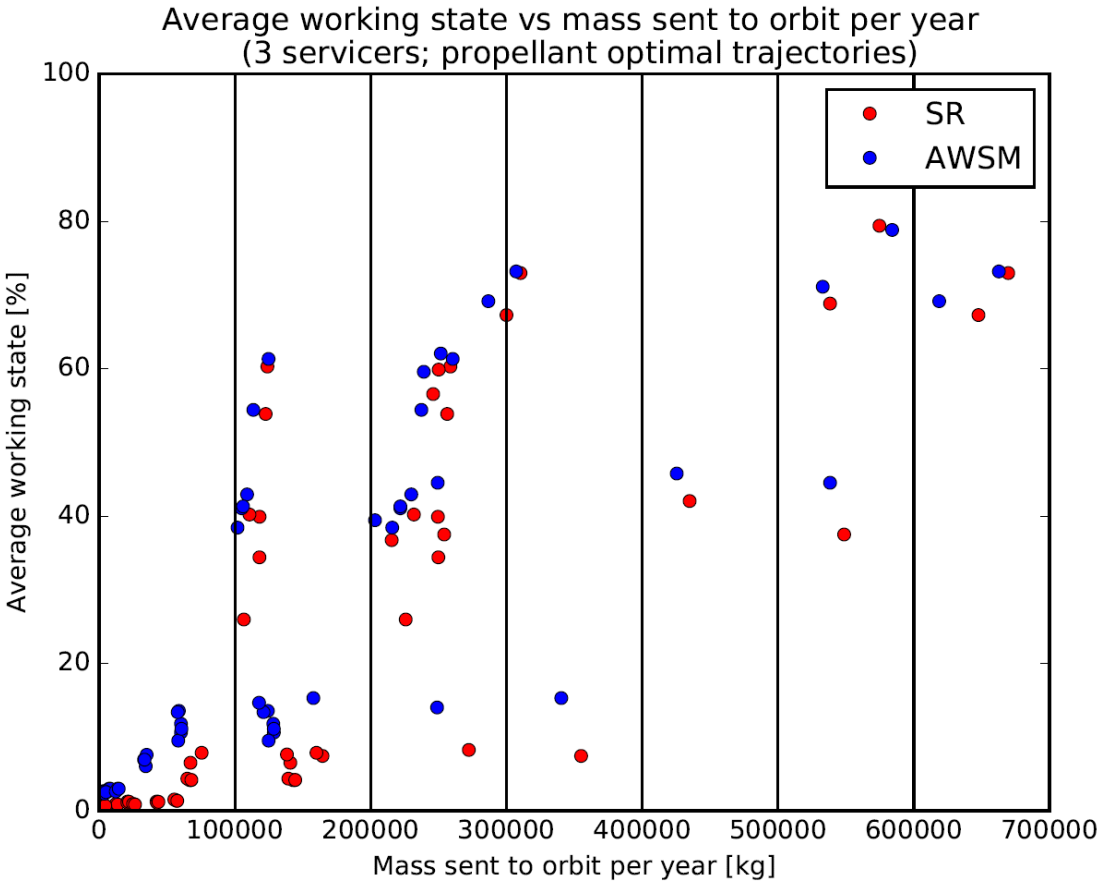

Figure 5.3: Data clouds for three servicers and propellant optimal trajectories

Figure 5.4 gives an idea of the trend which can be observed per design group. The scales on figure 5.4 are not consistent with that on figure 5.3. A design group is defined as in subsection 5.4.1. They are attributed the same indices as defined in subsection 5.4.1. They are recalled below for convenience:

1 : (2000, 1, 35000)

2 : (5000, 1, 35000)

3 : (2000, 2, 35000)

4 : (5000, 2, 35000)

5 : (5000, 4, 35000)

6 : (2000, 3, 35000)

7 : (5000, 3, 35000)

8 : (2000, 1, 30000)

9 : (2000, 2, 30000)

10 : (2000, 3, 30000)

11 : (5000, 1, 30000)

12 : (5000, 2, 30000)

13 : (5000, 3, 30000)

14 : (5000, 4, 30000)

15 : (2000, 1, 35786)

16 : (5000, 2, 35786)

17 : (5000, 1, 35786)

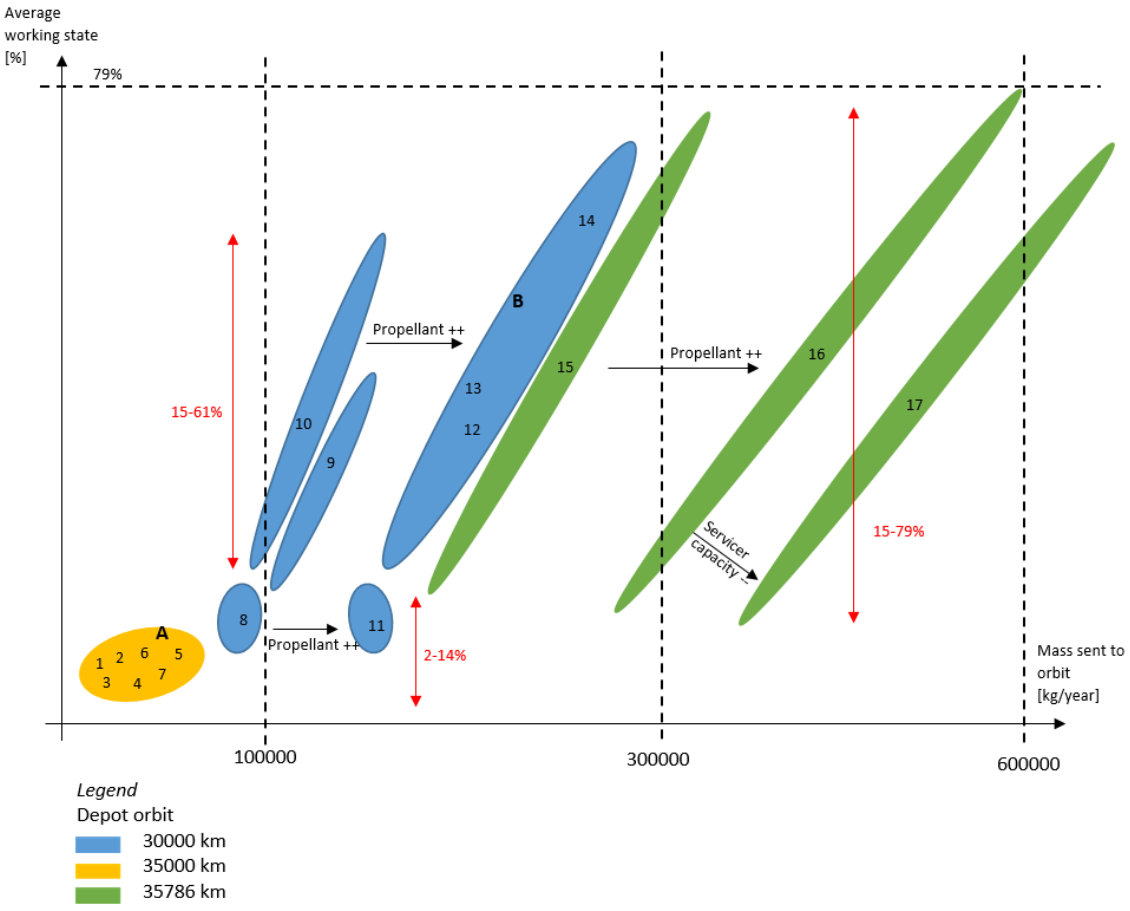

Figure 5.4: Data analysis for three servicers, propellant optimal trajectories and AWSM strategy

We designate by A the bubble including the design groups 1, 2, 3, 4, 5, 6, 7, and by B the bubble including the design groups 12, 13, 14.

We first observe that all the design groups associated with a depot at 35000 km yield a very low responsiveness due to a very low service frequency. Increasing the number of servicers up to 3 does not seem to have as much impact

as the choice of the orbit for the responsiveness to be sensitive to the capacity of the depot.

The second observation one can make is that, as compared to the OOS design groups with only one servicer, the responsiveness becomes very sensitive to the capacity of the depot when it is placed at 30000 km, i.e. further away from GEO. This also increases the cost due to a higher service frequency. For instance, the highest level of average working state with design group 10 and 3 servicers is 61.3% for a cost of 124725 kg/year, whereas it is 7.16% and a cost of 26458 kg/year with 1 servicer. This increase in responsiveness, however, seems to happen only for the case of servicers that can welcome more than 1 EU. Indeed, the design groups 8 and 11 does not follow the same trend as the number of servicers increases. This is again because of the AWSM service dispatch strategy. In other words, design groups 9 and 10, among others, draw more benefit from AWSM than design groups 8 and 11 do.

As for the design groups with 1 servicer only, increasing the size of the propellant tank of the servicers does not improve the responsiveness but costs more. Data points associated with a propellant tank containing 2000 kg are more or less translated along the horizontal axis to the right to get the version with 5000 kg.

Finally, as shown in figure 5.4, increasing the capacity of the servicer from 1 EU to higher levels dramatically increases the sensitivity of the responsiveness with respect to the capacity of the depot. For instance, in the 1-servicer case, switching from design group 8 to design group 10 with a depot capacity of 20 for each type of EUs, yields a quasi-null gain in responsiveness. However, in the 3-servicer case, the gain in average workings state in design group 10 is about 4.5 times that of design group 8. This effect is a combination of the addition of servicers and the AWSM strategy.

The next subsection will present a similar analysis with 5 servicers and propellant optimal trajectories

5.4.3 Scenarios with five servicers

All the data points for the case with five servicers are plotted on figure 5.5. For the data points corresponding to the most responsive OOS designs, there is, again, a slight difference between using the SR strategy and the AWSM strategy. But overall, AWSM remains the most responsive while limiting the increase in cost. Since we are interested in the most responsive OOS infrastructure, we will keep only the AWSM data for the remaining analysis of the scenarios with one servicer and propellant optimal trajectories.

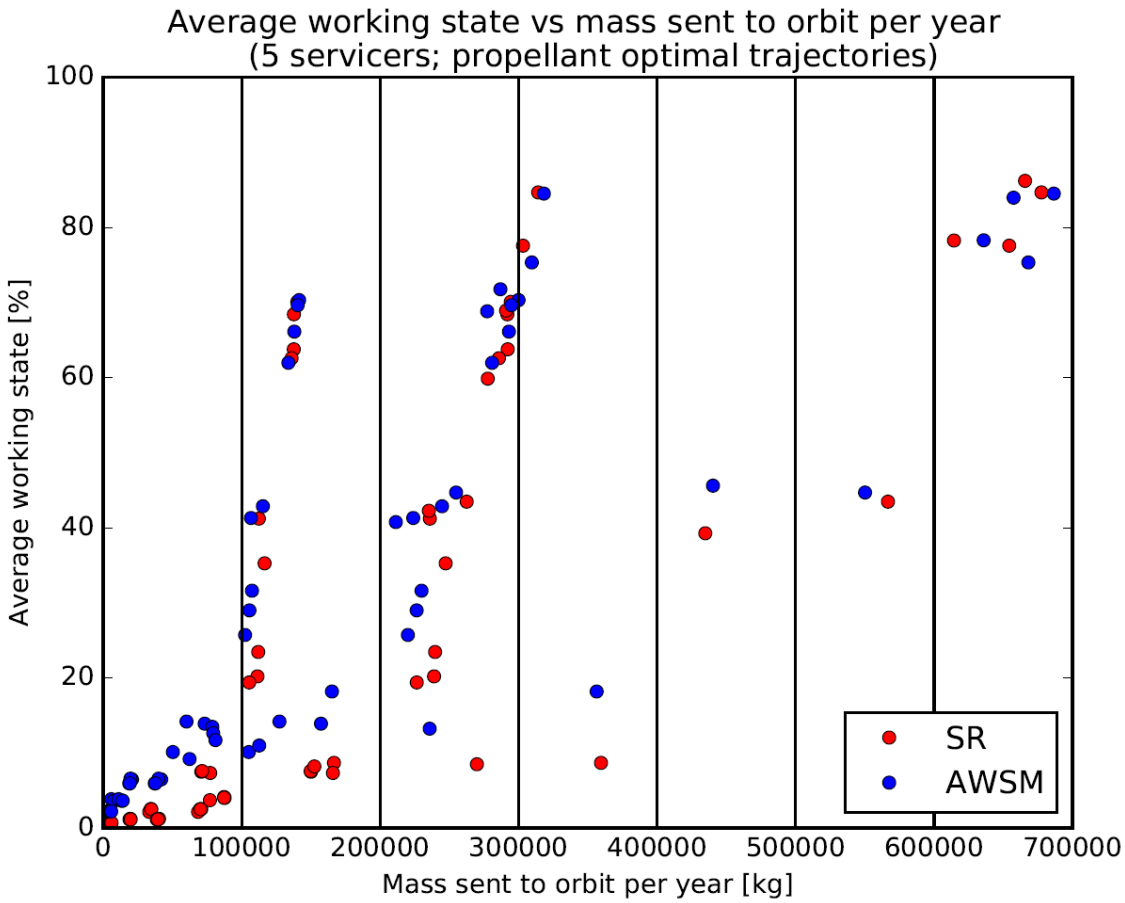

Figure 5.5: Data clouds for five servicers and propellant optimal trajectories

Figure 5.6 gives an idea of the trend which can be observed per design group. The scales on figure 5.6 are not consistent with that on figure 5.5. A design group is defined as in subsection 5.4.1. They are attributed the same indices as defined in subsection 5.4.1. They are recalled below for convenience:

1 : (2000, 1, 35000)

2 : (5000, 1, 35000)

3 : (2000, 2, 35000)

4 : (5000, 2, 35000)

5 : (5000, 4, 35000)

6 : (2000, 3, 35000)

7 : (5000, 3, 35000)

8 : (2000, 1, 30000)

9 : (2000, 2, 30000)

10 : (2000, 3, 30000)

11 : (5000, 1, 30000)

12 : (5000, 2, 30000)

13 : (5000, 3, 30000)

14 : (5000, 4, 30000)

15 : (2000, 1, 35786)

16 : (5000, 2, 35786)

17 : (5000, 1, 35786)

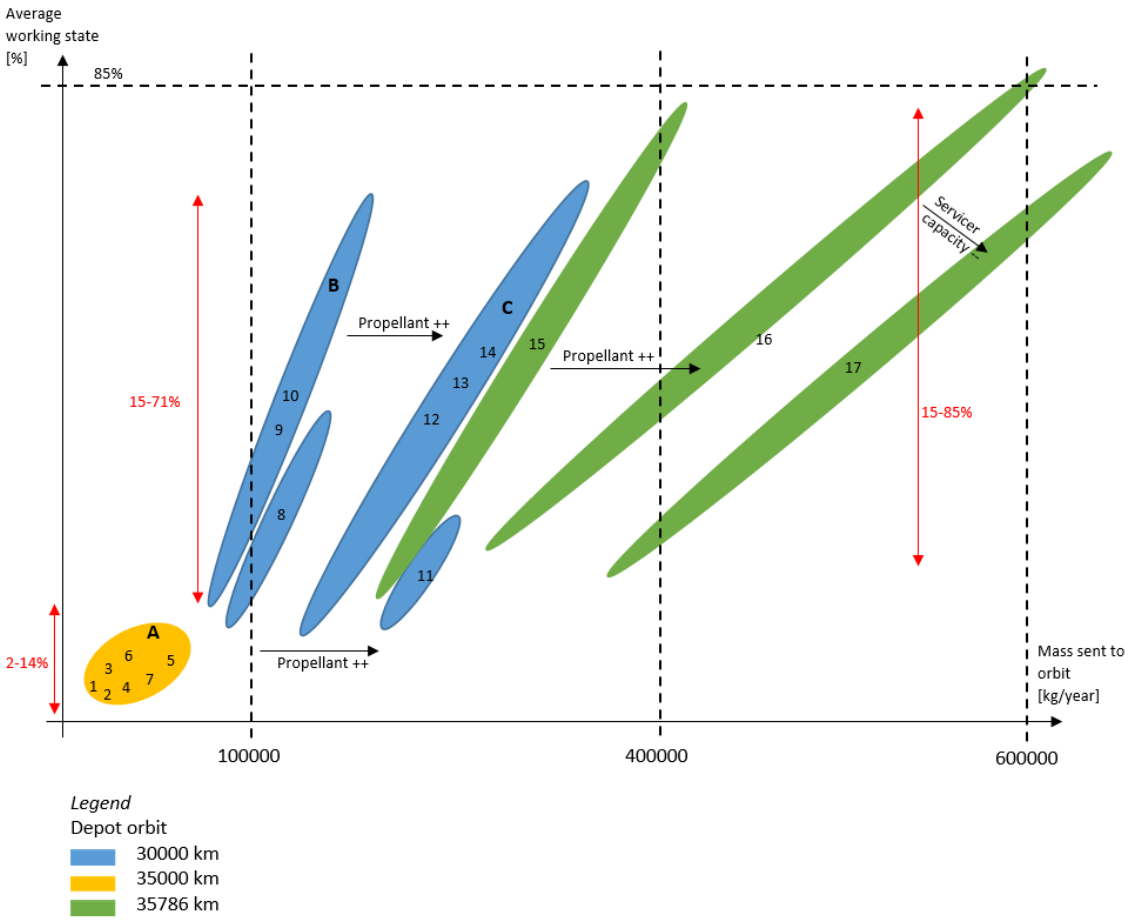

Figure 5.6: Data analysis for five servicers, propellant optimal trajectories and AWSM strategy

We designate by A the bubble including the design groups 1, 2, 3, 4, 5, 6, 7; by B the bubble including the design groups 9, 10; and by C the bubble including the design groups 12, 12, 14.

Overall, the same observations as for the scenarios with 3 servicers can be made. More insight is provided below.

The first observation is that, once again, the design groups associated with a depot at 35000 km are not responsive. In particular, the highest average working state reached by bubble A is about 14%, which is very similar to the case with 3 servicers. This means that, when the choice of the depots orbit induces long waiting times, increasing further the number of servicers will lead to slight changes or no change at all in responsiveness.

The second observation is that increasing the number of servicers from 3 to 5, further increases the overall responsiveness when the orbital depot is in an orbit allowing for frequent services such as at an altitude of 30000 km or in GEO. For instance, with 3 servicers, the highest level of average working state reached with design group 10 is 61.3% for a cost of 124725 kg/year, whereas it is 69.6% for a cost of 140377 kg/year. This corresponds to a 13.5%-increase in responsiveness for a 12.5%-increase in cost.

The last interesting observation is that, when switching from 3 servicers to 5 servicers, certain design groups experience more improvement in responsiveness than others. This can be seen by comparing design groups 8 and 9. In the case of design group 8, with a servicer capacity of only 1 EU, increasing the number of servicers from 3 to 5 leads to a 185%-increase in responsiveness although it was only a 56%-increase between 1 servicer and 3 servicers. Those results about design group 8 can be compared to the results of design group 9, which considers servicers with a capacity of 2 EUs. For this design group, we observe a 588%-increase in responsiveness between the cases with 1 servicer and with 3 servicers while it is only a 63.8%-increase in responsiveness between the 3-servicer and 5-servicer cases. From the data, no general trends can be consistently given but there are scenarios (e.g design groups 8 and 9) for which the responsiveness is more or less sensitive to the increase in number of servicers.

The next section will deal with the analysis of the data in the case of propellant-time-traded trajectories.

5.5 Propellant-Time-Traded Trajectories

The OOS infrastructure designs investigated in this section were simulated using a combination of propellant optimal and Lambert trajectories to trade travel times against propellant expenditure. The motivation for this trade-off between propellant optimal and Lambert trajectories is to see what is the impact, on the Pareto fronts, of saving time but consuming more propellant. This effect will be analyzed in section 5.6. The decision making algorithm used to trade travel times against propellant expenditure is explained in section 3.4.3. The first, second and third subsections of this section deal with the analysis of the data obtained for one servicer, three servicers and five servicers respectively.

5.5.1 Scenarios with one servicer

All the data points for the case with only one servicer are plotted on figure 5.7. Although it is not often as clear as with propellant optimal trajectories, the AWSM service dispatch strategy increases the responsiveness of the OOS infrastructure while limiting the cost with respect to SR. As an example, lets choose the following design: a propellant tank capacity of 5000 kg, a capacity of the servicer equal to 2, a depot located in GEO and with a capacity of 15 for each EU type. In this case, using the AWSM service dispatch strategy yields an average working state of the satellites equal to 34.2% for a mass sent to orbit per year equal to 482359 kg. In the SR case, average working state is 12.3% only for a mass sent to orbit per year of 395908 kg. The responsiveness in the AWSM case increases by 178% with respect to SR while the cost only increases by 21.8%. The trend outlined in this example can be generalized to almost all data points. Since we are interested in the most responsive OOS infrastructures, we will keep only the AWSM data for the remaining analysis of the scenarios with one servicer.

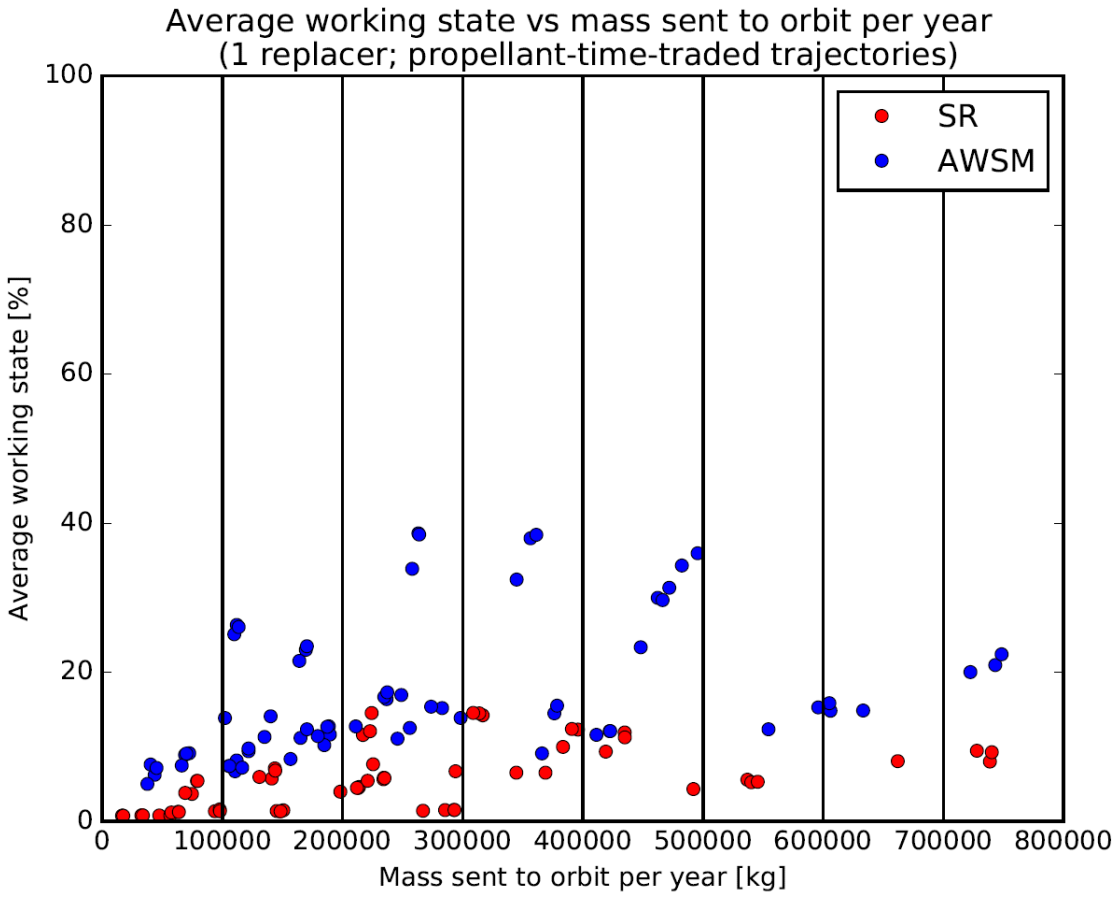

Figure 5.7: Data clouds for one servicer and propellant-time-traded trajectories

Figure 5.8 gives an idea of the trend which can be observed per design group. It must be noted that the scales

on figure 5.8 are not consistent with that on figure 5.7. A design group is defined exactly as in section 5.4. For this section we kept the same assignment of design indices to the design groups as in section 5.4. This may allow the reader to compare each design group more easily between the propellant optimal and propellant-time-traded cases. For convenience, the design groups are recalled below:

1 : (2000, 1, 35000)

2 : (5000, 1, 35000)

3 : (2000, 2, 35000)

4 : (5000, 2, 35000)

5 : (5000, 4, 35000)

6 : (2000, 3, 35000)

7 : (5000, 3, 35000)

8 : (2000, 1, 30000)

9 : (2000, 2, 30000)

10 : (2000, 3, 30000)

11 : (5000, 1, 30000)

12 : (5000, 2, 30000)

13 : (5000, 3, 30000)

14 : (5000, 4, 30000)

15 : (2000, 1, 35786)

16 : (5000, 2, 35786)

17 : (5000, 1, 35786)

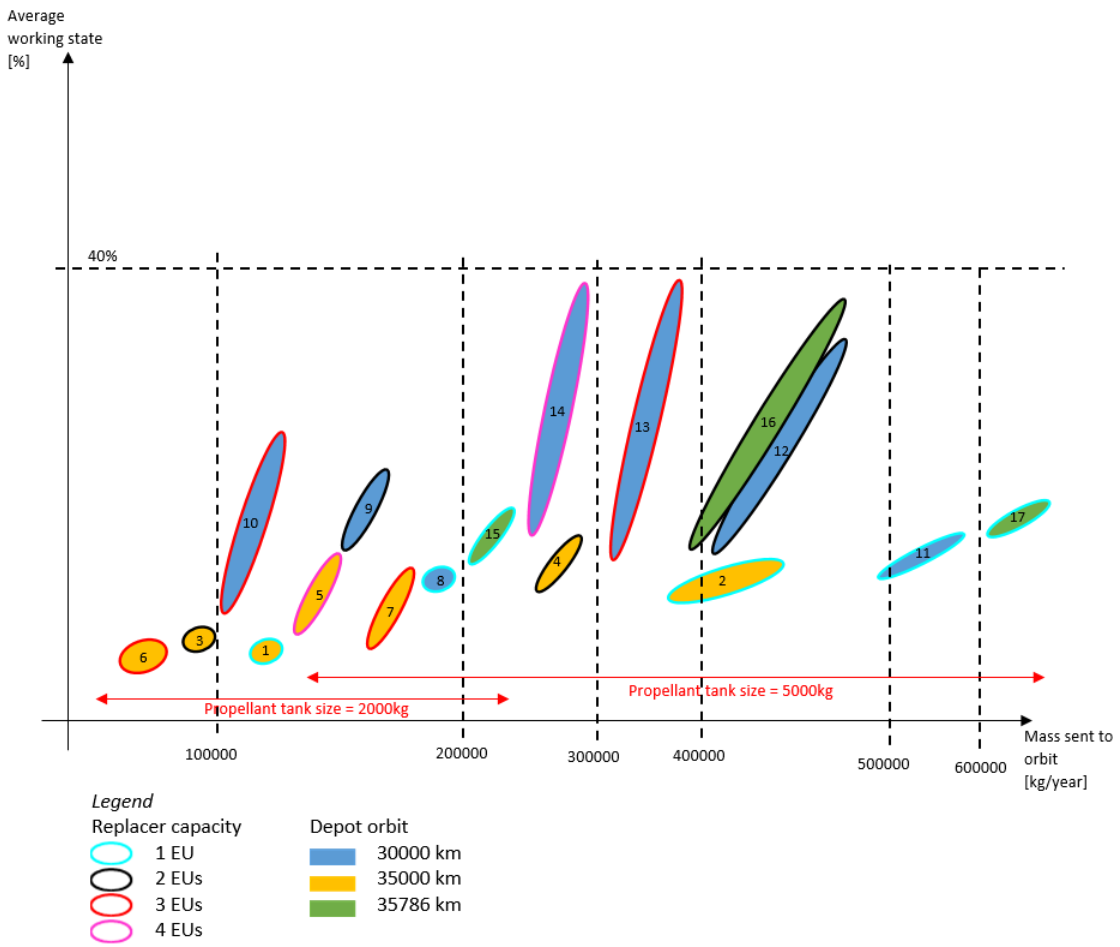

Figure 5.8: Data analysis for one servicer, propellant-time-traded trajectories and AWSM strategy

We first observe that, as compared to the scenarios using propellant optimal trajectories only, there are not three clear zones associated with the three different orbits of the depot anymore. Instead, there seems to be two clearly delimited zones. The first one, on the left side of figure 5.8, is associated with a capacity of the servicers tank equal to 2000 kg. The second one, extending from the middle to the right side of figure 5.8, is associated with a propellant tank capacity of 5000 kg. This can be explained by the fact that giving more propellant to the servicers also allows more flexibility in the trade between travel times and propellant expenditures. The bigger the propellant tank is, the more likely the servicer will be able to travel on fast but expensive trajectories. This explains why the left-most zone shows lower levels of responsiveness than the right-most zone.

We can make the same observation as in section 5.4 regarding the benefit of using the AWSM service dispatch strategy. Again, the trend is very clear, the bigger the EU capacity of the servicer, the more responsive and less expensive. This is illustrated with design groups 14, 13, 12 and 11 in the decreasing order of EU capacity of the servicer.

The third observation we can make is between the design groups with a depot at 30000 km and a depot at 35000 km. The bubbles for an altitude of 30000 km are usually more stretched and tilted towards the vertical direction than those associated with 35000 km, showing a better sensitivity of the responsiveness to the increase in depot capacity. This is because for a depot at 35000 km, if a Hohmann transfer is chosen instead of a Lambert trajectory, due to a lack of propellant in the tank for instance, it will result in a very long waiting time impacting dramatically the final average working state of the customer satellites. In other words, because the trade between propellant optimal and Lambert trajectories sometimes lead to the choice of a Hohmann transfer, the service frequency for a depot at 30000 km remains much higher than for a depot at 35000 km.

Another interesting observation is between design groups 14 and 16. These two design groups share the same propellant tank capacity but have different servicer EU capacities and orbits. In the most responsive case, with a depot of 15 EUs for each EU type, design group 14 costs 263088 kg/year for an average working state of 38.6%. With a depot of 20 EUs per EU type, design group 16 consumes 495483 kg/year for an average working state of 40%. Thus for a similar level of responsiveness, design group 16 consumes 88% more mass than design group 14. This is because design group 14 leverages the benefits of AWSM more efficiently than design group 16 due to a higher EU capacity of the servicer. This trade-off shows that with propellant-time-traded trajectories, it might make more sense to design group servicers with high EU capacities and place the depot in a non-GEO orbit able to provide responsive services, such as 30000 km as opposed to 35000 km. This might be of interest if we want the depot not to be too far from Earth to limit the launch costs for instance.

The next subsection will present a similar analysis for the scenarios with 3 servicers and propellant-time-traded trajectories.

5.5.2 Scenarios with three servicers

All the data points for the case with 3 servicers are plotted on figure 5.9. It is not always clear whether AWSM is more beneficial than SR. However, from how SR and AWSM were implemented, AWSM should give an upper bound of responsiveness beyond which the responsiveness due to SR should never go. The reason why there are rare cases for which this happens is because of randomness. From this observation, SR and AWSM would yield equivalent levels of performance when the design of the OOS infrastructure is such that it can provide frequent services. To remain consistent with the previous section, the AWSM dispatch strategy is considered for the remaining of this subsection.

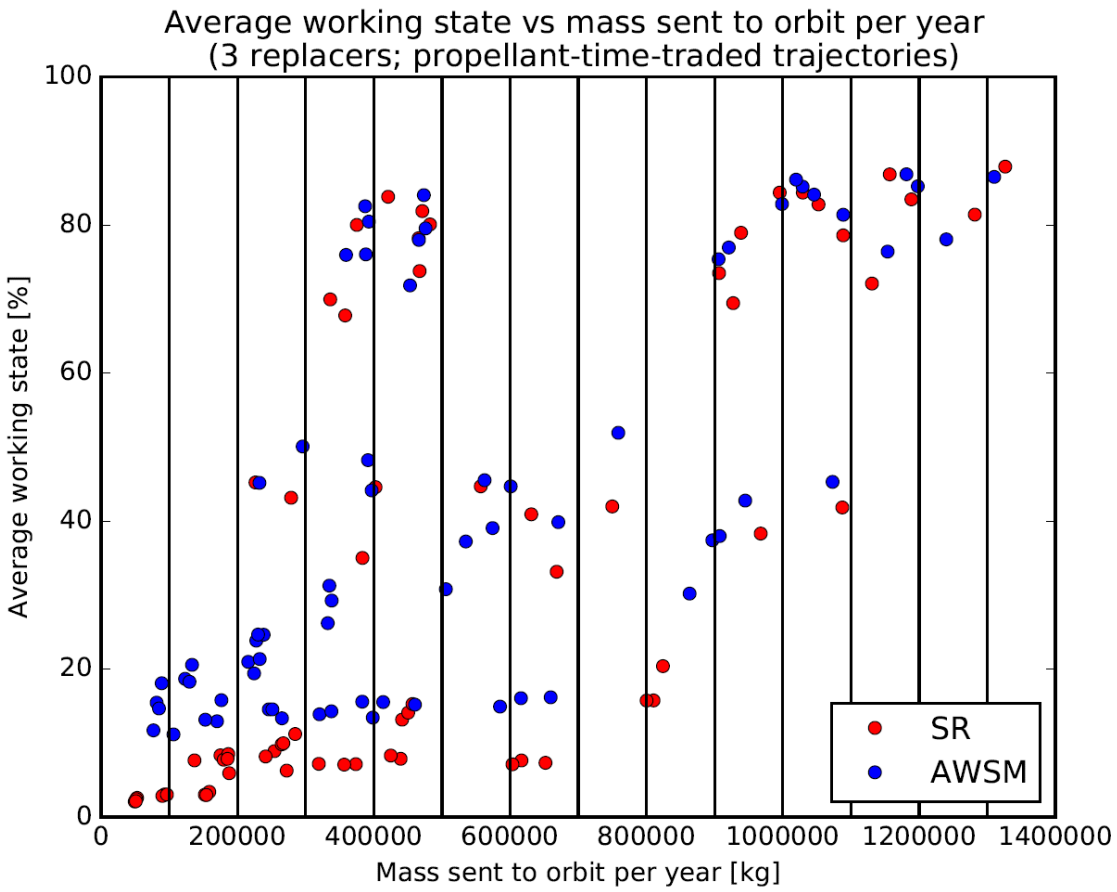

Figure 5.9: Data clouds for three servicers and propellant-time-traded trajectories

Figure 5.10 gives an idea of the trend which can be observed per design group. It must be noted that the scales on figure 5.10 are not consistent with that on figure 5.9. A design group is defined exactly as in section 5.4. For this section we kept the same assignment of design indices to the design groups as in section 5.4. This may allow the reader to compare each design group more easily between the propellant optimal and propellant-time-traded cases. For convenience, the design groups are recalled below:

1 : (2000, 1, 35000)

2 : (5000, 1, 35000)

3 : (2000, 2, 35000)

4 : (5000, 2, 35000)

5 : (5000, 4, 35000)

6 : (2000, 3, 35000)

7 : (5000, 3, 35000)

8 : (2000, 1, 30000)

9 : (2000, 2, 30000)

10 : (2000, 3, 30000)

11 : (5000, 1, 30000)

12 : (5000, 2, 30000)

13 : (5000, 3, 30000)

14 : (5000, 4, 30000)

15 : (2000, 1, 35786)

16 : (5000, 2, 35786)

17 : (5000, 1, 35786)

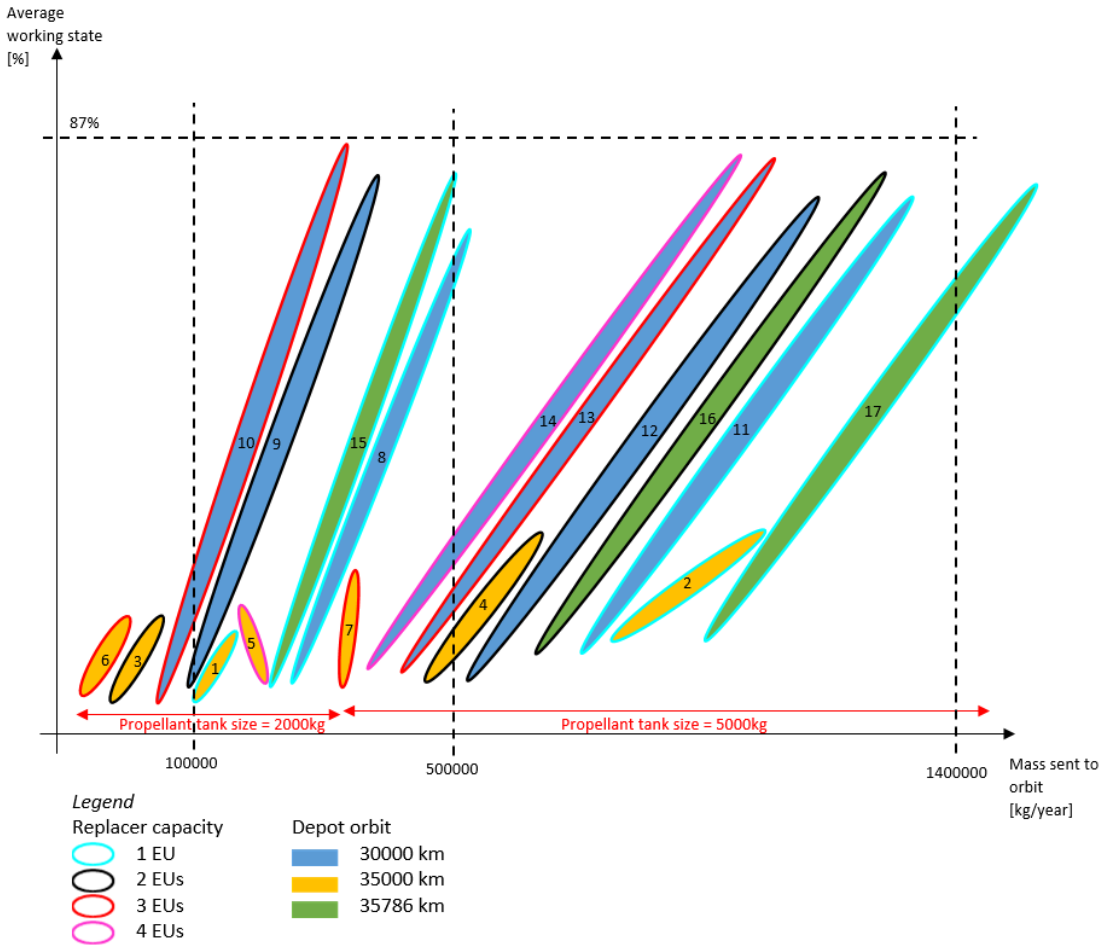

Figure 5.10: Data analysis for three servicers, propellant-time-traded trajectories and AWSM strategy

We first observe by increasing the number of servicers that the differences in responsiveness already existing in the 1-servicer analysis between the scenarios with a depot at 35000 km and a depot at 30000 km have significantly increased. For instance, this difference in responsiveness between design groups 4 and 13 has increased by 100% from

the case with 1 servicer to the case with 3 servicers. Increasing the number of servicers when the depot is at 35000 km does not have as much impact as when the depot is at 30000 km. This is because the service frequency is much higher when the depot is either at 30000 km or in GEO than when it is at 35000 km. The downside of this dramatic increase in responsiveness in some cases is the huge cost, the highest level of which has more than doubled from the 1-servicer case to the 3-servicer case.

The observations made for the 1-servicer case are still valid. In particular, it is still true that the AWSM strategy is all the more beneficial that the capacity of the servicers increases. A final observation which can support interesting design trade-offs, is that increasing the size of the propellant tank of the servicer does not improve significantly the responsiveness of the infrastructure but it can decrease the cost. For instance, design group 10 with a depot of 20 EUs per EU type yields an average working state of 82.5% at a cost of 387221 kg/year whereas design group 13 with a depot of 20 EUs per EU type yields an average working state of 85.2% for a cost of 1029077 kg/year. This represents a 3.3%-increase in responsiveness for a 165%-increase in cost.

The next subsection will present a similar analysis for the scenarios with 5 servicers and propellant-time-traded trajectories.

5.5.3 Scenarios with five servicers

All the data points for the case with 3 servicers are plotted on figure 5.11. It is not always clear whether AWSM is more beneficial than SR. However, on the Pareto front, the OOS design groups mainly use the AWSM dispatch strategy. Therefore the remaining of this subsection will focus on the analysis of the data with AWSM strategy only.

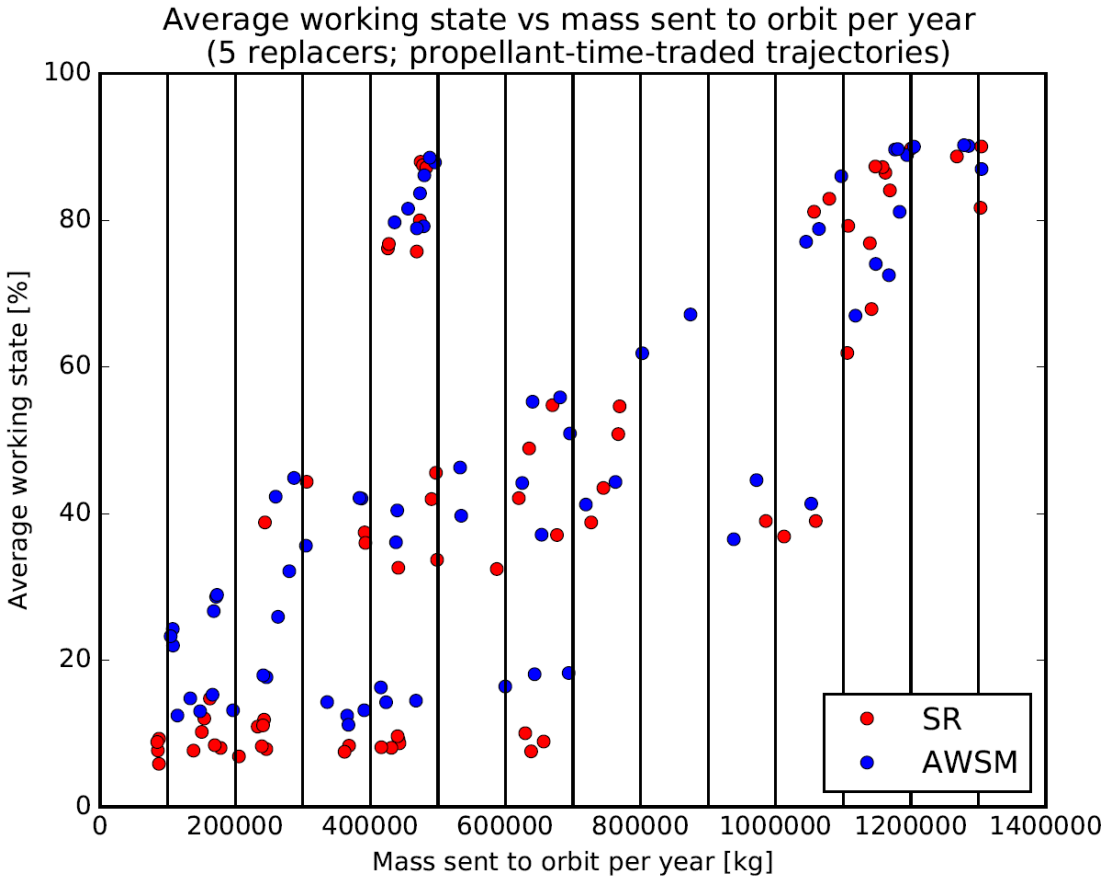

Figure 5.11: Data clouds for five servicers and propellant-time-traded trajectories

Figure 5.12 gives an idea of the trend which can be observed per design group. It must be noted that the scales on figure 5.12 are not consistent with that on figure 5.13. A design group is defined exactly as in section 5.4. For this section we kept the same assignment of design indices to the design groups as in section 5.4. This may allow the reader to compare each design group more easily between the propellant optimal and propellant-time-traded cases. For convenience, the design groups are recalled below:

1 : (2000, 1, 35000)

2 : (5000, 1, 35000)

3 : (2000, 2, 35000)

4 : (5000, 2, 35000)

5 : (5000, 4, 35000)

6 : (2000, 3, 35000)

7 : (5000, 3, 35000)

8 : (2000, 1, 30000)

9 : (2000, 2, 30000)

10 : (2000, 3, 30000)

11 : (5000, 1, 30000)

12 : (5000, 2, 30000)

13 : (5000, 3, 30000)

14 : (5000, 4, 30000)

15 : (2000, 1, 35786)

16 : (5000, 2, 35786)

17 : (5000, 1, 35786)

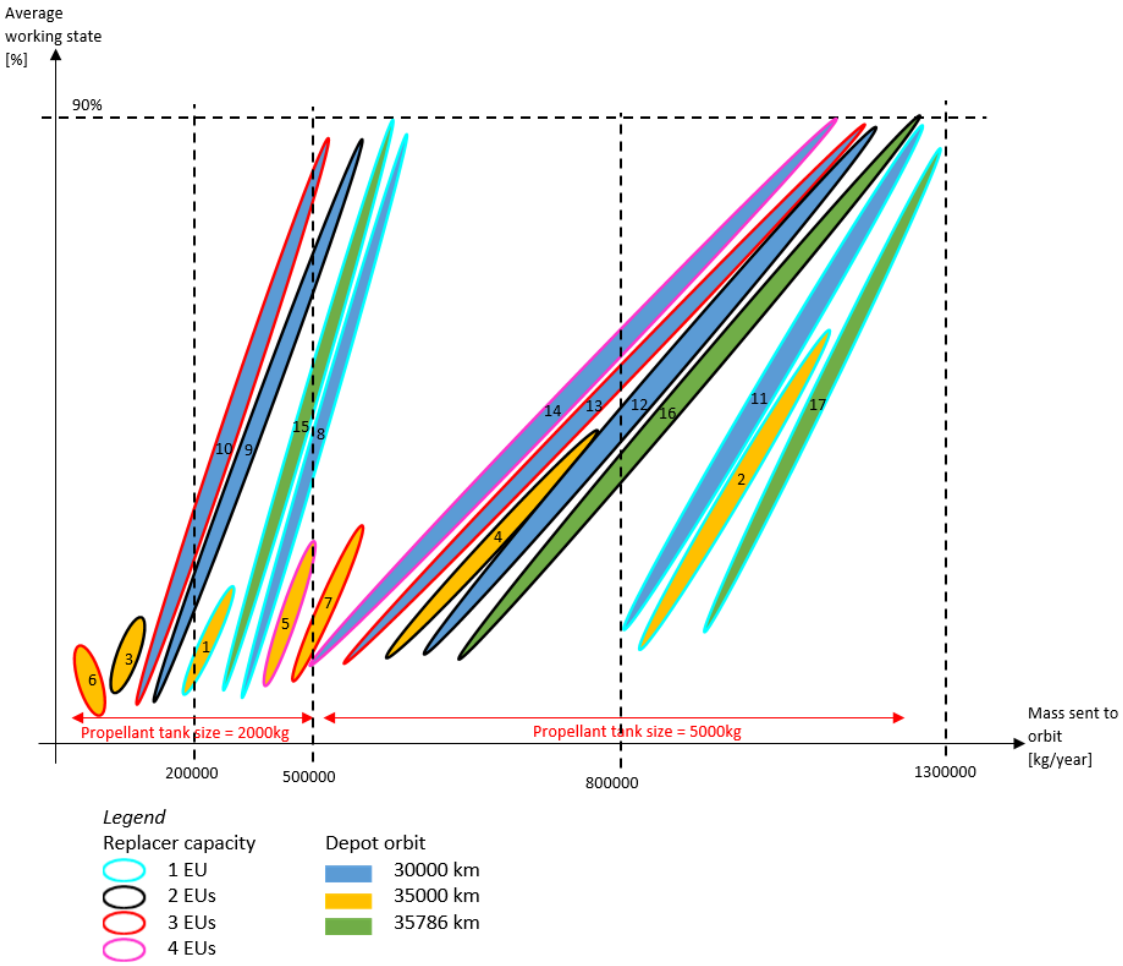

Figure 5.12: Data analysis for three servicers, propellant-time-traded trajectories and AWSM strategy

The trends observed on figure 5.12 are the same as those observed on figure 5.10. The maximum level of responsiveness is 90%, which is equivalent to the case with 3 replacers. Similarly, the cost does not significantly change. This shows that designing an OOS infrastructure with more than 3 servicers allowed to travel on Lambert trajectories would

be sufficient to yield a good responsiveness. Adding more servicers would cost more for no significant improvements in responsiveness.

The next section compares the scenarios with propellant optimal and propellant-time-traded trajectories, and gives examples of OOS design trade-off based on the results obtained with propellant optimal trajectories.

5.6 Comparisons and Discussions

After having analyzed the results in the previous sections according to the types of trajectories and the number of servicers, subsection 5.6.1 deals with comparing the scenarios with propellant optimal trajectories and propellant-time-traded trajectories, while subsection 5.6.2 focuses on the propellant optimal scenarios to show that finding a design maximizing the responsiveness while limiting the cost is not trivial, and that large scale trade space exploration is necessary.

5.6.1 Comparison between propellant optimal and propellant-time-traded trajectories

In order to compare the scenarios with propellant optimal and propellant-time-traded trajectories, figure 5.13 plots the corresponding Pareto fronts.

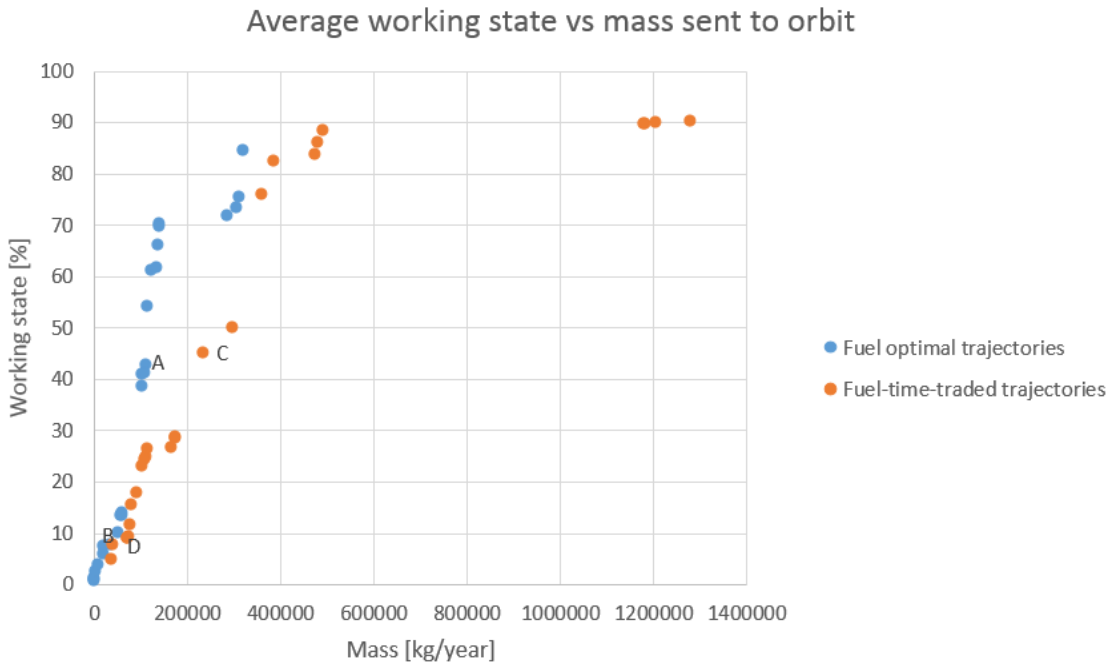

Figure 5.13: Pareto fronts obtained with propellant optimal and propellant-time-traded trajectories

The four designs A, B, C, D were chosen to perform the analysis presented in this subsection. Their characteristics are presented in table 5.2.

Table 5.2: Characteristics and performance of the designs A, B, C, D introduced in figure 5.13

Design	A	B	C	D
# servicers	3	5	3	1
Servicer propellant capacity [kg]	2000	2000	2000	2000
Servicer EU capacity	2	3	3	2
Trajectory type	Propellant optimal	Propellant optimal	Propellant-time-traded optimal	Propellant-time-traded optimal
Depot EU capacity	20	5	10	10
Depot orbit [km]	30000	30000	30000	35000
Average working state [%]	42.9	10.1	45.2	9.14
Mass sent to orbit [kg/year]	108920	50262	232142	72394

Table 5.3 gives the average working state and mass sent to orbit for the same designs with the other type of trajectory. It also gives the increase in responsiveness and cost with respect to the data in table 5.2.

Table 5.3: Performance of designs A, B, C, D using the other trajectory paradigm

Design	A	B	C	D
Trajectory type	Propellant optimal	Propellant optimal	Propellant-time-traded optimal	Propellant-time-traded optimal
Average working state [%]	83.8	13	40.2	0.547
Mass sent to orbit [kg/year]	420936	147881	110666	5283
Increase in average working state	+95%	+30%	-11%	-94%
Increase in mass sent to orbit	+286%	+194%	-52%	-93%

Tables 5.2 and 5.3 show that propellant-time-traded trajectories increase the responsiveness. The downside is that the cost is also significantly increased from propellant optimal trajectories to propellant-time-traded trajectories.

By observing the Pareto fronts in figure 5.13, the propellant-time-traded trajectories enable a higher level of responsiveness than the propellant optimal trajectories. However, the cost is also multiplied by 4. In fact, one can see that the Pareto front obtained with propellant optimal trajectories is slightly more towards the left hand side of the plot than is that associated with propellant-time-traded trajectories. The reason why we observe this trend is because scenarios with propellant-time-traded trajectories will consume more than with propellant optimal trajectories for a same amount of gained responsiveness. For instance, lets consider the designs associated with the A, B, C, D letters on figure 5.13. From the data in table XX, the slope of the line between points B and A is 0.559 %/ton/year whereas it is

0.226 %/ton/year between points D and C. This shows that for a given increase in mass, the scenarios with propellant optimal trajectories will better improve the responsiveness. Because of this trend, OOS designers may prefer consider propellant-optimal trajectories with a more distributed infrastructure, i.e. more servicers and a depot in GEO or far from GEO, to yield a satisfying level of responsiveness with less mass sent to orbit a year.

5.6.2 Trade-off discussion based on propellant optimal trajectories

Given the large number of design variables in this problem, it may be interesting to know whether it is trivial to find a design with a high level of responsiveness at a rather limited cost from an initial design intuited by the OOS designers to be responsive.

The most intuitive design for an OOS infrastructure would be to oversize the fleet of servicers and the orbital depot, and place the depot in GEO. More precisely, lets assume that the OOS designers choose an initial design with 5 servicers with an EU capacity equal to 2 and a propellant capacity equal to 5000 kg, and a depot capable of storing 20 EUs of each type. We want to know whether we can increase the initial responsiveness and decrease the initial cost by changing the value of one variable at a time with respect to the initial design, called design 0. Table 5.4 gives the results for this sensitivity analysis.

Table 5.4: Performance of designs A, B, C, D using the other trajectory paradigm

Design	*# servicers*	*Servicer propellant capacity* [kg]	*Servicer EU capacity*	*Depot EU capacity*	*Depot orbit* [km]	*Average working state* [%]	*Mass sent to orbit* [kg/year]
Design 0	5	5000	2	20	35786	84.0	657390
Design 1	3	5000	2	20	35786	78.8	584059
Design 2	1	5000	2	20	35786	22.8	235166
Design 3	5	2000	2	20	35786	–	–
Design 4	5	5000	1	20	35786	84.5	686466
Design 5	5	5000	2	15	35786	78.3	635811
Design 6	5	5000	2	10	35786	45.6	440312
Design 7	5	5000	2	5	35786	13.2	235715
Design 8	5	5000	2	20	35000	3.65	13999
Design 9	5	5000	2	20	30000	70.3	299916
Optimal	**5**	**2000**	**1**	**20**	**35786**	**84.5**	**318190**

Note that in table 5.4, design 2 does not display responsiveness and cost-effectiveness values because a propellant tank capacity equal to 2000 kg is not sufficient to provide services for the most expensive servicing operations.

As can be seen from table 5.4, there is no case increasing both responsiveness and cost-effectiveness. Based on these results only, OOS designers would be likely to choose design 4 to maximize the responsiveness. However, by

exploring the trade space in a more systematic way, i.e by varying several design variables at a time, they would realize that a similar responsiveness as design 4 can be achieved at a much lower cost. This is demonstrated in figure 5.14 where the designs in table 5.4 have been indicated.

Figure XX shows that there is a design with a similar responsiveness but a lower cost than design 0 or 4. It is called Optimal for the purpose of this analysis. Its characteristics are 5 servicers, a servicer propellant tank capacity equal to 2000 kg, a servicer EU capacity equal to 1, a depot located in GEO and capable of storing 20 EUs of each type.

This analysis shows that non intuitive designs may exist and which can be found either by exploring the trade space by discretizing the ranges of the variables, as done in this research, or by using an optimization algorithm, such as genetic algorithm, to find the optimal design in a more rigorous way.

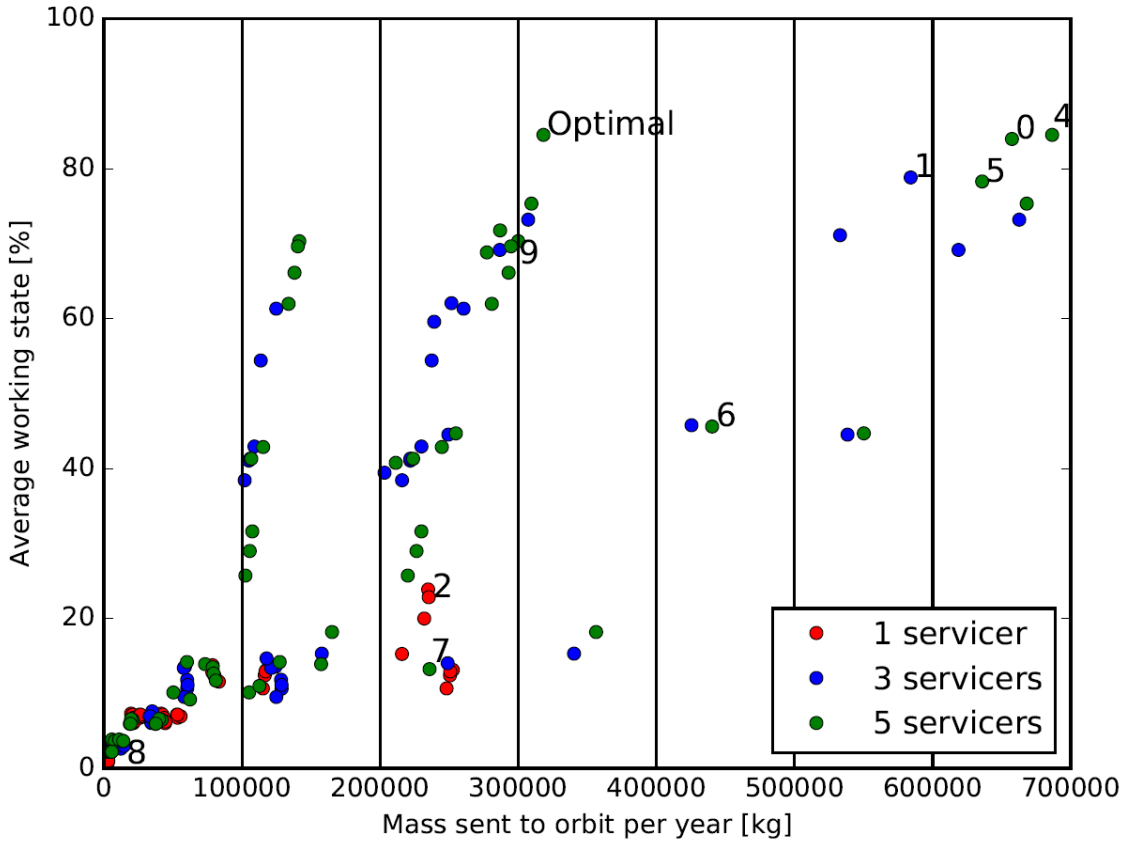

Figure 5.14: Average working state versus mass sent to orbit per year (propellant optimal trajectories)

Chapter 6

Conclusion

This research proposes and evaluates two different robotic on-orbit servicing (OOS) infrastructures using event-driven simulations. The first infrastructure (WoD) is comprised of a launch vehicle and a robotic servicer. The second (WD) integrates a launch vehicle, a robotic servicer, and an orbital depot. The simulations for both infrastructures are run over a period of 10 years of operations. In order to yield realistic results, the launch and platform failures are modeled using probabilistic distributions.

Several MEO and GEO platforms are modeled and simulated by the Simulink-based and Python-based simulation frameworks developed in this research. These platforms are coplanar and identical, owning 11 Elementary Units (EUs) of four different types dedicated to the specific functions of a classical satellite. The failure of those EUs is modeled as a Poisson process, with the average failure times of interest found in the literature [5].

The first study carried out in this research about the trade-off between the WoD and WD infrastructures leads to the conclusion that WD is much more responsive than WoD, but also likely to be more expensive. Regarding responsiveness, the WD infrastructure is shown to provide services about 12 times more rapidly with a service completion rate about 10% greater than that of WoD. More particularly, WD provides a service in one month on average, whereas a service with WoD takes one year. This is explained by the presence of the orbital depot which plays the role of a buffer before the next launch of spare EUs from Earth. It is also observed that the performance of both WD and WoD is robust to variations in the parking orbit of the servicer if it is not near MEO or GEO. Regarding cost-effectiveness, it is first shown that, when there are as many GEO as MEO satellites and when their failure frequency is high enough, both servicing infrastructures are equally costly in terms of propellant expenditure per service provided. What distinguishes WD from WoD regarding cost-effectiveness is that, when only one servicer is used, WD will require more maintenance cost than WoD because the utilization ratio of the servicer for WD is 70% greater than for WoD.

The second study showed the impact of the different OOS design variables on the responsiveness and cost-effectiveness of the WD infrastructure with propellant optimal and propellant-time-traded trajectories. A WD in-

frastructure is characterized by a service dispatch strategy to distribute the EUs to the different servicers, a fleet of servicers, and a depot of EUs. Two types of service dispatch strategies were considered: the "Average Working State Maximization" strategy (AWSM) aiming to maximize the responsiveness and cost-effectiveness of the OOS infrastructure, and the "Servicer Rotation" strategy (SR) distributing the EUs among the servicers in the order of the failure times. The design variables of the fleet of servicers considered in this research are the orbit of the orbital depot, i.e. the parking orbit of the servicers, the number of servicers, and the EU and propellant capacities of the servicers. The design variable of the orbital depot is its capacity for each EU type.

It is first demonstrated that the "Average Working State Maximization" (AWSM) dispatch strategy is by far more responsive and cost-effective than the "Servicer Rotation" (SR) dispatch strategy. Indeed, in some cases, the responsiveness due to AWSM increases by 178% with respect to SR, while the cost only increases by 22%. Consequently, OOS designers should design the service dispatch strategy in such a way that it maximizes a particular responsiveness metric and minimizes a particular cost function.

Second, it is shown that the fleet of servicers is responsible for giving the OOS infrastructure the potential to provide frequent services. The orbital depot cannot provide services by itself but its presence is necessary to allow the OOS infrastructure to exploit its full servicing potential provided by its fleet of servicers. A fleet of servicers providing frequent services is useless when the depot becomes empty before the next replenishment from Earth. More particularly, the choice of the orbit of the depot has a significant impact on the responsiveness of the OOS infrastructure. There are designs of the OOS infrastructure for which switching the altitude of the depot from an altitude of 35000 km to GEO multiplies the responsiveness by a factor of 6. The inevitable downside is that the propellant expenditure also increases due to a higher service frequency. The number of servicers also has a significant impact on the responsiveness as long as the orbit of the depot does not involve long waiting times before performing the impulse required to send the servicers on a Hohmann transfer. For instance, when only propellant optimal trajectories are considered, switching from 1 servicer to 3 servicers with a depot at 30000 km multiplies the responsiveness by a factor of 4.35. However, increasing further the number of servicers does not necessarily increase the responsiveness of the OOS infrastructure. Indeed, switching from 3 servicers to 5 servicers only multiplies the responsiveness by a factor of 1.15 for the most responsive designs. The EU capacity of the servicer has a somewhat desired impact on the responsiveness and cost-effectiveness when the AWSM dispatch strategy is used. The more EUs the servicers carry, the better the responsiveness is at a lower cost. This is because servicers carrying more EUs draw more benefits from AWSM. Finally, a larger propellant tank of the servicers allows them to provide services in more expensive scenarios, i.e. with larger EU capacities and a depot in GEO, and perform more flexible trade-offs between travel time

and propellant expenditure when propellant-time-traded trajectories are considered. This consequently increases both responsiveness and cost.

Finally, it is shown that performing sensitivity analysis from an initial, intuitively responsive design is not effective to find more responsive and cost-effective designs. It is necessary to search a wide area of the trade space by varying several design variables at a time to find more efficient designs.

The research presented in this thesis dealt with the assessment of the responsiveness and cost-effectiveness of two different concepts of permanent OOS infrastructure providing EU replacement services to modularized satellites. However, in the early days of OOS, the traditionally-designed satellites will need more basic services such as refueling, subsystem deployment or orbital maneuvers. Similar analysis as those presented in this thesis could be performed by modeling and simulating such satellites and services to inform decision making on the design of early OOS infrastructures. Future studies could include, among other ideas, a complete modeling of the supply chain of the servicers' propellant from Earth to the orbital depot, the addition of other depots, the modeling of on-orbit additive manufacturing to increase responsiveness, and the uncertainty in the servicers' reliability.

Appendix A

Satellite Subsystems Failure Data

Table A.1: Failure rates of each type of EU

Subsystems	*Average time before fatal failure* [years]	*Average time before major failure* [years]	*Average time before minor failure* [months]
Gyro/Sensor/Reaction wheel	11.18	5.97	3.88
Thruster/Propellant	1	2.63	1.59
Beam/Antenna Operation/Deployment	-	0.15	2.42
Mechanisms/Structures/thermal	-	0.15	1.08
Payload instrument/Amplifier/On-board data/Computer/transponder	-	2.89	1.56
Battery/Cells	-	0.99	-
Electrical distribution	-	4.03	0.38
Solar array deployment[1]	-	0.1	-
Solar array operating	5.83	2.51	0.68
Telemetry, tracking and command	-	3.18	3.3

[1] Solar array deployment is not modeled in this research.

References

[1] JAXA, "Manipulator Flight Demonstration," *Manipulator Flight Demonstration*, Available: http://iss.jaxa.jp/shuttle/flight/mfd/index_e.html. [Accessed 7/14/2016].

[2] NASA, "Robotic Refueling Mission," *Robotic Refueling Mission*, Available: https://sspd.gsfc.nasa.gov/robotic_refueling_mission.html. [Accessed 7/14/2016].

[3] DARPA, "Phoenix," *DARPA RSS*, Available: http://www.darpa.mil/program/phoenix. [Accessed 7/14/2016].

[4] "Space Launch Report," *Space Launch Report*, Available: http://www.spacelaunchreport.com/. [Accessed 7/14/2016].

[5] R. A. Haga, J. H. Saleh, "Epidemiology of satellite anomalies and failures: a subsystem-centric approach," *2011 Aerospace Conference*, 2011.

[6] A. Ellery, J. Kreisel, B. Sommer, "The case for robotic on-orbit servicing of spacecraft: spacecraft reliability is a myth," *Acta Astronautica*, vol. 63, 2008, pp. 632-648.

[7] R. C. Larsen, A. R. Odoni, "Urban Operations Research by Larson/Odoni," *Urban Operations Research by Larson/Odoni*, Available: http://web.mit.edu/urban_or_book/www/book/index.html. [Accessed 7/14/2016].

[8] Satellite Industry Association, "State of the Satellite Industry Report," Available: http://www.sia.org/wp-content/uploads/2015/06/Mktg15-SSIR-2015-FINAL-Compressed.pdf. [Accessed 4/13/2017].

[9] A. W. Verstraete, N. M. St.Louis, J. Hudson, "GEO Robotic Servicer Trajectory Optimization," AIAA 2016-5242, *AIAA SPACE 2016 Conferences and Exposition*, Long Beach, CA, 13-16 September 2016.

[10] J. H. Saleh, "Weaving Time into System Architecture: New Perspectives on Flexibility, Spacecraft Design Lifetime, and On-Orbit Servicing," Ph.D. Dissertation, Aeronautics and Astronautics Dept., MIT, Cambridge, MA, 2004.

[11] A. R. Graham, J. Kingston, "Assessment of the Commercial Viability of Selected Options for On-Orbit Servicing (OOS)," *Acta Astronautica*, vol. 117, 2015, pp. 38-48.

[12] B. R. Sullivan, D. L. Akin, "A survey of Serviceable Spacecraft Failures," AIAA 2001-4540.

[13] D. E. Hastings, B. L. Putbrese, P. A. La Tour, "When will On-Orbit Servicing be part of the space enterprise.", *Acta Astronautica*, vol. 127, 2016, pp. 655-666.

[14] Orbital ATK, "Mission Extension Vehicle," Available: https://www.orbitalatk.com/space-systems/human-space-advanced-systems/mission-extension-services/default.aspx. [Accessed 7/14/2016].

[15] MDA, "Robotics and On-Orbit Servicing," Available: http://mdacorporation.com/isg/robotics-automation/space-based-robotics-solutions/robotics-and-on-orbit-servicing. [Accessed 7/14/2016].

[16] D. Barnhart, L. Hill, M. Turnbull, P. Will, "Changing Satellite Morphology through Cellularization," AIAA 2012-5262, *AIAA SPACE 2012 Conferences and Exposition*, Pasadena, CA, 11-13 September 2012.

[17] D. Barnhart, L. Hill, E. Fowler, R. Hunter, L. Hoag, B. Sullivan, M. Turnbull, P. Will, "A Market for Satellite Cellularization: A First Look at the Implementation and Potential Impact of Satlets," AIAA 2013-5484, *AIAA SPACE 2013 Conferences and Exposition*, San Diego, CA, 10-12 September 2013.

[18] A. A. Kerzhner, M. O. Khan, M. D. Ingham, J. Ramirez, J. Hollman, J. de Luis, S. Arestie, D. Sternberg, "Architecting Cellularized Space Systems using Model-based Design Exploration," AIAA 2013-5371, *AIAA SPACE 2013 Conferences and Exposition*, San Diego, CA, 10-12 September 2013.

[19] T. Jaeger, W. Mirczak, "Satlets - The Building Blocks of Future Satellites - And Which Mold Do You Use," AIAA 2013-5485, *AIAA SPACE 2013 Conferences and Exposition*, San Diego, CA, 10-12 September 2013.

[20] W. Yao, X. Chen, Y. Huang, M. von Tooren, "On-orbit servicing system assessment and optimization methods based on lifecycle simulation under mixed aleatory and epistemic uncertainties," *Acta Astronautica*, vol. 87, 2013, pp. 107-126.

[21] S. Zhao, P. Gurfil, J. Zhang "Optimal Servicing of Geostationary Satellites Considering Earth's Triaxiality and Lunisolar Effects," *Journal of Guidance, Control, and Dynamics*, vol. 39, No. 10, October 2016.

[22] G. A. P. Horsham, G. R. Schmidt, J. H. Gilland, "Establishing a Robotic, LEO-to-GEO, Satellite Servicing Infrastructure as an Economic Foundation for Exploration," AIAA 2010-8897, *AIAA SPACE 2010 Conferences and Exposition*, Anaheim, CA, 30 August - 2 September 2010.

[23] INFORMS, "Operations Research & Analytics," Available: `https://www.informs.org/Explore/Operations-Research-Analytics`. [Accessed 4/14/2017].

[24] R. C. Larsen, A. R. Odoni, "Urban Operations Research by Larsen/Odoni," Available: `http://web.mit.edu/urban_or_book/www/book/index.html`. [Accessed 4/14/2017].

[25] J. Sztrik, "Basic Queueing Theory," Faculty of Informatics, University of Debrecen, Hungary, Available: `http://irh.inf.unideb.hu/~jsztrik/education/16/SOR_Main_Angol.pdf`. [Accessed 4/14/2017].

[26] "Space Launch Report," Available: `http://www.spacelaunchreport.com/`. [Accessed 7/15/2016].

[27] "Falcon 9 Users Guide," Available: `http://www.spacex.com/sites/spacex/files/falcon_9_users_guide_rev_2.0.pdf`. [Accessed 7/15/2016].