Uncertainty-Based Design Optimization and Decision Options for Responsive Maneuvering of Reconfigurable Satellite Constellations

by

Charlotte E. Lowey
B.Eng., Queen Mary University of London, 2015
S.M., Massachusetts Institute of Technology, 2017

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
May 2022

Author .....
Department of Aeronautics and Astronautics
May 17, 2022

Certified by .....
David W. Miller, Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by .....
Paulo Lozano, Professor of Aeronautics and Astronautics
Thesis Committee Member

Certified by .....
Malcolm Macdonald, Professor at University of Strathclyde
Thesis Committee Member

Certified by .....
Dr. Robert Legge
Thesis Committee Member

Accepted by .....
Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Uncertainty-Based Design Optimization and Decision Options for Responsive Maneuvering of Reconfigurable Satellite Constellations

by

Charlotte E. Lowey

Submitted to the Department of Aeronautics and Astronautics
on May 17, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

There are many time-sensitive mission applications for persistent satellite coverage, including dynamic and unpredictable events such as natural disasters, oil spills, extreme weather events, or geopolitical conflicts, which may progress rapidly and require frequently-updated information to co-ordinate the ground response. Reconfigurable satellite constellations can provide on-demand regional coverage by maneuvering orbits to focus passes over the area of interest. In contrast, traditional satellite constellations cannot maneuver to pass over specific ground locations, meaning that achieving persistent coverage spanning all possible locations of interest globally results in a requirement for thousands of satellites. This would present prohibitive costs for many applications, as well as contributing to worsening issues of space traffic management and congestion in Low Earth Orbit (LEO).

Incorporating reconfigurability into constellation design allows for responsive maneuvering of satellites into repeating ground tracks (RGTs) over a location of interest, simultaneously reducing the required constellation size by improving the utilization of individual satellites and providing flexibility in the achievable ground coverage. Past work on reconfigurable constellations (ReCon) demonstrated average cost savings of 20-70% compared to iso-performance static constellations, although the complexity of the solution space for the design optimization process limited the maximum size of constellations that could be evaluated.

In this thesis, a probabilistic performance metric is developed to compare constellation designs, adopting principles of reliability-based design optimization to quantify the confidence level that reconfigurable designs will outperform iso-cost static alternatives and by what margin of performance. The results show that 74.2% of reconfigurable designs outperform iso-cost static designs with a confidence level of 90% or higher, and with a margin of at least 10% improvement in the level of performance achieved. Computational intensity of the model presents the major constraint upon the size and complexity of simulation cases that may be modelled, so variance reduc-

tion techniques are applied to lower the standard error of mean performance in the output, allowing for a reduction in optimization size and runtime while maintaining the same level of error in the predicted results. Decision options for the operational phase of a reconfigurable constellation are presented and assessed to characterize how satellite operators must weigh mission priorities to evaluate trade-offs between propellant conservation and improved coverage of high-value targets.

Thesis Supervisor: David W. Miller
Title: Professor of Aeronautics and Astronautics

Acknowledgments

This research has been supported by the Aerospace Corporation, the MIT Department of Aeronautics and Astronautics, and the MIT Office of Graduate Education, through a combination of research assistantships and fellowships.

I would like to begin with particular thanks to my committee members, Dr. Robert Legge, Prof. Paulo Lozano, Prof. Malcolm Macdonald, and especially my thesis advisor Prof. David Miller. I am extremely grateful to my thesis readers, Prof. Olivier de Weck and Prof. Richard Linares. My sincerest gratitude also goes to all the members of the Aerospace-MIT ReCon project team, especially Kyle Hanifen at the Aerospace Corporation and Alex Straub.

Thank you to all in the SSL who have made my time in the lab so great. Tonio, I look forward to your next marathon fueled by deep dish pizza. Kat, Kit and Pratik, thanks for all the board games. Keenan, Hailee, Rosemary, Alex, and everyone else past and present who I have been lucky enough to overlap with in the lab. Thanks to Catherine and Oliver for several years of companionable office-sharing in 37-307.

Many campus resources have helped me out along the way; I am particularly grateful to Dr Melissa Mackel, Kathleen Monagle at DAS, Jason McKnight, and Suraiya Baluch at GradSupport. I would also like to thank Marilyn Good for the endless administrative support that holds the SSL together; Anthony Zolnik for huge amounts of work around the department; Jean Sofronas for a lot of scheduling help; Joyce Light for everything she does to keep us sane in AeroAstro; Pam Fradkin for being the friendliest conversationalist in 37; and Beata Shuster for ushering this thesis to the end. The MIT Writing and Communication Center's 'Writing Together Online' sessions were invaluable in completing this work, and I would especially like to thank Juliana Cherston and Shawn Hu for their work as facilitators. Lauren Milechin from the MIT Supercloud support team was endlessly helpful, and I would like to acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC resources that contributed to the research results reported within this thesis.

Beth Marois has earned her own paragraph in these acknowledgments: her calm expertise, support and knowledge throughout the last seven years have really done more for me and for this thesis than I can put into words here. Thank you for your guidance and advocacy, and for making the department feel like home.

Deep appreciation must go out to several years of GA^3 committees, for providing such a strong sense of community, so many friendships, and so much work that went into making the department a better place to be: special thanks to Ben (my pset buddy from the start), Aaron (but can you model snow?), Jacob (I owe you one thank), Prash (the final forum group members!), Daniel (it's time for a cutback week), Jess, Amelia, Regina, Cadence, Christine, Cory, and Hugh and Parker for memorably introducing me to GA^3 and the department in the first place. Thank you to all the GWAE committees over my time here, with particular shoutouts to Elizabeth, Rosemary and Sydney for all your hard work. Thank you to all the students who founded AeroAstro dREFS and those who keep it going.

There are an overwhelming number of other people I must thank for their support, friendship, general advice, or other help over the course of the PhD: Anne, Kirsty, Eva; Tyler, Mark, Erik and Ryan for saving the world with me; Mike and the rest of the Muddy staff for the most relaxing place (and best workplace) on campus. Thank you to all my friends from TMIRCE:BOS and Runfellow for the last four years of running camaraderie and great memories; I hope there are many more ahead.

I cannot express enough gratitude for the friendship of Mary Tellers Strawser since my arrival at MIT; I feel so lucky to have met my best friend within my first week here. It seems almost inevitable that we ended up defending within a week of one another, and I am excited to see what the amazing Dr Strawser does next. Harriett Feenstra has been there for me for the last 16 years through many ridiculous scrapes, even across the Atlantic, and I am so grateful for her presence in my life. I have known Ross Fowkes almost as long, and am indebted for our many years of board games, Tuesday Tilt, fancy coffees, mutual judgement, and assorted adventures.

It is impossible to sign off without thanking Pax and my parents for all of their support, listening and advice, and belief that I could do whatever I wanted. Finally,

thank you to John, Bo and Rosie for your love, endless affection and enthusiasm, laughter, and a lot of listening to me talking about satellites.

Contents

List of Tables

List of Figures

Nomenclature

Constants

$\dot{L}$	Earth's rotation rate, 360°/sidereal day
μ	Earth's gravitational constant, 398600.44 km^3/s^2
D^*	Length of a sidereal day, 86164.1 seconds
J_2	Earth's oblateness factor, 0.0010826269
R_E	Earth's equatorial radius, 6378.1 km

Variables

$\bar{P}$	Mean performance
$\bar{x}$	Sample mean
ΔP	Change in performance
ΔV	Change in velocity [m/s]
ΔV_T	Total change in velocity [m/s]
ΔV_{recon}	Change in velocity for reconfiguration [m/s]
$\dot{\Omega}$	Rate of change of the right ascension of the ascending node [deg/sidereal day]
$\dot{\omega}$	Rate of change of the argument of perigee [deg/sidereal day]

$\dot{M}$	Rate of change of the mean anomaly [deg/sidereal day]
ϵ	Satellite elevation angle [deg]
ϵ_{min}	Minimum elevation angle [deg]
η	Nadir angle [deg]
λ	Earth central angle [deg]
λ_{max}	Maximum Earth central angle [deg]
Ω	Right ascension of the ascending node, see also RAAN [deg]
ρ	Angular radius of the Earth [deg]
σ	Standard deviation
σ_x	Sample standard deviation
$\sigma_{\bar{x}}$	Standard error of the mean
a	Semi-major axis [km]
$c_4(n)$	Bias correction factor
D	Slant range [km]
D_{max}	Maximum slant range [km]
e	Eccentricity
H	Orbital altitude [km]
h	Orbital altitude [km]
H_0	Estimated RGT orbital altitude [km]
i	Orbital inclination [deg]
I_{sp}	Specific impulse [s]

j	Number of orbits
k	Number of days
n	Mean angular motion [deg/sidereal day]
n	Number of orbits over a specified number of days in a repeating ground track orbit, presented as $n/1$ or $n/2$ RGT options
n	Number of samples
NN	Number of nodes
$NPPN$	Number of processes per node
$NTPP$	Number of threads per process
P_{ReCon}	Reconfigurable design performance
P_{static}	Static design performance
q	Quantile
w_{swath}	Satellite coverage swath width [km]
x_{diff}	Desired performance margin

Acronyms

CI	Confidence interval
CPU	Central processing unit
EP	Electric propulsion
GA	Genetic algorithm
GOM	Global observation mode
GPU	Graphics processing unit

GSD	Ground sample distance
HPC	High performance computing
KDE	Kernel density estimation
LEO	Low Earth orbit
LLSC	Lincoln Laboratory Supercomputing Center
MOEA	Multi-objective evolutionary algorithm
NICM	NASA Instrument Cost Model
NRE	Non-recurring engineering
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PDF	Probability density function
PSMF	Propulsion system mass fraction
RAAN	Right ascension of the ascending node, see also Ω
ReCon	Reconfigurable constellations, or a reference to the ReCon codebase created by Robert Legge[8]
RGT	Repeating ground track
ROM	Regional observation mode
SSCM	Small Satellite Cost Model
SSP	Sub-satellite point
UMDO	Uncertainty-based multidisciplinary design optimization
UQ	Uncertainty quantification
USCM8	Unmanned Spacecraft Cost Model, version 8
VoR	Value of Reconfigurability

Chapter 1

Introduction

1.1 Motivation

Persistent satellite coverage has a broad range of mission applications, from Earth imaging, to weather monitoring, to global communications and data transmission. Time-sensitive data from orbit may be required to respond to rapidly unfolding events on the ground, such as extreme weather events, natural disasters, or conflicts. Where these events may be changing rapidly over time, persistent coverage is highly desirable for supplying up-to-date information to co-ordinate the necessary ground response, while obtaining only intermittent imagery may be insufficient to reach full understanding of the dynamic details of the situation. Persistence is defined for these applications in terms of how closely the achieved coverage matches the desired revisit cadence needed to track events as they develop over a prolonged period of interest.

For some scenarios, a rapid response to gather data via ground transportation may be possible, but in other cases the local conditions may rule this out. Aircraft and airborne systems may also be able to supply air-based support across a regional area, but cannot respond at a global scale and may encounter access issues over certain regions. In both cases, if vehicles must be brought to the relevant region from elsewhere, the timescale involved may take longer than utilising satellites which are already in orbit.

Static constellations extensive enough to provide a high level of persistent coverage over any possible ground location of interest are associated with costs that would be prohibitive for many applications. This is primarily due to inefficient utilization of the individual satellites, especially when measured against the performance of reconfigurable systems; many more satellites must be launched into static orbits to achieve a comparable level of coverage. As static constellations cannot flexibly respond to uncertainty in future operating contexts, attempting to optimize the design variables for static architectures over a range of target decks will result in the selection of more conservative (and therefore costly) designs, in order to build in the necessary margins to achieve an acceptable level of coverage when target regions of interest are not known in advance.

Although the issue of space debris has been known for decades, with Kessler's work in the 1970s describing the potential for an exponential cascade in collisions creating a debris belt,[1] there have been increasing concerns raised about the problem as space becomes more congested.[2] Studies have been carried out on the potential impact of the mega-constellations planned by various companies including Boeing (1396 satellites, with a later 1560 to be added), SpaceX's Starlink (over 2100 satellites already launched of a planned 4425 V-band satellites, with a further 7518 intended within Ka- and Ku-bands) and OneWeb (428 satellites already launched of a planned constellation of 648 satellites, later intended to rise to 882),[3] concluding that greater regulatory oversight may be needed to prevent a large increase in future debris levels,[4] with simulations suggesting that the first generation of such a large-scale constellation would have a 35% probability of catastrophic fragmentation, and that hundreds or even thousands of collision avoidance maneuvers would be required over the mission lifetime depending on the selected orbital altitude.[5] Further investigation of whether the proposed large-scale communications constellations could achieve similar performance by launching significantly fewer reconfigurable satellites would make for an interesting and relevant optimization scenario to investigate using

the upgraded reconfigurable constellation codebase, if runtime can be sufficiently reduced to enable analysis of such large systems.

Although many satellite applications involve delivering coverage of areas that are well-defined in advance, there are also numerous applications which require flexibility due to a lack of advance knowledge of the desired coverage locations. Such types of missions include weather monitoring, storm tracking, wildfire tracking, oil spill containment, or regionally persistent communications capacity over seasonal or emerging events (such as other forms of natural disaster). By the very nature of these types of event, their locations will not be known ahead of time, and so a constellation operator wishing to attain persistent coverage from orbit must design flexibility into their approach, aiming to "improve results by recognizing that the future is inevitably uncertain".[6] A crucial distinction must be drawn here in order to avoid the 'flaw of averages': that "plans based on average assumptions are wrong, on average".[7] It may be possible to determine a probable distribution for some applications even when it is impossible to know exactly where the next event will occur; for example, hurricanes tend to fall within a similar spatial distribution each year, and knowledge of this distribution may be used to inform the design choices made for a constellation that seeks to acquire persistent hurricane coverage. However, designing a satellite constellation to achieve optimal coverage of an imagined perfectly average hurricane is likely to be "wrong, on average"; it must be appreciated in making such design decisions that the mean of the distribution is not the same thing as the distribution of the mean.

Where the areas of interest are not known ahead of time, the constellation must be designed to perform well under uncertain conditions, despite the fact that the initially-selected satellite orbits are unlikely to prove optimal in achieving coverage of a randomized target location. Incorporating reconfigurability into the design allows for flexibility in the achievable ground coverage due to the ability to modify satellite orbits throughout the mission lifetime. Using a reconfigurable constellation therefore allows for improved satellite utilization, as all or part of the constellation can be ma-

noeuvred into appropriate repeating ground track (RGT) orbits to achieve persistent coverage over the desired target region.

Past work on reconfigurable satellite constellations has demonstrated that they can provide greater responsiveness than traditional static constellations due to their ability to change orbits over time in response to ground events and changes in desired coverage. Robert Legge determined the Value of Reconfigurability (VoR) for a reconfigurable constellation to be around 20 to 70% cost saving, when compared to a static architecture offering the same level of performance. Legge also demonstrated that both reconfigurable and static constellations performed better than other architectures that might be suggested as potential solutions for these applications, such as sun-synchronous or rapidly-launched satellites.[8]

However, Legge treated predicted constellation performance as a point metric, despite the fact that by the very nature of operating a reconfigurable constellation in unpredictable circumstances, any optimized design will give a range of utility values under a variety of scenarios. No two mission scenarios for the constellation lifetime will provide exactly the same performance value, with the selection of different subsets of satellites being maneuvered at different times and to different drift orbits in order to achieve the desired coverage of ground locations. Recognizing the statistical nature of this performance as a range and not a point metric allows for a more informative characterization of the utility supplied by the constellation under a range of mission scenarios. This allows for the establishment of confidence metrics that not only predict the percentage of scenarios in which reconfigurable constellations will outperform static designs, but also quantify by how much.

Due to the computational intensity of the models required to optimize a constellation design for an uncertain operating context, simplifications must be made in order to arrive at a computationally tractable problem. However, with the application of previously unused uncertainty quantification techniques to the parameter sampling

within the simulation model, improvements in efficiency may be made to decrease the runtime of the design optimization code. Using techniques such as a quasi-Monte Carlo method[9] or Latin Hypercube sampling[10] may allow for the statistical confidence of the input distribution to be maintained while reducing the number of inputs needed, thereby reducing the simulation runtime. Another addition that may improve the efficiency of the simulation is importance sampling, a technique which recognizes that some input variables in a model have a greater impact on the output parameters than others, and weights sampling resources accordingly to reduce the estimator variance.

Rather than focusing exclusively on optimizing the design of reconfigurable constellations, considering the options and variability introduced to the operational phase of the system by the reconfiguration decision is another area of interest for this research. Quantifying the trade-off between fuel use and time taken to reconfigure is recognized in the literature as a difficult problem for multiple reasons, including the dependence of the solution on the orbit of the particular satellite under consideration and the desired final orbit, which may vary widely between operating scenarios.[11] However, delaying reconfiguration maneuvers may save fuel and even lead to the selection of different drift orbits that affect the total time taken to reconfigure the constellation over a desired imaging location. This motivates the necessity of developing decision tools which may be used by satellite operators, supplying details of potential trade-off between maneuvers which may be balanced according to the particular needs of the mission at hand.

Another topic of interest in the realm of operator decisions is that of crossover latitudes: the ground track latitudes over which both ascending and descending passes occur. These crossovers can be defined in two ways: the ground track of an individual satellite passing over its own previous tracks, and the ground track of an individual satellite passing over the ground track of the satellite in the adjacent plane within the constellation. Due to the fact that a notable performance increase is shown over

these latitudes, there is a strong motivation to determine operator ability to shift and redefine them after launch in order to improve performance over areas of interest as they arise. Depending on the mission objective under investigation, there are also a wide number of other design and performance criteria that may be considered to be more or less important by the operator, with the effect of propulsion constraints on constellation design and performance being one such factor examined here.

1.2 Research objectives

The previous section highlights some of the limitations in the existing simulation model and past research conducted on optimizing the designs of reconfigurable constellations, leading to the selection of research objectives for this work where potential improvements were identified.

Legge developed a custom performance metric to place a value on the utility provided by both reconfigurable and static constellations under a range of operating scenarios. He compared simulation results for optimized designs on an iso-performance basis, showing cost savings of 20–70% for a reconfigurable constellation to provide the same utility as a static constellation design. However, these performance values are primarily presented as point-value predictions, rather than characterizing the statistical range of values that result from simulating these designs under a selection of varied operating scenarios.

The first objective of this work is to establish the statistical parameters of this performance comparison, which will allow for not only calculating the level of confidence in how often reconfigurable designs outperform static designs across a range of scenarios, but also for the quantification of how much these designs outperform by. Rather than simply showing that reconfigurable designs are 'better' over some percentage of scenarios, a metric can be established to show confidence that reconfig-

urable architectures are specifically at least 10% better, or 20% better, or any specific desired value by which the performance excess may be quantified.

As computational intensity is the limiting factor in the size and complexity of simulation cases that can be modelled, another objective of this research is improving the efficiency of the model in order to retain the level of confidence in the simulation results while decreasing the runtime required. A hypothesis for this research is that applying uncertainty quantification techniques, stratified sampling and importance sampling to the model will allow for such an objective to be achieved.

Finally, past research has focused heavily on optimizing the design of reconfigurable constellations, but less so on the operational phase of such a constellation after it is launched. Beyond characterizing maneuver selection, little work has been carried out on satellite operator choices that may have significant impacts on the performance of a reconfigurable constellation during its operational lifetime. A final objective of this work is to investigate the impact of some of these factors on performance and present methods of incorporating these into decision tools for satellite operators. Specific decisions considered include the effect of reconfiguration delays on the maneuvers that are then selected as optimal, changes to the ΔV budget for reconfiguration (including broadening the available choice of RGTs for gathering imagery), and the ability to adjust crossover latitudes (at which both ascending and descending passes occur) during the operational phase of the mission.

Research objectives, in summary:

1. Develop a statistical performance metric and compare it to previous custom performance metrics to evaluate the proportion of cases where reconfigurable designs outperform static constellations.

2. Use the statistical performance metric to establish the level of confidence that reconfigurable designs outperform comparable static designs and quantify by

how much.

3. Improve simulation efficiency with the application of uncertainty quantification, stratified sampling and importance sampling techniques to the model.

4. Identify decisions options during the operational phase and characterize how much these satellite operator decisions can significantly affect constellation performance.

1.3 Thesis roadmap

The rest of this thesis is organized in the following order. Chapter 2 reviews the existing literature on satellite constellations, focusing in particular on the areas of constellation design and reconfiguration, as well as the topic of multi-objective constellation optimization. Other work examined in the same chapter includes uncertainty analysis and designing for flexibility, especially where it has been applied to space systems, and the analysis of performance metrics as statistical ranges. Chapter 3 reviews the methodology of this work, providing an overview of the existing simulation model and custom performance metric inherited from Robert Legge and modified by the author, as well as details of the scenario modelling inputs, and resources used on the MIT Supercloud. Chapter 4 discusses statistical performance metrics and their application to confidence calculations to compare the utility of reconfigurable constellations against more traditional static designs. Chapter 5 applies uncertainty quantification techniques to the reconfigurable design optimization for the improvement of input sampling efficiency. Chapter 6 introduces the topic of operational decision options, considering how a reconfigurable constellation would actually be operated once designed and launched, and presenting the effects of several types of operator decisions on constellation performance. Finally, Chapter 7 discusses the conclusions of this work, and presents suggestions for future work in this area of research.

Chapter 2

Literature Review

This chapter presents a review of relevant previous work in fields related to the current research. Literature is reviewed here relating to the areas of constellation design, reconfiguration and optimization; design for flexibility/real options; multi-objective optimization; uncertainty quantification; and probabilistic approaches to performance metrics. The following sections summarize the state-of-the-art in pertinent research areas, identify research gaps, and highlight work that contributed to the research covered within this thesis.

2.1 Satellite constellations

2.1.1 Constellation design

Satellites are typically launched into static orbits, where the only maneuvering that occurs is that required for initial deployment into the desired orbit, and for station-keeping and drag makeup. While the concept of operations for a mission is being designed, it must be decided whether the project's objectives can be achieved by a single satellite (with the benefit of lower cost) versus a constellation (bringing benefits of coverage and reliability).

Historically, many applications have used constellations of satellites, which act in

concert to achieve a shared objective, dating back as far as the Transit constellation for the US Navy's NAVSAT (Navy Navigation Satellite System), which began launching in 1959 and provided a continuous satellite navigation system for Navy ships and submarines from 1964.[12] Other constellations followed, such as the Global Positioning System (GPS) and Global Navigation Satellite System (GLONASS), with a significant global impact for military and later civilian navigation. Satellite communications systems were initially located in higher geostationary orbits for the most part, due to advantages in the number of satellites required to achieve global coverage, but transmission time delays and power requirements led to greater interest in Low Earth orbit (LEO) constellations over time, especially for communications and data applications.[13]

There are many possible constellation geometries which may be appropriate for achieving a given mission objective. One of the best known is Walker Delta pattern constellations, consisting of a set of circular orbits which ensure global coverage with a minimum specified elevation angle,[14] with the geometry later expanded for larger patterns of satellites with up to seven-fold continuous satellite coverage of any ground point.[15] Walker introduced a 'pattern unit' ensuring minimum satellite separation between adjacent planes, allowing for the constellation geometry to be uniquely described using only a small number of parameters (total number of satellites, number of orbital planes, relative spacing between planes and orbital inclination).[16] Other authors recognize the inherent difficulty of arriving at an optimal constellation design due to the sheer quantity of parameter combinations, but criticize the narrowness of the restrictions adopted by researchers such as Walker, often offering alternative approaches such as evaluating parameter sensitivity in order to determine significant effects on potential architectures.[17]

A more recent constellation geometry of interest are the Flower constellations, which are described in the literature as similar to Walker constellations with some constraints removed to lead to greater efficiency for certain applications.[18] Satellites

in a Flower constellation are not limited to circular orbits as Walker patterns are, but will all show an identical repeating ground track (RGT).[19] In these constellations, satellites follow the same track with respect to a rotating reference frame, with orbits described by a set of five parameters determining synchronization and phasing.[20] Optimizing the constellation design for a particular application is recognized to be a complex task, but published research comparing the performance of a Flower constellation with a Walker pattern found that the Flower constellation provided better performance due to optimizing against a higher number of possible configurations, and also showed lower costs for deployment and station-keeping.[21] More recent work on Lattice Flower constellations expanded the design space further to include constellations that did not have identical repeating ground tracks, providing an expansion of potential designs for future optimization of constellations.[22]

Other researchers have adopted different approaches to constellation design depending on the application of interest, with examples including designing constellations based on the resulting ground tracks without constraining the uniformity of the orbital parameters,[23] selecting LEO designs for regional positioning applications that prioritize long local coverage durations and a minimized Geometric Dilution of Precision (to characterize positioning accuracy),[24] or introducing other application-based constraints such as achieving repeating sun-synchronous orbits using single or multi-plane constellations.[25] Paek developed several variants upon the sun-synchronous RGT orbits useful for remote sensing applications. These include sun-synchronous orbits with drifting ground tracks at a predefined speed; 'multi-sun-synchronous' orbits with RGTs that supply identical solar angles at multi-day intervals, relaxing the requirement upon sun-synchronous orbits to achieve these every day; and a combined multi-sun-synchronous orbit with drifting ground tracks.[26]

2.1.2 Constellation reconfiguration

Early work on constellation reconfiguration appears to originate in the idea of planning for a staged deployment, where an existing constellation can be expanded on orbit by adding capacity and reconfiguring the satellites on orbit to accommodate the newly launched additions to the constellation.[27, 28, 29] Chaize identified architectural paths that may be built up as additional capacity is added, calculating the value of flexibility compared to the cost of incorporating real options into the system design.[27] Scialom considered planning of the necessary launches and orbital transfers required to transition from an initial lower-capacity constellation to a new higher-capacity constellation, while simulating reconfiguration time and calculating cost and coverage during the orbital transfer phase.[28] Appel carried out a simple modelling of the same problem, transferring newly launched satellites into an existing constellation and establishing equal spacing.[30] Siddiqi also examined the problem of staging satellite deployment, determining the optimal initial constellation design for later reconfiguration into a higher-capacity system, while balancing initial costs against later reconfigurability and overall lifecycle costs,[31] with additional research carried out with de Weck and Scialom on developing a framework for orbital reconfiguration using the auction algorithm to assign satellites to vacant 'slots' in a new constellation while minimizing ΔV usage.[32] Davis focused on analyzing constellation designs using a new method of orbit propagation, addressing the need to optimize transfers of satellites from their initial orbit to slots in the final desired orbit, also using the auction algorithm to assign satellites once transfers are computed.[33]

Other research focuses on reconstitution of a constellation after the loss of one or more satellites, and which maneuvers may be undertaken to restore performance. Ferringer developed a framework for constellations which have lost capacity, in order to determine what trade-offs may be made to arrive at acceptable performance after a catastrophic event.[34] Additional work in this area involved the use of a multi-objective evolutionary algorithm to explore multiple loss scenarios where a future

constellation must be assembled from remaining assets; this led to conclusions regarding the need to avoid unnecessary constraints upon the design space due to the discovery of multiple non-intuitive architectures.[35] Ferringer's thesis work highlights that the future state of the reconfigured constellation is not assumed, but arrived at based on an optimization of the remaining assets and propellant using objectives of performance, cost and risk.[36]

Moving towards focus on responsive space-based architectures rather than only considering orbital transfers between an earlier and later constellation design, Paek introduced the concept of using reconfiguration to enable rapid coverage of targets of interest, as well as the ability to switch between supplying wider global coverage and more frequent regional coverage. The feasibility of such constellations is established, and optimal reconfigurable designs are identified for some example cases, as well as initial work being carried out on the ReCon framework with Robert Legge.[37]

Co also examined the idea of maneuvering satellites to manipulate the ground track, developing a methodology for overflights of desired target locations using either chemical or electric propulsion.[38] With similar interest in responsive orbits, Ingraham assessed the feasibility of maneuvering between circular and eccentric orbits to achieve high resolution coverage, while considering the necessary ΔV budget for such a mission.[39] Due to high costs for responsively launched satellites, as well as higher costs for chemical propulsion than for electric propulsion (EP), Co argues for the use of EP in reconfigurable systems, exploring optimal solutions to attain desired coverage,[40] and developing the necessary algorithm and tools to demonstrate the achievability of timely maneuvering to overfly desired targets using EP, while allowing for over five times as much maneuvering as a chemical system.[41]

McGrath described a fully analytical method for reconfiguration, making use of the Earth's J_2 effect to reduce the ΔV cost of maneuvering, at the cost of a longer reconfiguration duration but enabling the use of low-thrust propulsion.[42] Other work

describes a fully analytical solution for a low-thrust in-plane maneuver with the objective of reducing revisit time over a specific ground location, while acknowledging the complexity of the solution space and the fact that increasing ΔV usage does not automatically equate to a reduced revisit time.[43] McGrath published more recent research on using low-thrust EP for reconfiguration, demonstrating that reconfiguring a single satellite can more than double the available flyovers of a desired region.[44] Later publications show improved coverage of more than ten times compared to a static constellation using only 35% of the available propellant,[45] and developing a fully analytical solution that allows for solutions to be rapidly found for the maneuver optimization problem.[11]

Morgan and McGrath addressed the issue of mobile target tracking using a reconfigurable constellation, optimizing a design to track moving targets of interest such as oil spills, hurricanes, and other weather events. Trade-offs must be made between metrics such as reduced distances to targets during satellite passes, total access time to the mobile target, and total ΔV usage. An optimizer is applied to explore possible responsive maneuver options in a continuous design space, rather than using graph theoretical techniques to evaluate discrete possibilities, as is carried out in McGrath's other work.[46] Morgan also independently validated and extended McGrath's work, exploring trade-offs between ΔV usage and pass distances for low-thrust maneuver options, as well as outlining a proposed ReCon demonstrator mission.[47]

Straub explored additional trade-offs for reconfigurable constellations, examining the effect on performance and cost scores when electric propulsion is used for reconfiguration instead of chemical propulsion. Staged and responsive launch options are also considered to take advantage of changing launch costs and adapt to variation in constellation demand over time. Straub also expands image scheduling for ReCon to consider off-nadir imaging targets, for which slewing is required to expand the coverage footprint of each satellite.[48] Gentgen extended some of the propulsion focus of Straub's work, evaluating the performance and feasibility of using hybrid chemical-

electric propulsion systems for reconfigurable constellations, and concurrently optimizing chemical and electric systems to arrive at improved hybrid architectures.[49]

The current project builds upon past work by Robert Legge, who developed a framework for the concurrent design optimization of reconfigurable constellations under uncertain future operating conditions.[8] The simulation model concurrently optimizes individual satellite design, overall constellation pattern, and aspects of the operations design, and is described in more detail in Section 3.2.

Further publications of Paek's describe the development of an optimization tool for both individual satellite design and constellation geometry, as well as the use of a genetic algorithm to identify optimal solutions in the tradespace, relating these along potential deployment paths that could be used to expand a constellation over time and thereby relating back to earlier reconfigurability research.[50] The performance of simulated annealing and a genetic algorithm in concurrently optimizing the design is compared in recent research, as well as results similar to Legge's[8] showing that reconfigurable constellations outperform static constellations under uncertain operating conditions.[51]

2.1.3 Multi-objective constellation optimization

Multi-attribute tradespace exploration has been proposed as a crucial stage in effective space system design, in order to better resolve ambiguity and capture decision maker preferences.[52] La Tour investigated a merger of tradespace exploration with system dynamics modelling in order to simulate the effect of changing design variables and policy choices and incorporate learning effects, enabling the examination of performance against possible futures.[53] Uebelhart also proposed model-based analysis in order to take uncertainty into account in investigating design options, assessing both performance and robustness across multiple uncertain operating conditions while also taking into account critical uncertainty parameters.[54]

A common problem in constellation design is that the tradespace for distributed satellite systems or constellations is frequently too large to analyze and compare all possible candidate architectures. Jilla outlined a seven-step methodology using multi-disciplinary design optimization techniques to find Pareto-optimal architectures, providing an integrated assessment of performance, reliability and cost models.[55, 56] For simple mission types, Paek proposed an algorithm using a semi-analytical approach to determine an optimal design, which allows for trade-offs between performance and deployment cost to be examined, providing additional insight for a decision-maker in selecting a constellation design.[57]

The discontinuous, nondifferentiable or nonlinear characteristics of the metrics used to evaluate potential architectures can introduce difficulty in arriving at a constellation design, as traditional optimization methods may fail.[58] However, the use of genetic algorithms (GAs) or multi-objective evolutionary algorithms (MOEAs) has demonstrated the ability to generate optimal solutions under these circumstances, while balancing multiple objectives and saving on computational resources required.[58, 59] Using GAs with parallel computing resources allows for highly effective searching within the tradespace of constellation design and performs well in finding Pareto-optimal solutions.[59] A study using MOEAs to maximize successful search specifically examining a constellation design coverage problem also found that simple worker-leader arrangements often outperform more complex schemes using multiple populations.[60]

2.1.4 Research gaps

Although there is considerable literature examining reconfiguration of satellite constellations, much of this is focused on staging system deployment or reconstituting a useful constellation after loss of one or more satellites. Due to the complexity of optimizing large amounts of design variables, many constellation designs are selected from a simplified tradespace, or multiple assumptions are made in the course of architecture selection in order to reduce the complexity of the available options. This

results in a gap in the literature where certain complex mission types cannot be simulated by existing models, due to dynamic aspects of the targets of interest, such as moving or expanding locations.

There is some focus on the added utility gained by using reconfigurable systems, but in most cases maneuvering is assigned without attention to the trade-offs between parameters that may be of varying importance to the satellite operator for different applications. The opportunity costs of reconfiguration decisions are not presented to operators, although the relative prioritization of different elements could usefully be presented to decision makers when considering operational options for responsive maneuvering. Some maneuvering options may have significantly higher ΔV costs beyond those considered reasonable under normal circumstances, but present desirable performance gains under conditions featuring extremely high-value targets. There is also a research gap when it comes to the investigation of time elements related to the decision to reconfigure (or not reconfigure).

2.2 Uncertainty analysis/designing for flexibility

2.2.1 Flexibility and real options

Engineering systems are commonly designed in the context of uncertainty, as requirements, environments or technologies change over time,[61] but despite this, standard design methods often rely on static requirements which take an entirely deterministic view of the future operating conditions.[62, 63] Complex projects may need to consider sociological or technical changes over time as well as uncertainty over the course of the deployment and operation of the systems they are designing.[63] Designing changeability or flexibility into a system may be motivated by the admission of uncertainty in future operating conditions and the desire for system flexibility in such an environment. As the mission lifetime progresses, estimated qualities can be updated in response to more accurate information, and system response calibrated

accordingly.[64, 65]

In many contexts, uncertainty is perceived as risk, despite the fact that uncertainty may offer opportunities which can be leveraged, as well as downsides to be mitigated.[61, 62, 63, 66, 67] Tradespace exploration may aid in assessing design performance across a varied environment, as flexibility under uncertain conditions is hard to assess when considering only one or a few potential operating environments out of many future possibilities.[61, 66] Uncertainty may actually increase performance where system flexibility is able to take advantage of potential benefits offered by new opportunities, while also mitigating potential losses.[62, 63, 67, 68, 69]

Definitions of system changeability incorporate aspects of flexibility, adaptability, scalability and robustness, as well as the ability of a system to be modified. [61, 64, 65] Robust systems should possess adaptability to changes in the operating environment or mission, be scalable to accommodate new capability, and modifiable in order to take advantage of new technological developments.[64, 65, 70] The advantage of robustness is in the ability to sustain a system's value when inevitably, change occurs over time. This can be demonstrated by designs that retain high value for a variety of prospective future scenarios.[61, 64, 65] Flexibility in design may be defined simply as the ability to change under uncertainty,[63] or as a system quality that characterizes the feasibility of implementing multiple alternative behaviors under different conditions,[70] or as an externally initiated possibility for change in a system[61, 64, 65] such as the option presented by a reconfigurable constellation.

Real options analysis is one widely used way of assessing the value of potential changes to a system under some specified future uncertainty about its operating environment.[61, 67, 68, 71, 72] "Real options" are the possibility of taking some physical action for a system (without the obligation of doing so), which may be used to ease the transition when a system undergoes changes, thereby improving its flexibility.[61, 63, 68, 69] Reconfigurable constellations provide an example of an en-

gineering system where real options may be applied "in" the system, meaning that the option is internal to the physical design of the system considered, and analysis may be used to aid in determining whether exercising the option is valuable or not under some condition of future uncertainty.[69, 71, 72] However, real options analysis focuses more on ascribing a monetary value to flexibility, rather than specifically quantifying the extent of the available flexibility in a system.[66, 68]

Real options literature contains examples of the use of Monte Carlo or other large-scale simulations for evaluating system value under a variety of potential future circumstances,[67, 68, 72] in much the same method as adopted by Robert Legge in his work on the ReCon framework.[8] De Neufville states that estimating the value of flexibility requires estimating the risks to an inflexible version of the project, calculating the additional value provided by including options to the system, and identifying strategies for exploiting the new flexibility of the system, and discusses using simulations on the scale of thousands of run to assess the results of deploying real options within a system.[68]

2.2.2 Uncertainty analysis for space systems

Literature on flexible design suggests that notable cost improvements may be made across a range of engineering projects, from a flexible concept of the Iridium constellation design that would have saved up to 20% in development costs,[73] to Robert Legge's findings of a 20 to 70% cost saving for reconfigurable satellite constellations compared to similarly performing static architectures,[8] to de Neufville's work indicating that design flexibility can improve performance by 10 to 30%.[6]

Uncertainty analysis has been applied to small models for space systems applications previously, such as in Hassan's work comparing static and flexible designs for a satellite fleet to quantify the value of flexibility, in this case by assuming the ability to provide a variety of communications capabilities to disparate potential markets dur-

ing the mission lifetime. A model was developed to simulate uncertainty in market demand and generate several simple constellation architectures that met the requirements of the project. Using real options analysis, Hassan found that making different changes in the architecture supplied the ability to alternately reduce the maximum possible financial loss, increase the maximum possible financial gain, or reduce the standard deviation in the assessed value, according to what might be prioritized by the operator.[74]

Uebelhart used uncertainty analysis and design of experiments techniques to determine critical design parameters within the tradespace (for optical structures designs), using a model-based analysis to assess designs under uncertain conditions. Uncertainty bounding techniques were used to correlate specific design variables with the ability to move along the Pareto front of non-dominated designs.[54]

Walton studied tradespace management for satellite system architectures by applying uncertainty analysis, developing a framework for analyzing the uncertainty of each candidate architecture. By simulating vectors of design variables and design constants, output measures were modelled for thousands of designs, including generating cost and performance metrics. Sources of uncertainty were categorized according to relevance to the model, including the difficulty of simulation and the impact on the design, so that output distributions could be generated for the designs over a variety of potential scenarios.[75] Walton's model appears to possess similarities to Robert Legge's simulation work, though at a simplified level in order to be applied to a range of applications. However, more detailed uncertainty quantification could be applied to the design parameters determined to have the most significant effects on the output distribution.

2.2.3 Research gaps

Although uncertainty analysis has previously been applied to smaller simulation models for comparing the performance of constellation designs, such models were heavily simplified for the purposes of reducing the computational intensity required and keeping the investigated design space tractable. Previous research reviewed did not use complex design optimization models for constellations while applying uncertainty quantification techniques in order to improve the sampling efficiency from realistic parameter distributions. These techniques could be combined with the flexible decision options presented by reconfigurable satellite constellations. Uebelhart's research on optical structures may supply useful examples of the application of uncertainty quantification and bounding techniques. Although the literature on real options analysis is mostly regarding how to generate ideas for where flexible options might be added within engineering systems and assessing them in monetary terms, its techniques could potentially be applied in providing metrics for evaluating the relative value of reconfiguration options for presentation to satellite operators.

2.3 Probabilistic metrics for system design optimization

Yao presented a comprehensive overview of uncertainty-based multidisciplinary design optimization (UMDO) for aerospace vehicles, reviewing existing approaches and challenges and including details on optimization under uncertainty as well as modelling, propagation, and analysis of uncertainty. This recognizes the sheer volume of potential uncertain variables for any system under consideration, including the environment, operating conditions, and even the vehicle system itself. These uncertainties may cause minor variations in system performance, severe deviations from predicted performance, or even mission failure. This review paper emphasizes how crucial it is to include consideration of uncertainty at all stages of system design.[76]

Traditional system design methods consolidate an assortment of potential uncertainties by assigning a single safety factor to the system.[77] However, imposing a large safety factor results in conservative designs with excessive redundancy, while imposing a small safety factor may result in a design for which the level of reliability is unsatisfactory. Advanced analytical approaches with greater accuracy are needed, enabling the consideration of uncertainty in a systematic manner during the design phase. Yao divides such approaches into two major categories: robust design optimization and reliability-based design optimization. Robust design optimization aims to increase system robustness, decreasing sensitivity to variation in inputs and improving performance stability under uncertainty. This approach primarily considers events occurring around the mean of the probability density function (PDF) of conditions, and the response observed to such fluctuations around the nominal status of the system. Reliability-based design optimization aims to improve system reliability and reduce the probability of failure under extreme conditions, assigning a required level of probability for the system to maintain a normal operating state even under critical conditions. This approach primarily considers events occurring in the tails of the PDF of conditions, and the response observed to such extreme events in the operating environment.[76]

An assortment of research focuses on reducing the variation in predicted performance for various applications, adopting the robust design optimization approach to UMDO. Sondecker focused on 'stochastic process decision methods', identifying uncertainty as the largest cause of cost and schedule overruns for complex engineering systems, and aiming to determine and measure the largest sources of uncertainty and strategically reduce them during system design.[78] Stout aimed to improve the validation of spacecraft thermal models using rigorous uncertainty quantification and propagation through models to determine the ultimate effect on outputs of interest. This is used to reduce the variance in the estimated performance with the goal of improved accuracy in assigning adequate system margin, reducing mass and cost by improving overly conservative design modelling.[79] Sankararaman developed a com-

putational framework for robust design optimization of engineering systems, characterizing and propagating uncertainty. The model focuses particularly on sensitivity analysis with the objective of reducing uncertainty in the predicted system performance, refining the effect of significant variables sequentially in order to decouple covariance within their impact upon model output.[80]

The reliability-based approach to design optimization was defined by Yao as "A methodology to optimize design which is reliable with small chance of failure under predefined acceptable level."[76] Such an approach is adopted in the work of Shaw, propagating input statistical distributions to assess sensitivity and capability of the resulting system as a probabilistic measure. Shaw states that "To be unambiguous and quantifiable, performance should represent the likelihood that the system can satisfy the functional requirements." An acceptable minimum standard of availability is specified for the system, and the optimization is carried out with the objective of maximizing the probability of the system being in an operational state according to the specified standard.[81, 82]

Input probability distributions have been applied in evaluations of lasercom designs, such as Biswas's work assessing link performance for a deep-space optical communications downlink from Mars to Earth while incorporating pointing losses, receiver losses, space loss, background-noise estimation and atmospheric attenuation. This project estimated the best, worst and nominal data rates for the mission, while posing the question of what fraction of the time such data rates might be achievable, leaving the development of a fully probabilistic performance metric for future work.[83]

Clements extended research on the use of probabilistic metrics to estimate the distribution of performance for additional lasercom missions, predicting the volume of data delivered during a specified time interval to reframe the performance metric and arrive at a fully probabilistic metric. The probabilistic case with the objective of

minimizing the probability of failing to meet the data requirements was compared to a deterministic optimization case with the objective of maximizing average data volume per day. Optimization results are compared using the worst-case input conditions, full input distributions, and the probabilistic metric. The usefulness of a probabilistic design approach for complex missions is emphasized, while acknowledging the challenges of implementing sufficiency levels of accuracy in the input distributions and models. Clements added a statistical component to the systems engineering process for nanosatellite lasercom systems, optimizing designs by prioritizing a high probability of meeting mission requirements under uncertainty. The project found that using probabilistic metrics for design optimization resulted in designs with similar performance at lower cost and improved manufacturability.[84]

2.3.1 Research gaps

Both the robust design optimization and reliability-based design optimization approaches within uncertainty-based multidisciplinary design optimization have a history of application to other areas within aerospace research. Robust design optimization has been successfully demonstrated to lead to reduced variance in estimated performance, as well as reductions in mass and cost for aerospace systems when improved accuracy in modelling allows for the reduction of margins in overly conservative designs. Reliability-based design optimization has been demonstrated to result in designs for lasercom nanosatellite systems with similar performance to those produced by deterministic approaches to design optimization, while lowering costs and improving manufacturability.

Reconfigurable satellite constellations present a new application for robust design optimization and reliability-based design optimization, while being extremely well-suited to UMDO due to the objective of designing a constellation which can flexibly initiate responsive maneuvers following an unplanned event of interest. Robust design optimization may be used to improve the robustness of reconfigurable constellation designs, reducing the variation in performance achieved under a range of

unpredictable mission scenarios. Reliability-based design optimization may be used to evaluate reconfigurable design performance in probabilistic terms rather than the single performance scores given by Legge, acknowledging the statistical distribution of the range of performances that will be attained in widely-varying operating conditions. This would allow for the coverage achieved to be compared to a pre-specified minimum standard of persistence, or for evaluation of the margin by which reconfigurable designs outperform traditional static alternatives from the same region of the design space.

Chapter 3

Methodology

Past research on reconfigurability has demonstrated that the use of maneuverable satellites can provide on-demand regional coverage with greater persistence and lower cost than traditional static constellations. However, providing a very high level of persistent coverage over unpredictable ground locations of interest would require a static constellation to launch so many satellites that the costs would be unreasonably high for many potential applications. As reconfigurable constellations can change the orbits of their satellites in response to demand, they are able to improve the amount of utility provided by each satellite in the constellation, rather than relying on being able to acquire static coverage from a much higher number of satellites, most of which will not be providing coverage of the area of interest.

The advantage afforded by reconfigurable constellations is the ability to flexibly respond to uncertain future operating conditions, because reconfiguration grants the ability to change orbital coverage after the constellation is launched. With static constellation designs, achievable ground coverage is decided by the choice of design variables, and can no longer be altered after launch. If the locations of desired coverage are not known in advance, any attempt to optimize a static constellation design to achieve a desired minimum level of coverage over a broad range of potential ground locations will result in the selection of architectures that attempt to achieve global coverage at all times, resulting in more costly designs. In comparison, a reconfigurable

constellation can repeatedly maneuver its satellites into repeating ground tracks to obtain persistent coverage of the desired location, regardless of where that location happens to be.

3.1 Existing simulation model

Robert Legge developed a detailed optimization and valuation model for reconfigurable satellite constellations (ReCon) as part of his PhD work in the MIT Space Systems Laboratory (SSL).[8] The ReCon framework concurrently optimizes the individual satellite design, overall constellation architecture, and aspects of the reconfiguration maneuvering, as well as explicitly considering uncertainty in the future operating conditions that such a constellation will encounter. The simulation is written in MATLAB, and the orbital propagation was validated by Legge using Systems Tool Kit. This codebase was inherited by other graduate students in the SSL and used to further additional research on the ReCon project, with upgrades, modifications, and extra capabilities added as described within this thesis and others.[48, 47] The structure of Legge's original model is described here to provide context on the unmodified original framework, with modifications and contributions of the current author described in later chapters.

Legge's framework generates a range of constellation designs and evaluates these in a campaign-based model, optimizing performance over the full mission lifetime rather than attempting to perfect the utility gained during each individual event of interest. The structure of the ReCon mission lifetime assumes an initial launch and deployment of the satellites into a Global Observation Mode (GOM), where the constellation supplies global coverage (within the latitude bands determined by the orbital inclination of the selected design) at a low level of persistence. Once an event of interest occurs, the satellites maneuver into intermediate drift orbits to achieve the correct phasing prior to maneuvering again to reach their specific orbital slots within the desired repeating ground track. Placing the satellites within the RGT is

used to achieve persistent coverage over the target location of interest, at the cost of losing global coverage of areas which are not relevant to the mission objective. This RGT mode is referred to as the Regional Observation Mode (ROM). After the period of interest in a target location ends, the constellation maneuvers back into GOM. This return to GOM is necessary for several reasons: to re-acquire global coverage while waiting for the next unpredictable target location to be determined (as it is very unlikely that the next target will coincidentally occur along the same RGT), to avoid large relative drift in the right ascension of the ascending node (Ω) across the constellation, and to ensure that the satellites within the constellation do not end up clustering within their orbital planes and degrading the pass time options for future events of interest.

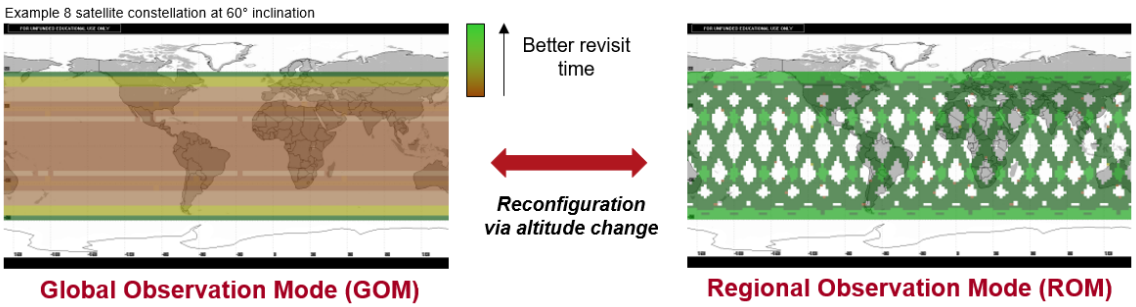


Figure 3-1: Comparison of satellite revisit times and ground coverage achieved in Global Observation Mode (left) versus Regional Observation Mode (right). Image credit: Robert Legge[8]

As shown on the left in Figure 3-1, a satellite constellation in GOM provides partial global coverage (within the latitude bands imposed by the orbital altitude of the constellation) but with considerably worse revisit times of any particular location that is selected within the coverage area. Maneuvering into ROM as shown on the right of Figure 3-1 allows for a drastic improvement in the revisit time (and therefore achievable level of persistence) at a specific location along the RGTs, at the cost of degrading the global coverage so that some locations now receive no satellite passes at all. Once an event of interest has been selected, this is considered to be an acceptable tradeoff, as the target location has now been determined and all other locations

become irrelevant for the purposes of gathering observations.

Legge's goal in developing this codebase for satellite constellation design and optimization was to avoid many of the artificial limitations and arbitrary constraints placed on the design space by assumptions made in previous research. The framework was created in order to find globally efficient constellation designs, by using the explicit consideration of uncertainty in the constellation's operating conditions, and the application of comprehensive and multidisciplinary modelling to accurately represent subsystem interconnections, in order to avoid many common past simplifications.

The basic structure of Legge's framework requires the selection of several inputs: a constellation architecture (choosing from static, reconfigurable, sun-synchronous, or rapid launch options); some fixed design parameters (such as propulsion system specific impulse or propellant module structural mass fraction) which are not changed during the design optimization; and system constraints (such as minimum orbital altitude or maximum mass fraction of the propulsion system) used to place bounds on the design optimization process. The model takes these specifications from the operator and uses them to generate a set of optimized designs which are assessed along a cost/performance curve, with the objective of maximizing performance while minimizing cost. This is referred to as a Pareto curve, an example of which is shown in Figure 3-2.

The cost model is based on a combination of payload cost, spacecraft cost, launch cost, and quantities of scale effects, but excludes operations costs.[8] For modelling payload costs, Legge combined the NASA Instrument Cost Model (NICM) with an optical telescope assembly cost model developed by Stahl[85] to arrive at an estimated payload cost based on aperture diameter. Spacecraft cost was estimated using a combination of the Small Satellite Cost Model (SSCM) and the Unmanned Spacecraft Cost Model, version 8 (USCM8).[86] Launch costs are based on a database Legge compiled of existing US launch vehicles available at the time of writing in 2014. Fi-

nally, Legge's cost modelling incorporates amortization of non-recurring engineering costs, and learning curve effects that represent a fractional reduction in cost per satellite for each doubling in the number of satellites produced.

All of the cost values are presented in US dollars for FY2010 for consistency of comparison with Legge's past results, meaning that if inflation alone is taken into account, realistic 2022 cost values would be 32.6% higher. Straub modelled updated launch costs for the ReCon model in 2020,[48] but these are not incorporated here as this work was already underway at the time of Straub's findings, although this would be a highly useful addition to future work using the ReCon codebase. Other future work should consider updating the cost modelling, quantifying the amount of variation in constellation costs rather than predicting a single value, and incorporating operations costs into the overall cost estimates.

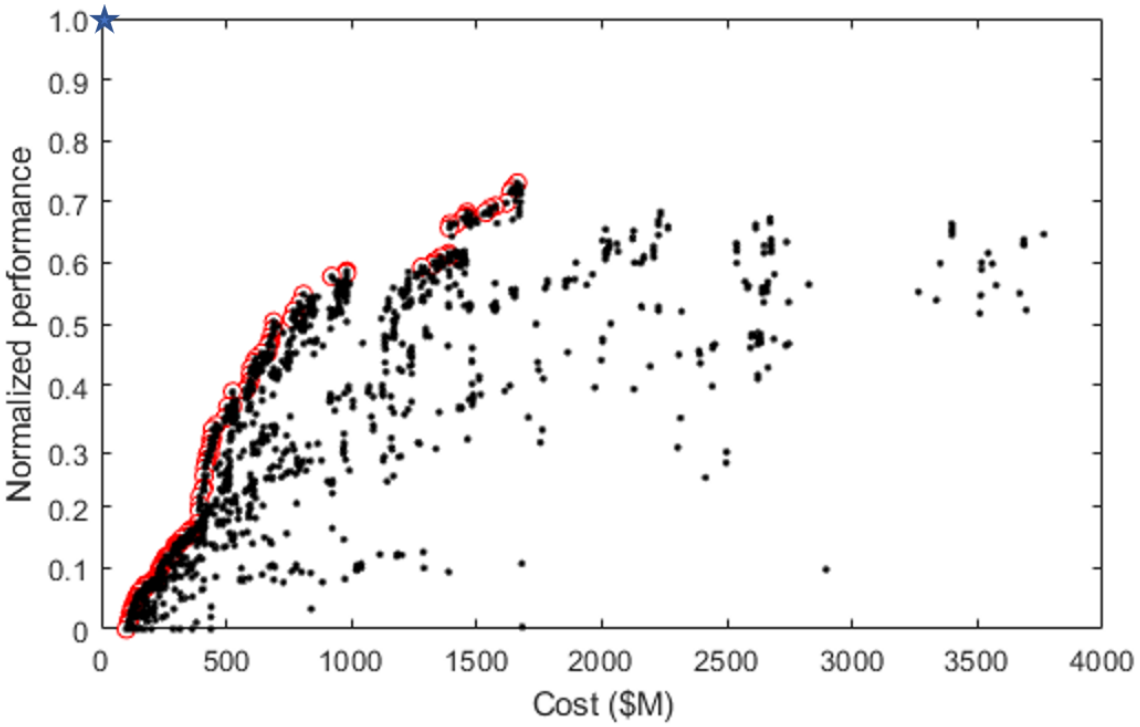


Figure 3-2: Example of a Pareto chart, showing a range of constellation designs plotted by normalized performance against cost

The Pareto chart in Figure 3-2 shows sample output from a single run of the Re-Con codebase. All of the generated constellation designs are plotted as black dots, compared along the axes of constellation cost and normalized performance (where 1.0 is the maximum achievable ideal performance, if every single desired satellite pass of the location of interest was achieved at exactly the desired time and at the desired resolution). The blue star in the upper left of the figure represents the ideal point, delineating an imaginary theoretical best design which would achieve perfect performance at zero cost. The Pareto front in Figure 3-2 consists of the designs on the left highlighted by red circles, which are the set of designs which fall closest to the ideal point. These designs are 'Pareto optimal' or 'non-dominated', meaning that an improvement in one metric can only be achieved by a reduction in the other metric. Taking any one of these highlighted non-dominated designs as a starting point, another design that gives higher performance must cost more, or an alternate lower cost design will perform more poorly in comparison to the original. The designs shown by plain black dots are 'Pareto dominated', meaning that they have been outperformed on both metrics of interest by some alternative option, and are therefore judged to be non-optimal. The set of optimal non-dominated designs are saved as the output of Legge's framework, and the non-optimal designs are discarded.

Thoroughly detailed descriptions of the various layers of the simulation model may be found in Robert Legge's PhD thesis,[8] but a useful summary is shown in Figure 3-3, consisting of three main layers. The first layer (shown in blue) is a multidisciplinary simulation model which outputs performance and cost figures for the mission lifetime of a satellite constellation design. The second layer (shown in red) incorporates Monte Carlo sampling of various uncertain parameters, using an input parameter distribution for uncertain event locations and timings to arrive at a performance and cost distribution from the simulation model layer. The third layer (shown in green) is a multi-objective optimization used to find a performance/cost Pareto front of potential constellation architectures satisfying the specified set of objectives and constraints.

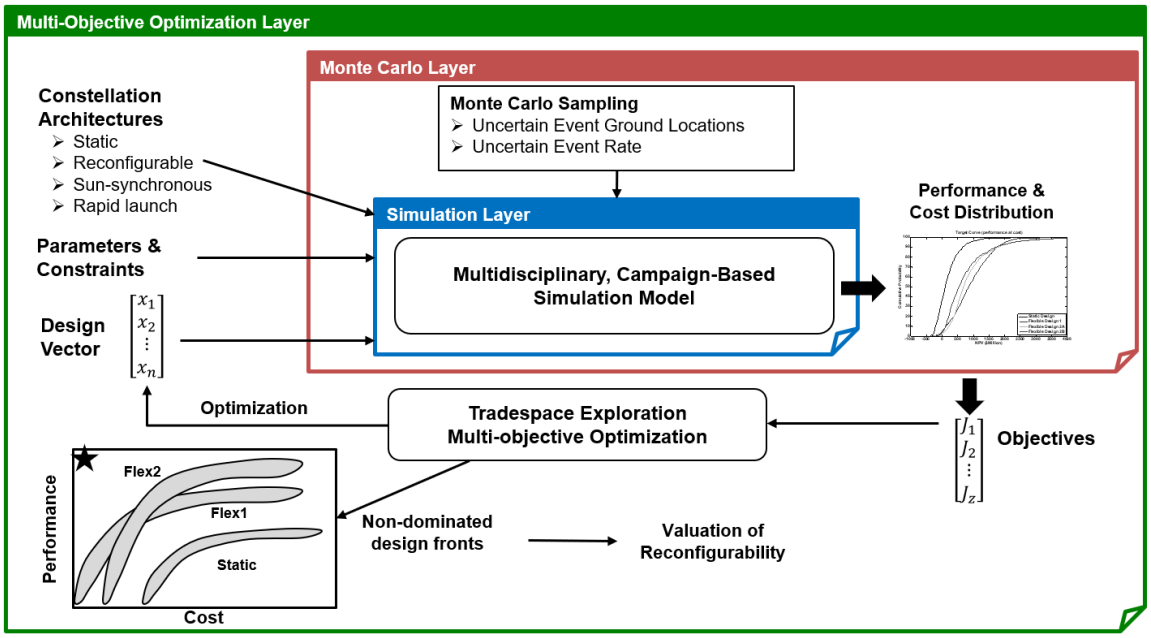


Figure 3-3: Structure of Legge's ReCon framework for constellation design and optimization. Image credit: Robert Legge[8]

3.1.1 Simulation layer

The ReCon simulation model is by far the most detailed of the three layers described above, using a set of inputs that includes the choice of constellation architecture, a vector of design variables, and various system parameters that are assigned fixed values rather than being varied during the design optimization process. This model layer can be envisioned as the simulation of a single constellation design over the entire course of a single possible mission lifetime scenario. The simulation model takes in the selected constellation architecture, any fixed system parameters or constraints that must be fulfilled in order for the design to be feasible, and a vector of design variables that specifies the individual design under consideration. The simulation output consists of the performance and cost for this individual design under one scenario, as well as highlighting any constraint violations that may affect how realistically the design could actually be used.

The design vector includes variables such as the repeating ground track (RGT) from which the constellation will make observations, the variation in altitude between ROM and GOM orbits, the number of orbital planes, the number of satellites per orbital plane, the orbital inclination, the ΔV budget for all reconfiguration maneuvers over the mission lifetime, the aperture diameter for optical imaging, a parameter to determine satellite phasing between orbital planes, and a decision weighting variable. This weight-based decision model is used to simulate the balancing of resources during the mission lifetime, between the competing objectives of achieving coverage as quickly as possible versus conserving propellant for use later in the mission. Legge chose to include this variable in the design optimization to allow for variation in preferences over the duration of the mission as propellant use becomes known, rather than assigning a single fixed value at the start of the mission. Designating a fixed value might result in the system running out of propellant early and being unable to react to events of interest at the end of the constellation lifetime, or conversely over-emphasizing the conservation of propellant and failing to generate what could have been useful coverage by leaving extra unused propellant leftover at the end of the mission.

The structure of the simulation model is shown in Figure 3-4, consisting of five distinct modules. The simulation setup module takes the input given for the constellation architecture, the ranges for design variables, and the fixed system constraints. These are used to generate a constellation pattern and set of design variables that define a single potential constellation design. For symmetric architectures, these designs consist of Walker Delta pattern constellations (as discussed in Section 2.1.1), with asymmetric architectures derived from restricted asymmetric and asymmetric Walker patterns.

This design and pattern is passed to the spacecraft module, which sizes the satellite dry mass and launch volume as a function of the aperture diameter (one of the initialized design variables), using curve-fitting based on data from past Earth obser-

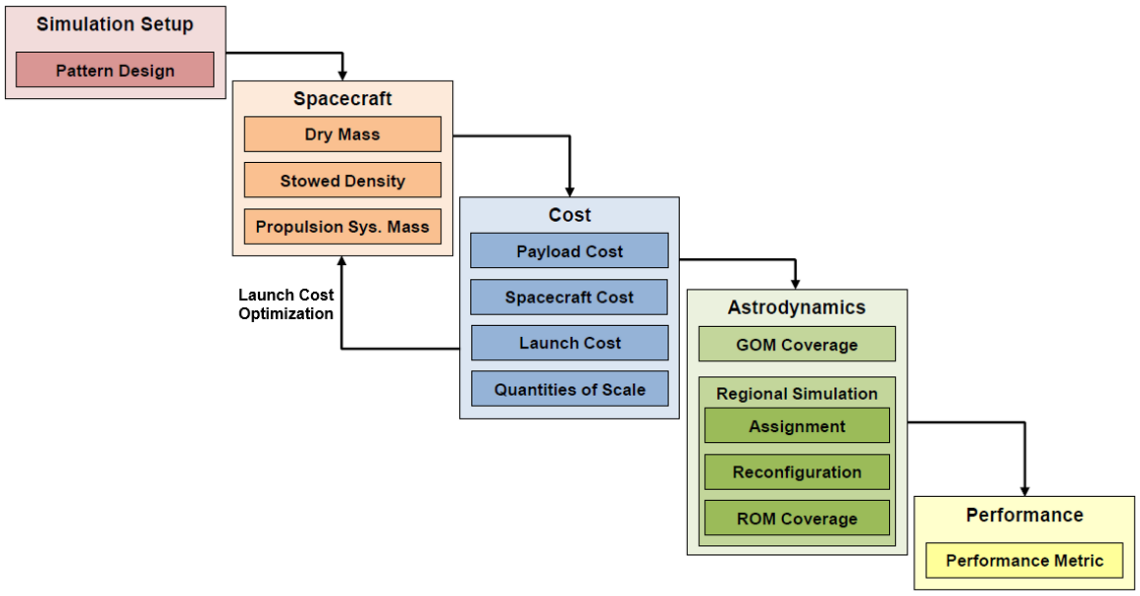


Figure 3-4: The five modules that make up the simulation model layer of the ReCon codebase. Image credit: Robert Legge[8]

vation satellites. The vector of design variables is used to model and appropriately size the satellite bus and payload. The propulsion system is sized based on the ΔV requirements for initial constellation deployment and phasing, station-keeping and drag makeup, the ΔV budget for reconfigurations over the whole mission lifetime (another of the initialized design variables), and satellite disposal at the end of the mission lifetime. The total ΔV budget is used to calculate the propellant mass and total propulsion system mass, and then spacecraft dry mass can be estimated.

The cost module combines optical payload, spacecraft and launch costs, and applies effects for economies of scale, although operations costs are excluded. The economies of scale are implemented by amortizing non-recurring engineering (NRE) costs over the whole constellation and applying a learning curve effect to the per-satellite costs, for efficiency gained in producing more units of the same satellites. The optical payload and spacecraft costs are estimated based on a comparison of several existing models, with launch costs calculated using a customized model assigning satellites to specific launch vehicles and carrying out a launch cost optimization process in conjunction with the spacecraft module (shown in Figure 3-4 as the one upward

link between modules). This optimization is carried out to examine the tradeoff between increasing the onboard ΔV budget (thereby increasing spacecraft costs) and decreasing launch costs. Minimizing overall costs requires finding the balance between these factors, as increasing the amount of propellant available for constellation deployment can reduce launch costs by imposing less stringent requirements on the initial orbit into which the satellites are launched. The launch assignment process in the cost module is iterated using a range of values for the initial deployment ΔV budget in order to find the global minimum for overall constellation cost.

The astrodynamics module uses the constellation pattern and design vector to calculate ground coverage in GOM. This module uses a single target deck of events of interest to supply the scenario for the constellation's mission lifetime, modelling the constellation response to each event of interest as a sequential campaign, propagating the satellite orbits and optimizing propellant usage over time. For reconfigurable designs, a process of maneuver selection is carried out to move satellites into ROM and achieve more persistent coverage over the target location during the period of interest. For static designs, the achievable event coverage in GOM is calculated for this same period. A satellite assignment optimization is carried out within this module, determining how many satellites will be maneuvered for an event response, which individual satellites will be maneuvered, which drift orbits will be used for phasing, and which RGT each satellite will be placed into, which determines whether ascending or descending passes will occur over the target location. A dynamic programming optimization is carried out to determine optimum satellite assignment, using the decision weighting variable from the design vector that was previously described in this section. In an operational reconfigurable system, satellite operators could use the output of this optimization to make decisions on the prioritization of achieving coverage of the current event of interest as quickly as possible versus conserving propellant for later reconfigurations, but here the decision model is used for the purposes of automating this factor to enable full design optimization. This optimization process also applies a penalty for maneuvering any satellites that have a lower than average amount of

remaining propellant, in order to balance propellant use across the constellation as a whole and avoid situations in which some satellites prematurely deplete their entire ΔV budget before the end of the mission lifetime is reached.

The fifth and final module within the simulation layer is the performance module, which takes in all of the information from the previous modules on constellation design and pattern, mass and cost calculations and event response and calculates the overall lifetime performance of the system. This is computed as the mean performance generated over all events of interest during the mission lifetime scenario. A custom performance metric is used to assess how well the persistence and resolution of the satellite coverage achieved by the constellation matches the desired levels of temporal and spatial event coverage, and this is described in more detail in Section 3.1.1.1 below.

3.1.1.1 Performance module and custom performance metric

Satellite constellation performance is commonly evaluated using metrics such as average revisit time and maximum revisit time over the location of interest. However, Legge and other previous literature[8, 58, 87, 88, 89] has demonstrated that optimizing for one of these metrics often worsens the other, despite the fact that they are usually used together as a pairing to assess the performance of satellite ground coverage. Average revisit time is improved by merely increasing the number of observations within the specified time window, with no attention paid to when they occur or the spacing of observations within this window. Maximum revisit time is improved by simply reducing the duration of the single longest gap in coverage. Neither of these metrics account for whether a desired level of persistence is achieved. An alternate metric of response time, which is defined as the mean time to coverage in a given time window, allows for comparisons to be made with a desired persistence level, but due to its statistical nature, it cannot take other factors into account such as varying spatial resolution across different satellite passes.

Legge defined a custom 'persistence metric' by combining two utility functions, assessing whether the desired levels of persistence and spatial resolution are achieved, as well as incorporating illumination constraints for optical imaging applications which require daylight passes over the location of interest to gather useful imagery. Persistence is defined as sustained coverage of a target location during a period of interest, achieving a stated and consistent frequency of observations, and so the persistence metric is designed to assess how well the constellation performance matches this desired frequency.

The first utility function incorporated into the persistence metric is a temporal utility function derived from the amount of time elapsed since the last observation was made. This is used to assign a utility value based on whether the time at which passes occur matches up well with the desired cadence of observations. The temporal function starts at a value of zero when the time since last observation is zero, and increases at a constant rate, reaching a maximum value of 1 once the time since last observation reaches the desired observation cadence. For example, for a desired revisit cadence of one hour, the temporal utility function would start at zero after zero minutes, climb to 0.5 after 30 minutes, and reach a value of 1 after one hour. This means that if the target location was recently observed, the utility of an additional observation is low. The temporal utility does not increase beyond its maximum value after the desired revisit time is surpassed, so as not to reward undersampling and missed observations.

The second utility function integrated into the persistence metric is based on whether the desired spatial resolution, or ground sample distance (GSD), is being achieved by the constellation observations that occur. Rather than imposing a fixed minimum requirement for useful GSD, Legge observed that a reconfigurable constellation may be spread across a range of orbital altitudes, where different satellite passes would provide imagery at varying levels of spatial resolution. Incorporating a spatial resolution utility function allows for the utility supplied by each pass to be scaled

to the gradual degradation of optical information as this resolution declines, rather than selecting an arbitrary cutoff limit beyond which no value is gained from the imagery. This function gives a utility of zero for cases where the achieved GSD is four or more times greater than the desired GSD, and a utility of 1 where the desired GSD level is met or improved upon, again with no increase in utility given above the target value, to avoid rewarding oversampling in this case. Between the maximum utility value of 1× the desired GSD and the minimum utility value of 4× the desired GSD, Legge used a scaling parameter to give a curve that matches the level of degradation in utility to the National Imagery Interpretability Rating Scale.[90, 91]

These two utility functions were combined to create a two-dimensional utility surface, and a dynamic correction term was added to ensure symmetry of the surface. This addition was made to account for cases where later observations may be occurring too soon to provide significant temporal utility, but are supplying significantly improved spatial resolution above that acquired by previous passes that did not meet the desired GSD value. Finally, Legge applied a rectangular window function for daylight hours, specifying that only daylight observations are able to provide utility for a mission application that requires optical imaging.

3.1.2 Monte Carlo layer

While the simulation layer of the ReCon codebase can be envisioned as modelling a single constellation design over one possible mission lifetime, the Monte Carlo layer is then used to iterate that simulation layer to assess how the constellation design performs over a range of possible scenarios for its operating lifetime. Rather than providing a performance score for a single scenario, the Monte Carlo layer uses the uncertain factors of where and when events of interest will occur (in the form of randomized target decks, described in more detail in Section 3.1.2.1) to estimate a distribution of constellation performance. This can be used to make predictions about the expected performance the constellation will achieve under uncertain operating conditions.

Due to the fact that the simulation model is non-linear, simply using expected values of the uncertain factors as inputs will not give a correct expected value of system performance.

$$F(E(x)) \neq E(F(x)) \tag{3.1}$$

The inequality shown in Equation 3.1 can be illustrated by considering any non-linear function; Legge gives the example of $f(x) = x^2$ for a uniform distribution on the interval $0 \leq x \leq 1$.
This means $E(x) = 0.5$, which gives the result $F(E(x)) = 0.25$. However, finding the value for $E(F(x))$ by integrating over the specified interval gives the result:

$$E(F(x)) = \int_0^1 x^2 dx = \frac{1}{3} \tag{3.2}$$

The discrepancy between these two results demonstrates how using expected values for the inputs to model a non-linear system will not give the correct expected values of system output; the distribution of uncertain parameters must be modelled sufficiently to give an accurate estimation of the resulting non-linear system response.

Intuitively this makes sense, as thinking about trying to find the 'average location' on the Earth's surface and use it as a simulation input is a nonsensical endeavor. If a set of completely random locations are defined in terms of latitude and longitude and then averaged, the result is likely to be the intersection of the equator and the prime meridian due to the co-ordinate system used. If the averaging was carried out in three dimensions, the mean location would be the center of the Earth. Neither of these results will supply a useful input for optimizing constellation design, nor reflect the full variation in system performance under uncertainty.

The objective of using Monte Carlo simulation is to find the distribution of constellation performance that occurs against the distribution of uncertain event locations and event timings. This is then presented by Legge as a single statistical performance

measure of the fitness of the resulting constellation designs. Legge uses median performance as the condensed performance metric for optimized designs, while observing that average or percentile values could be used based on stakeholder needs. Using the median performance will display the center point of the distribution, while using percentile values can show some information about the spread of the distribution itself. However, any single value taken alone can only present limited information about how a design may perform against a range of unknown operating scenarios.

3.1.2.1 Target decks for Monte Carlo simulations

If operating conditions are unknown, the question then arises as to how a constellation may be designed for them. For the ReCon project, a set of target decks are used, presenting different theoretical 5-year scenarios for the mission lifetime. A target deck is an array containing a single 5-year timeline with a set of randomized locations of interest, that are also randomly spaced in time, though not occurring simultaneously. Time between events is modelled as a normal distribution with a mean of 3 months and standard deviation of 1 month. As the default event duration is set to 14 days, the minimum time between events is set to 2.5 weeks to avoid the occurrence of any overlap. This randomized temporal spacing results in sets of 14–26 events of interest over a 5-year mission lifetime (mean: 19.5 events, median: 19 events, standard deviation: 1.5 events).

Events of interest are distributed within a set latitude range (minimum: 44°S, maximum: 65°N) and were generated by Legge based on data from a World Bank global risk analysis of natural disaster hotspots with effects weighted by economic impact.[92] This results in a probability density function (PDF) with clear global emphasis on areas of both high population (because disasters in uninhabited regions will have very little economic impact) and high natural disaster risk (because highly populated areas with few or no disasters will also suffer little economic impact). This is illustrated in Figure 3-5 by plotting 1000 random locations generated from the PDF, showing clear trends for densely populated regions of the world with increased

risks of natural disaster occurrence. Legge's research also compared this PDF to a uniform distribution of random locations between the latitudes of 40°S and 60°N, an area selected to include over 99.6% of the global population.

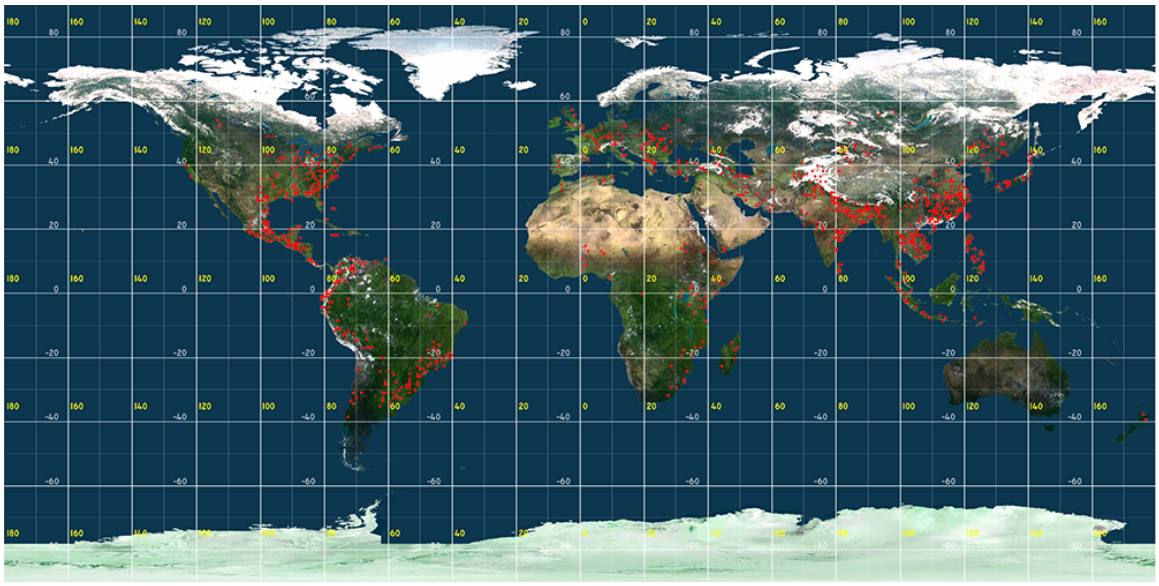


Figure 3-5: Probability density function of natural disaster risk weighted by economic impact, represented here by 1000 locations randomly generated using the PDF from the World Bank analysis of natural disaster hotspots[92]

Other latitudinal distributions may be used, but the specification must be justified due to the obvious effect on inclination selection for the constellation. Aside from locations of interest specified by latitude and longitude co-ordinates and event timing, these arrays also include the duration of interest (default is 14 days), desired ground sample distance (default is 1 m), daylight hours during which imagery will be captured (default is 6am–6pm local time), and desired revisit frequency (default is 1 hour). Target decks are used to evaluate performance, cost and constraint fulfilment for different constellation designs in an assortment of theoretical scenarios. If a design performs satisfactorily under a range of different operating scenarios, it is concluded that that design is well-suited to flexibility in its operating context.

The decision to use target decks rather than individual target locations may not be immediately clear, however it is operationally correct to optimize constellation designs based on a sequence of events rather than individual occurrences. Target

events cannot be treated as single stand-alone incidents due to the fact that each maneuver affects the future fuel availability and orbital position for each satellite. Balancing fuel consumption across the constellation forms part of the overall optimization strategy for the mission lifetime, meaning that the optimal reconfiguration may not be selected for each individual event in order to prioritize preserving a functional level of maneuverability until the mission duration is complete. Maneuvering between Global Observation Mode and Regional Observation Mode repeatedly over the course of multiple events also causes a drift in RAAN for each group of satellites that is moved, meaning that it is necessary to balance reconfiguration maneuvers across the constellation over time in order to maintain spacing between the orbital planes occupied by the satellites. Multiple operating scenarios must be used so as to avoid the aforementioned 'flaw of averages'; using a single 'average' 5-year mission to assess constellation performance will fail under almost every single scenario where the design is then exposed to real uncertainty, as it will result in a design optimized only for the exact mean of the distribution rather than incorporating responsive flexibility that can succeed under the variance found in real-world scenarios.

An important question that arises is how many target decks must be used to reach the conclusion that a design has been assessed against an adequate range of scenarios. Legge considers the level of Monte Carlo sampling error introduced by a given number of target decks by assessing the sampling error of the mean, median and lower 20th percentile values for constellation performance against the number of target decks the design is evaluated upon and including 68% and 95% confidence intervals for each of these three metrics. The confidence interval size is expressed using a percentage variation from the median, and it is concluded that using 24 target decks results in a 68% confidence level of 3% of the median value. This means that there is a 68% confidence level that the actual median value for constellation performance will fall within $\pm 1.5\%$ of the median performance value predicted by the model (the value that was used to evaluate the design during the process of determining "population fitness", as part of the design optimization process).

For the purposes of comparing Legge's confidence levels to those calculated in later chapters of this work for updated sampling alternatives, the use of 24 decks is treated as the pre-existing standard for an acceptable level of sampling error. However, as the updated calculations do not seem to exactly replicate the standard error levels calculated in Legge's work, the question of what constitutes an appropriate confidence level is considered in Chapter 5 alongside the results for this area of work.

If the number of target decks used for performance evaluation is altered, a balance must be struck between considering how much time and computational resources may be saved by using a smaller number of decks versus what effect a reduction in decks may have on the confidence level of the optimization results. It is also necessary to compare the performance confidence levels between reconfigurable and static constellation designs. The overlap in distribution between the two confidence curves represents the region for which it cannot be stated with confidence that reconfigurable designs will outperform static designs.

This is highlighted in Figure 3-6, showing a comparison of performance for a single reconfigurable design (on the right, in blue) against performance for a single static design (left, in orange) over 500 randomized scenarios. These designs were selected as an iso-cost pairing for this performance comparison and compared for their performance under uncertainty. Although the reconfigurable design achieves a mean normalized performance of 0.4 to the static design mean performance of 0.25 and both have comparable standard deviations of 0.04, there remains a region where it cannot be stated with 100% confidence that reconfigurable designs will outperform static designs, and this area is highlighted by the red box in Figure 3-6. Characterizing these regions and assessing confidence levels that reconfigurable designs will outperform static designs of comparable cost is an important question with this research, and this topic is covered in extensive detail in Chapter 4.

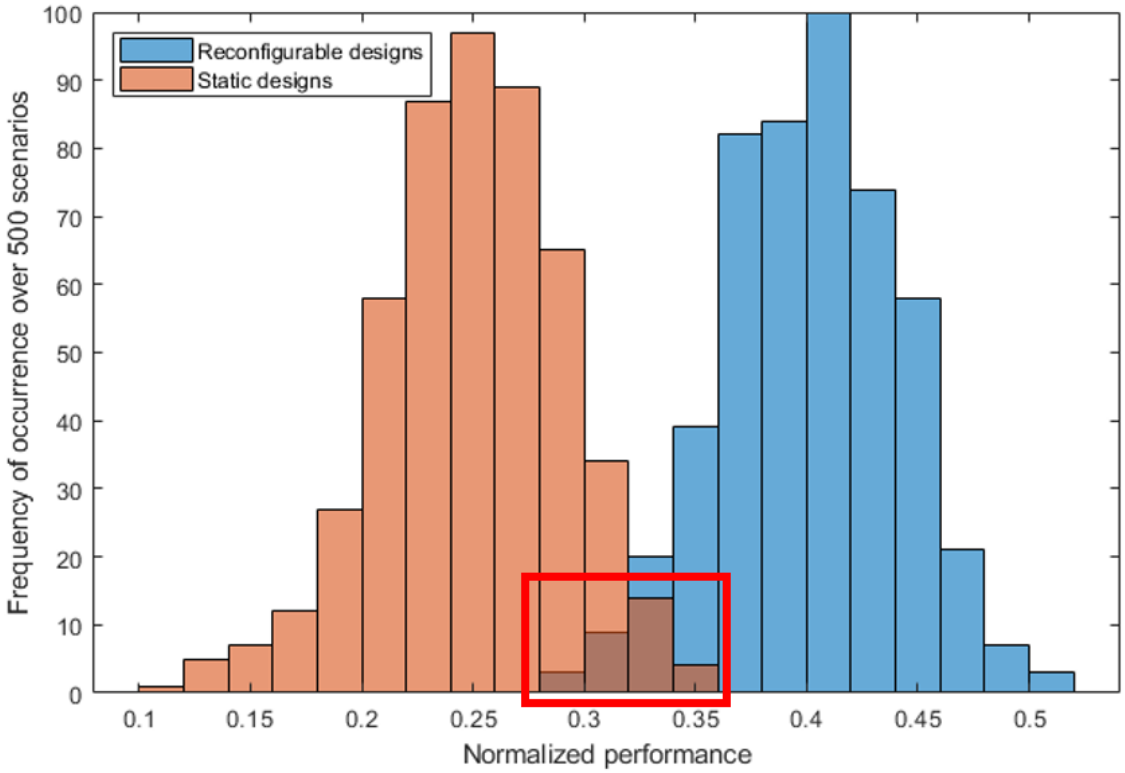


Figure 3-6: Overlapping histograms showing normalized performance for a pair of iso-cost designs (one static, one reconfigurable) over 500 random scenarios, with highlighted region of uncertainty for which it cannot be stated with confidence that the reconfigurable design will outperform the static design

3.1.2.2 Optimization versus re-evaluation decks

Target decks are used in the Monte Carlo layer of the framework as explained in the previous section, with each deck representing one possible mission lifetime for the constellation. For a five-year mission lifetime, the target deck consists of a list of event locations (specified in latitude and longitude) along with a sequence of timings for each event that are randomly distributed throughout the five-year period. In this work, overlapping events are not considered due to the additional complexity of dividing resource allocation between simultaneous targets, but this is recognized to be an important area for future work on reconfigurable constellations.

There are two different areas within the ReCon model where target decks are used, and it is important to distinguish between them. During the process of optimizing

the constellation design, the performance of each design is iteratively assessed during the Monte Carlo layer of the framework in order to calculate a distribution of cost and performance values across a range of scenarios. This step is carried out to examine the consistency of system performance under uncertain conditions, as part of the objective of designing for flexibility. Design fitness is assessed against 24 random target decks, and the performance against these scenarios provides the basis on which constellation designs are optimized during each code run. Legge uses the cost and performance scores from this output as the final scores upon which to judge the utility of a set of optimized, non-dominated designs that are provided as the output from a ReCon code run.

In contrast to Legge, the current work presented here chose to draw a distinction between the target decks used for assessment during the optimization stage of the framework (referred to hereafter as the 'optimization decks') and the target decks used to calculate a final performance score for the selected set of non-dominated designs (referred to hereafter as the 're-evaluation decks'). An extra script was created and integrated with the inherited ReCon codebase to select a new set of target decks to use for re-evaluating the performance scores of the output set of optimized, non-dominated designs and comparing these scores to the original utility scores assigned during the design optimization. This change was made to build in an extra layer of randomized assessment in the constellation design scoring, to ensure that selected designs did not only perform well against the target decks against which the design was optimized. As the optimization decks are used to drive the development of the design, a different set of re-evaluation decks are used as a form of validation for the performance scoring.

As an example, a theoretical scenario may be envisioned where the constellation design was optimized against a set of target decks, in which due to some quirk of randomization all the locations of interest were close to the equator. In this situation, the optimal designs selected are likely to be those at a low orbital inclination,

which will spend a greater amount of time passing over low latitudes compared to orbits at a higher inclination. If the designs are then scored only against this set of near-equatorial targets for which the constellation was optimized, they will naturally perform well. However, if the design performance is re-evaluated against a different set of target decks which include a much broader range of target latitudes, the performance score will naturally drop considerably once it becomes clear that the designs were only optimized for a narrow set of scenarios and not over the full range of desired flexibility in the operating conditions. This provides the motivation for the use of separate re-evaluation decks as an extra final stage in the framework scoring, which costs little in terms of computational resources while placing a check upon the design scoring to ensure that it is validated across a wider range of potential mission scenarios than just the set used for the initial optimization process.

3.1.3 Multi-objective optimization layer

For a satellite constellation with several competing objectives, there is no way to find a single 'globally optimal' design, which excludes the use of traditional optimization techniques focused on achieving a single objective. Multi-objective optimization techniques must be used instead, winnowing down the design space of potential constellations to arrive at the subset of non-dominated designs where one objective cannot be improved without another objective performing more poorly. The ReCon codebase sets dual objectives of simultaneously maximizing the median system performance and minimizing the total system cost, and potential constellation designs are assessed on these fronts.

The multi-objective optimization layer is the third and outermost layer of the ReCon codebase, wrapping around the previous two layers to iterate potential design variables against constellation objectives in an attempt to determine an optimal set of designs. The optimization layer calls upon the other code layers many times, iterating the simulation of individual constellation designs against many different mission lifetime scenarios, to assess performance and cost distribution under uncertain con-

ditions and discard any non-optimal designs.

Legge considered a range of multi-objective evolutionary algorithms (MOEAs) for use in the ReCon codebase, eventually opting to combine features from multiple MOEAs to customize the algorithm for this application. Early approaches to MOEA optimization converted multi-objective problems into a weighted set of single-objective problems which could then be solved using more traditional optimization methods. This approach is computationally inefficient and ineffective for non-linear or computationally intensive models, and therefore poorly suited to the ReCon optimization.

Developments in the second generation of MOEAs attempted instead to approximate the entire Pareto front of non-dominated designs, simultaneously improving this front towards the ideal point and spreading out the designs along it to reflect the full breadth of the non-dominated design space. These methods improved computational efficiency but required additional parameters to avoid becoming trapped in locally optimal solutions.[93] The two conflicting goals of achieving convergence of the optimization alongside maintaining diversity of the non-dominated solution set are often in tension, with the improvement of one leading to a deterioration in the other. Fixed population size can eventually lead to the algorithm having to discard non-dominated designs and ultimately sacrifice convergence for the preservation of diversity in the results. However, Ferringer's past work using the NSGA-II (Non-dominated Sorting Genetic Algorithm II) for a satellite constellation optimization found that the results were still stagnating prematurely in locally optimal solutions, rather than finding the true Pareto front.[58]

Work on the third generation of MOEAs led to a range of modifications to attempt to solve the issues of premature stagnation in local optima and deterioration where non-dominated designs are discarded. Legge's implemented solution combines features of several of these third-generation algorithms, taking features from the ϵ-MOEA

and Borg-MOEA algorithms and incorporating these into the ϵ-NGSA-II algorithm. These attributes were selected to avoid deterioration and premature stagnation, improve convergence, measure and improve convergence speed, reduce computational intensity, and account for complex interactions between interrelated or correlated design variables. More detail on the implementation of ϵ-NSGA-II and other features can be found in Chapter 5 of Legge's thesis, as well as an extensive comparison of the advantages and disadvantages of a range of multi-objective evolutionary algorithms.[8]

Finally, it was necessary for Legge to modify ϵ-NSGA-II to implement parallel computation in the optimization layer of the ReCon codebase. Due to the large amount of time devoted to functional evaluations in order to arrive at an optimized set of designs, it was decided to structure the code to use multiple processors in parallel, thereby providing the option to divide the computational demands across a number of cores and reduce the overall running time of the code.

Legge implemented a version of the simplest form of parallelization, using a leader-worker structure (formerly referred to as a "master-slave" structure, under deprecated terminology). This method designates a leader processor to implement the optimization functions that make up the modified ϵ-NSGA-II technique and send out assignments of functional evaluations to be carried out on the worker processors. The results of these evaluations are returned to the leader processor once completed, where they are compiled into the overall optimization output in an iterative process, with new evaluation tasks assigned to the worker processors to continue the cycle until the termination criterion of the optimization is reached and the code run ends. This termination criterion was chosen based on the rate of improvement within the constellation design population and averaged over a number of evaluations to smooth out convergence spikes. The rate of improvement is calculated as the percentage of the population archive that has been improved per 100 functional evaluations, averaged over the most recent 2500 functional evaluations. The optimization is terminated once the improvement rate drops below 2.5, meaning that over the last 2500 func-

tional evaluations, the algorithm is now improving less than 2.5% of the archive for every 100 evaluations it carries out.

Under ideal circumstances, parallelizing the processing would lead to a speedup factor that increases linearly with the number of processors used, but in practice this level of efficiency is rarely achieved. Limitations occur due to communication time taken to pass assigned tasks from leader to worker and to pass back output data from worker to leader, and due to idle time where workers must wait for a new task to be assigned by the leader processor due to computational bottlenecking.

Idle time may also occur due to generational synchronization, where all the functional evaluations for the current optimization generation must be completed before the leader processor can move on to assigning tasks for the next generation. Legge considered and rejected an asynchronous execution method for the ReCon codebase due to the large difference in duration taken to evaluate different constellation sizes, with a 9× increase constellation size found to result in a 16× increase in functional evaluation time. Such a large duration increase would lead to an asynchronous method carrying out many more evaluations for smaller constellation sizes, leading to their overrepresentation in the optimized output design set. Legge instead chose a batch processing approach to divide up the computational burden for each population generation and divide it into pieces that can be discretely evaluated. These are sorted by estimated evaluation time based on the constellation design size, and then the blocks are assigned to worker processors starting with the largest processing times. This synchronous method allows for smaller job blocks to be assigned to worker processors that have already completed their initial evaluation, thereby reducing idle time due to generational synchronization.

Parallelized optimization runs were carried out by Legge in 2014 on MIT Lincoln Laboratory's LLGrid computing cluster, described in more detail alongside its external equivalent in Section 3.2. Legge had access to a total of 1024 processors on LLGrid,

using these to carry out four simultaneous ReCon runs that were each allocated 256 processors. Over a period of three months, Legge completed 85 optimization runs, comprising a total of almost 40 million functional evaluations, equivalent to over 150 years of computation time on a single processor: thoroughly proving the necessity of parallelization for the ReCon codebase.

3.2 MIT Supercloud

MIT Lincoln Laboratory runs an internal-use computing cluster called LLGrid for use by its staff and affiliates, and an externally accessible version called the MIT Supercloud for use in research collaborations between Lincoln Laboratory and MIT and other academic institutions. This resource is provided to facilitate research with intensive computing, memory or data requirements. In 2019, the Supercloud cluster provided 1000 CPU cores and 10 GPUs across approximately 50 nodes for research purposes, with a per-user limit of 64 cores at any one time. By the end of 2021, following a series of upgrades, the Supercloud provided 32,000 CPU cores and 448 GPUs across 704 nodes, with the per-user limit upgraded to 1152 cores.[94, 95]

The facility is part of the larger MIT Lincoln Laboratory Supercomputing Center (LLSC), which had 40,000 cores as of 2018[94] and has been through several significant upgrades since then, making it the most powerful supercomputer in New England, and the third most powerful university supercomputing resource in the United States.[96] LLSC is connected to the Massachusetts Green High Performance Computing Center in Holyoke, Massachusetts, which is a renewable-energy collaboration between MIT, Harvard, Boston University, Northeastern, and the University of Massachusetts that provides hundreds of thousands of cores for academic research.

The work carried out in this thesis made use of MIT Supercloud resources to run the ReCon codebase, working in MATLAB and pMATLAB to carry out parallelized tasks across a series of cores. pMATLAB was developed at MIT Lincoln Laboratory

to remove standard obstacles preventing broader uptake of high-performance parallel computing, and enable straightforward access to this using the MATLAB language by interfacing with the libraries needed to distribute computational tasks in parallel at a large scale. Optimization runs were carried out between 2018 and 2022, with various upgrades made to the inherited codebase to bring it up to date with current versions of MATLAB and pMATLAB as the Supercloud went through upgrades over time. Other upgrades and modifications were made to improve sampling and other metrics within the ReCon codebase, as well as investigating how a range of design and operation decisions affect the performance of reconfigurable satellite constellations, and these are described in the following chapters.

Chapter 4

Probabilistic metrics for statistical distribution of performance

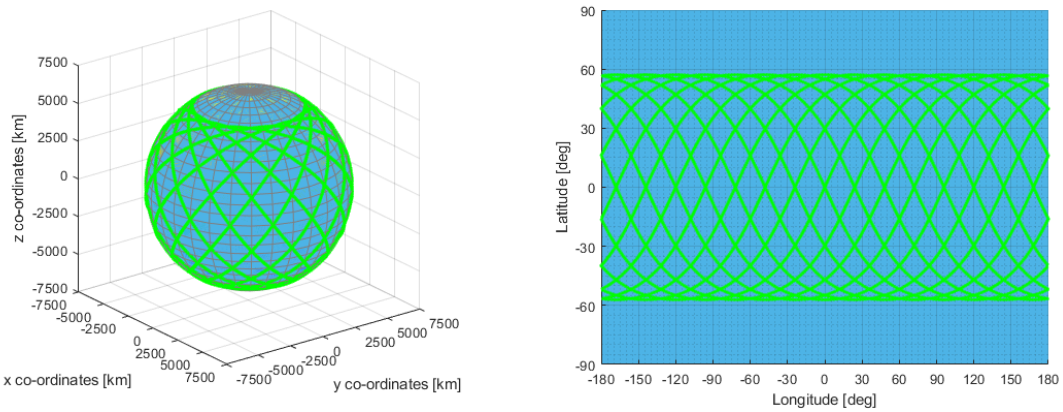
As the ReCon codebase is designed to optimize reconfigurable constellation designs to perform well under uncertain conditions, the performance metric of a constellation must be considered in statistical terms. Unless the exact location and timing for every event of interest can be determined ahead of time, the performance for any reconfigurable constellation will always be a probability distribution that must be analyzed statistically, rather than a deterministic score that can be predicted precisely for each design. In circumstances where the exact coverage requirements are known ahead of time, a static constellation may be designed to achieve such coverage without the need for reconfiguration or its attendant fuel and propulsion system costs, as the advantages of reconfigurable architectures lie instead in their ability to respond to unpredictable operating circumstances.

In summarizing a set of performance scores obtained against a range of possible scenarios that results in a distribution of outcomes rather than a single value, it is common to use multiple metrics to communicate a simple description of the data with more detail than any single metric can provide alone. In addition to a metric of central tendency (such as the mean or median), a metric of statistical dispersion around the central tendency (such as standard deviation, variance, or range) is fre-

quently used to give an idea of how the results are spread out around the average value. For non-Gaussian distributions, an additional metric to describe the shape of the distribution (such as skewness or kurtosis) may be added. If multiple variables are considered together in the output, a metric to describe the extent of any statistical dependence between these variables may be useful (such as a correlation coefficient).

There are many existing metrics for analyzing satellite ground coverage, each with their own advantages and disadvantages in visualizing the performance of a constellation. Some of these are more suited to evaluations of a single scenario, such as ground track plots or analytic coverage functions for a single orbit. Numerical simulations are generally necessary to establish statistical data and determine coverage figures of merit over a wider area. Satellite coverage is not a randomly distributed parameter; it tends to be achieved in clusters, as consecutive passes of the same satellite or adjacent passes of satellites in neighboring orbital planes are more likely to occur within range of the same ground location. It is necessary both to conduct analyses that assess patterns in coverage over time and to establish a qualitative physical sense of how coverage is provided by a particular orbit.

Figure 4-1 shows two examples of a ground trace plot for a 15/1 repeating ground track orbit (which completes 15 orbits of the Earth in 1 day), with Figure 4-1a showing this orbit in a spherical 3D representation and Figure 4-1b showing the same orbit in a flat map projection with longitude and latitude co-ordinates labelled along the axes. Advantages of ground track plots include their simplicity and usefulness in physical insight into the nature of the coverage supplied by a particular orbit. Comparing Figures 4-1a and 4-1b also illustrates the distortions introduced by using a rectangular map projection, which minimizes the proximity of the ground tracks towards the upper and lower latitude limits of the orbit. The disadvantage of ground track plots is the lack of ability to compare quantitative results between different orbit options; although they supply a useful illustration of the ground coverage achieved by a given orbit, they do not provide any metric of performance that can be easily



(a) Globe projection of the Earth (b) Flat latitude/longitude map projection

Figure 4-1: Ground track plots shown for a single satellite in a 15/1 repeating ground track (RGT) orbit on two different projections

evaluated against the results for other alternative orbit options.

Coverage may be examined as a function of latitude by creating a coverage histogram as shown in Figure 4-2. This shows what percentage of the coverage over the course of one day occurs within each 10° latitude band. The symmetry of the coverage north and south of the equator can be observed in this histogram, as well as the concentration of passes that occur at the upper and lower limits of the orbit's inclination just below 60°. This may be explained by visual comparison with the high proportion of crossovers seen between 50 and 60° latitude (and symmetrically between -50 and -60° of latitude) on the ground track plots shown in Figures 4-1a and 4-1b. Similarly to ground track plots, coverage histograms are straightforward to calculate and provide useful physical insight into the ground regions receiving coverage from a specified orbit, but suffer equally from difficulty in comparing multiple orbit options and the lack of a single numerical metric to enable comparisons to be drawn on quantitative terms.

Although visual representations of coverage such as those shown in Figures 4-1 and 4-2 are simple to generate and useful in supplying physical understanding of the coverage achieved by particular orbits, the lack of a quantitative metric that can eas-

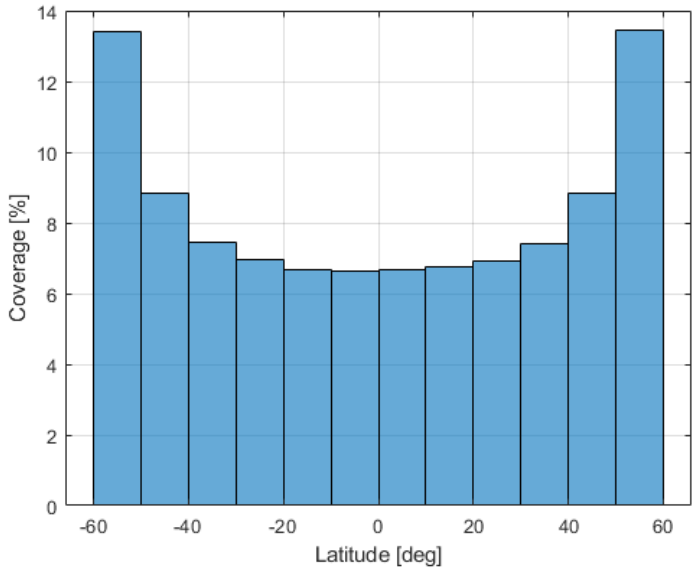


Figure 4-2: Histogram of coverage over the course of one day, as a function of latitude

ily be compared between different constellations or orbits means that they are poorly suited to evaluating the quality of ground coverage that is provided. Running numerical simulations allows for statistics on coverage to be gathered so that quantitative metrics may be calculated and compared across different design options.

Common standard coverage metrics include mean coverage gap, maximum coverage gap, mean response time, and percentage of time that coverage is achieved, as well as different visual formats for presenting coverage figures of merit. It is generally recommended to compare multiple figures of merit rather than relying on the results given by a single metric. Points of interest must be simulated on the ground as the orbits are propagated, so that coverage statistics may be collected and analyzed over time for specific locations.

Mean coverage gap refers to the mean length of gaps where no coverage is occurring over the location of interest. This is calculated by dividing the total length of time without coverage by the number of discrete gaps that occur over the period of interest.

Maximum coverage gap is the longest coverage gap encountered during the period of interest. This metric provides information on the worst-case coverage, though as the value depends entirely on a single coverage gap over the period simulated, this metric is not especially useful in ranking constellations without being able to compare performance across other coverage variables.

Mean response time is the mean length of time taken from receiving a demand for imagery of the location of interest to the next period of coverage over that point. A response time of zero corresponds to already having coverage of the point of interest at the time it is requested. If there is currently no satellite coverage of the target location, the response time is the length of time until the end of the coverage gap, when the next period of coverage of this location begins. Mean response time is averaged over all response times calculated for all time steps during the period of coverage simulation. This is one of the most useful coverage metrics for comparing responsiveness between different system options.

Percentage of coverage is calculated from the total amount of coverage time over the point of interest divided by the total time duration of the simulated coverage period of interest. This metric assesses how much of the time the point of interest is within coverage view of the simulated system. Although useful in supplying a figure for how much of the orbital period a particular location can be imaged, this metric does not supply any information about the coverage distribution, or how periods of coverage and gaps may be spaced out over the period of interest.

Figure 4-3 shows three simulated coverage plots along a nondimensional timeline for sample constellations labelled A, B and C. Constellation A (shown in blue) has 5 evenly spaced observations over the simulated time period, with equal gaps between each observation, and a proportion of one-third coverage time to two-thirds coverage gaps. Constellation B (shown in orange) has 4 observations, with the same total amount of coverage time as constellation A but with a larger gap between the first

two coverage periods. Constellation C (shown in yellow) has the largest proportion of coverage over the period of interest but divided across only 3 observations, with a very large gap between the second and third observation periods.

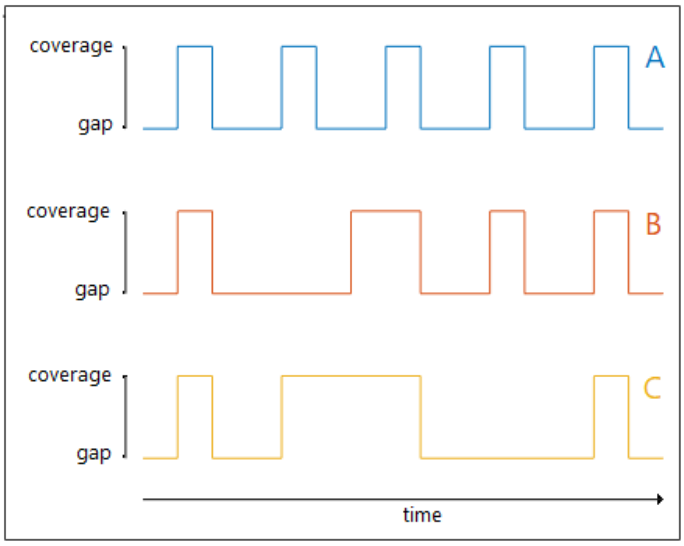


Figure 4-3: Three sample coverage charts comparing observations and coverage gaps for designs A, B and C

Table 4.1 shows several coverage figures of merit, which are calculated based on the sample coverage charts shown in Figure 4-3 and presented for each of the three constellations. This table shows how constellation A has the best performance for the metrics of mean coverage gap, maximum coverage gap and mean response time. Although constellation C has the best percentage of coverage overall, it performs worse than constellations A and B on all other metrics, due to the clustering of coverage into a very long second observation, followed by a very large observation gap between the second and third coverage periods, resulting in extremely inconsistent coverage. This highlights the need for consideration of gap statistics when ranking coverage performance, rather than relying solely on the total amount of coverage achieved. Mean response time is one of the most useful figures of merit for comparing different constellation options, due to the overall assessment view it provides of system performance averaged over every point in the simulated time period. As every figure of merit evaluates a different characteristic of the coverage that is achieved, the recom-

mendation is to combine two or three metrics in order to arrive at a comprehensive comparison of different constellation options.

Table 4.1: Figures of merit for the coverage shown in Figure 4-3 for sample constellation designs A, B and C

Constellation	Coverage figures of merit			
	Mean coverage gap	Maximum coverage gap	Mean response time	Percent coverage
A	2.00	2.00	0.93	33.3
B	2.67	4.00	1.21	33.3
C	3.50	5.00	1.36	40.0

Rather than using a standard coverage figure of merit, Robert Legge developed his own custom persistence metric (described in Section 3.1.1.1) to compare the achieved coverage to the desired levels of persistence and spatial resolution for each event of interest. This is presented as "a single statistical performance measure" used to express the distribution of design performance, for which Legge selects the median value as the center of the performance distribution. The mean value could also be used (although it is more susceptible to being skewed by outliers in the results set), or a desired percentile value within the performance range.

Figure 4-4 shows Legge's comparison of this persistence metric with other traditional coverage figures of merit. The first of these is average revisit time, which is calculated simply by dividing the time period of interest by the number of observations that occur during this time. This is analogous to the mean coverage gap metric explained above; average revisit time differs in that the revisit time is calculated from the entire period of constellation simulation, rather than only from the average of the gaps in coverage as was used to arrive at the mean coverage gap metric. Maximum revisit time similarly corresponds to maximum coverage gap, representing the longest time period between the start of two observations, rather than the longest time period between coverage as was used to calculate the maximum coverage gap metric. Mean response time is the third and final coverage metric used for comparison, and this

does not differ from the definition given previously.

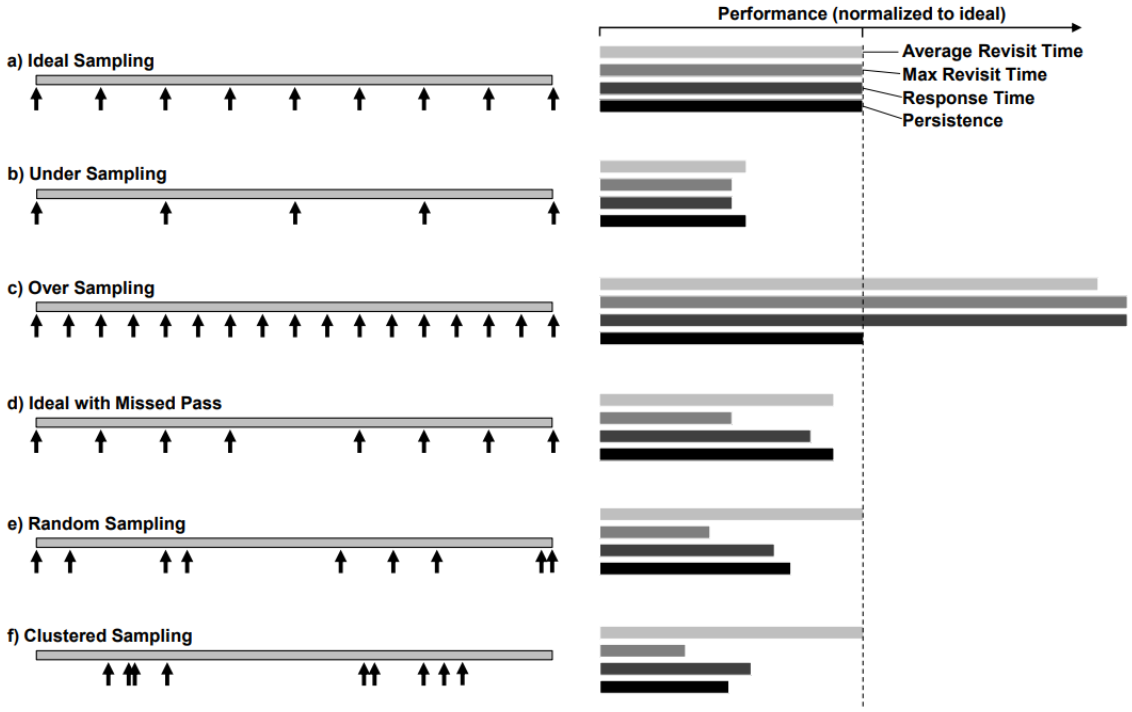


Figure 4-4: Comparison of Legge's persistence metric with traditional coverage metrics of average revisit time, maximum revisit time and mean response time. Image credit: Robert Legge[8]

Figure 4-4 compares these four metrics for six different coverage scenarios shown against a nondimensional timeline, where the desired coverage is shown in Figure 4-4a), with observations occurring at exactly regular intervals over the entire simulated period. In this subfigure, all four metrics are identical due to the completely regular nature of the coverage shown, with the same value calculated for average revisit time, maximum revisit time, mean response time and persistence. These values are used as the baseline to normalize the rest of the coverage metrics calculated for the following five scenarios.

Figure 4-4b) shows an undersampled scenario, where the observations occur at half the desired frequency. This corresponds to coverage performance of roughly half of the ideal performance for all four of the metrics shown. Maximum revisit time and

mean response time give values of 50% of the ideal performance, and average revisit time and persistence perform slightly better than 50% of ideal due to the lack of end gaps in the timeline period shown (with the first observation occurring immediately at the start and with no further gap in coverage after the last observation).

Figure 4-4c) shows an oversampled scenario, where the observations occur at twice the desired frequency. This corresponds to coverage performance of roughly double the ideal performance for the three traditional coverage metrics shown. Maximum revisit time and mean response give values of 200% of the ideal performance. Average revisit time performs slightly less than 200% of ideal, again due to the lack of end gaps, as the oversampled case equates to $2n-1$ observations, compared to n observations for the ideal sampling case. The persistence metric has an upper limit placed on it to prevent rewarding oversampling, resulting in a performance score equal to the ideal sampling scenario. Such a constraint could also be applied to the other traditional coverage metrics if desired, to avoid allocating a higher score for oversampling than for the ideal sampling cadence.

Figure 4-4d) shows a scenario which almost matches the ideal sampling scenario in Figure 4-4a) except for a single missed pass. This is reflected by a slight reduction from the ideal performance value for average revisit time, mean response time and persistence. However, this subfigure highlights the weakness of the maximum revisit time metric, which drops to 50% of the ideal performance value. This is due to the fact that this metric relies solely on the single largest coverage gap during the period of interest, and does not reflect performance achieved more broadly across the entire time period simulated.

Figure 4-4e) shows the same number of observations as in the ideal sampling case, but randomly scattered through the time period of interest. This highlights the weakness of the average revisit time metric, which gives an ideal performance value for this scenario despite the irregular nature of the coverage Figure 4-4f) similarly shows

the same number of observations but in an even more clustered random distribution, again highlighting the weakness of using only the average revisit time metric as it does not reflect the temporal spacing of the coverage that is achieved.

Comparing the four coverage metrics plotted in Figure 4-4 shows the weakness of using average revisit time or maximum revisit time alone as the basis of a comparison between the coverage supplied by different systems. Mean response time performs somewhat better in most scenarios, with the exception of the oversampling plot shown in Figure 4-4c). However, this could be remedied by applying an upper limit to the metric to avoid rewarding oversampling, similarly to the constraint that is applied to Legge's persistence metric. Based on a comparison of the current forms of the four metrics shown, the persistence metric gives the most consistent quantitative evaluation of the provided coverage, as it does not reward oversampling, does not drop sharply due to a single missed pass, and provides a performance score that rewards a regular temporal cadence of sampling over irregularly clustered observations.

4.1 Probabilistic versus deterministic coverage metrics

It is a severely limiting restriction to attempt to use any single coverage metric to represent the range of performance scores that will result from evaluating a constellation design against multiple possible five-year mission scenarios. Using the mean or median performance value from a variety of scenarios will simply provide a central value for this performance range. The average performance score will not provide details as to the consistency of the utility that is achieved against an unpredictable set of ground locations of interest for which coverage is desired. By the very definition of the median performance, the constellation design will have achieved a worse performance 50% of the time even when only the initial evaluations carried out as part of the optimization are considered. When performances are re-evaluated against

a different set of scenarios, even the average performance is then usually found to be a slightly lower value than the initial performance prediction that was generated as part of the optimization results.

Figure 4-5 shows the results of an optimization run of the ReCon code, which generates a set of reconfigurable constellation designs. These designs are plotted by their cost against the mean normalized performance achieved when the simulated design is evaluated against 24 different target decks; 5-year mission scenarios consisting of a set of 14–26 locations of interest. The non-dominated designs are circled in red; these represent the designs for which no alternative scores more highly in performance without also costing a higher amount (or alternatively: designs for which there is no lower-cost substitute without also sacrificing some amount of performance). The uncircled black dots represent dominated designs; these are designs for which at least one objectively better alternative exists, which simultaneously surpasses this design in both performance and cost. A total of 4772 constellation designs were generated by this optimization run, of which 182 are non-dominated.

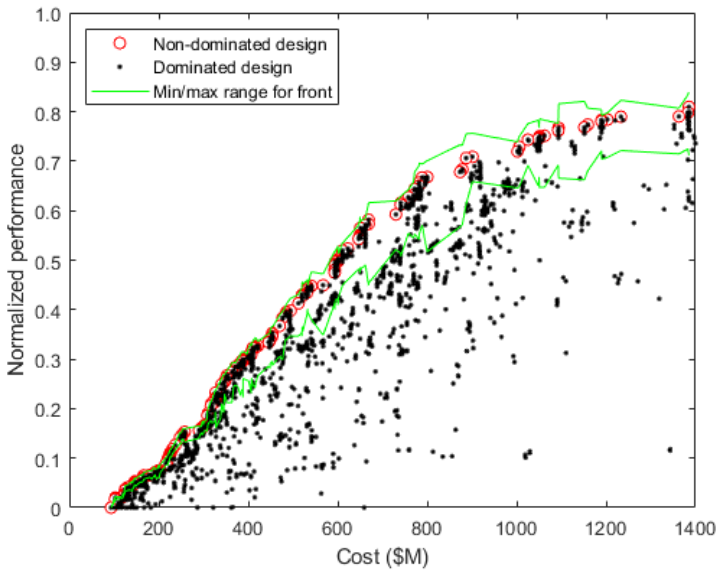


Figure 4-5: Pareto chart showing the output from a ReCon design optimization run: a range of constellation designs plotted by cost against normalized performance

After completing the initial optimization run, the set of 182 non-dominated designs along the Pareto front (circled in red in Figure 4-5) were re-evaluated against a completely different set of 24 target decks, to assess how much the performance scores varied from the initial optimization estimate. The upper and lower limits of the range of performances observed when the designs are re-evaluated against 24 different 5-year scenarios are delineated in bright green in Figure 4-5, to show the amount of variation around the value of the single performance prediction made by the original optimization score. It must be observed that this band is broad enough that it overlaps with many of the dominated designs from the original optimization output, which may outperform the 'optimal' set of non-dominated designs under some of the possible operating conditions. One area of interest for future work would be to take the subset of the dominated designs that fall within the range outlined in green in Figure 4-5 and re-evaluate their performance, to determine the extent of the overlap in performance ranges and establish whether any of these designs might exhibit more reliable performance than the supposedly non-dominated candidates highlighted in red in the same figure.

Figure 4-6 shows the percentage change in the mean performance for each of the non-dominated designs, compared to the original mean performance predicted as part of the optimization output. The mean change in performance is a decrease of 1.22% from the original score. The largest decrease is a drop of 7.87% compared to the original score, occurring for a constellation design with a cost of $327M. The largest gain in performance is an increase of 2.85%, occurring for a constellation design with a cost of $122M. Only 16.5% of the constellation designs (30 out of 182 designs) achieved an increase in the predicted performance. The designs with an increase in the mean performance score are almost entirely within the low-cost low-performance region of the Pareto curve, falling between costs of $93M and $221M. The only notable exceptions to this trend are two designs occurring at costs of $298M (which makes a gain of 0.17%) and $879M (which makes a gain of 0.13%), and these are both small enough increases to fall within the amount of expected error about the mean, given

the sample size of 24 target decks.

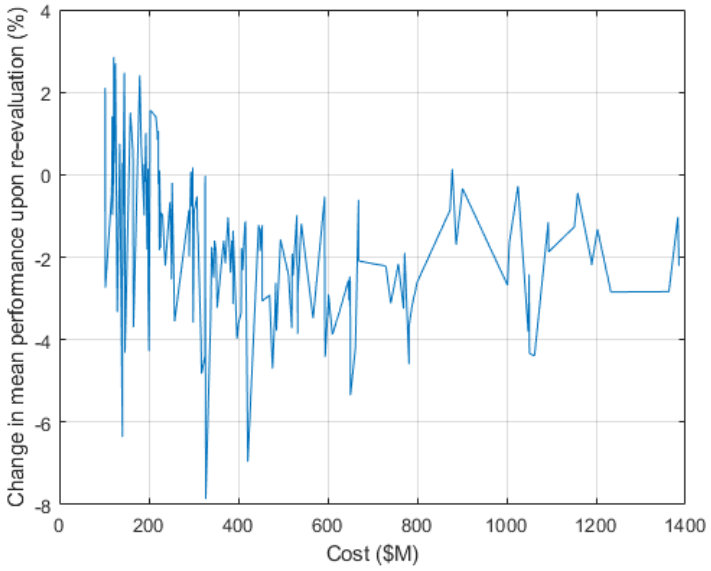


Figure 4-6: Percentage change from original optimization performance score for 182 non-dominated designs, when re-evaluated against 24 different target decks

The amount of variation seen in the predicted performance scores from the output of the ReCon model design optimization highlights the problems in relying on a single value for performance to represent an entire statistical distribution of results. Each point in Figure 4-5 represents a single design, but the performance of these designs cannot be accurately represented as a single point. One way of including more information in such a plot could be to add error bars showing either the full range of the data, 95% confidence brackets, or interquartile ranges to highlight the variability of the performance achieved under a range of different operating scenarios.

An area for future work is to consider the propagation of uncertainty in the cost models used in the ReCon codebase. Straub explored cost and performance trades between the use of chemical vs electric propulsion for reconfigurable constellations, and also investigated updates to available launch providers in the ReCon model.[48] This is an area of cost modelling that continues to evolve from year to year due to commercial pressures, particularly as vehicles are developed or taken out of service.

In addition, operations costs were not modelled in the original ReCon cost model, due to the lack of available data but also with the assumption that these costs would be similar between static and reconfigurable constellations. If this assumption is false and operations costs are found to be significantly higher for reconfigurable architectures, this may result in a reduction in the utility of reconfigurability across some of the constellation design space. This area of future work would allow for the consideration of uncertainty in two dimensions and the addition of error bars or confidence brackets to Pareto charts such as those shown in Figure 4-5. Each constellation design might be represented by a + shape showing the error bars in both dimensions, or by an oval showing the area of 95% confidence in the scores assigned by the optimization.

4.1.1 Re-evaluated performance scores

When optimized designs are re-evaluated against a new set of target decks, the performance score assigned by the optimization output is usually found to be an overestimate by a small amount. This is explained by a tendency for the optimization process to select designs that are well-suited to the initial set of target deck scenarios used to evaluate performance, rather than being suited to the full distribution of all possible scenarios. This tendency diminishes as an increasing number of target decks are used in the initial optimization process, but does not disappear entirely within the range of decks used within this work. The number of decks used could be increased further, but this would result in runtime increases beyond an amount which was considered tenable for the optimization sizes carried out within this research, whereas performance re-evaluation can be carried out separately from the design optimization, making it a much less computationally-intensive solution.

The reassessment of design performance was carried out as part of this work as described in Section 3.1.2.2, to build in an extra stage of performance evaluation for the output set of constellation designs. This step was added to ensure that the set of non-dominated designs resulting from an optimization run do not only perform well against the target decks that were used during the optimization, but exhibit consistent

performance against a wide range of possible scenarios. Rather than scoring designs only against the scenarios which were initially used to drive the optimization process, constellation designs are re-evaluated against a completely different set of scenarios, to ensure a consistent level of performance against unpredictable operating conditions.

When re-evaluation was carried out upon the non-dominated designs resulting from an optimization using 24 target decks to evaluate design performance, the mean change in performance was found to be a drop of 1.22% compared to the original score. Figure 4-6 showed the changed performance values resulting from a re-evaluation carried out upon this 24-deck optimization output, with 83.5% of the designs showing a drop in performance score compared to the optimization results.

Figure 4-7 extends this analysis of re-evaluation results for sets of non-dominated designs generated by optimization runs using 2–24 target decks. Each set of reconfigurable designs is re-evaluated against 500 different target decks, and plotted by its constellation cost on the x-axis and the mean percentage change in the performance score from the original optimization output value on the y-axis. Darker purple and blue lines show the smaller optimization sizes where designs are optimized against only 2–8 target decks, with mid-range optimization sizes shown in aqua and green, and the largest optimization sizes using 20–24 target decks shown in shades of orange and yellow. The smaller optimization sizes show the least amount of reliability in the performance scores, with the magnitude of variation from the original score reaching as high as 14% below the original value. Larger optimization sizes show greater reliability, with the magnitude of variation from the original score generally staying within 5%. Re-evaluated performance scores were found to decrease compared to the original prediction for 83.1%–92.5% of the designs from each optimization set.

Figure 4-8 shows the same re-evaluation analysis for sets of static constellation designs generated by optimization runs using 2–24 target decks. Each set of static designs is re-evaluated against 500 different target decks, and plotted by its constellation

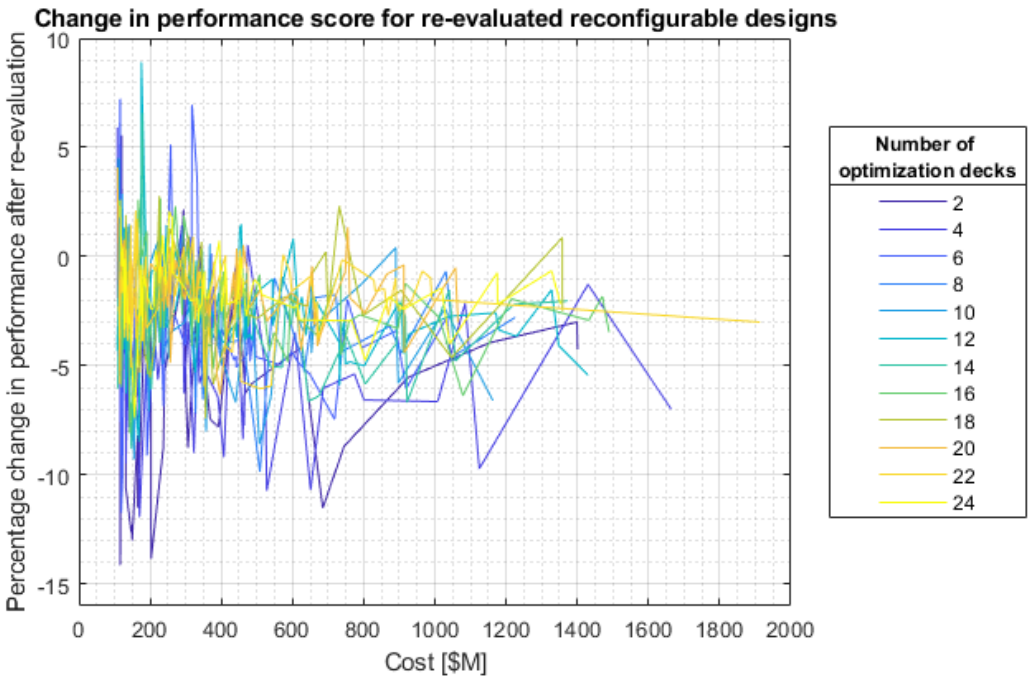


Figure 4-7: Change in performance score for re-evaluated reconfigurable designs compared to optimization performance values, for 12 different optimization sizes re-evaluated against 500 target decks

cost on the x-axis and the mean percentage change in the performance score from the original optimization output value on the y-axis. This figure shows the same trend as Figure 4-7, with smaller optimization sizes showing the least amount of reliability in the performance scores, with the magnitude of variation from the original score reaching as high as 16% below the original value. The largest optimization sizes show greater reliability, with the magnitude of variation from the original score generally staying within 4%. Re-evaluated performance scores were found to decrease compared to the original prediction for 69.2%–88.2% of the designs from each optimization set.

Figure 4-9 shows the mean change in performance score for each optimization size from 2–24 target decks, for re-evaluations of both reconfigurable and static designs. Static designs (plotted in orange in Figure 4-9) show less change in performance score than reconfigurable designs. For the smallest optimization size using only 2 target decks to optimize constellation designs, the static performance score dropped by an

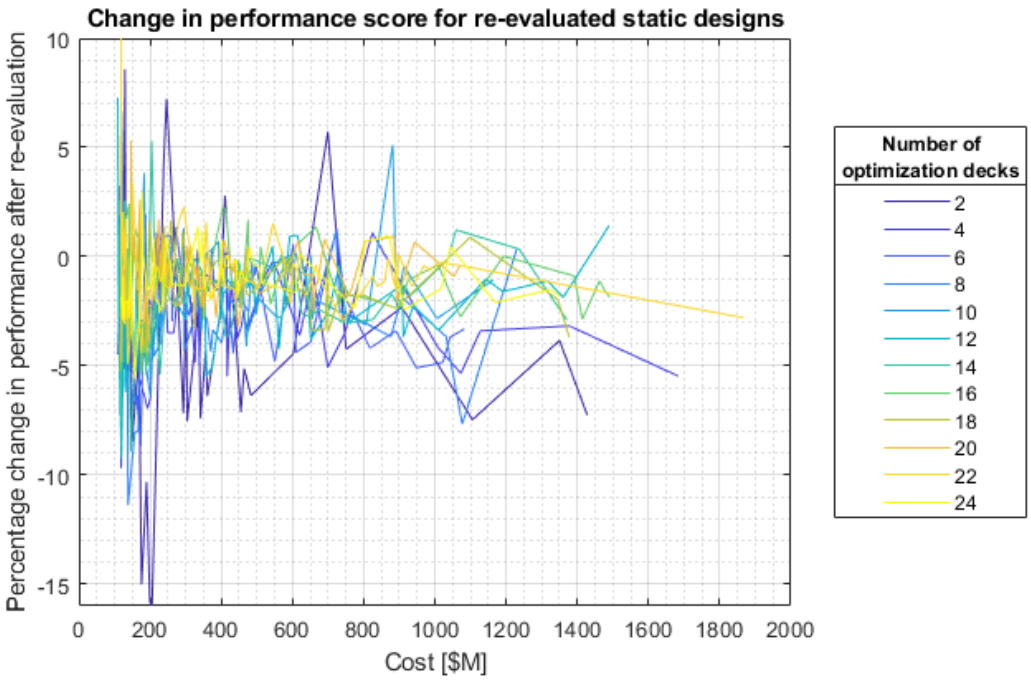


Figure 4-8: Change in performance score for re-evaluated static designs compared to optimization performance values, for 12 different optimization sizes

average of 4% when re-evaluation was carried out against 500 different target decks. This change dropped to less than 1% for optimization sizes of 16, 20, 22 and 24 target decks, with the smallest change of -0.51% seen for an optimization using 22 target decks. The lower amount of variation for static design scores is theorized to be due to the lack of reconfigurability of these designs; performance is predicted to be more consistent across a range of scenarios due to the use of passive coverage by the static constellation designs for all scenarios. As reconfigurable designs make use of responsive maneuvering, this creates a greater amount of variation in the event response to different target scenarios.

For reconfigurable designs (plotted in blue in Figure 4-9), the smallest optimization size using only 2 target decks to optimize constellation designs had an average performance decrease of 5.85% compared to the original score from the optimization output. The change in performance steadily decreases as optimization size increases, showing improvements in the amount of overestimation made by the original perfor-

mance score. The change in performance dropped to less than 2% for optimization sizes of 18–24 target decks, with the smallest change of -1.53% seen for the largest optimization size using 24 target decks.

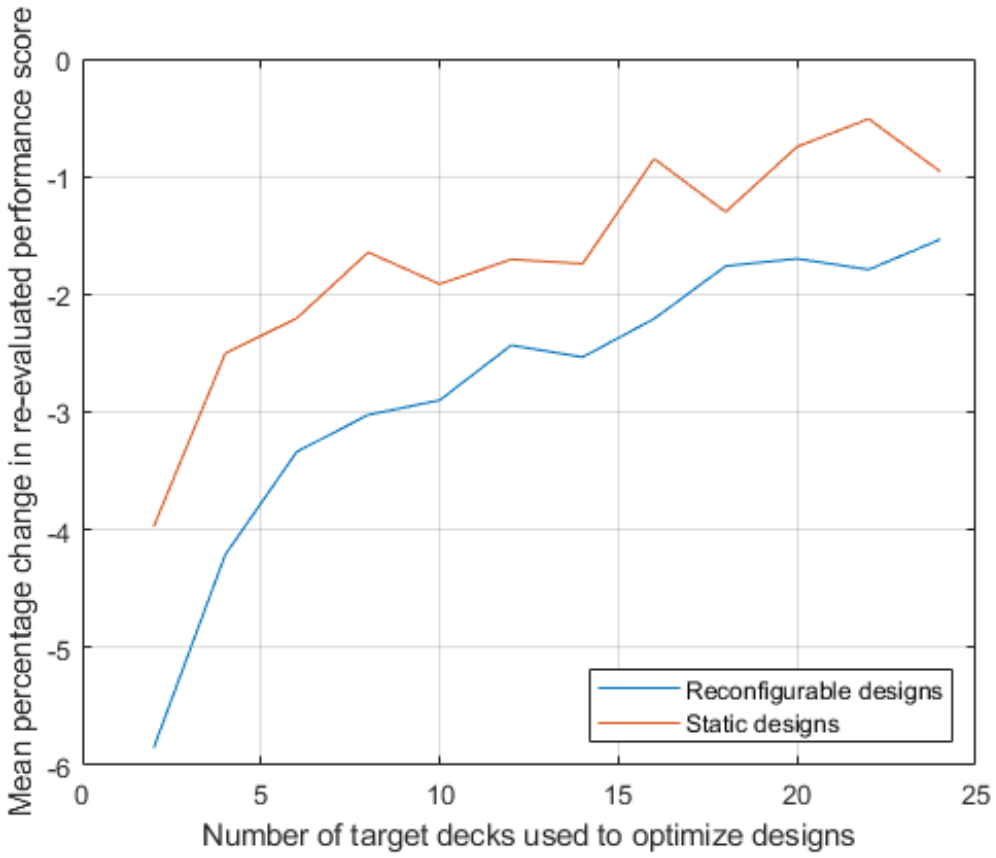


Figure 4-9: Mean percentage change in re-evaluated performance score compared to original optimization score for different optimization sizes

Figure 4-9 highlights the importance of carrying out design optimization using a sufficient number of target decks to evaluate and optimize the performance of constellation designs. For smaller optimization sizes, the overestimation of performance scores in the optimization output is sufficiently large that designs which were initially dismissed as dominated may actually provide preferable candidates when performance re-evaluation is carried out. The most significant drops in performance score reliability are seen when optimization sizes are reduced below 16 target decks, with relatively consistent amounts of prediction error seen for optimizations using 20–24 target decks.

The conclusion drawn from this analysis is that the use of separate re-evaluation decks provides a useful check upon the performance scores provided by the initial optimization output, while separating this function from the computationally-intensive optimization code run avoids adding to the runtime, which is already of significant duration. Although re-evaluation can be used to provide a more accurate performance score than the initial optimization, achieving a reasonable level of accuracy in the optimization score is desirable to ensure that the non-dominated designs have been correctly identified by the optimization run. This avoids the requirement to re-evaluate the entire output of several thousand dominated and non-dominated designs, which would present a much more computationally-intensive task than the current re-evaluation script focusing on only the non-dominated design set. To achieve this end, it is concluded that the minimum optimization size should not fall below 20 target decks, to maintain the level of performance overestimation error within 1% for static constellation designs and 2% for reconfigurable constellation designs.

4.1.2 Iso-performance versus iso-cost metrics

Legge establishes the Value of Reconfigurability (VoR) as a useful metric for comparing iso-performance reconfigurable and static designs. This is defined as "the reduction in total system cost for the same performance level by incorporating reconfigurability into the system". Non-dominated designs from the reconfigurable and static Pareto fronts are selected with approximately identical performance scores and compared on the basis of total constellation cost. The difference in cost is used to show a quantifiable scoring of the advantages provided by using a reconfigurable architecture. The improved utilization of each satellite in a reconfigurable constellation results in the ability to match static constellation performance using fewer satellites. Although the additional maneuvering capability required for reconfiguration will add to propulsion system costs, these are far exceeded by the cost savings achieved by launching a smaller constellation.[8]

Figure 4-10 compares the Pareto fronts generated by two different optimization runs of the ReCon code. The first non-dominated set of designs is for reconfigurable satellite constellations, and these are plotted as solid black dots in the figure. The second non-dominated set of designs is for static satellite constellations, and these are shown as white dots in the figure. Between the two Pareto fronts, the VoR is highlighted in blue at three different points, showing how the amount of benefit to be gained from reconfigurability varies throughout the design space.

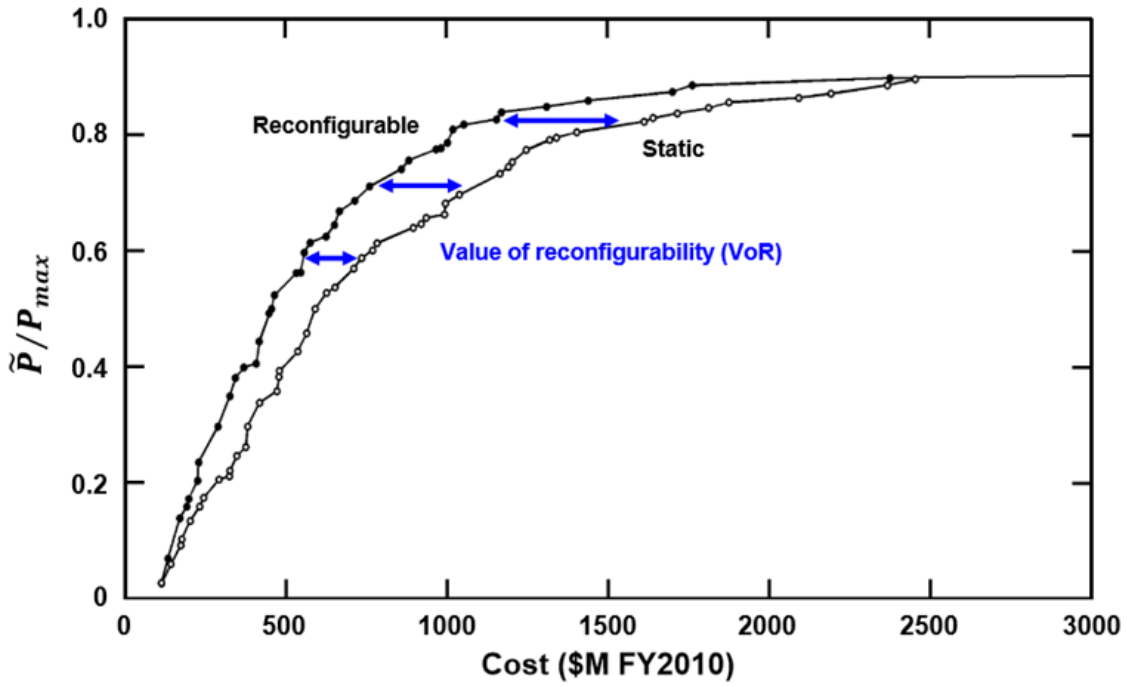


Figure 4-10: Pareto chart comparing cost and performance for two sets of non-dominated designs: one for reconfigurable constellations and another for static constellations. The Value of Reconfigurability (VoR) is highlighted in blue. Image credit: Robert Legge[8]

One weakness with analyzing the VoR in terms of cost savings is that this requires matching up pairs of reconfigurable and static designs for comparison on an iso-performance basis. However, the performance achieved by a constellation design can vary significantly when evaluated against different operating scenarios. This was highlighted in Figure 4-5 when the set of non-dominated reconfigurable designs along

the Pareto front were re-evaluated against a different set of 24 target decks, with the worst-case performance for some designs dropping by as much as 25% from the initial score predicted as part of the optimization output.

As described in the opening of this chapter, the performance of a reconfigurable constellation against a range of possible mission scenarios must be considered as a statistical distribution, especially as one of the motivations for using reconfigurability is the ability to achieve consistent performance under unpredictable operating conditions. Unless the exact locations where imaging is desired are known ahead of time, constellation performance must be analyzed in statistical terms, and is not guaranteed to reach a certain value under all mission conditions. Due to the amount of variation in performance achieved under different circumstances, the choice was made to switch from an iso-performance analysis of cost savings to an iso-cost analysis of the performance differential between reconfigurable and static constellation designs. This change is additionally justified by the fact that constellation costs are set by the initial choice of design and do not vary between different evaluation scenarios. For example, even though different amounts of propellant are used in responding to different scenarios, the propulsion system costs are set by the propellant budget that was selected at the point the design was finalized, and therefore these do not vary when repeatedly evaluating the same design against a range of operating conditions.

Figure 4-11 shows a Pareto chart with cost and performance plotted for non-dominated sets of reconfigurable (in blue) and static (in red) constellation designs. The Pareto fronts in this figure were created by combining the sets of non-dominated designs from four separate optimization runs carried out to generate reconfigurable designs, and another four optimization runs carried out to generate static designs. Each of these optimization runs was carried out using a different set of 24 target decks to represent a range of unpredictable operating scenarios for the constellation. Designs that were previously non-dominated but now became dominated due to the combination of multiple sets were discarded, leaving only the best-optimized

candidates from all four reconfigurable runs as a single super-set of reconfigurable constellation designs, and the best-optimized candidates from all four static runs as a single super-set of static constellation designs.

These super-sets of reconfigurable and static designs were then compared along the cost axis. A script was written to evaluate the relative costs of constellations from each set and find iso-cost pairings of reconfigurable and static designs. Any designs that did not have a pairing from the opposite set within 3% of the same cost were discarded. Finally, any designs that showed up as duplicate pairings (e.g. three adjacent static designs might all most closely match on cost with the same reconfigurable design) were matched with the closest candidate from the opposite set, and the other less-close pairings were eliminated.

Figure 4-12 shows a Pareto chart comparing cost and performance for these iso-cost sets of reconfigurable and static designs. Each reconfigurable design (shown in blue) in the figure has a corresponding iso-cost static design pairing (shown in red), with a cost difference no larger than 3% (and in many cases lower than 0.1%, with the closest pair only separated in cost by 0.005%). Between the two Pareto fronts, the difference in performance (ΔP) is highlighted at three points, showing how the amount of performance to be gained from reconfigurability varies throughout the design space.

The metric of ΔP is directly equivalent to Legge's Value of Reconfigurability as a single-axis comparison, but rather than focusing on the cost savings to be made by switching between iso-performance static and reconfigurable designs, this metric focuses on the amount of performance that could be gained by adopting reconfigurability at iso-cost points in the design space. Instead of assuming that performance stays constant across all possible scenarios, ΔP allows for statistical analysis to be factored into the comparison of distributions of performance from static and reconfigurable designs. This iso-cost metric is also consistent with the assumption made in

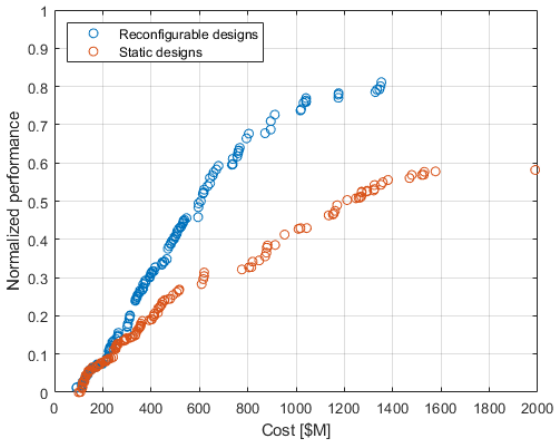


Figure 4-11: Pareto chart showing cost and performance for two distinct Pareto fronts, for a non-dominated set of reconfigurable constellation designs (plotted in blue) and a non-dominated set of static constellation designs (plotted in red)

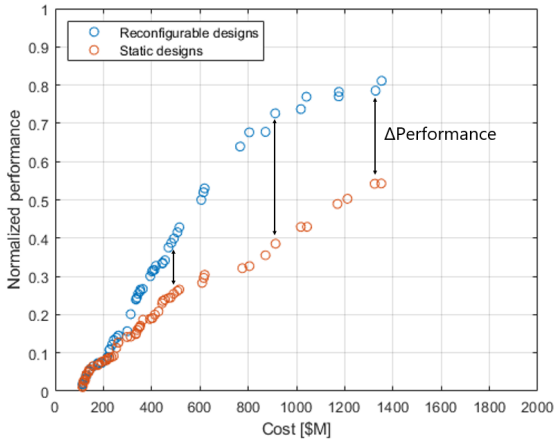


Figure 4-12: Pareto chart for an iso-cost subset of the two sets of designs shown in Figure 4-11, with each reconfigurable design (plotted in blue) paired with a static design (plotted in red) with a cost difference of less than ±3%

the ReCon codebase that constellation cost is fixed once design parameters have been fixed. In fact, as long as cost variation between scenarios is limited to less than 25%, assessing the relative value of reconfigurable and static designs along an iso-cost axis will exhibit greater reliability and lower variation along the 'fixed' axis than the distribution of performance values seen when constellation performance is re-evaluated against a new set of target decks, as was highlighted in Figure 4-5.

An area of interest for future work is to add the simulation of operating costs for different mission scenarios to the ReCon cost model, as these are not currently included and may not be identical across different mission lifetimes, even when re-evaluating the same constellation design. Adding operating costs to the model will reduce uncertainty in the total system cost and enable a statistical comparison of the distributions of cost and performance scores simulated for different scenarios. This would allow for more thorough evaluation of the relative benefits of scoring constellations on iso-performance and iso-cost metrics.

A useful comparison in evaluating operating costs may be to compare the perfor-

mance of a constellation against target decks that feature the minimum number of 14 events of interest over a 5-year period against decks that feature the maximum number of 26 events of interest in the same period of time, as these represent the extremes of the temporal spacing between desired coverage that are modelled in the ReCon codebase. A thorough characterization of the uncertainties in cost modelling would allow for assessment of the dispersion of scores about the mean cost. This could result in an additional useful scoring metric for non-dominated constellation designs, as some candidates may have more consistent costs across many different scenarios while others may vary significantly if the operating costs fluctuate by a larger quantity between target decks.

4.2 Calculating statistical dispersion of performance

The statistical dispersion around the average score for a metric may be represented in various ways. These include standard deviation, variance, full range, interquartile range, or other percentile values designed to illustrate the spread of the distribution. Even if a particular metric is preferred for evaluating the statistical distribution of the data, there are many possible ways in which this may then be presented usefully for decision makers.

Figure 4-13a shows a recreation of the Pareto chart previously shown in Figure 4-12, with iso-cost pairs of reconfigurable and static designs generated by a combination of eight optimization runs. A sample pair of designs is highlighted, and these are re-evaluated against 500 different scenarios. The distribution of the re-evaluated performance scores is shown on the right in Figure 4-13b for the paired designs, with the range of reconfigurable performances shown in blue and the range of static performances shown in red.

The selected reconfigurable design features 11 satellites in 11 planes, with an orbital inclination of 86.6°. This constellation has a total cost of $766.75M, with

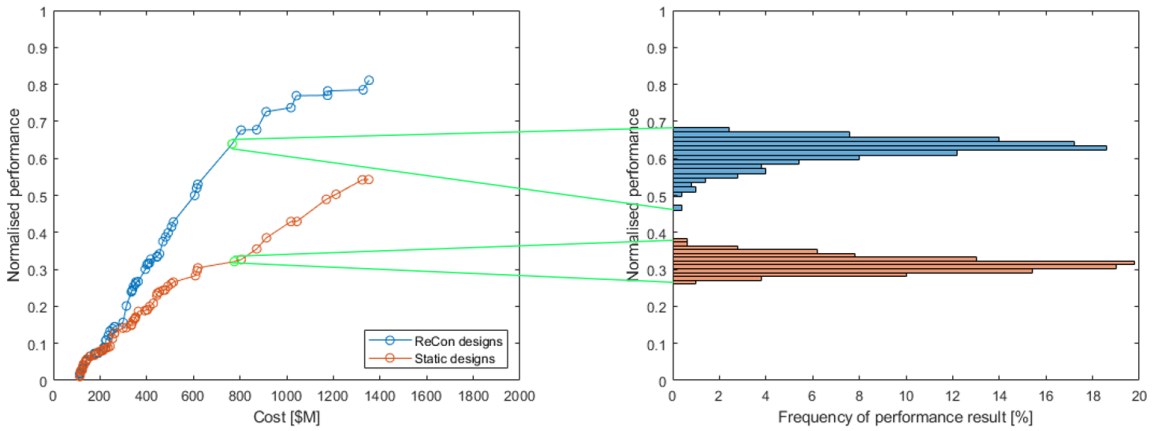


Figure 4-13: a) Left-hand plot shows a recreation of the Pareto chart from Figure 4-12, with iso-cost pairs of reconfigurable (in blue) and static (in red) constellation designs
b) Right-hand plot takes a sample iso-cost pairing with a 1.14% cost difference and shows histograms of the distribution of performance scores from 500 re-evaluations of these two designs

an initial performance score from the optimization of 0.6394 (equating to 63.94% of the ideally-desired persistence). When re-evaluated against 500 different scenarios as shown in Figure 4-13b (in blue), the mean performance of the design was found to be 0.6205: a drop of 2.9% compared to that predicted by the optimization. The re-evaluated scores varied from a minimum of 0.4662 to a maximum of 0.6829, giving a range of 0.2167. The standard deviation for the performance distribution was 0.0352, and the median performance was 0.6285.

The selected static design features 16 satellites in 16 planes, with an orbital inclination of 121.4°. This constellation has a total cost of $775.57M, with an initial performance score from the optimization of 0.3212 (equating to 32.12% of the ideally-desired persistence). When re-evaluated against 500 different scenarios as shown in Figure 4-13b (in red), the mean performance of the design was found to be 0.3136, a drop of 2.4% compared to that predicted by the optimization. The re-evaluated scores varied from a minimum of 0.2635 to a maximum of 0.3841, giving a range of 0.1206. The standard deviation for the performance distribution was 0.0213, and the median performance was 0.3126.

The standard error of the mean, $\sigma_{\bar{x}}$, may be calculated in order to place confidence intervals on the data obtained from these distributions of n samples. As the actual standard deviation, σ, is unknown, the standard error of the mean may be approximated using the sample standard deviation, σ_x, as follows:

$$\sigma_{\bar{x}} \approx \frac{\sigma_x}{\sqrt{n}} \tag{4.1}$$

Using Equation 4.1 results in a standard error of mean performance of 1.57×10^{-3} for the reconfigurable design and 9.52×10^{-4} for the iso-cost static design. Confidence intervals can then be estimated using the sample mean $\bar{x}$ and a standard statistical table of z-values, assuming a normal distribution of performance values. These z-values are often referred to using the shorthand of the '68-95-99.7 rule', where a z-value of 1 corresponds to an interval estimate covering 68.3% of a normal distribution, a z-value of 2 corresponds to an interval estimate covering 95.4% of a normal distribution, and a z-value of 3 corresponds to an interval estimate covering 99.7% of a normal distribution. A z-value of 1.96 corresponds to a confidence interval of 95% for a normally-distributed parameter, meaning that upper and lower bounds for a confidence interval of 95% ($CI_{0.95}$) on the mean are given by the following equation:

$$CI_{0.95} = \bar{x} \pm 1.96\sigma_{\bar{x}} \tag{4.2}$$

For the iso-cost pair of designs described above, if a normal distribution of performance is assumed and the approximated standard error of mean performance falls within an acceptable error range, the mean performance can be calculated and could be presented with a confidence interval in the following ways:

- 95% confidence that the mean performance of the reconfigurable design falls within the interval 0.6175–0.6236, or equivalently, 95% confidence that the mean performance of the reconfigurable design is 0.6205 ± 0.00309
- 95% confidence that the mean performance of the static design falls within the

interval 0.3118–0.3155, or equivalently, 95% confidence that the mean performance of the static design is 0.3136 ± 0.00187

It is possible to assess whether the assumption of a normal distribution is reasonable for these two performance distributions, based on the sample size of conducting re-evaluations against 500 scenarios. The Shapiro-Wilk test is a statistical test of normality, which assesses the probability of whether a sample originated from a normally-distributed population to a desired level of significance. Monte Carlo simulations are used to calculate the expected values for the order statistics of independent random samples from a normal distribution, with the test statistic used to compare the probability of the sampled data occurring from such a distribution. The significance level is usually set at a p-value of 0.05 for the null hypothesis that the data is normally distributed, meaning that there is a 5% probability of obtaining sampling results at least as extreme as the observed results if the data is actually from a normal distribution. If there is less than a 5% chance of obtaining the observed results from a normal distribution, the hypothesis is rejected, and the original distribution cannot be assumed to be normal.[97] This test was implemented in MATLAB and used to assess the normality of the performance distributions obtained by repeatedly re-evaluating different designs.

When the normality of the performance distributions was evaluated for the sample iso-cost pairing considered here, both distributions scored a p-value of <0.01 on the Shapiro-Wilk test. This means that there is less than a 1% probability of observing these results by chance from a normal distribution, and so neither set of performance scores can be assumed to be normally distributed. Examining the histograms shown in Figure 4-13b, it may be observed that the reconfigurable performance distribution appears to be visibly skewed towards higher performance values, and the static performance distribution appears to be visibly skewed towards lower performance values, and so the rejection of the null hypothesis of normality is unsurprising for both of these cases.

Fortunately, there are alternative methods of evaluating standard error for various statistical parameters which are not dependent on assumptions of normality. Bootstrapping is a statistical resampling technique which can be used to calculate standard error or confidence intervals for almost any desired test statistic, without making assumptions about the underlying distribution. The bootstrap method involves taking the existing set of sample data and resampling this with replacement to create additional, identically-sized sample sets from the same approximated distribution. This is combined with a Monte Carlo algorithm to independently and randomly resample the original data set many times, calculate the test statistic of interest for each new data set, and then find the sample standard deviation of the test statistic. As the number of resampled data sets increases, the sample standard deviation of the test statistic approaches the standard error for that statistic. This procedure may be summarized as calculating an estimation of the standard deviation using Monte Carlo sampling.[98, 99, 100, 101]

Bootstrapping is considered to be the benchmark against which other approximation formulae for standard error and confidence intervals are measured. An additional benefit of bootstrapping is that it can be applied to any test statistic of interest. Standard error of the mean and confidence intervals are simple enough to calculate with the assumption of normality, as described by Equations 4.1 and 4.2, but this result may now be compared with the non-parametric bootstrapped estimate to see the effect on accuracy of the assumptions made in this calculation. Other test statistics do not have similarly simple standard error formulae, such as median or percentile values of interest, but bootstrapping allows for reliable estimates of standard error to be calculated just as easily for these metrics. This enables the application of confidence intervals to metrics other than the mean performance.[98]

Bootstrapping was applied to the reconfigurable and static designs chosen as a sample iso-cost pairing and described in detail above. The 500 performance scores that were generated by re-evaluating each design against 500 target decks were resam-

pled using replacement with a Monte Carlo algorithm, to create 1000 resampled data sets (each containing 500 scores) for the bootstrap analysis. The standard deviation of the 1000 means calculated from these data sets was found to be 1.52×10^{-3} for the reconfigurable design, or about 3.2% lower than the standard error of the mean calculated using Equation 4.1. For the static design, the standard deviation of the 1000 means from resampled data sets was found to be 9.75×10^{-4}, or about 2.4% higher than the standard error of the mean calculated using Equation 4.1. The conclusion drawn from comparing these values to the benchmark set by the bootstrapped values is that the assumptions made in using Equation 4.1 were an acceptable approximation, resulting in only a small error rate when compared to the bootstrapped standard error of the mean. However, if especially precise estimates are required for a particular application, bootstrapping may be used in place of this approximation.

Confidence intervals may be applied to the mean using the bootstrapped standard error values. These confidence intervals will be applicable to the distribution of the sample means due to a combination of the Central Limit Theorem with the bootstrap method. The Central Limit Theorem states that when sample sets are obtained from a population by randomly sampling many times, as the number of sample sets obtained is increased, the distribution of the sample means will approximate more and more closely to a normal distribution. This theorem holds true even if the distribution from which the data sets are sampled is not normally distributed.[102, 103]

The Central Limit Theorem is an extremely important concept in probability, as it enables the use of statistical methods based on normality to be applied to calculations involving the mean and standard deviation of non-normal distributions.[102, 103] In order to validate the applicability of the Central Limit Theorem with the bootstrapped performance data sets, the normality of the distribution of sample means for the reconfigurable and static designs was checked using the Shapiro-Wilk test. Even though the original re-evaluated performance data sets were not normally distributed and failed the Shapiro-Wilk test of normality, it was found that the distribution of

sample means from the bootstrapped data sets passes the Shapiro-Wilk test. The resampled data sets are therefore confirmed to approximate a normal distribution, validating the use of the Central Limit Theorem for applying confidence intervals to the mean performance.

The bootstrapped standard error of the mean was used to recalculate 95% confidence intervals for the mean performance so that these could be compared to the previously calculated values. This resulted in the following results:

- 95% confidence that the mean performance of the reconfigurable design falls within the interval 0.6176–0.6235, or equivalently, 95% confidence that the mean performance of the reconfigurable design is 0.6206 ± 0.00299
- 95% confidence that the mean performance of the static design falls within the interval 0.3117–0.3155, or equivalently, 95% confidence that the mean performance of the static design is 0.3136 ± 0.00191

These results fall within 3.2% and 2.4% of the previously calculated confidence brackets for the reconfigurable and static designs respectively, as would be expected based on the amount of change in the calculated standard error values. These low error rates are acceptable for the present applications of the performance estimation, and validate the assumptions made in the original calculation of the standard error of mean performance.

After validating the original calculation of standard error for the mean performance values and the 95% confidence intervals for mean performance that were estimated based on these figures, these confidence intervals were plotted onto a Pareto chart, shown in Figure 4-14. These intervals are plotted as areas covering the full width of the confidence interval, with a solid color area filled in between the upper and lower bounds of this interval. It can be seen from this figure that the 95% confidence interval for mean performance is very narrow compared to the overall performance score, varying by around $\pm 0.5 - 0.6\%$ of the mean. This interval region

resembles a single thick line within the figure, due to the narrowness of the 95% confidence interval that is achieved. There is some overlap in the mean performance scores for reconfigurable and static designs in the low-cost area of the design space, up to a constellation cost of approximately $220M. Above this region, there is a significant amount of separation for the performance scores of iso-cost reconfigurable and static designs, and especially above constellation costs of $300M, this gap becomes extremely large.

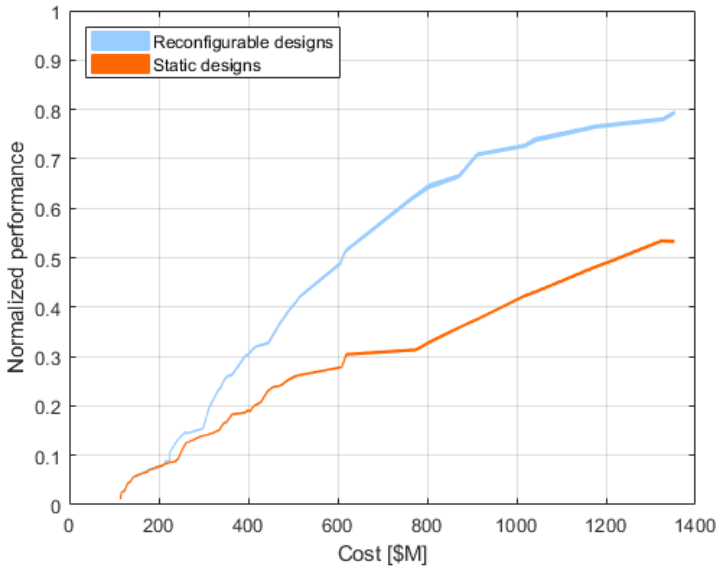


Figure 4-14: 95% confidence interval for the mean performance of iso-cost pairs of reconfigurable and static designs, plotted onto a Pareto chart showing cost against normalized performance

The resampled data sets were then used to calculate the bootstrapped standard error of the median for the iso-cost pair of designs. The standard error of median performance was found to be 1.26×10^{-3} for the reconfigurable design and 1.35×10^{-3} for the static design. Although a widely-cited formula for estimating the standard error of the median states that it is generally equivalent to around 1.253 times the standard error of the mean, the accuracy of this approximation is heavily reliant on the assumption of a normal distribution in the data set under consideration.[104] The amount of kurtosis in a distribution is a parameter that describes the degree to

which data is clustered in the peak(s) or tail(s); a mesokurtic (moderate breadth and medium peak height) distribution with a kurtosis value of 3 is a normal distribution, considered to set a baseline level compared to other distributions.[105] For leptokurtic distributions (which possess a higher kurtosis value) where the data is more heavily concentrated about the mean, the median is likely to have lower error than the mean, due to being less affected by sampling variance.[104]

Examining the heights of the peaks in Figure 4-13b, it appears that both performance distributions shown are leptokurtic. When the value of kurtosis was calculated for these performance data sets, it was found to be 4.66 for the reconfigurable design (significantly leptokurtic) and 3.10 for the static design (mildly leptokurtic; not far from a normal distribution). This correlates with what would be expected of the relative values of the standard error of mean and median performance for the two distributions; for the reconfigurable design, the standard error of the median was found to be 17.3% smaller compared to the standard error of the mean, suggesting that this distribution is significantly leptokurtic enough that the median is a more reliable measure of central tendency than the mean, therefore having a lower standard error for this case. For the static design, the standard error of the median was found to be 38.0% larger compared to the standard error of the mean, suggesting that this distribution is close enough in kurtosis to a normal distribution (i.e. more mesokurtic in nature) that the mean is a more reliable metric of central tendency than the median for this case.

Although the standard error of the median may be bootstrapped as described above, the distribution of the sampling medians from the resampled data sets is not normally distributed. The Central Limit Theorem does not apply to the median of the sampling distribution, and so alternative methods of applying confidence intervals must be adopted for this metric. Literature on estimating the standard error and confidence intervals of percentiles[106, 107] supplied the following equation for calculating the upper and lower bounds on a 95% confidence interval for the quantile

q, using n samples:

$$CI_{0.95} = nq \pm 1.96\sqrt{nq(1-q)} \tag{4.3}$$

The output of Equation 4.3 is the sample numbers within the sorted original data set that correspond to the limits of the confidence interval. For example, calculating the 95% confidence interval of the median performance based on the set of 500 re-evaluated performance scores would set the inputs for Equation Z as $n = 500$ and $q = 0.5$. This gives a result of 228.1 and 271.9, meaning that the 95% confidence interval for the median is estimated by the range set by taking the 228th and 272nd performance results from the sorted data set of 500 performances.

Equation 4.3 uses the same z-value of 1.96 for a 95% interval as would be applied with the assumption of normality, and this is justified in the literature as a reasonable approximation for the median of non-normal distributions. The error introduced by this approximation remains small, as long as the estimation is made for a large number of samples n and not applied to "extreme percentiles" of the distribution.[106]

Applying Equation 4.3 to the performance medians for the iso-cost design pair resulted in the following confidence intervals:

- 95% confidence that the median performance of the reconfigurable design falls within the interval 0.6251 – 0.6306
- 95% confidence that the median performance of the static design falls within the interval 0.3096 – 0.3144

Rather than automatically assuming the validity of the approximation supplied by Equation 4.3, an additional validation was carried out using bootstrapping to provide another standard for benchmarking. The median performance was calculated for each of 1000 resampled data sets, and these 1000 medians were sorted in numerical order. The bounds of a $(1-\alpha)$ confidence interval using n bootstrapped samples are calculated as follows:

$$CI_{1-\alpha} = \frac{\alpha}{2}n, \left(1 - \frac{\alpha}{2}\right)n \tag{4.4}$$

Using Equation 4.4, the bounds of a 95% confidence interval using 1000 bootstrapped medians were found to be the 26th and 975th samples in the list of sorted medians.[103] Applying Equation 4.4 to the bootstrapped medians for the iso-cost design pair resulted in the following confidence intervals:

- 95% confidence that the median performance of the reconfigurable design falls within the interval 0.6254 – 0.6306
- 95% confidence that the median performance of the static design falls within the interval 0.3097 – 0.3144

Comparing the bootstrapped 95% confidence intervals for the median with the estimated confidence intervals using techniques from literature,[106, 107] less than 0.05% discrepancy was found in any of the 95% confidence interval bounds. The conclusion drawn from the extremely small observed error level is that the assumptions made in approximating confidence intervals for the median using Equation 4.3 are justified, resulting in a highly accurate estimate.

After validating the estimated calculation of 95% confidence intervals for the median performance values against the bootstrapped confidence intervals, the estimated intervals calculated using Equation 4.3 were plotted onto the Pareto chart shown in Figure 4-15. Comparing this to Figure 4-14, it was found that the median performance is very slightly higher than the mean performance values, with a slightly broader 95% confidence for the median compared to the mean. This can be seen in the figure as a slightly thicker confidence region for each set of designs, though still strongly resembling a single line rather than a broad interval. This is due to the narrowness of the achieved 95% confidence interval, which varies by about $\pm 0.4 - 0.8\%$ of the median in Figure 4-15. This means that the median performance can be very reliably predicted to fall within a narrow range. Once again, there is some overlap in the confidence intervals for reconfigurable and static designs up to a cost of around $220M, though the scores begin to separate above this region and show significant differences for all designs costing $300M or higher.

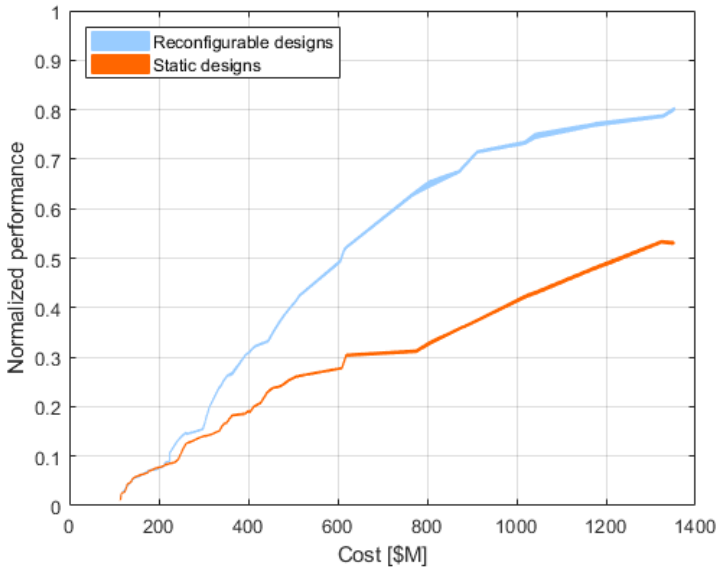


Figure 4-15: 95% confidence interval for the median performance of iso-cost pairs of reconfigurable and static designs, plotted onto a Pareto chart showing cost against normalized performance

Percentiles may also be calculated for these performance distributions, to supply a broader impression of the distribution of performance scores achieved by the designs against a range of scenarios. 25th percentile scores were calculated for the iso-cost design pair based on the original set of 500 re-evaluated performance scores, giving a result of 0.6043 for the reconfigurable design and 0.2984 for the static design. 75th percentile scores were calculated as 0.6451 for the reconfigurable design and 0.3267 for the static design.

Confidence intervals may also be placed on these percentile scores, and these were calculated using both the estimation given by Equation 4.3 and the bootstrapped standard given by Equation 4.4, resulting in the values given in Tables 4.2 (for the reconfigurable design) and 4.3 (for the static design). The amount of error between the two techniques is again extremely small, with less than 0.1% discrepancy between any of the bounds calculated.

After validating the estimated calculation of 95% confidence intervals (using Equa-

Table 4.2: Comparison of estimated and bootstrapped 95% confidence intervals for 25th and 75th percentile reconfigurable design performance

Percentile of performance distribution	**Estimated 95% confidence interval**	**Bootstrapped 95% confidence interval**
25	0.5971 – 0.6090	0.5975 – 0.6092
75	0.6416 – 0.6478	0.6417 – 0.6480

Table 4.3: Comparison of estimated and bootstrapped 95% confidence intervals for 25th and 75th percentile static design performance

Percentile of performance distribution	**Estimated 95% confidence interval**	**Bootstrapped 95% confidence interval**
25	0.2953 – 0.3012	0.2955 – 0.3013
75	0.3237 – 0.3292	0.3234 – 0.3292

tion 4.3) for the 25th and 75th percentile performance values against the bootstrapped confidence intervals (calculated using Equation 4.4), the estimated confidence intervals for all iso-cost design pairs were plotted on Pareto charts. The 95% confidence interval for the 25th percentile performance values (i.e. halfway between the minimum score and the median) is shown in Figure 4-16, and the 95% confidence interval for the 75th percentile performance values (i.e. halfway between the median and the maximum score) is shown in Figure 4-17. Once again, these are plotted as full confidence intervals where a solid color region is filled in between the upper and lower limits of the 95% confidence region, but it may be observed that these intervals are narrow enough in width to resemble single thick lines within these plots.

The breadth of the 75th percentile confidence interval in Figure 4-17 is comparable to the median confidence interval shown in Figure 4-15 for both reconfigurable and static designs. The breadth of the 25th percentile confidence interval in Figure 4-16 is comparable to the width of the median confidence interval for the static curve, but slightly wider for the reconfigurable curve. This suggests a greater variation within the lower-performance group of scores and a narrower range in the higher-performance score quantiles for the reconfigurable designs, which corresponds to the broader tail in the lower-performance half of the reconfigurable performance distribution shown

in Figure 4-13b.

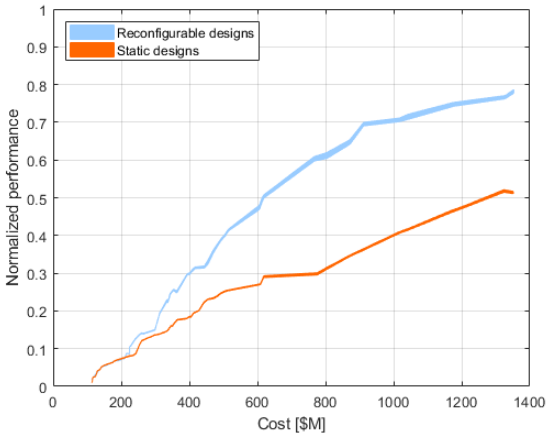


Figure 4-16: 95% confidence interval for the 25th percentile performance of iso-cost pairs of reconfigurable and static designs, plotted onto a Pareto chart showing cost against normalized performance

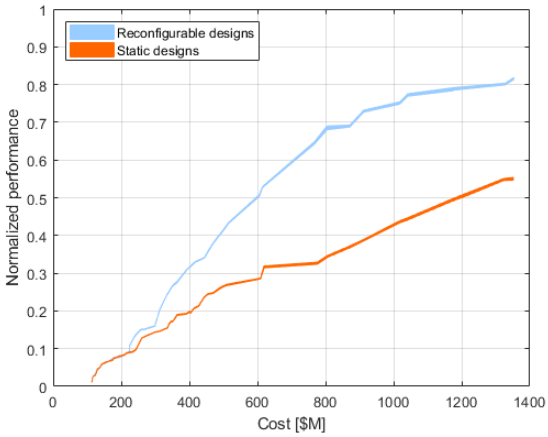


Figure 4-17: 95% confidence interval for the 75th percentile performance of iso-cost pairs of reconfigurable and static designs, plotted onto a Pareto chart showing cost against normalized performance

A variety of estimation techniques and bootstrapped metrics were compared in order to establish confidence intervals for measures of statistical dispersion in the performance distributions for iso-cost reconfigurable and static constellation designs. These metrics of dispersion include the mean, median, 25th percentile and 75th percentile of performance. 95% confidence intervals were placed upon these metrics to assess the amount of variation in performance when the constellation designs are evaluated against a range of operating scenarios, varying about the estimated metrics by less than $\pm 1\%$. The width of these confidence levels establish that the distribution of performance scores can be predicted with high reliability over a range of target decks.

4.3 Calculating confidence that ReCon outperforms static designs

The confidence intervals shown in the previous section establish a narrow range within which some of the metrics of central tendency (such as mean and median performance)

are expected to fall. 95% confidence intervals for the 25th percentile and 75th percentile of performance are also calculated in order to provide more information about the distribution of performances that are achieved in different scenarios, and how far these may differ from the median value for different constellation designs. However, all of these confidence intervals only place a range upon a single measure of constellation performance, representing some part of the distribution.

Figure 4-18 shows the full range of performances achieved by iso-cost pairs of reconfigurable (in blue) and static (in orange) designs when re-evaluated against 500 different operating scenarios. Although the reconfigurable designs consistently outperform iso-cost static designs with a constellation cost of around $400M and above, there is some overlap in the performance distributions below this point, and almost total overlap in performance for the lowest cost constellation designs (below $220M). Figure 4-19 shows a magnified view of this region, highlighting the $100M–400M region of the design space to show the overlapping range of performance more clearly. From the lowest cost pair at around $113M to around $220M, there is almost total overlap in the performance ranges for reconfigurable and static designs, and then decreasing amounts of overlap up to around $380M.

The distinct cost regions within Figures 4-18 and 4-19 may be compared to the design variables generated for constellations within these regions. Where there is almost total overlap in performance for a constellation cost below $220M, there is also almost total overlap in design variables, suggesting little difference in the architectures generated at this cost point. The ΔV budget for reconfiguration at this cost point is also extremely inconsistent, varying between 0 and 290 m/s. Where the two distributions begin to separate in performance between a constellation cost of $220M to $400M but still exhibit some overlap, the largest change in design variables is that the ΔV budget for reconfiguration increases to a range of 170 to 450 m/s. Above a constellation cost of $400M, the number of satellites increases steadily for reconfigurable designs, the number of satellites per plane drops to 1, the optimiza-

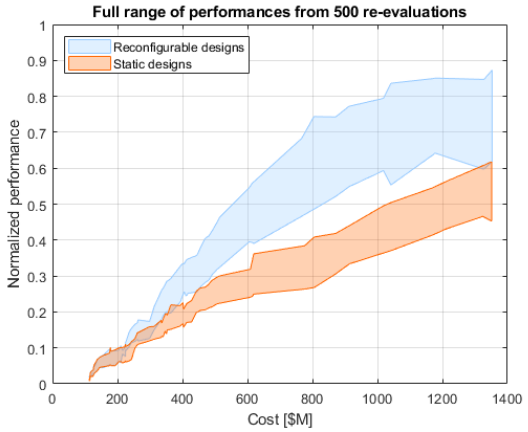


Figure 4-18: A Pareto chart of iso-cost sets of reconfigurable and static designs, showing the full range of performance scores from 500 re-evaluations

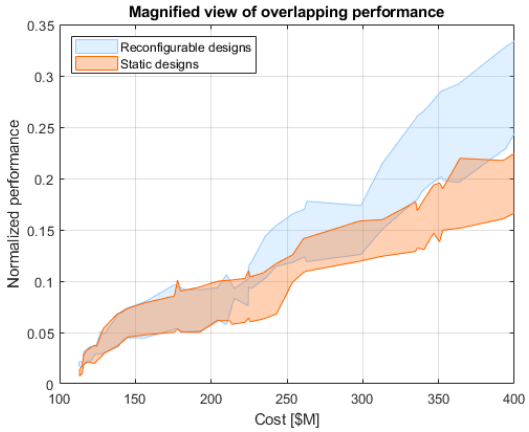


Figure 4-19: A magnified view of the low-cost region in Figure 4-18 where performance distributions overlap for reconfigurable and static designs

tion consistently adopts a near-polar orbit with an inclination of around 87°, and the GOM altitude increases to approximately 540 km.

The strongest correlation between improved performance and a single design variable is still that of the ΔV budget for reconfiguration. Reconfigurable designs consistently outperform static designs once the ΔV_{recon} budget exceeds 200 m/s. This makes sense as the largest distinction between the reconfigurable and static designs generated is that one set of constellations can maneuver while the other cannot. The minimum value of 200 m/s to achieve consistent performance may be explained by the fact that below this budget, reconfigurable constellations cannot consistently maneuver to respond to every event of interest over the mission lifetime, and so this establishes a minimum standard at which responsive reconfigurability may be achieved.

The confidence intervals described in Section 4.2 help to establish specific and predictable ranges for average constellation performance. However, the particular numerical value of mean performance is of limited use in establishing the benefits of reconfigurability over static designs. In Figure 4-19, a reconfigurable constellation costing $400M achieves a range of normalized performance values between 0.24 –

0.33, meaning that it achieves 24 to 33% of the ideally desired level of persistence; this is not a particularly high-performing constellation when compared to the ideal coverage score. A static constellation of the same cost achieves a range of normalized performance values between 0.17 and 0.22, meaning that this design achieves only 17 to 22% of the ideal amount of coverage. Even if the minimum reconfigurable performance of 0.24 happened to occur for the same scenario as the maximum static performance of 0.22, this would equate to an iso-cost improvement in performance of 9% due to reconfigurability.

Having established confidence intervals for the mean and median performance values when averaged over a range of scenarios, it was concluded that a more useful confidence metric in demonstrating the benefits of reconfigurability to satellite operators would be to establish confidence intervals for reconfigurable designs outperforming iso-cost static designs. Rather than establishing confidence in predicting specific performance values, it should be predicted at what point in the design space reconfigurable designs can consistently outperform iso-cost static alternatives. The first step was to establish confidence that reconfigurable designs outperform static designs by any amount. Additional utility could be gained by then extending this analysis to find the confidence level that reconfigurable designs outperform iso-cost static designs by a set margin, for example by 10 or 20% higher performance.

4.3.1 Confidence metrics for comparing ReCon and static designs

Several methods of calculating confidence were considered for comparing the performance distributions of reconfigurable and static designs at the same cost points. Confidence levels in the performance difference observed between iso-cost design pairs was calculated using five different methods in order to enable comparisons between the results and cross-validate these techniques to assess whether the assumptions made were reasonable. Although confidence intervals were calculated within Section

4.2, these are only appropriate for use with single-value parameters, and other types of interval estimation must be carried out for parameter ranges. It is appropriate to place a **confidence level** upon an interval, but not to use a **confidence interval** upon the population of performances. To explain the distinction between interval types, some definitions are included here in order to clarify several different terms within the field of interval estimation.

A confidence interval provides bounds on a single chosen population parameter, such as the mean, median, or a quantile point in the distribution, as calculated in the previous section. The size of a confidence interval is entirely due to sampling error, rather than actual variation in the population. If the sample size approaches the population size, the population parameter will approach its true value. At this point the sampling error approaches zero and the width of the confidence interval will also approach zero. A confidence interval is not an appropriate estimate of the full range of future sampling.

A prediction interval provides bounds for a single future sample from the population. It predicts the distribution of individual future samples from the population, based on the past sample observations. These are also sometimes referred to as 'predictive confidence intervals', specifying bounds within which a specific proportion of the population will fall on average, based on knowledge of the distribution of past sampling.

A tolerance interval provides bounds for the data range that includes a specific chosen proportion of the population. This is equivalent to constructing a set of prediction intervals: a tolerance interval places bounds on the whole population, while a prediction interval places bounds only upon a single future sample from this population. The size of a tolerance interval is partly due to sampling error and partly due to the actual variation in the population being sampled. If the sample size approaches the population size, the width of the tolerance interval will approach the probability

interval of the underlying population as the sampling error approaches zero. The bounds of a tolerance interval specify not only a range within which a specific proportion of the population will fall, but also a confidence level upon that range (not merely 'on average' as in the case of a prediction interval).

In the case of comparing the performance of reconfigurable and static designs, a tolerance interval may be calculated with the parameters p for the desired proportion of the population and c for a desired confidence level. For example, a tolerance interval might be constructed to include 50% of the population with a confidence of 95%. A lower tolerance limit refers to the confidence limit at the 0th percentile of the population, and an upper tolerance limit refers to the confidence limit at the 100th percentile of the population, effectively placing bounds on the minimum and maximum population limits at the desired level of confidence.

4.3.1.1 Independent hypothesis test

The first method of calculating a confidence metric for constellation performance is an independent hypothesis test, assuming that the distributions of reconfigurable and static designs are independent, i.e. the paired performances are not correlated across target decks. This technique involves comparing the overlap between the two distributions of reconfigurable and static design performances, and quantifying the confidence level that reconfigurable performance is higher than static. This overlap in distributions can be thought of as the distribution of ΔP.

Figure 4-20 shows two histograms of the distributions of normalized performance for a pair of iso-cost designs, with scores for the reconfigurable design shown in blue and scores for the static design shown in orange. A different sample pair of iso-cost designs is shown for this example, in order to highlight a pair of designs that have some overlap in the performance scores achieved. The reconfigurable constellation design has a cost of \$253.8M and the static design has a cost of \$253.7M, with a difference

in cost of 0.05% between the two designs. The performance distributions shown in Figure 4-20 are the results from re-evaluating the two designs against 500 different target locations. The area of overlap between the two distributions is highlighted by a red box, and this is the area that will be characterized using the independent hypothesis test.

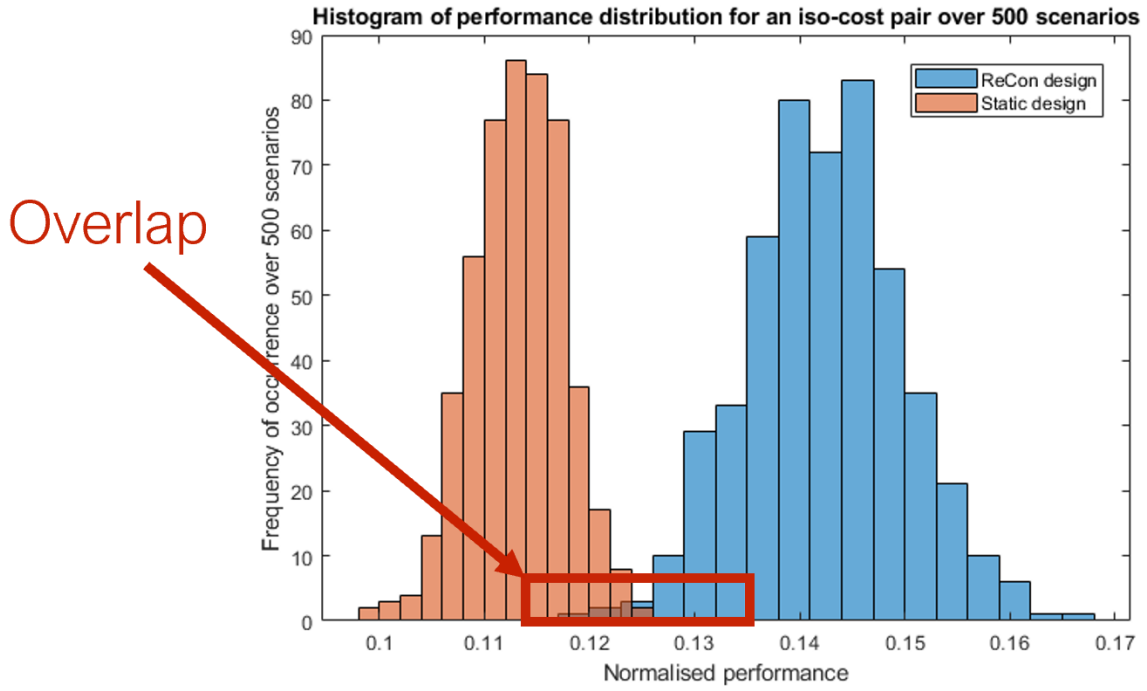


Figure 4-20: Histograms of normalized performance distributions over 500 scenarios for an iso-cost pair of designs, with the overlap in performance scores highlighted in red

The mean of the difference in performance, written here as the expected value of the overlap, $E[overlap]$, is given by subtracting the mean of the static distribution, $E[static]$, from the mean of the reconfigurable distribution, $E[ReCon]$, as follows:

$$E[overlap] = E[ReCon - static] = E[ReCon] - E[static] \tag{4.5}$$

The variance of the distribution for the difference in performance may be calculated from the variances of the two distributions, assuming they are independent of one

another:

$$Var[ReCon - static] = Var[ReCon] + Var[static] \tag{4.6}$$

and the square root of Equation 4.6 may also be used to find the standard deviation of the difference in performance:

$$\sigma_{overlap} = \sqrt{{\sigma_{ReCon}}^2 + {\sigma_{static}}^2} \tag{4.7}$$

After using equations 4.5–4.7 to characterize the distribution of the overlap, a one-tailed hypothesis test is used to determine what proportion of the distribution meets the failure criterion. In this case, the failure criterion is met for scenarios where the static performance exceeds the reconfigurable performance, meaning that ΔP gives a negative result. To estimate what proportion of the distribution falls in this region, the following equation is used:

$$A = \frac{E[overlap] - x_{diff}}{\sigma_{overlap}} \tag{4.8}$$

In Equation 4.8, A is the number of standard deviations away from the mean that the failure boundary is located, and x_{diff} is the desired amount by which the reconfigurable design outperforms the static design. For this calculation $x_{diff} = 0$ was used to find the confidence that reconfigurable designs outperform static designs by any amount, but a different value may be used for calculation of confidence that reconfiguration designs outperform static alternatives by a specific desired amount.

The percentage confidence that reconfigurable designs outperform static is given by the following equation, using the quantity A from Equation 4.8:

$$\text{Probability}(R > S) = \frac{1 + \text{erf}(\frac{A}{\sqrt{2}})}{2} \tag{4.9}$$

The result from Equation 4.9 may also be found by consulting a standard statistical table of z-values, as long as the one-sided failure criterion is factored into the result.

For example, a value of $A = 1.96$ equates to a 95% probability of success for a two-sided failure criterion, but a 97.5% probability of success for a one-sided failure criterion. This is due to the fact that the failure probability is only occurring for one tail of the distribution.

4.3.1.2 Hypothesis test with covariance

The second method of calculating confidence is a hypothesis test factoring in covariance between the two performance distributions, rather than assuming independent performances for reconfigurable and static designs when evaluated against the same target deck. This technique also involves comparing the overlap between two performance distributions for the two iso-cost sets of designs, but in this case the covariance of the distributions is factored into the calculation. A confidence level is placed upon the probability that reconfigurable designs outperform iso-cost static designs, accounting for the correlation coefficient between the two distributions.

For a continuous random variable X, the expected value $E[X]$ (equivalent to the mean) is given by the following equation:

$$E[X] = \int_{-\infty}^{\infty} x f_x(x)\, dx \tag{4.10}$$

and the variance for X is given by:

$$Var[X] = E[X - E[X]^2] = E[X^2] - E[X]^2 \tag{4.11}$$

The covariance between two distributions of variables X and Y is calculated as follows:

$$Covar[X, Y] = E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y] \tag{4.12}$$

If X and Y are independent variables, then:

$$E[XY] = E[X]E[Y] \tag{4.13}$$

so for the case shown in Equation 4.13, the calculation of covariance in Equation 4.12 cancels to zero.

Equation 4.11 may be used to calculate the variance for the overlap of the two distributions, $Var[X-Y]$, as follows:

$$\begin{aligned} Var[X-Y] &= E[(X-Y)^2] - (E[X-Y])^2 \\ &= E[X^2 - 2XY + Y^2] - (E[X]^2 - 2E[X]E[Y] + E[Y]^2) \\ &= E[X^2] - E[X]^2 + E[Y^2] - E[Y]^2 - 2(E[XY] - E[X]E[Y]) \\ &= Var[X] + Var[Y] - 2(Covar[X,Y]) \end{aligned} \tag{4.14}$$

If X and Y are independent variables with zero covariance, the covariance term equals zero and the result from Equation 4.14 is identical to Equation 4.6.

Equation 4.12 is used to calculate the covariance for all iso-cost design pairs. The covariance result for each pair may then be used with Equation 4.14 to calculate new values of variance for each iso-cost pair. In situations where these pairs of designs have extremely low correlation coefficients, the correction factor added to the variance calculation by using Equation 4.14 will simply be extremely tiny. A correlation coefficient of +1 equates to a perfect positive correlation, -1 means a perfect negative correlation, and 0 means there is no correlation between variables. Calculating the correlation coefficients for all 62 iso-cost design pairs, it was found that the performance distributions for 24 pairs (38.7%) were correlated at a p-value of 0.05. The calculated correlation coefficients varied in magnitude from 5.7×10^{-4} up to 0.98, with both positive and negative correlations found in the range of design pairs.

4.3.1.3 Monte Carlo tally of ΔP results

The third calculation of confidence is a Monte Carlo method based on the re-evaluation of each iso-cost pair of designs against 500 different target locations. 62,000 performance scores were produced by evaluating 62 reconfigurable designs and 62 iso-cost

static designs against 500 scenarios. The difference in performance, ΔP, is calculated for each individual scenario, providing a dataset of 500 ΔP results for each of the 62 iso-cost design pairings. This sample set of 31,000 ΔP values is used to calculate a straightforward confidence value for the frequency at which reconfigurable designs outperform iso-cost static designs.

4.3.1.4 Statistical properties of ΔP distribution

The fourth calculation of confidence uses the same set of 31,000 ΔP values as the previous method. Instead of simply tallying the percentage of scenarios where reconfigurable designs outperformed iso-cost static alternatives, the set of sample results is used to estimate properties of the ΔP distribution and model the probability that reconfigurable designs show higher performance. This method is expected to have a high degree of agreement with the previous method, due to the use of the same data set with a slightly different technique.

Figure 4-21 shows two histograms of normalized performance evaluated over 500 scenarios for the same sample pair of iso-cost designs previously used to provide an applicable example for the calculation of various metrics in Section 4.2. Figure 4-22 shows a histogram of ΔP between the reconfigurable and static design, calculated for each of these 500 scenarios to show the distribution. The distribution shown in Figure 4-22 represents the difference between the two distributions shown in Figure 4-21; for each of the 500 scenarios, the static performance was subtracted from the reconfigurable performance achieved against the same target location. This ΔP value represents the amount by which the reconfigurable design outperformed the iso-cost static design. All of the ΔP values shown in Figure 4-22 are positive, meaning that the reconfigurable design outperform the static design in all 500 scenarios for this specific design pair. A negative ΔP value would denote a scenario in which the static design outperformed its reconfigurable alternative.

To clarify the difference in techniques used, the methods described in Sections

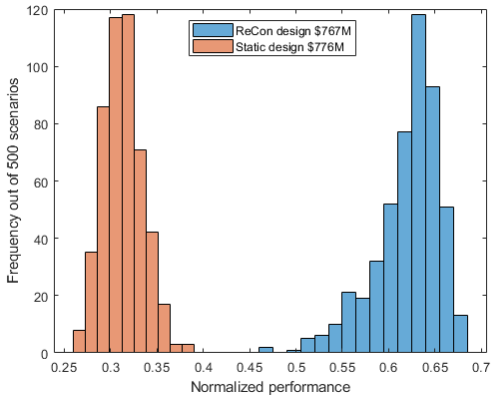


Figure 4-21: Histograms of normalized performance distributions over 500 scenarios for an iso-cost pair of designs

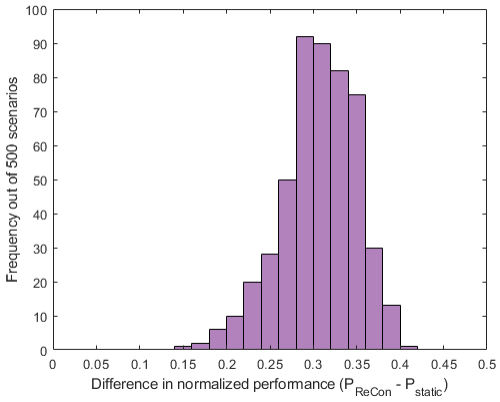


Figure 4-22: Histogram of ΔP evaluated for the same pair of designs and over the same 500 scenarios shown in Figure 4-21

4.3.1.1 and 4.3.1.2 are based on comparing the two distributions shown in Figure 4-21 for all iso-cost sets of designs. The method described in Section 4.3.1.3 involves a straightforward count of how many times the ΔP value shown in Figure 4-22 is above zero. The method used in the present section is to characterize the mean and standard deviation of the distribution shown in Figure 4-22, and calculate the confidence level for ΔP being greater than zero.

It must be acknowledged that the ΔP distribution shown in Figure 4-22 is clearly skewed, and the Shapiro-Wilk test confirms that this distribution of values does not approximate a normal distribution (failing the Shapiro-Wilk test when evaluated for a p-value of 0.05). Although confidence calculations will be made based on the assumption that z-values are applicable to this distribution, the results will be compared to the other techniques used in order to assess the level of error introduced by this approximation.

However, when the improvement in performance is instead evaluated as a percentage improvement of the static constellation performance score ($\frac{\Delta P}{P_{static}} \times 100$) as shown in Figure 4-23, the distribution visibly appears much closer to normality. Re-evaluating this distribution with the Shapiro-Wilk test confirms that it now approx-

imates a normal distribution, passing the test when applied with a p-value of 0.05.

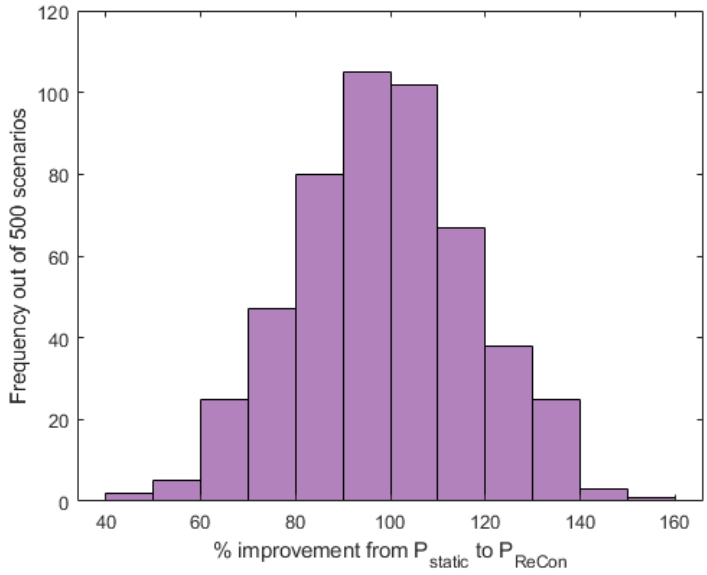


Figure 4-23: Histogram of improvement in performance due to reconfigurability, calculated as percentage improvement from the static performance score

The confidence calculation for this method will be carried out using the ΔP distribution shown in Figure 4-22 and compared to the other confidence metrics to evaluate the level of error resulting from this technique. If the error level is unacceptably high, the percentage improvement metric used in Figure 4-23 will be used as an alternative for calculating confidence in the improved performance as a percentage instead. In this case, the test method will require some modifications to find results in the appropriate units (as a percentage improvement over static performance).

4.3.1.5 Bootstrap method

The fifth and final method of calculating confidence levels is the bootstrap method. The 500 re-evaluated performance scores for each one of the iso-cost pairs are repeatedly resampled with replacement to create 1000 bootstrapped data sets, each consisting of 500 resampled performance scores. These sets are used to calculate confidence levels for tolerance intervals to be placed upon the performance ranges for

each iso-cost design pair.

These confidence levels may be applied as previously discussed in Section 4.2, using Equation 4.4 to calculate and apply the bounds of a 95% confidence level (or any other percentage level) to the desired metric of performance. The first application of this technique is to calculate the level of confidence that reconfigurable designs outperform static by any amount. Bootstrapping the original set of 500 ΔP scores into 1000 resampled data sets (each containing 500 ΔP scores) gives a large number of sample sets from which to calculate the probability that $\Delta P > 0$.

Beyond calculating the probability of a ΔP that is greater than zero, the confidence level may also be calculated for outperforming static designs by a specified proportion (e.g. reconfigurable designs must outperform static designs by at least 10%, or 20%, or X%). Some options for applying confidence levels to other performance metrics that may be useful to satellite operators include the minimum ΔP value for each design pairing; a two-sided quantile that includes a specific proportion of the population, e.g. bounds within which 90% of the ΔP values occur with 95% confidence; or a one-sided quantile that includes a specific proportion of the population, e.g. a lower bound above which 90% of the ΔP values occur with 95% confidence.

4.3.2 Results for confidence level that ReCon outperforms iso-cost static designs

The first two methods used to calculate the confidence level that reconfigurable designs outperform iso-cost static designs were the hypothesis testing techniques (described in detail in Sections 4.3.1.1 and 4.3.1.2). The first of these methods assumes that the performance distributions for reconfigurable and static designs are independent and uncorrelated, and the second method calculates the covariance between the two distributions and factors this into the calculation of confidence.

Figure 4-24 shows the calculated confidence that reconfigurable designs outperform static designs across a range of constellation costs from around $113M to $1352M. The independent hypothesis test method results are shown in blue, and the hypothesis test method results with covariance are shown in red. For every design with a cost greater than $223M, the confidence that ReCon outperforms static is over 97.5% according to the independent method and over 97% according to the covariance method. There is one exception to this result, which is a single iso-cost pair of designs at a cost of $298M. For this pair, there is a confidence level of 93.3% that the reconfigurable design outperforms the iso-cost static alternative according to the independent hypothesis testing method, and 93.1% according to the covariance hypothesis testing method.

Figure 4-25 shows a magnified view of the confidence results for constellation design pairs with costs between $100M and $350M. This figure highlights the variation in the confidence curve in the lower-cost design space, as well as the difference in results between the two confidence calculation methods. The largest differences in the predicted confidence levels are seen for designs with a cost between $140M and $156M, and between $181M and $204M. For designs with a total constellation cost between $127M and $211M, the confidence that ReCon outperforms static is predominantly below 50%, meaning that in this region the static designs are likely to outperform the reconfigurable alternatives. These confidence curves will be compared to later methods of calculating confidence levels in order to assess which of the two hypothesis testing methods gave a more accurate result.

The next two methods used to calculate the confidence level that reconfigurable designs outperform iso-cost static designs were the Monte Carlo tally of ΔP scores and modelling the statistical properties of the ΔP distribution (described in detail in Sections 4.3.1.3 and 4.3.1.4). Figure 4-26 shows the calculated confidence levels from these two methods, with the Monte Carlo method results shown in blue and the

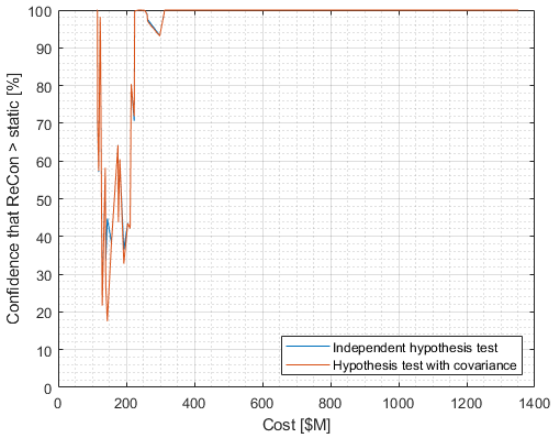


Figure 4-24: Two plots showing the level of confidence that ReCon outperforms static, with independent hypothesis test results plotted in blue and covariance hypothesis test results plotted in red

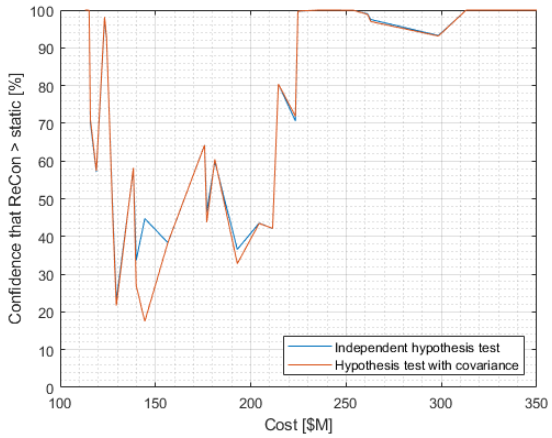


Figure 4-25: A magnified view of the confidence levels from Figure 4-24 showing differences between the two methods for the predicted confidence within the cost region from $100M–350M

statistical distribution results shown in red.

Using the Monte Carlo method, the confidence level that ReCon outperforms static is over 97.6% for every design with a cost greater than $223M. Using the statistical distribution method, the confidence that ReCon outperforms static is over 97% for every design with a cost greater than $223M. These are both very similar results to the previous predictions of 97.5% and 97% from the hypothesis testing methods. Once again, there is a single exception to this result: the same single design pair at $298M, this time with ReCon predicted to outperform static with 93.1% confidence for the statistical distribution method and 93.6% for the Monte Carlo method, compared to 93.3% and 93.1% from the previous hypothesis testing methods.

Figure 4-27 shows a magnified view of the lower-cost design region from Figure 4-26, highlighting the area with the lowest confidence level in reconfigurable designs outperforming static alternatives. The confidence levels calculated for ReCon outperforming static are in greater agreement between these two methods than between the two hypothesis testing methods shown in Figures 4-24 and 4-25, though a small

amount of variation between the predicted confidence values can be seen in Figure 4-27.

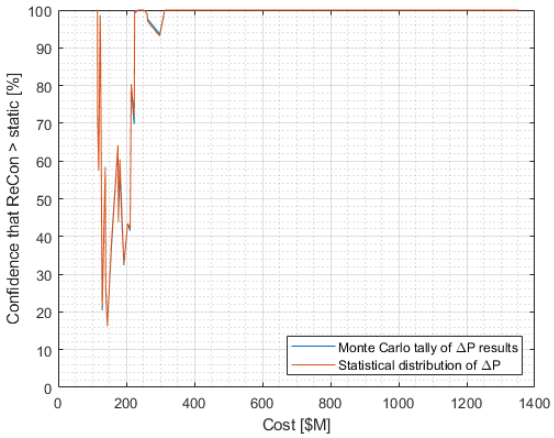


Figure 4-26: Two plots showing the confidence level that ReCon outperforms static, with the Monte Carlo tally of ΔP plotted in blue and results using the statistical distribution of ΔP plotted in red

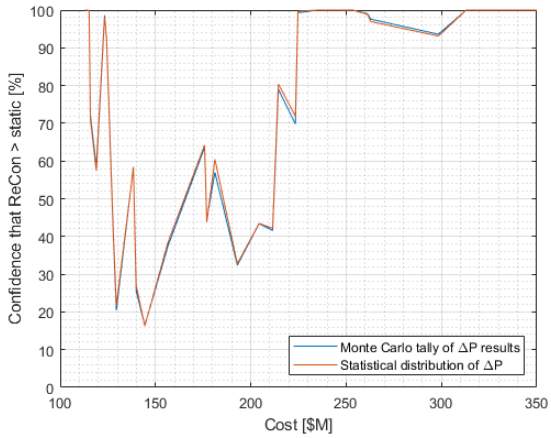


Figure 4-27: A magnified view of the confidence levels from Figure 4-26 showing differences between the two methods for the predicted confidence within the cost region from \$100M–350M

The fifth and final method used to calculate the confidence level that reconfigurable designs outperform static iso-cost alternatives was the bootstrap method (described in Section 4.3.1.5). Figure 4-28 shows the bootstrapped confidence level that ReCon outperforms static, with an extremely similar confidence curve shown compared to all four of the previous methods plotted in Figures 4-24 and 4-26. The confidence that reconfigurable designs outperform the iso-cost paired static designs is over 97.6% for all designs with a cost greater than \$223M, aside from the design pair at \$298M which has a confidence level of 93.6%.

Figure 4-29 shows a comparison of the calculated confidence levels from all five methods, with the independent hypothesis test method plotted in blue, the hypothesis test with covariance plotted in red, the Monte Carlo method plotted in yellow, the statistical distribution of ΔP method plotted in purple, and the bootstrap method plotted in green. Figure 4-30 shows a magnified view of Figure 4-29, focusing on the

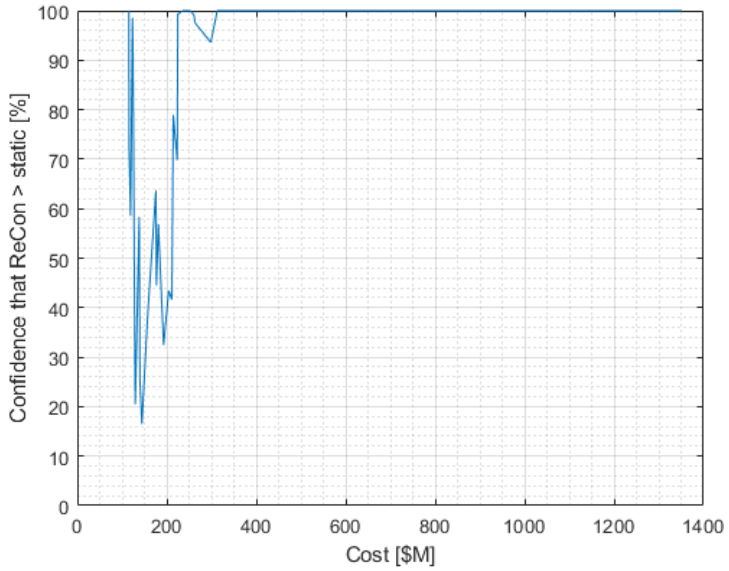


Figure 4-28: Bootstrapped confidence level that ReCon outperforms static

confidence levels for designs with a cost between $100M and $350M to show the area of greatest variation.

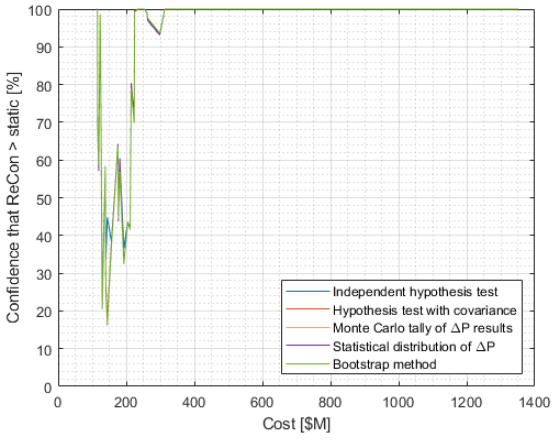


Figure 4-29: Comparison of results for the confidence level that reconfigurable designs outperform static, calculated from all 5 methods

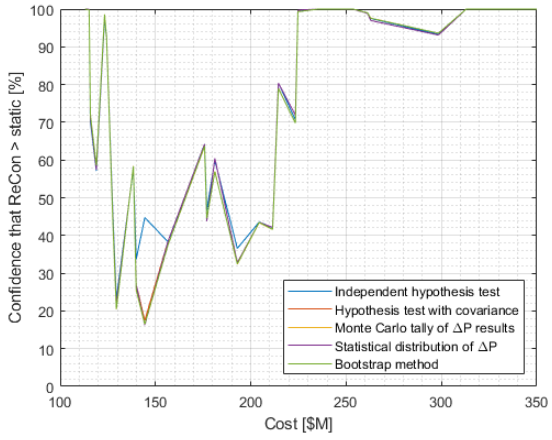


Figure 4-30: Magnified view of the confidence levels from Figure 4-29 showing differences in the predicted confidence within the cost region from $100M–350M

From these figures it may be observed that the independent hypothesis method has the largest amount of error, with large deviations in the predicted confidence

in the cost regions between \$140M–156M and \$181M–204M when compared to the results from the other four methods. It is concluded from the large amount of error introduced by this method for certain design pairings that the distributions of performance were not independent, and so more accurate confidence calculations were obtained from the hypothesis method accounting for covariance in the performances between the paired designs.

The other four methods all show low amounts of variation in the predicted confidence values, with a maximum discrepancy of 3% between the results obtained using different techniques. The Monte Carlo method shows the greatest correlation with the bootstrap method, and the hypothesis test with covariance shows the greatest correlation with the statistical distribution method. This may be due to the fact that the Monte Carlo method and the bootstrap method are both iterative techniques where the probabilities are modelled based on repeated large data sets, while the covariance hypothesis test method and statistical distribution method are both techniques based on calculating statistical properties of the available data and modelling the underlying distribution.

Using the bootstrap method on available sets of performance data from re-evaluating optimized constellation designs against a larger set of target scenarios is concluded to provide the lowest amount of error in calculating the confidence level that reconfigurable designs outperform iso-cost static designs. However, this technique may be computationally intensive, depending on the number of designs under consideration, the number of target scenarios used for re-evaluation, and the number of resampled data sets used for the bootstrapping technique. If computational resources are limited, the covariance hypothesis test method is proposed for use as an acceptable alternative, with error levels of less than 2% when compared to the bootstrapping method that is considered to provide a benchmark for confidence level calculations.

Comparing all of the five methods of confidence level calculation for the 62 iso-

cost pairs of designs, it was found that reconfigurable designs outperformed static designs with a confidence level of more than 93% for all design pairs with a cost greater than $223M, and a confidence level of more than 97% for all design pairs in this region barring one. There were also four design pairs in the lowest-cost region of designs between $113M and $124M where the reconfigurable design outperformed the static design with a confidence of at least 92.3%. However, the cost region from $116M to $223M does not have a strong confidence level that reconfigurable designs outperform static alternatives, with 50% of the design pairs in this region having a higher confidence level that static designs will outperform reconfigurable alternatives.

4.3.3 Results for confidence level that ReCon outperforms static designs by a specified margin

After comparing the amount of error from several techniques, levels of confidence that reconfigurable designs outperformed iso-cost static designs at various constellation cost were established. This completed the first objective in this area of work, showing the confidence levels that ReCon designs outperform static designs by any amount, even if the difference is infinitesimal. The next goal was to extend these calculations to find the confidence level that reconfigurable designs outperform static designs by a specified margin of performance.

Due to the range of normalized performance values through the design space, it was decided to apply performance margins as percentages of the static performance. For example, for a reconfigurable design to outperform an iso-cost static design with a normalized performance of 0.220 (achieving 22% of the ideal coverage performance) by a margin of 10%, it would have to achieve a normalized performance score of at least 0.242 (achieving 24.2% of the ideal coverage performance). This decision was made to keep margins as a consistent percentage across the wide range of static performance values, rather than applying performance margins as a proportion of normalized performance.

If a flat performance margin was applied, e.g. a margin of 0.10 normalized performance between the static and reconfigurable designs, this would represent a huge relative increase for designs in the low-performance region. Compared to a static design with a normalized performance score of 0.080 (achieving 8% of the ideal coverage performance), imposing a margin of 0.10 normalized performance so that the reconfigurable iso-cost alternative has to achieve a performance of 0.180 would represent a performance increase of 125% over the static score. In contrast, for a high-performing static design with a normalized performance score of 0.550 (achieving 55% of the ideal coverage performance), imposing a 0.10 normalized performance margin so that the reconfigurable iso-cost alternative has to achieve a performance of 0.650 would only represent a performance increase of 18.2% over the static score.

The margins that were applied were set at levels of 10%, 20%, 30%, 40% and 50% above static performance. This means that for a static design achieving 0.30 normalized performance (30% of ideal coverage), the iso-cost reconfigurable design would have to achieve a normalized performance score of at least 0.33, 0.36, 0.39, 0.42 and 0.45 respectively to meet these desired margins. For the previous confidence calculations, the reconfigurable design would only have had to achieve at least 0.30 to match or exceed the static performance, and this 0% margin is presented alongside the other results for comparison.

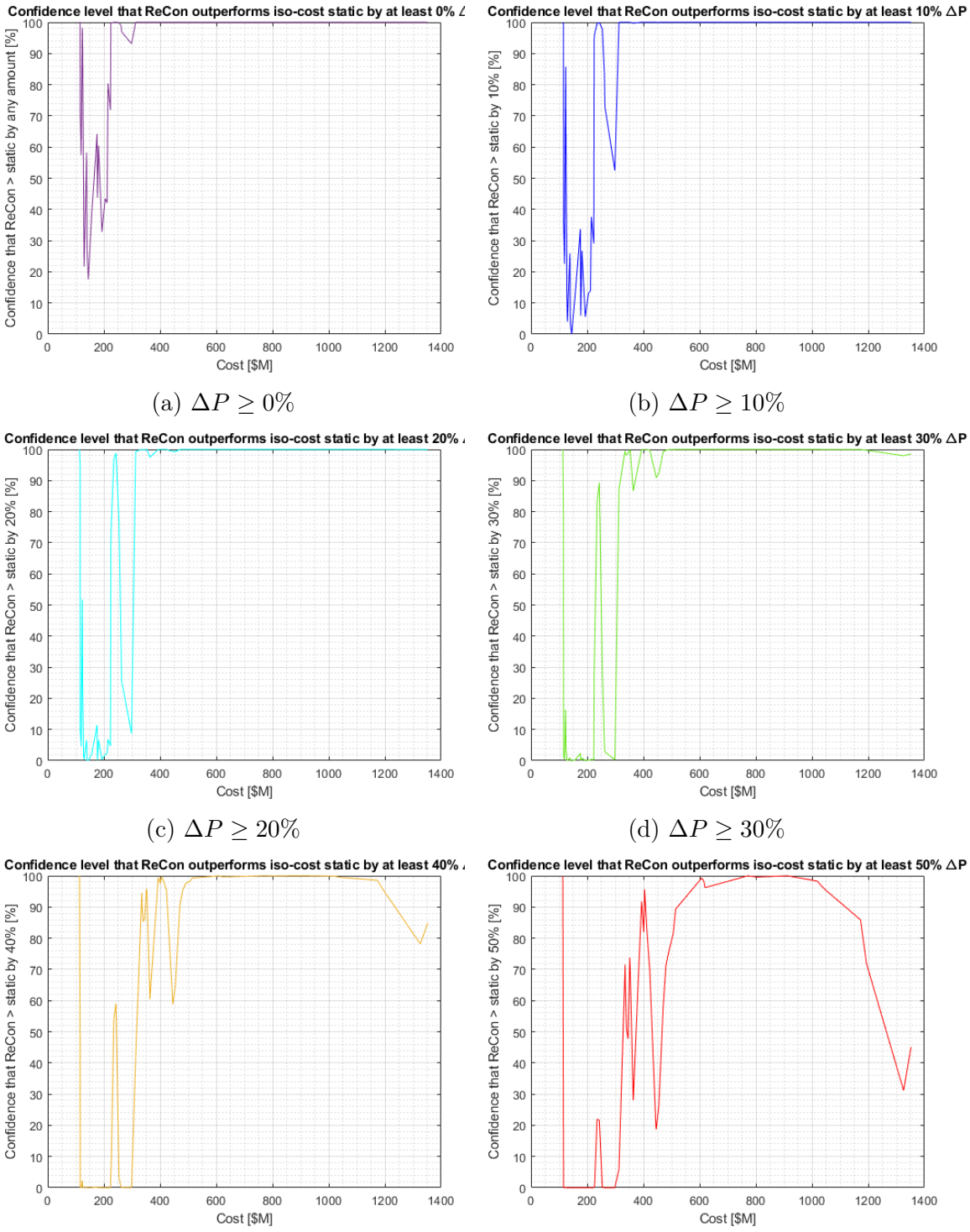
Figure 4-31 shows a set of subplots for all of the specified performance margins, with the 0% margin plotted in purple in Figure 4-31a (equivalent to the confidence levels shown in Section 4.3.2), the 10% margin plotted in dark blue in Figure 4-31b, the 20% margin plotted in aqua in Figure 4-31c, the 30% margin plotted in green in Figure 4-31d, the 40% margin plotted in amber in Figure 4-31e, and the 50% margin plotted in red in Figure 4-31f. The expected result was that each confidence curve would drop as increasingly stringent performance margins were imposed upon the reconfigurable designs, and this was observed to be the case in the results shown by

this series of figures.

For a 0% performance margin as shown in Figure 4-31a, it was found that all reconfigurable designs above a constellation cost of $223M outperformed iso-cost static designs with at least 90% confidence. There were also four designs in the lower-cost region of $113M–124M that outperformed static alternatives with at least 90% confidence. Of the remaining 16 design pairs, reconfigurable designs were predicted to have higher performance in 43.75% of the iso-cost pairs, and static designs were predicted to have higher performance in the remaining 56.25% of pairs, but with levels of confidence that were lower than 90% in both directions of prediction. Across the entire design space, it was found that for 74.2% of iso-cost pairings there was more than 90% confidence that the reconfigurable design would outperform the static alternative. Including lower levels of confidence, the reconfigurable designs were predicted to outperform the static pairing in 85.5% of cases, with static designs outperforming reconfigurable in only 14.5% of cases.

For a 10% performance margin as shown in Figure 4-31b, it was found that all but one pair of reconfigurable designs above a constellation cost of $223M outperformed static designs by this amount with at least 90% confidence. The single exception is a design pair at a cost of $298M, which dropped to a confidence level of only 52.5% that ReCon would outperform static by a minimum margin of 10%. Two of the four lower-cost designs (at constellation costs of $113M and $115M) retained confidence levels above 90% with the increased performance margin. Across the entire design space, it was found that for 66.1% of iso-cost pairings there was more than 90% confidence that the reconfigurable design would outperform the static alternative by a performance margin of at least 10%. Including lower levels of confidence, the reconfigurable designs were predicted to outperform the static pairing by at least 10% in 74.2% of cases.

For a 20% performance margin as shown in Figure 4-31c, it was found that all reconfigurable designs above a constellation cost of $298M outperformed static designs



(a) $\Delta P \geq 0\%$

(b) $\Delta P \geq 10\%$

(c) $\Delta P \geq 20\%$

(d) $\Delta P \geq 30\%$

(e) $\Delta P \geq 40\%$

(f) $\Delta P \geq 50\%$

Figure 4-31: Bootstrapped confidence levels that ReCon outperforms static by a specific performance percentage of at least a) 0% b) 10% c) 20% d) 30% e) 40% and f) 50%

by this amount with at least 90% confidence, as well as a pair of designs at costs of $235M and $243M, and the same pair of lower-cost designs at $113M and $115M. Across the entire design space, it was found that for 61.3% of iso-cost pairings there was more than 90% confidence that the reconfigurable design would outperform the static alternative by a performance margin of at least 20%. Including lower levels of confidence, the reconfigurable designs were predicted to outperform the static pairing by at least 20% in 67.7% of cases.

For a 30% performance margin as shown in Figure 4-31d, it was found that all reconfigurable designs above a constellation cost of $313M outperformed static designs by this amount with at least 90% confidence, except for a single design pair at $364M which dropped to a confidence level of only 86.7% that ReCon would outperform static by a minimum margin of 30%. Other designs at $313M and $243M achieved confidence levels of 87.3% and 89.3%, also narrowly missing the 90% confidence mark. A single lower-cost design at $113M retained over 90% confidence in outperforming static by at least 30% margin. Across the entire design space, it was found that for 53.2% of iso-cost pairings there was more than 90% confidence that the reconfigurable design would outperform the static alternative by a performance margin of at least 30%. Including lower levels of confidence, the reconfigurable designs were predicted to outperform the static pairing by at least 30% in 61.3% of cases.

For a 40% performance margin as shown in Figure 4-31e, it was found that all reconfigurable designs between a constellation cost of $455M and $1326M outperformed static designs by this amount with at least 90% confidence, with other designs costing $113M, between $313M and $340M, and between $345 and $444M (excluding one design pair at $364M) also achieving this confidence level. Across the entire design space, it was found that for 41.9% of iso-cost pairings there was more than 90% confidence that the reconfigurable design would outperform the static alternative by a performance margin of at least 40%. Including lower levels of confidence, the reconfigurable designs were predicted to outperform the static pairing by at least 40% in

58.1% of cases.

For a 50% performance margin as shown in Figure 4-31f, it was found that all reconfigurable designs between a constellation cost of $515M and $1173M outperformed static designs by this amount with at least 90% confidence, with other designs costing $113M, $393M and $404M also achieving this confidence level. Across the entire design space, it was found that for 19.4% of iso-cost pairings there was more than 90% confidence that the reconfigurable design would outperform the static alternative by a performance margin of at least 50%. Including lower levels of confidence, the reconfigurable designs were predicted to outperform the static pairing by at least 50% in 43.5% of cases.

Figure 4-32 shows all of the subplots from Figure 4-31 overlaid onto a single diagram, illustrating how the confidence level drops in stages for individual design pairings as more stringent performance margins are imposed. The constellation cost region between $515M and $1173M retains a high level of confidence, even for a required margin of 50% higher performance for reconfigurable designs compared to iso-cost static constellations, suggesting a very large ΔP for designs in this region. The cost region between $223M and $515M shows a much greater amount of variation, with the confidence dropping rapidly for some designs as more stringent margins are imposed, while for others the level decreases much more gradually.

Figure 4-33 shows the minimum ΔP values for each design pairing in units of normalized performance, estimated with a confidence level of 95% using the bootstrap method. These units are equivalent to the subtracting the normalized performance score of each static design from the normalized performance score of its iso-cost reconfigurable paired design. Figure 4-34 shows the same bootstrapped minimum ΔP values with 95% confidence, but with ΔP expressed as a percentage of the static design performance. A 10% minimum ΔP on this chart equates to a reconfigurable design achieving 10% higher normalized performance than its static equivalent, for

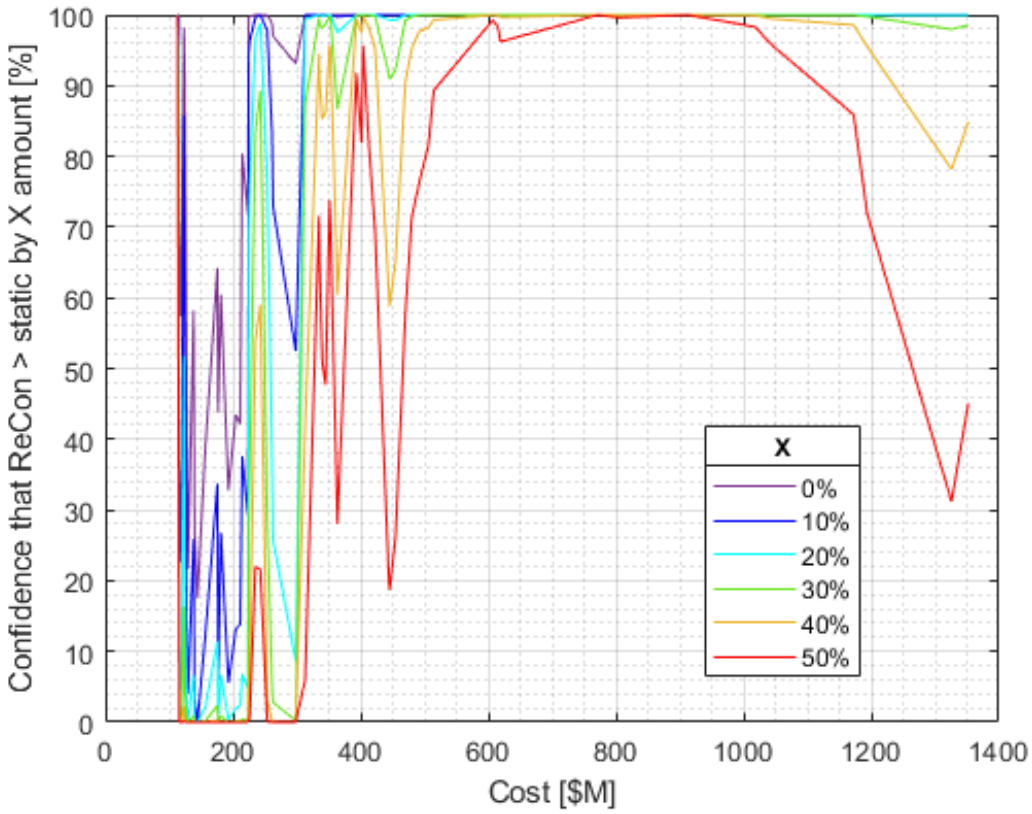


Figure 4-32: Bootstrapped confidence levels that ReCon outperforms static by a specific margin of performance, expressed as a percentage of the static design performance between 0–50%

example with a normalized performance score of 0.44 for ReCon relative to a score of 0.40 for static at the same cost point. Rather than expressing the full 95% confidence range for the interval within which the minimum ΔP may occur, both of these figures simply show the lower bound, as for a one-sided metric such as minimum ΔP, this represents the boundary above which 95% of minimum performance results will lie.

These figures explain some of the variation in confidence levels seen in Figures 4-31 and 4-32 across different regions of the design space. When assessed against 500 different target scenarios of interest, the worst-case performance for reconfigurable designs costing approximately \$500M to \$1250M still achieved 20% higher performance than iso-cost static designs, with a 95% confidence level in this result. For designs costing between \$113M and \$300M, the worst-case scenarios are often negative ΔP

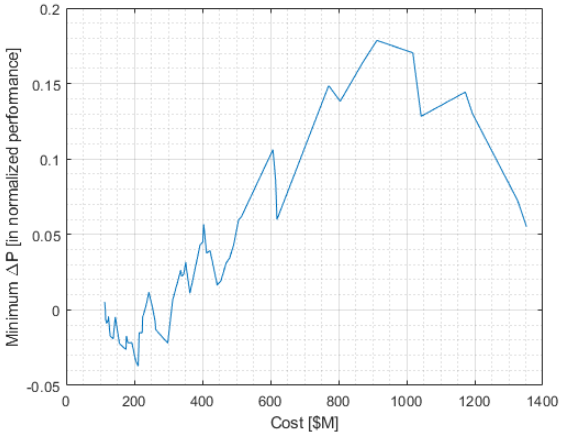


Figure 4-33: Bootstrapped minimum ΔP values (with 95% confidence) for each design pairing, expressed in terms of normalized performance

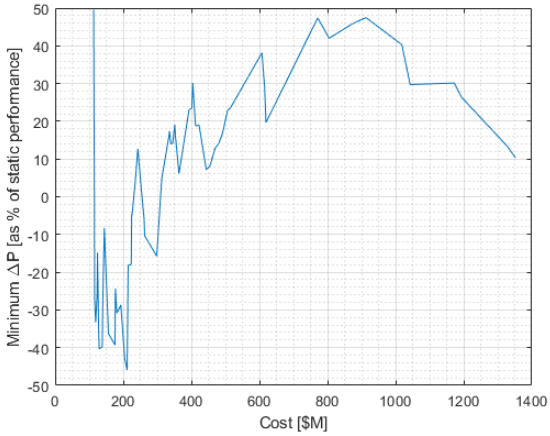


Figure 4-34: Bootstrapped minimum ΔP values (with 95% confidence) for each design pairing, expressed as a percentage of static performance

values, meaning that these are situations in which the static design outperformed the reconfigurable design. Although this may happen with varying levels of frequency depending on the design, this explains the lower levels of confidence in reconfigurable designs outperforming static designs within this cost range.

Pearson correlation coefficients were calculated for the confidence level of ReCon outperforming static against the design variables used to define the reconfigurable and static constellations under consideration. A correlation coefficient of 1 equates to a perfect positive correlation between variables, a coefficient of -1 equates to a perfect negative correlation, and a coefficient of 0 represents no correlation between the variables. The static variables were found to be only weakly correlated with the confidence level, but the correlation was much more significant for the reconfigurable design variables.

Table 4.4 shows the correlation coefficients between the level of confidence that ReCon outperforms static and six variables from the set used to define the reconfigurable design. All of these variables have a positive correlation with the confidence level, meaning that an increase in GOM altitude, orbital inclination, number of satel-

lite planes, aperture diameter, ΔV budget for reconfiguration, or total number of satellites is correlated with an increase in confidence that ReCon outperforms static. The variables are ordered by the strength of the correlation, meaning that the GOM altitude is the most strongly correlated variable with the confidence level, and the total number of satellites demonstrates the weakest (though still positive) correlation. The p-value of each result is shown, representing the probability of obtaining such a result by chance if the variables were not actually correlated. The significance level was set to 0.05 for this calculation, meaning that any correlation with a p-value greater than 0.05 was dismissed as insignificant, due to the possibility of obtaining such a result by chance.

Table 4.4: Correlation coefficients and p-values for six reconfigurable design variables against confidence level that ReCon outperforms static

Design variable	**Correlation coefficient**	**p-value of result**
GOM altitude	0.8372	2.29e-17
Inclination	0.7308	1.56e-11
Number of satellite planes	0.5259	1.14e-5
Aperture diameter	0.4725	1.05e-4
Reconfiguration ΔV	0.4711	1.11e-4
Total number of satellites	0.2817	2.65e-2

The results in Table 4.4 show that the confidence value in the performance gained due to reconfigurability cannot be correlated with any single design variable, but that many aspects of the optimized designs contribute to ΔP with varying amounts of influence. Often these design variables are interrelated, and trade-offs must be made between different aspects of the design to balance cost, performance, and other constraints placed upon the design space to ensure that selected designs are feasible for real-world operations. These include limitations on variables such as minimum orbital altitude, maximum constellation size, or propulsion system mass fraction. Although performance benefits may be positively correlated with specific design variables, increasing the values for these variables may have other negative effects on the constellation in terms of cost, feasibility, or interactions with other aspects of the

design.

4.4 Conclusions

Using iso-performance comparisons of prospective designs such as Legge's Value of Reconfigurability does not account for the amount of variation that occurs in performance scores when constellations are evaluated against a range of different target scenarios. Due to the fact that the ReCon model does not account for operating costs, constellation cost is a fixed number for each design regardless of evaluation against different operating scenarios or locations. A new iso-cost comparison metric of ΔP is presented as an alternative to Legge's VoR metric.

ΔP is the difference in performance between iso-cost reconfigurable and static constellation designs, calculated by subtracting the static performance from the reconfigurable performance. A positive ΔP value represents a situation in which the reconfigurable design outperforms the static design, and a negative ΔP value represents a situation in which the static design outperforms the reconfigurable design. A ΔP value of zero represents a situation in which the paired designs achieve exactly the same performance. ΔP may be considered as a simple difference expressed in terms of the normalized performance (based on Legge's custom persistence metric described in 3.1.1.1, and calculated as a fraction of the ideal performance when the desired revisit cadence and ground sample distance are perfectly matched).

ΔP may alternatively be expressed as an improvement in performance in terms of a percentage of the static design performance; for example, if a reconfigurable and static iso-cost pair achieve normalized performance scores of 0.5 and 0.4 respectively, ΔP expressed as a simple difference would be +0.1 in terms of normalized performance, or a +25% improvement over the static performance due to the incorporation of reconfigurability in the iso-cost design.

Figure 4-35 shows 95% confidence intervals for two useful ΔP metrics in evaluating the relative merits of different constellation design options. The mean ΔP (as a percentage improvement over the static design performance) is plotted in dark blue, and the minimum ΔP (expressed in the same terms) is plotted in lilac. Considering these metrics in tandem supplies greater information than a single estimated performance value, providing a sense of the overall distribution of performances in different coverage scenarios. The mean ΔP provides an estimate of the average performance improvement to be gained over a static design at the same constellation cost, with a 95% confidence level that the actual mean achieved will fall within the specified range. The confidence interval on this metric is relatively narrow, due to the averaging effect of comparing a large number of means calculated using the bootstrap method. The minimum ΔP provides an estimate of the worst-case performance improvement to be gained over a static design at the same cost, with negative values representing cases where the static design outperformed its reconfigurable alternative. The 95% confidence interval on this metric is somewhat wider than for the mean, due to minimum performance calculations relying on a single value in the distribution rather than the average of many values.

64.5% of the design pairs shown in Figure 4-35 have a mean ΔP of 25% or higher, and 74.2% have a mean ΔP of 10% or higher. 61.3% of the design pairs have a minimum ΔP that is greater than 0. The highest values of ΔP are seen for iso-cost design pairs in the \$750M–800M region of Figure 4-35, where the mean ΔP shows a 95–100% improvement for the reconfigurable design performance compared to the score of the static alternative. For these pairs, the reconfigurable design achieves a 45–60% improvement above the static design even in the worst-case scenario represented by the minimum ΔP range.

The type of trends in performance data shown for designs in the \$220M–300M range may be of particular interest in developing methods of selecting a constellation design. Although the mean ΔP varies between 0 and 38% in this region, the

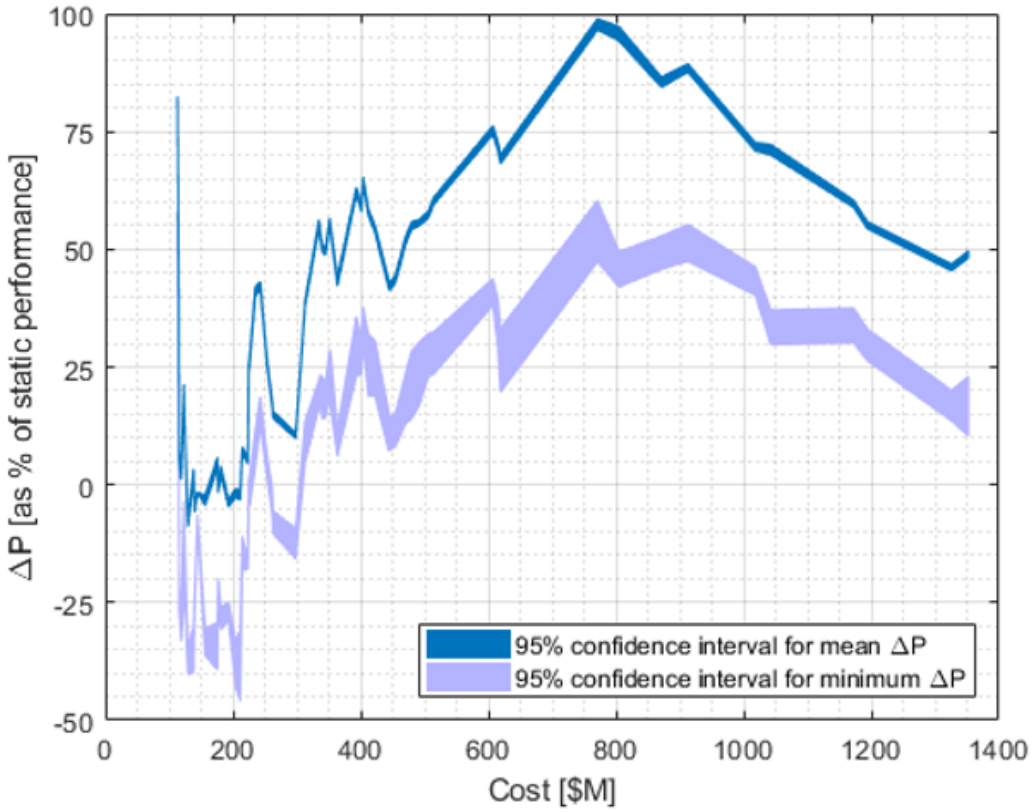


Figure 4-35: 95% confidence intervals for mean and minimum values of ΔP, expressed as a percentage of the static constellation design performance

worst-case minimum ΔP shows that for some scenarios, the ΔP that is actually achieved by these designs will be in the range of -18% to +18%. This demonstrates the need to consider the range of performance data and how it is dispersed, rather than assessing only the mean value in evaluating and comparing constellation designs.

Figure 4-35 illustrates how stakeholders who are considering adopting reconfigurability in designing and launching satellite constellations must ensure that the selected designs achieve a reliable level of performance, as well as an improvement in mean performance. In quantifying the benefits of reconfigurability, it must be acknowledged that for some areas of the design space there is a large amount of overlap in the expected range of performance when compared to iso-cost static alternatives. This occurs particularly in the lower-cost, lower-performance region of the design space,

where lower constellation budgets mean that reconfigurable architectures may not possess sufficient satellites or ΔV for fully responsive maneuvering. At these levels, the performance of reconfigurable designs is comparable to that of iso-cost static designs.

Examining the subset of the 14.5% of static designs in Figure 4-31a that were not predicted to outperform reconfigurable designs at the same cost, there is not a high level of confidence in any of this set of results, with confidence levels varying from 56 to 82% in the prediction that static designs would have the better performance in these cases. All of these designs are also within the cost region from \$130M to \$214M, where mean ΔP has a range within $\pm 5\%$ of static performance. These results demonstrate that for the worst-performing reconfigurable designs, the mean performance is at least on par with the performance of static constellations. Observing the amount of variation in performance over different scenarios shown by the minimum ΔP interval, these reconfigurable designs are likely to be significantly worse in some scenarios and significantly better in others, averaging out to similar mean performance when compared to static designs.

ΔP was modelled using confidence levels and interval estimation rather than approximating a single performance score, in order to properly represent the range of performance results that are achieved when a design is re-evaluated against different target decks. Several methods of calculating confidence levels were compared, and it was established that bootstrapping or Monte Carlo methods provided the highest accuracy in results, although these techniques required the most computational resources to execute. Hypothesis testing with covariance or modelling the statistical distribution of ΔP were also found to supply low error rates in their estimates, and would supply acceptable alternative methods where computational resources are limited or a large number of iterations of the calculations must be carried out.

While VoR assumes that performance can be approximated as a single constant

score across all possible operating scenarios, ΔP is an equivalent single-axis comparison that focuses on the amount of performance to be gained from reconfigurability, evaluated at iso-cost points in the design space of possible constellation designs. ΔP also has the benefit of factoring in statistical dispersion of the constellation performance, acknowledging that different performance scores will be achieved in different target scenarios and presenting this information to decision-makers as an additional data available to evaluate and rank constellation design candidates.

Adopting an iso-cost metric such as ΔP accounts for the lack of variation in estimated constellation costs when evaluated against different scenarios in the ReCon model, and the presence of a large amount of variation in the estimated performance scores. Once design parameters are fixed in the model, the cost estimate is also fixed, regardless of changes in the target locations and timings used for performance re-evaluation. If future work focuses on modelling operational costs as part of the ReCon model and introducing cost variability for different scenarios, comparing the relative value supplied by reconfigurable and static designs on an iso-cost basis will still supply greater reliability and lower variance than comparisons on an iso-performance basis as long as the constellation cost variation between scenarios is found to be less than 25% of the existing cost estimates.

An interesting area for future work in the area of reliability-based design optimization would be to implement use of the confidence metric in the actual process of optimizing the designs generated by the ReCon codebase. This would allow for a probabilistic approach to be taken to the optimization, maximizing the chance of ReCon outperforming static along the Pareto front of non-dominated design candidates. However, this would be a complex task due to the fact that the ΔP metric is based upon a comparison between reconfigurable and static designs at the same cost point in the architecture tradespace. A useful step in investigating this implementation might be to first run the optimization to generate a set of static designs, and then use these as the basis for a second design optimization incorporating reconfigurability

to improve upon the initial performance scores. The optimization objective would be targeted towards achieving the greatest percentage improvement in score at each cost point. Enacting repeatability in the target decks used to evaluate the relative performance of both sets of designs would be beneficial to ensure that ΔP is actually evaluated from the comparison in performance achieved in response to identical target events of interest.

Chapter 5

Sampling improvements and uncertainty quantification for design optimization

With the development of increasingly advanced computer hardware over time, the ability to computationally simulate complex real-world processes has become an important tool across every STEM field. These improvements in simulation capabilities have enabled the study and analysis of phenomena that it was previously impossible to research, or that could only be investigated through highly resource-intensive experimentation. Such simulations are developed through several stages of modelling: first, a real-world, physical system must be represented by a mathematical model that captures the quantities of interest, which must be validated as sufficiently accurate for the intended uses, although it is rarely a perfect representation of the physical reality. Next, the mathematical model is approximated into a computational model which can be solved algorithmically to find an approximate solution. This must be verified by comparing the approximated computational solutions with the mathematical model to determine the level of accuracy, taking into account truncation errors, finite difference approximations, and other errors introduced by the use of numerical methods to solve the model. Finally, the computational model must be verified against the original real-world process, comparing outputs computed by the simulation for quan-

tities of interest against real, physical data from observations or experiments.[108] In cases where this is prohibitively difficult or expensive, newer computational models are often compared against other pre-existing computational models which have already been verified against real-world data.

Although computational models of this kind have led to many scientific and technological advances, the ability to quantify uncertainty in such models and the predictions made from their outputs is much less well-developed than the process of creating these approximate simulations.[109] Computational simulations diverge from reality in a variety of ways, which must be acknowledged in assessing how useful the simulation output may be. Often there may be inadequate data, a lack of precise detail, or uncertainty in the knowledge of essential inputs supplied to the model such as initial conditions, boundary conditions, constraints on certain parameters, or even values for variables of interest that have been shown to have a significant effect on model output. Discrepancies between the mathematical and computational models also add a degree of uncertainty, due to the error introduced by the necessary approximations that must be made to numerically solve the real-world system, such as representing physical spaces with a finite element structure, or modelling a continuous span of time using an approximation with discrete timesteps. On a basic level, simulations must omit or simplify certain aspects of reality in order to focus on the quantities of interest and avoid becoming so complex as to be computationally intractable.[108]

As stated by George Box, "All models are wrong, but some may be useful"; it is only by quantifying exactly how wrong the model predictions are (and in what ways) that it can be decided what use may be made of the results.[110] Understanding the limitations, flaws, assumptions and uncertainties built into the computational model is crucial to arriving at an appropriate level of confidence in the simulation result. Comparing the computed output for a quantity of interest and a value for its real-world physical equivalent, the uncertainty in value between the two must be quantifiable for the simulation output to be of any practical use; "The uncertainty

is as important a part of the result as the estimate itself... An estimate without a standard error is practically meaningless."[111]

Due to the fact that uncertainty is omnipresent in any computational model's approximation of reality, the necessity arises for quantifying these uncertainties, by assessing how the model-based estimation of the output variables of interest is affected by errors and uncertainties of numerous origin.[108] The field of uncertainty quantification occurs where probability and statistics must intersect with physical reality, recognizing the ubiquity of uncertainty and therefore the necessity of identifying its sources and quantifying the relative amounts of their contributions to overall uncertainty, in order to assess the validity of a given model.[112]

The US Department of Energy gives the definition as follows:

> "Uncertainty quantification (UQ) studies all sources of error and uncertainty, including the following: systematic and stochastic measurement error; ignorance; limitations of theoretical models; limitations of numerical representations of those models; limitations of the accuracy and reliability of computations, approximations, and algorithms; and human error. A more precise definition is UQ is the end-to-end study of the reliability of scientific inferences."[109]

The field of uncertainty analysis is strongly tied to that of sensitivity analysis. While UQ has a greater focus on quantifying and propagating uncertainty, sensitivity analysis may also involve identifying sources of uncertainty in a model and quantifying the degree of their impact on system output, and so the two analyses are often carried out together. Sensitivity analysis may also involve modelling the relationships between model inputs and outputs, identifying errors, and attempting to reduce uncertainty, though it differs from UQ in its focus on other objectives such as model simplification, redundancy reduction and optimization of model parameters.[113]

Many of the recurring limitations and issues that arise across a wide range of simulation applications should be considered as part of the processes of validation, verification and uncertainty quantification of the model. The amount or relevance of the empirical data used to verify the model may be limited, or the computational model may aim to extrapolate well beyond the physical conditions available for validating its output, going as far as to simulate untested or unobservable scenarios. Computational models cannot provide perfectly accurate representations of the real world, and the more complex the actual physical system is, the more approximations, inadequate estimations, simplifications or omissions are likely to be made to arrive at a computational representation. Such approximations also typically feature unknown parameters and boundary conditions which must be estimated for the desired application.[108, 109]

The starting point for UQ is the quantification of uncertainty in such inputs to the model, which may be subject to factors such as random variability in operating circumstances, a lack of knowledge about the exact parameter values, or unknown boundary conditions. These uncertainties are most commonly quantified using probability distributions or by defining ranges within which a given variable may fall.[112] These inputs may be situation-invariant, such as gravitational acceleration or other physical constants, or entirely dependent on the situation under consideration and its associated initial conditions or parameter constraints.[108]

The computational model must then be used to propagate these uncertainties in input through the simulation to quantify the effect on the computed output variables of interest and understand the mapping between model inputs and outputs. As an 'end-to-end' reliability study, UQ focuses on the relationships between pieces of information in the model, taking into account the assumptions and approximations made along the way. Rather than aiming to state that a given model is exactly 'true' (which would be in any case an impossibility), UQ quantifies the validity of the model to a specified degree, leading to the logical conclusion that the validity of the simulated

output may also be quantified and accepted to a specified degree.[112]

The process of UQ must aggregate uncertainties from a variety of different sources within the model, including uncertainty in the model inputs, real-world physical variability of an unpredictable nature, numerical error, and error from the computational model itself. After identifying and quantifying sources of error and uncertainty in both the inferences and predictions that went into constructing the model and its inputs, and identifying the assumptions made in creating the uncertainty quantification itself, UQ results in a summary of these sources and a final combined metric for uncertainty. All of the uncertainties determined in this process are incorporated into this overall quantitative assessment of the uncertainty in the simulated prediction of the quantity of interest.[108, 109]

In order to design for the uncertain operating circumstances for the proposed reconfigurable constellations of the ReCon model, uncertainty propagation is used. For this case, designing for uncertainty requires quantifying it and propagating the effects of locational uncertainties through the ReCon model, to characterize the effects of uncertainty on the computed performance. As the model is highly non-linear and non-deterministic, a probabilistic forward uncertainty propagation approach is taken to arrive at an accurate estimation of system performance. The use of Monte Carlo methods and other techniques for model uncertainty propagation is described in more detail in Section 5.1.

5.1 Designing for flexibility using Monte-Carlo-based model propagation

Using forward uncertainty propagation involves observing the uncertainty in the inputs to be propagated through the full model to estimate overall uncertainty in the system response. This can be used to find the mean and standard deviation of the

output data, the reliability of the output response, or to evaluate the full probability distribution of the output information. Where model inputs are uncertain and treated as random variables, an appropriate input distribution may be used based on whatever data is available. For the ReCon model, locations of interest are generated using the World Bank natural disaster PDF; Legge generates target decks using simple random sampling from this distribution.

A key aspect of UQ is the propagation of input uncertainties through the full set of calculations in order to quantify the overall effect of uncertainty on the final output of interest. Mapping input distribution to output distribution improves the understanding of model behavior needed to assess uncertainty in the final prediction made by the simulation. The simplest method of gathering information on the output distribution is to use Monte Carlo sampling, generating a large number of samples directly from the probability distribution of the model inputs, carrying out the forward propagation of this input uncertainty through the simulation, and assessing the output data based on each input. Once an output distribution is obtained for the quantity of interest, other relevant metrics may be calculated from it such as mean, median, confidence intervals, or percentile values of interest.

As Monte Carlo sampling does not depend on the model having a particular degree of complexity, linearity, or dimensionality of the inputs, it provides a flexible method of propagation usable by most simulation models. However, the speed of the computer model can impose a limit on the usefulness of Monte Carlo techniques, as this method can involve repeating hundreds or thousands of functional evaluations to arrive at a sufficient level of accuracy in estimating the output distribution. In these cases, the computational demands of the simulation will determine the number of times it can reasonably be iterated to gather data, particularly limiting the number of simulation runs that may be carried out for models that incur a high processing time for each individual input. Characterizing events with a low probability of occurrence but with high impact on output results is also difficult using standard Monte

Carlo methods, as such events are rarely generated in the course of simple random sampling but will have considerable consequences for system performance when they do occur.

The issue of runtime arose in Legge's work developing the ReCon framework, with acknowledgement made in his thesis[8] that the achievable confidence level (calculated as standard error of mean performance and standard error of median performance) was limited by a maximum of 24 Monte Carlo samples based on the available computational resources. This means that at most, each optimized constellation design could be assessed against 24 randomly sampled five-year operating scenarios. Legge's selection of the numerical time step for use in the simulation model was also motivated by balancing model runtime against modelling error, with error of 0.144% quantified for GOM propagation based on a 30-second timestep and error of 0.76% quantified for ROM propagation based on a 20-second timestep.

Although the number of Monte Carlo samples that can be carried out may be drastically limited depending on the complexity of the problem and the available computing resources, the embarrassingly parallel nature of the repetition (i.e. with no dependencies at all between the parallel iterations of the simulation model) lends itself extremely well to parallelization of the Monte Carlo method. For the ReCon model, this parallelization was implemented by Legge for the multi-objective optimization stage of the model described in Section 3.1.3, rather than more simply applying it only to the Monte Carlo layer described in Section 3.1.2. Legge separated the functional evaluations (one constellation design evaluated against 24 scenarios) down to the level of code executions (one constellation design evaluated against one scenario) in order to use a batch processing technique for parallelization. An illustration of how these two terms differ is shown in Figure 5-1 below: code executions consisting of the evaluation of a single constellation design against a single target are shown in the left-hand chart, e.g. evaluating the performance of design #4 against target deck C is carried out by the single code execution 4C. Functional evaluations are shown on the right-hand

side of Figure 5-1, consisting of one constellation design getting evaluated against a full set of target deck scenarios to arrive at an averaged performance value, e.g. evaluating the performance of design #3 against 24 target decks A–X (a total of 24 code executions) to arrive at an averaged performance value $\bar{P}_3$ for the constellation.

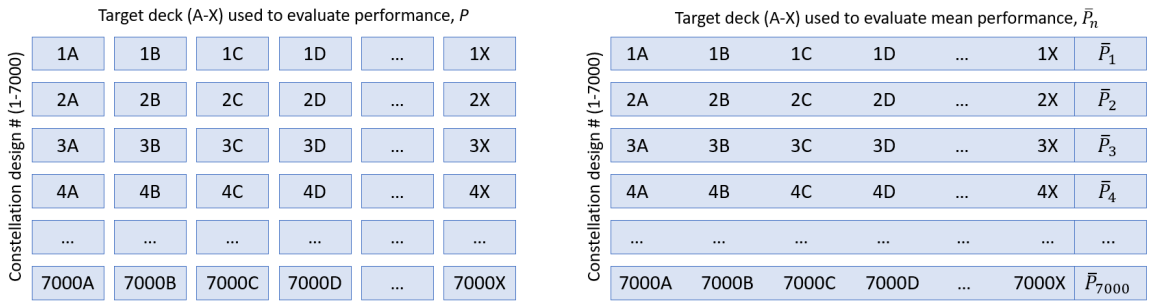


Figure 5-1: Visual comparison of code executions (left) consisting of one constellation design evaluated against one target deck, against functional evaluations (right) consisting of one constellation design evaluated against 24 target decks to arrive at an averaged performance value

These code executions were sorted by estimated runtime in order to balance the computational load more evenly across processors, with the highest runtime tasks assigned first and the lower runtime tasks used to supply new tasks for processors that finished their initial allocation. This approach was used due to greater runtime savings in parallelizing at the level of code executions rather than functional evaluations, which allows for large constellation designs with a higher runtime to be distributed more evenly across the available processors. If parallelization was carried out at the level of functional evaluations, tasks consisting of large designs being evaluated against 24 potential scenarios would be extremely hard to balance with small constellations evaluated against the same number of target decks, with some processors taking orders of magnitude longer runtimes to complete their tasks and return output to the leader processor for collation. Depending on the number of available processors, code executions can be combined into evaluation blocks, reducing communication time with the leader processor.

Aside from implementing simple parallel processing, Monte Carlo approaches are not well-suited to take advantage of any physical or mathematical structure in the model that could otherwise be used to speed up the calculations, due to the objective of retaining the functional dependence of output distribution on input distribution rather than a simple averaged result. However, depending on the degree of input variability and model linearity, alternative probabilistic approaches exist for uncertainty propagation that may be better suited to improving runtime, with some limitations that are contingent on the output data of interest.

The lack of dependence of Monte Carlo sampling on input dimensionality or model complexity means it is an extremely versatile technique, as long as the computational model is sufficiently fast that runtime does not pose a problem in executing the number of iterations required to arrive at the desired level of confidence in the model output. Where issues of high runtime start to become an issue for achieving sufficient functional evaluations of the simulation, variance reduction techniques can be used to improve the precision (as the name suggests, reducing the variance) of the estimated output distribution without increasing the level of computational resources needed. A range of methods may be employed to improve the statistical efficiency of the output data, increasing precision and reducing the size of confidence intervals for the estimated output variable of interest. Two of the most commonly used variance reduction techniques are importance sampling and stratified sampling, and these were selected for implementation in this work as part of efforts to improve the ReCon model by increasing simulation efficiency. The specifics of introducing importance sampling for the ReCon target decks are described in Section 5.2, and a comparison of several stratified sampling techniques for the model is given in Section 5.4.

The objective in this area of work was to retain the existing level of quantified uncertainty in the simulation results established in Robert Legge's past work, while reducing the amount of samples needed to achieve this confidence, thereby achieving savings in computational runtime. Several new methods of sampling from the World

Bank PDF of natural disaster locations are proposed for the generation of new target decks. The level of confidence in the estimated constellation performance is compared using different sampling methods, with conclusions drawn from the results about the reduction that could be made in number of target decks used. This possibility of using fewer scenarios for design optimization is then compared to the potential savings to be made in model runtime, based on runtime data collected from a series of ReCon model runs on the MIT Supercloud.

5.2 Importance sampling

Importance sampling is one of the most common variance reduction techniques that can be applied to Monte Carlo sampling of a simulation. Statistical sampling such as Monte Carlo is an effective technique for quantifying uncertainty in input and how it maps to uncertainty in the output parameters generated by a computational model. However, assessing this mapping shows that certain values of the input distribution have a greater impact on the output estimation than others. It is possible to reduce the variance in the estimated output by sampling these 'important' values at a higher frequency. The use of an input distribution that is deliberately biased in this fashion may lead to a bias in the estimated results, depending on if it is directly implemented upon the simulation input and by what method, but where this occurs it can be remedied by applying a corrective weighting to the results of the model.

For the ReCon model, the Monte Carlo layer of the simulation (described in Section 3.1.2) takes an individual constellation design and iterates the simulation of its mission lifetime to assess its performance against a range of potential operating scenarios. A single 'functional evaluation' is defined by Legge as the evaluation of one constellation design against a range of target decks to arrive at an averaged performance value, therefore involving an even greater number of 'code executions' (defined as the single execution of a code function that is iterated during a Monte Carlo

process[108]). A visual illustration of the distinction between these terms was shown previously in Figure 5-1.

Many thousands of code executions at this level of the model are carried out during each simulation run, evaluating the generated constellation designs against a number of target decks. By default 24 target decks are used, meaning that each single functional evaluation is itself made up of 24 code executions. These target decks each outline one possible five-year mission lifetime, and so one code execution scores the utility provided by one constellation design in one potential scenario. In contrast, one functional evaluation scores the **average** utility provided by one constellation design across all 24 potential scenarios used to supply a range of different operating conditions for the design optimization process.

To arrive at an optimized set of non-dominated designs, hundreds of possible constellation designs are evaluated in each simulation run. ReCon simulation output logs show that a range of 5,000–20,000 functional evaluations must be carried out to reach the optimization termination criterion, and as each of these thousands of evaluations is carried out against multiple target decks in the Monte Carlo layer, this equates to a total of 120,000 to 480,000 code executions. This high number of iterations means that any small decrease in the runtime of this layer of the simulation model would result in a huge decrease in overall runtime, and reducing the number of iterations itself is one way to achieve this effect. Legge's previous work on the ReCon model established a standard error based on the use of 24 target decks for design optimization, and therefore even a small reduction in the number of target decks that are needed to establish the same level of error will lead to a significant decrease in the number of code executions carried out and a corresponding decrease in computational runtime to complete the code run. In the example given, achieving the smallest possible reduction in target decks from 24 to 23 scenarios would allow for a reduction of 4.2% in the number of code executions (5,000–20,000 code executions for the scenario described).

In order to reduce the number of target decks used to optimize reconfigurable constellation designs in the ReCon model, the process of generating these decks was examined. These operating scenarios are pre-generated and saved to a large array consisting of 10,000 different 5-year scenarios, made up of a total of 195,301 individual events of interest. Pre-generating this array avoids the addition of an extra script within the ReCon model, which would increase the overall runtime due to the high number of times the model is iterated, even if the script itself runs relatively fast. A large number of scenarios are generated to supply an extensive array of target decks which can then be called upon by the ReCon model. Saving the entire array in this manner also means that the same decks can be called multiple times to compare the results of different optimization settings using identical inputs, or target decks can be sampled randomly from within the array in cases where randomization is desirable. The use of this data array also allows for side-by-side comparison of optimization results obtained using different techniques for target deck generation.

Many of the parameters specified in each target deck can be set to constant values for a standard design optimization run, such as the event duration (14 days by default), desired ground sample distance (1 m by default), daylight hours for optical imaging (6am–6pm local time), and desired temporal resolution of imaging (1 hour by default, i.e. the desired revisit frequency is once per hour). However, the other parameters used to define the events of interest will necessarily be changing constantly, due to the entire concept of a reconfigurable constellation providing the capability of responding to events with unpredictable locations and timings. These are defined using latitude and longitude co-ordinates for the event location, and time (in years) since the start of the mission lifetime. An example 5-year target deck is shown in Table 5.1, listing 20 events of interest that are spaced out in both the location and the time of occurrence. Table 5.1 shows how many of the parameters remain constant between events, specifying the desired satellite imagery to be collected at a revisit frequency of 1 hour, a ground sample distance of 1 m resolution, over a period of 14

days, between the hours of 6am and 6pm local time. These parameters may be altered as desired for more specific mission applications, but will not have a significant effect on the constellation design or reconfiguring decisions that are investigated in this work.

Table 5.1: An example target deck, consisting of the desired observation parameters for 20 events of interest occurring over a 5-year period

Event number	Event latitude [deg]	Event longitude [deg]	Event timing [years since mission start]	Duration of event [days]	Desired revisit frequency [hr]	Start of daylight hours [24hr clock]	End of daylight hours [24hr clock]	Desired GSD [m]
1	-33.1647	115	0.235452144	14	1	6	18	1
2	49.56269	108.3333	0.513384223	14	1	6	18	1
3	16.34871	105	0.82206289	14	1	6	18	1
4	31.12565	-90.8333	1.053947037	14	1	6	18	1
5	33.02325	-53.3333	1.1266568	14	1	6	18	1
6	-18.2162	-57.5	1.362483297	14	1	6	18	1
7	3.340536	-83.3333	1.567116898	14	1	6	18	1
8	19.92714	139.1667	1.806121182	14	1	6	18	1
9	27.78984	105.8333	2.198243997	14	1	6	18	1
10	34.92452	75	2.466719341	14	1	6	18	1
11	32.25029	-77.5	2.781293737	14	1	6	18	1
12	11.45422	99.16667	3.01506447	14	1	6	18	1
13	19.91067	115	3.35156814	14	1	6	18	1
14	39.43174	53.33333	3.570971787	14	1	6	18	1
15	35.38766	117.5	3.732875319	14	1	6	18	1
16	-13.4694	47.5	3.994842292	14	1	6	18	1
17	8.10348	-75.8333	4.327408069	14	1	6	18	1
18	35.31898	114.1667	4.578569801	14	1	6	18	1
19	18.09824	109.1667	4.781381663	14	1	6	18	1
20	40.78005	-71.6667	4.963216647	14	1	6	18	1

In order to apply importance sampling to the use of target decks for designing under uncertain conditions, and to improve model efficiency by reducing computational runtime, sampling resources must be prioritized to those inputs with the greatest effect on output variance. The process of uncertainty quantification demonstrates a drastically different impact of latitude variation vs longitude variation on constellation performance estimates from the ReCon model, and these effects are described at length in Section 5.2.1.

5.2.1 Latitude versus longitude importance sampling

It is important to accurately model the desired latitude distribution for target locations of interest for a variety of reasons, such as supplying realistic input scenarios for the design optimization process in order to ensure that design variables are being optimized for operating scenarios that are authentic and representative. The most crucial reason is that there is a strong coupling of latitude distribution to inclination selection for the constellation, due to the orbital inclination constraining what coverage is achievable, especially at very high and low latitudes towards the limits of the coverage range. It requires a large amount of ΔV to change the orbital inclination in order to pass over latitudes outside of the designed range, therefore the constellation should be designed with the intention that such a maneuver will not be required, as it is highly likely to fall outside the limits of a reasonable ΔV budget. It is necessary to ensure that an appropriate inclination is selected for the constellation at the design stage, so that all potential latitudes of interest are included in the range of coverage. This will avoid cases where a decision would have to be made whether to ignore a location of interest due to it falling outside of the coverage range (resulting in a significant drop in constellation performance when no orbital coverage occurs over a high-value target), or to maneuver despite the fact that the location is outside of planned-for parameters, resulting in unexpectedly high ΔV usage during the mission lifetime, which would have a significant negative effect on the remaining ΔV budget for future reconfigurations.

Accurately modelling the desired longitudinal distribution for target locations is less important, as longitude does not have a similarly strong coupled effect to orbital parameters (and therefore achievable coverage) as latitude does. Any desired longitude will eventually be achieved simply by waiting, but the length of time required to get there is a combined effect of the randomized target longitude, the random timing of each event occurrence, and the constellation phasing (changing ground tracks throughout the day). These factors combine to result in an essentially randomized

length of time taken to achieve longitude coverage at the start of an event of interest. This results in an operator choice as to how to balance length of time to achieve coverage against ΔV usage. Any desired longitude can be passed over either by using more time (with the consequence of a delay in achieving imagery, but at the benefit of lower ΔV use to do so) or more propellant (with the consequence of using some ΔV to conduct a maneuver, but with the benefit of passing over the desired longitude sooner).

Although accurate modelling of the longitudinal distribution is less important, the target distribution cannot be simply collapsed to latitude-only (creating a one-dimensional distribution) as information about the constellation performance is lost by not considering how coverage is achieved over two dimensions. This would lead to a supposedly optimal constellation design that may then supply completely different coverage than that simulated, once placed in a real-world target scenario. Using a single dimension to try to model ground coverage fails to reflect the reality of many orbits, including cases of polar and equatorial orbits, as well as the highly unlikely but theoretically possible scenario where every desired target lines up perfectly along the course of a potential repeating ground track orbit.

An example target deck (19 locations of interest for a 5-year mission lifetime) is shown in Figure 5-2, with the upper plot showing these locations plotted as red circles and overlaid onto a world map to illustrate the actual geographic distribution of these targets. The lower plot shows the locations of interest plotted as blue circles on a plain background, separating out the distribution from the map so that it can be considered alone without the distraction of continental features underneath. These same locations are plotted again in Figure 5-3, with the left-hand plot showing only the distribution in longitude and the right-hand plot showing only the distribution in latitude. These plots highlight the pathological cases mentioned above, where no distinction can be made between an equatorial orbit and the longitude-only data shown in Figure 5-3a, or between a polar orbit and the latitude-only data shown in

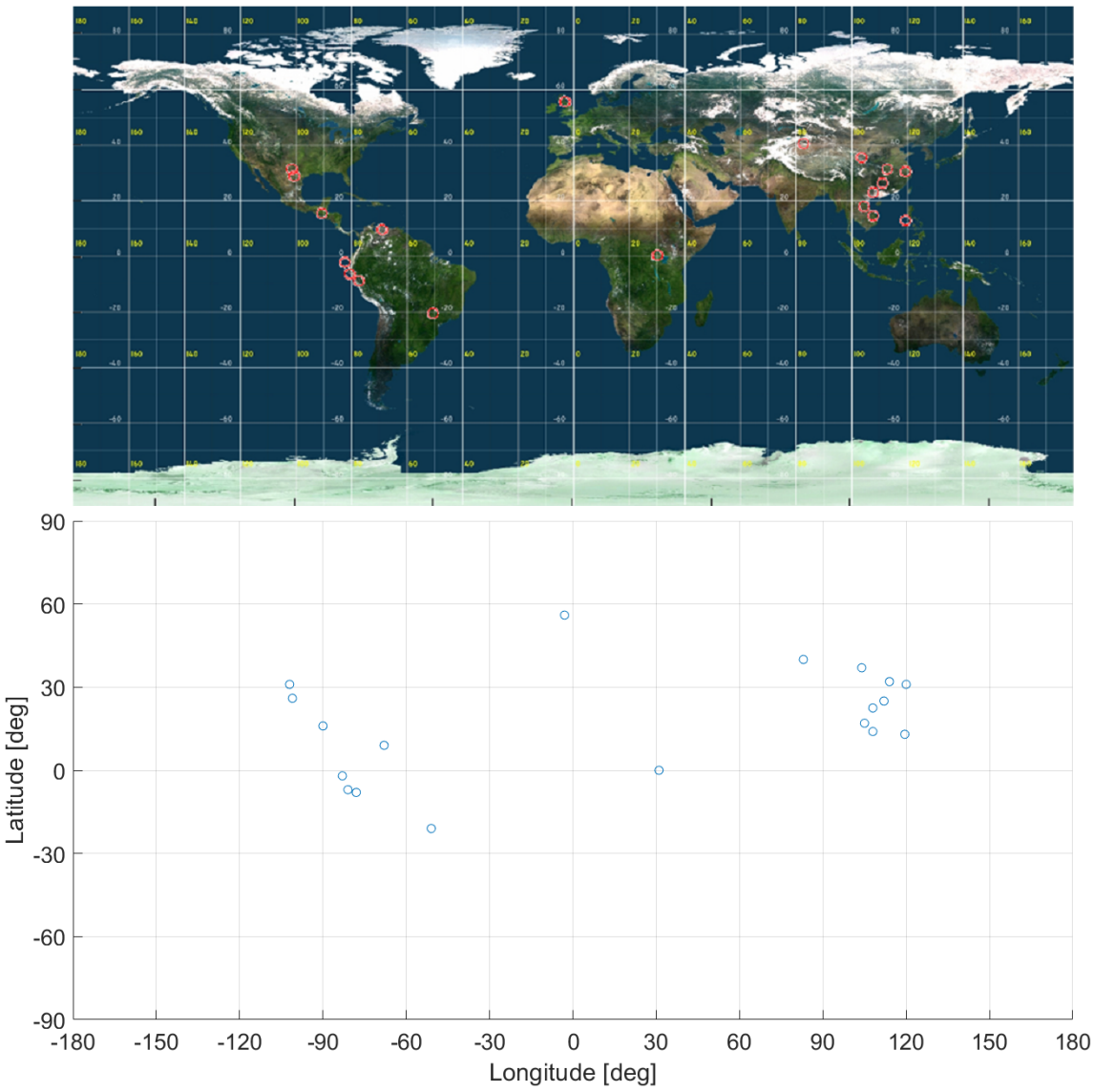
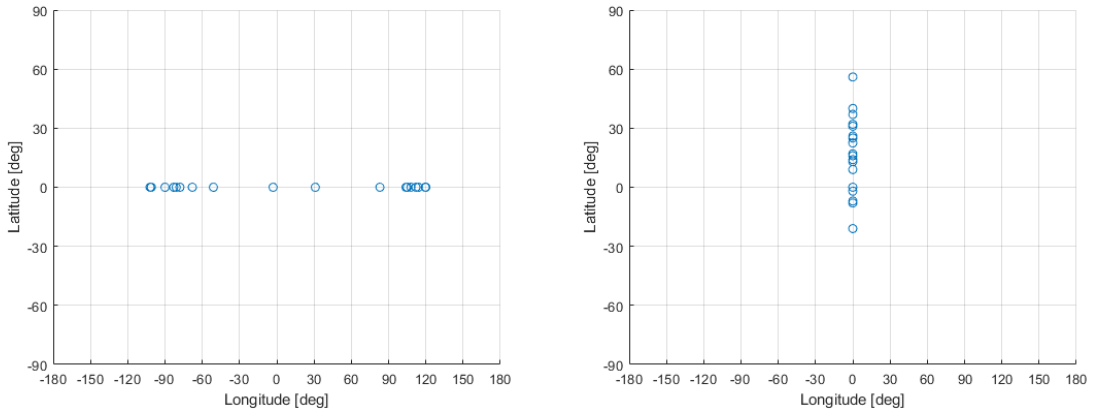


Figure 5-2: 19 target locations taken from a single example target deck, overlaid onto a world map (upper plot, targets shown as red circles) and shown separately against a plain background (lower plot, targets shown as blue circles)

Figure 5-3b.

The latitude and longitude data for the 19 example target locations was separated into latitude-only and longitude-only data and each of these data sets was sorted in ascending order, before being re-plotted in Figure 5-4. This figure illustrates how information is lost when only one dimension of the locational information is considered. Sorting the data in this fashion also shows how the locational data can be shaped into



(a) Longitude-only location distribution (b) Latitude-only location distribution

Figure 5-3: The same 19 target locations illustrated in Figure 5-2 are plotted as a longitude-only distribution (left plot (a)) and a latitude-only distribution (right plot (b)) to highlight the loss of information in only considering a single dimension of the location PDF

a curve that may start to resemble a satellite ground track, providing an additional pathological case that must be avoided during the course of the constellation design optimization.

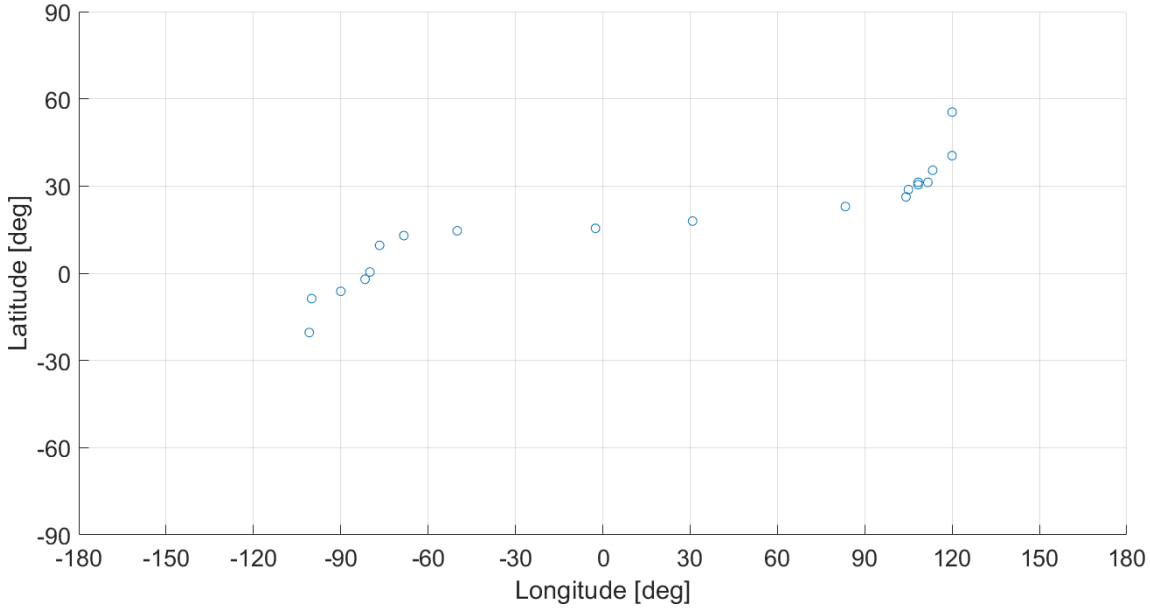


Figure 5-4: The location data from Figure 5-2 is replotted with the latitude and longitude co-ordinates now separately sorted in ascending order

Figure 5-5 shows the wide range of possible target locations that could be reconstructed from the disconnected latitudes and longitudes taken from the original set of target decks that was plotted in Figure 5-2. In Figure 5-5, the latitude of each location is plotted as a distinct color, showing a horizontal line of the scattered possibilities for the associated longitude co-ordinate, starting with light green circles for the northernmost target location (at 55.4°N) and continuing all the way down to blue circles for the southernmost target location (at 20.4°S). The associated longitude data for the 19 locations shown in the figure ranges from 100.8°W to 120°E.

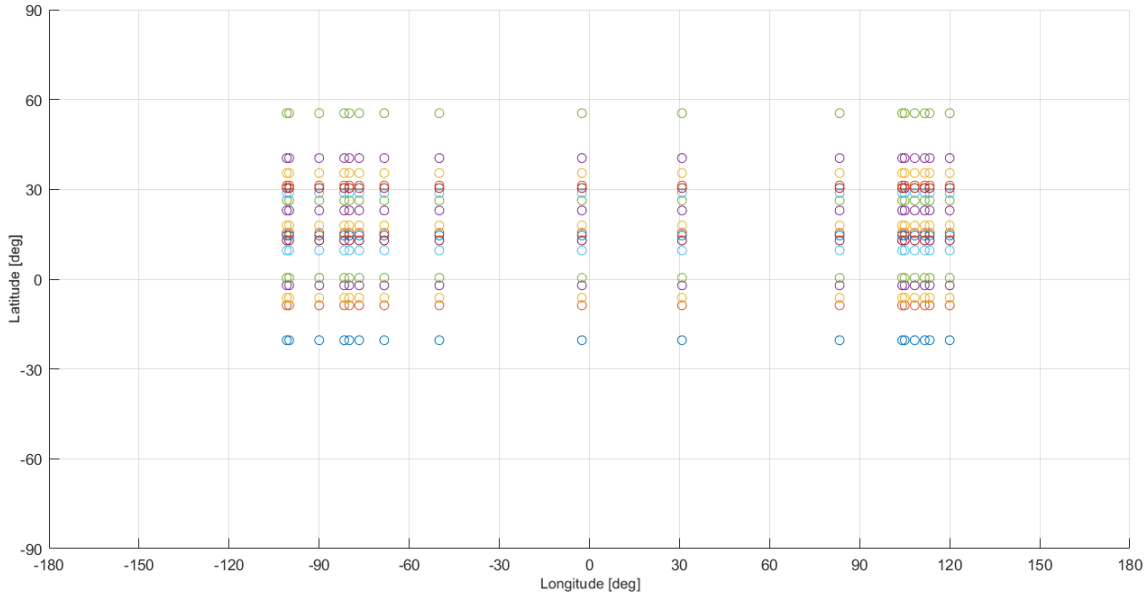


Figure 5-5: The location data from Figure 5-2 is replotted to show the range of possible target locations that may be recreated once the latitude and longitude distributions have been disconnected

These figures are used to illustrate the need to include associated longitude data alongside the latitude data, in order to avoid the loss of information necessary to determine the constellation's ground coverage performance in multiple dimensions. It is more important to prioritize accuracy in the modelling of the latitude distribution as this will have a much greater effect on constellation performance due to the strong coupling between orbital inclination and the bounds of latitude coverage, but associated longitude data must still be included in the target locations used to opti-

mize the constellation design. However, due to the lack of coupling between longitude coverage and constellation design parameters, employing a simple uniform longitude distribution in the location sampling is sufficient to avoid the pathological cases that occur with the use of latitude-only location data. Implementing the simplest possible representation of the longitudinal distribution avoids the need to use additional computational resources in modelling longitude when generating the target decks used with the ReCon model, instead devoting the available resources to improve accuracy in the latitudinal modelling. As long as the longitude distribution is randomly sampled in order to generate co-ordinates to pair with each sampled latitude co-ordinate and these co-ordinates are not numerically sorted, the probability of randomly generating a target deck that matches a possible repeating ground track orbit has an insignificant likelihood of occurring. This probability is then reduced even further by the use of multiple target decks to optimize the constellation design, avoiding the potential pathological cases highlighted here.

5.3 Non-parametric statistics

Non-parametric statistics is the branch of statistics which is not based on parameterized families of probability distributions (i.e. defined by known parameters such as mean and variance).[114, 115] It may be defined instead as a function of sampling, using distribution-free methods that make no assumptions about the probability distribution necessarily conforming to a specific template (such as a Gaussian or uniform distribution).[114] With a non-parametric model, the probability structure is not determined in advance but is instead established from the data.[116, 117]

For an application such as natural disaster monitoring, it is possible to approximate a probability distribution for certain kinds of events when these are separated out by type, such as earthquakes (mostly commonly found along tectonic plate edges or fault lines) or hurricanes (generally found within certain latitude bands and temporally associated with 'hurricane season'). However, when more general disaster

monitoring of multiple types is combined, it becomes difficult to mathematically estimate an appropriate distribution based on any specific pattern, other than simply using historical data. Using datasets derived from the locations of past disasters will be sufficient to supply a general overview, but with changes in climate over time and increasing incidences of extreme weather events, historical information is unlikely to provide a full picture of future event distribution. When human-made disasters are included (such as oil spills, building collapses, some wildfires, and even some earthquakes in areas where hydraulic fracturing or mining are carried out), the distribution becomes even less easy to predict.

However, under these circumstances, using historical data on disaster types of interest is likely to provide the most accurate information available compared to any other form of estimated distribution. Using historical data on natural disasters as the input for the model, a non-parametric estimate may be made to approximate the underlying distribution, without the need for assumptions to be made to arrive at an advance definition of parameters such as mean or variance from the location data. Histograms are one example of a simple non-parametric estimate of a distribution; kernel density estimation (KDE, described in more detail in Section 5.3.1) is another related method of estimating the PDF of a variable of interest, functioning similarly to histograms but converging much more quickly to the underlying probability density.[114, 115, 116, 117]

Using the aforementioned World Bank data[92] on natural disaster hotspots around the world, a KDE can be calculated to supply a considerably less discretized approximation of location data when compared to histograms created from the same data array.[115] The objective of using non-parametric statistical methods to characterize the PDF of this natural disaster location data is to arrive at the most accurate and undiscretized representation of the underlying distribution, to supply an accurate basis for the sampling that must then be carried out in order to generate target decks. Achieving a high degree of accuracy in how well these target decks represent the un-

derlying data is crucial, as the target locations of interest provide the input against which reconfigurable constellation performance is judged. This can be thought of as another area of focus within the UQ process carried out in this work; locating sources of uncertainty in the inputs to the ReCon model and working to improve these wherever possible.

Modelling assumptions could be made about the potential PDF of future events in various ways, for example by incorporating simulations of future changes in climate or increases in extreme weather in certain areas. Data for disasters with changing incidences over time (such as wildfires in areas suffering from drought, or flooding in areas with rising sea levels) could be weighted to more heavily emphasize datasets from recent years over historical data that may no longer accurately represent the current disaster distribution. The PDF of other disaster types such as (natural) earthquakes will change on longer geological timescales, and for these cases, historical data will remain a more accurate predictor for event modelling. In this thesis, historical disaster data from the World Bank is used (following the work carried out by Robert Legge) to create the PDF for randomly generated future events of interest, but the question of how to model changing future disaster probabilities could be an entire thesis in itself.

5.3.1 Histograms and kernel density estimations

Density estimation is the process of approximating a PDF which cannot be directly observed. The sample provided by the available observed data (usually assumed to be a random sample from the larger unobservable population) is used to construct an estimate of the underlying distribution. One straightforward and widely-used type of density estimator is a histogram, which makes no assumptions about the underlying distribution (therefore providing a non-parametric estimate, as discussed in the previous section) but simply divides the data range into an appropriate number of discrete bins, counts the datapoints that fall within each bin, and provides a simple visualization of the spread and shape of sample data. Basic histograms often supply a

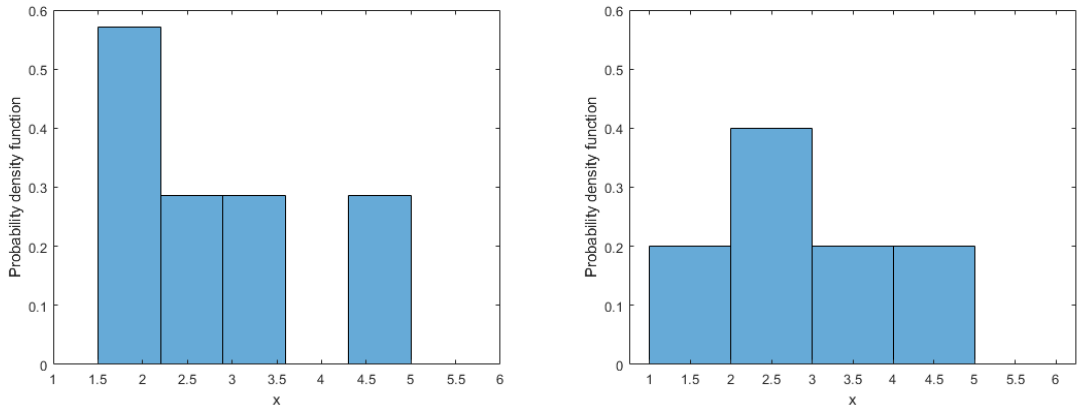


Figure 5-6: Two normalized histograms plotted using the same data set of 5 points but different bin widths and locations, to illustrate the contrasting impressions that are given of the underlying PDF

simple count of datapoints per bin on the y-axis, though this may also be normalized by relative probability so that the area of each bar equates to the relative proportion of observed data points. This supplies a somewhat discretized (due to the selection of uniform bin widths) form of PDF estimate, where the sum of all bar areas equals 1.

One issue with using histograms to estimate the underlying PDF is that the selection of bin size or the location of bin edges can lead to drastically different approximations made from the same data set. A very simple set of five data points was generated with values 1.90, 2.15, 2.24, 3.57 and 4.78. This data was plotted as a histogram in MATLAB twice, as shown in Figure 5-6. The histogram on the left in Figure 5-6 was generated by assigning the number of bins to 5 and allowing the MATLAB function to automatically select the bin width, and the histogram on the right was generated by assigning the bin width to 1.0 and allowing the MATLAB function to automatically select the number of bins. Without seeing the individual data points, it is not immediately clear to the observer that these histograms are visualizing the same data, and the underlying PDFs do not appear to be following the same shape.

A kernel density estimation is another non-parametric method of estimating the

probability density function of a random variable of interest. They are related to histograms in that both provide a way to visualize the spread of sample data and the shape of its underlying distribution without relying on advance assumptions about the character of that distribution. KDEs use a kernel function (frequently a standard normal density function, although other functions may be used if they are deemed more appropriate) and a bandwidth/smoothing parameter to improve the smoothness and continuity of the resulting estimation in comparison to the discrete nature of a histogram. This bandwidth parameter specifies the width of the kernel functions used to build the KDE, with larger bandwidths resulting in a smoother overall curve and smaller bandwidths resulting in a larger number of oscillations in the resulting KDE. Some pre-programmed KDE functions implement automatic bandwidth selection to avoid issues of under- or over-smoothing, although these are often based on rules of thumb that rely on assumptions about the underlying PDF, making them incompatible with attempts to approximate non-parametric density estimations. Without the use of such assumptions, any selected bandwidth value must be validated against the data set under consideration, to ensure that the generated KDE is not over-smoothed (losing features that are present in the actual distribution) or under-smoothed (creating additional curves in the estimate that are not present in the underlying PDF, as an artifact of the kernel functions used to arrive at the KDE).[118]

A comparison of the two histograms shown in Figure 5-6 above with a KDE generated from the same five data points is shown in Figure 5-7 below. The process of creating the KDE (shown as the blue line in the right hand plot in Figure 5-7) is illustrated here, with kernel functions (shown in red) placed centered at each of the five data points from the sample. These five red kernel functions are summed to arrive at the kernel density estimation shown in blue, which supplies an approximation of the underlying PDF. The KDE shape can be compared to the two histograms on the left to observe a more accurate reflection of the distribution shape, with much less variance occurring due to differences in sampling or binning.

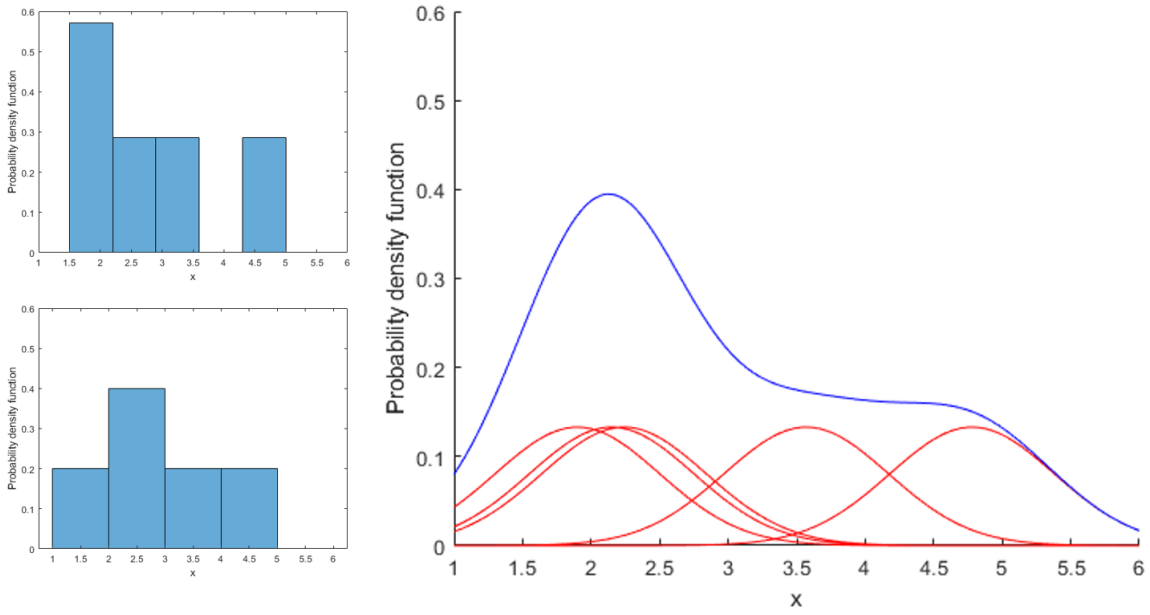


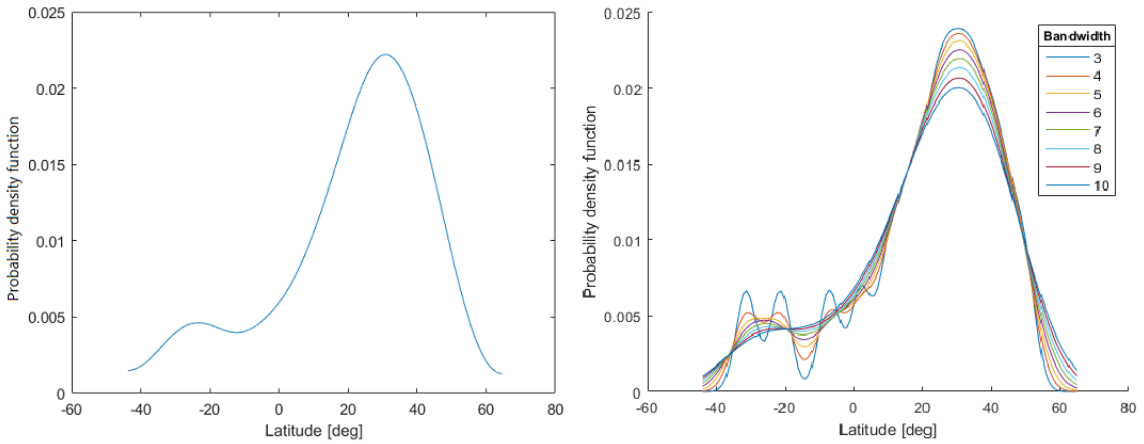
Figure 5-7: Comparison of the two histograms previously shown in Figure 5-6 (left) with a KDE (right) created using the same data set

As Figure 5-7 shows, KDEs are able to supply a much smoother estimate of the underlying PDF for continuous random variables than histograms can, converging much faster to the actual density of the distribution even using only a few samples. Due to the data binning that must occur to create a histogram, discretization issues with the PDF estimated using a histogram are much harder to avoid without modifying the technique itself into something inherently different. Using KDEs to estimate the underlying latitude distribution for events of interest will avoid these discretization issues and provide a more accurate probability estimation to sample for event locations of interest.

5.3.2 Kernel density estimation implementation

In-built MATLAB functions for generating KDEs are available, but these include automatic bandwidth selection by default, based on commonly-used rules of thumb that make assumptions about the underlying distribution and are therefore not useful for the present case of estimating a PDF non-parametrically. These functions also generate only the overall curve of the KDE, and do not include details of the individual

Gaussian kernel functions used to build it. Parameters and placement of these kernel functions are of interest in this work, due to the investigation of alternative sampling methods with the objective of reducing computational runtime for the ReCon model. Due to the need for additional information, a custom function was created in MATLAB to generate KDEs while providing more control over the parameters and retaining finer detail in the results of the estimation.



(a) KDE with automatic bandwidth (b) Custom KDE with a range of bandwidths

Figure 5-8: Comparison of two KDEs generated from natural disaster latitude data, with (a) the left-hand plot created using a built-in MATLAB KDE function featuring automatic bandwidth selection, and (b) the right-hand plot created using custom code and featuring a range of possible bandwidth values

Figure 5-8 shows a comparison of two KDEs generated from the natural disaster PDF latitude data, both using Gaussian kernel functions. An existing MATLAB function using automatic bandwidth selection was used to generate the left-hand plot Figure 5-8a, and this was found to select a bandwidth value of 7.0. The right-hand plot Figure 5-8b was generated by the custom KDE function written as part of this work, showing the effect of different bandwidth values on the resulting KDE curve. The green curve for a bandwidth of 7 in the right-hand plot can be compared to the left-hand plot, validating the custom KDE code against the existing MATLAB function by replicating the same results for this case.

The effects of under-smoothing can be seen in the custom KDE plot in Figure 5-8b where some of the plotted bandwidth values are too low to be useful; for bandwidths of 3–5, additional curves are present in the estimated PDF that do not exist in the underlying latitude distribution data, because the shape of the individual kernels is narrow enough to start becoming visible even in the stacked overall curve. The effects of over-smoothing can also be observed in Figure 5-8b, where some of the plotted bandwidth values are too high to provide a useful estimate; for bandwidths of 9–10, the second lower peak at the left side of the bimodal distribution is smoothed out to the point where this second mode is lost from the shape of the estimated PDF, due to the extremely wide overlap between the individual kernels that are summed to arrive at the KDE.

Figure 5-9 shows the breakdown of the latitude KDE into the individual Gaussian kernels that are stacked to arrive at the overall estimation curve, which is enabled by use of the custom KDE MATLAB function that was created for this work. The overall probability density function on the y-axis is defined to give an area of 1 under the entire KDE curve. This KDE was generated using a bandwidth of 7.0, which means that each individual Gaussian kernel is generated as a normal distribution using a standard deviation of 7.0. As 20 kernels are used to arrive at this KDE, each individual kernel accounts for 5% of the latitude distribution. The centerpoints of each kernel (the mean of each normal distribution curve) are placed at the mean latitude values from each of these 5% bins of latitude.

To give an example: the southernmost 5% of the latitude distribution falls between 43.7°S and 25.4°S, with the probability distribution skewed towards the northern end of this range, resulting in a mean value of 32.3°S and a median value of 31.2°S. This results in the leftmost kernel function in Figure 5-9 being generated using a mean of -32.3° latitude and a standard deviation of 7°, and with the amplitude selected to give a total area under the curve of 0.05 (5% of the total probability).

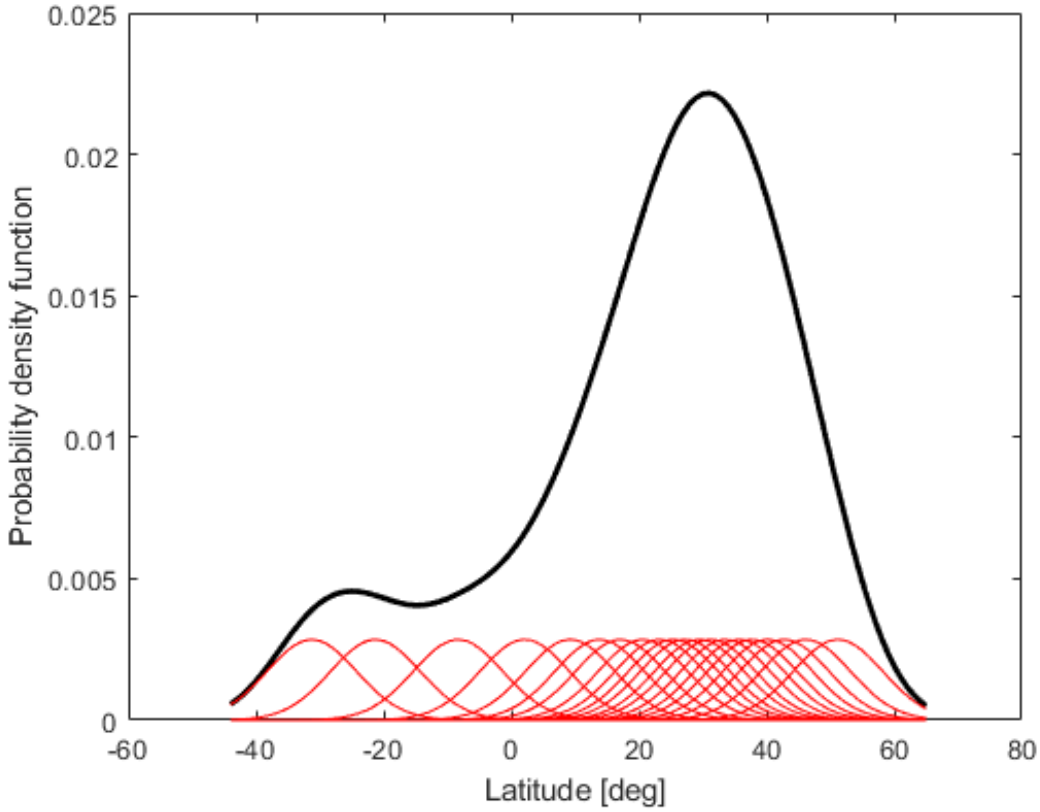


Figure 5-9: KDE of latitude data (plotted as the thicker black line) generated using a custom-built MATLAB function, also illustrating the 20 constituent kernels (plotted as the smaller red curves) that were stacked to build up this estimation of the underlying latitude PDF

For comparison to the KDEs shown previously, Figure 5-10 shows four histograms created from Legge's entire original target deck array of natural disaster location latitude data used as input for the ReCon codebase, plotted using a range of bin widths from 1–10° of latitude. The y-axis shows the number of occurrences of each location within the target deck array of 195,301 locations. The histograms created using 1 and 2° latitude bins show jagged estimations of the underlying PDF, highlighting the histogram discretization issues that were previously described, where artifacts of the binning process create visible spikes in the estimation that are not present in the underlying distribution. In these cases, the choice of bin width and the placement of bin boundaries can have a significant effect on the visualization of the estimated latitude PDF. The 5 and 10° latitude bins are noticeably less discretized, but it can be seen

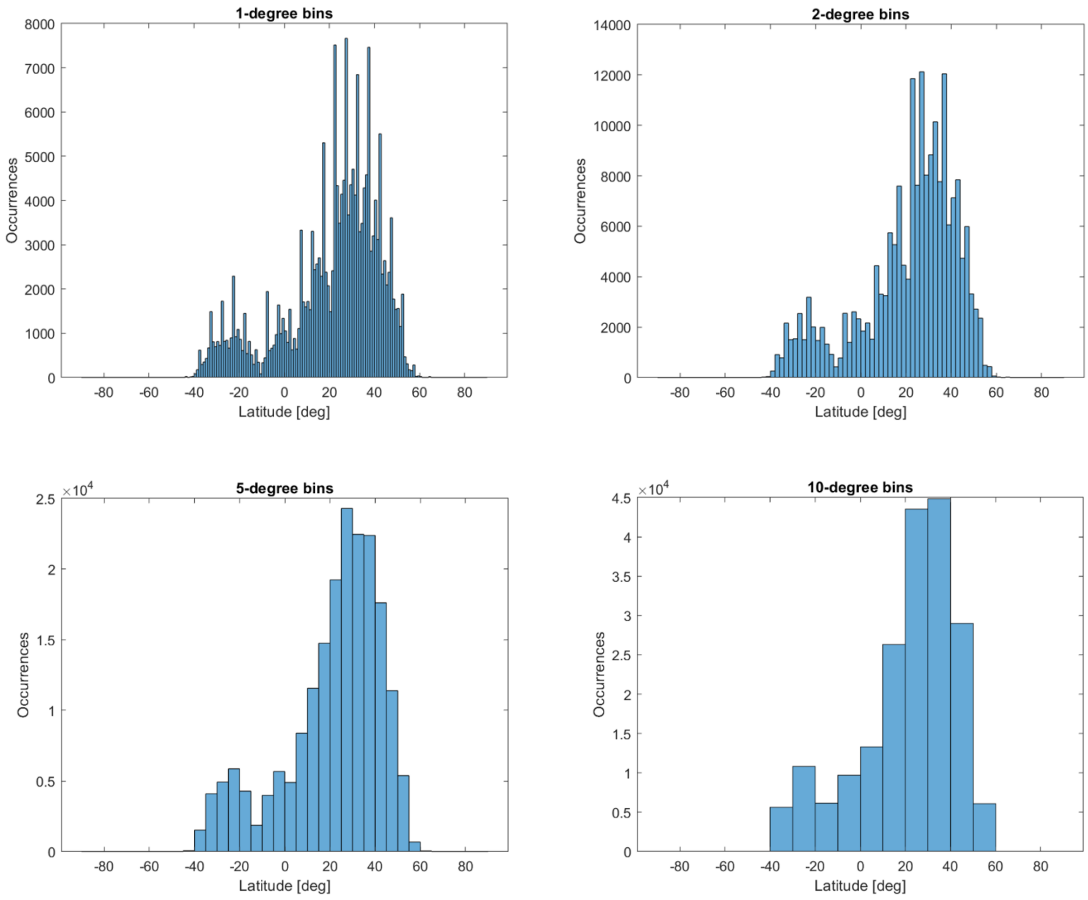


Figure 5-10: Four histograms of latitude data plotted using various bin widths (1/2/5/10° of latitude), illustrating the discretization issues inherent to histograms and the difficulty of converging to a smooth and accurate estimation of the underlying PDF

from the lower two plots in Figure 5-10 that finer details of the estimated distribution shape are gradually lost as the histogram moves towards wider binning, for example the way that the smaller left-hand peak is gradually beginning to merge into the slope of the larger right-hand peak in the distribution as the bin width increases.

Figure 5-11 shows a three-dimensional histogram generated from the World Bank natural disaster location data used by Robert Legge to generate 195,301 target locations, with the y-axis showing the frequency of events out of this total. A 3D histogram is used to represent both dimensions of the original data, showing both

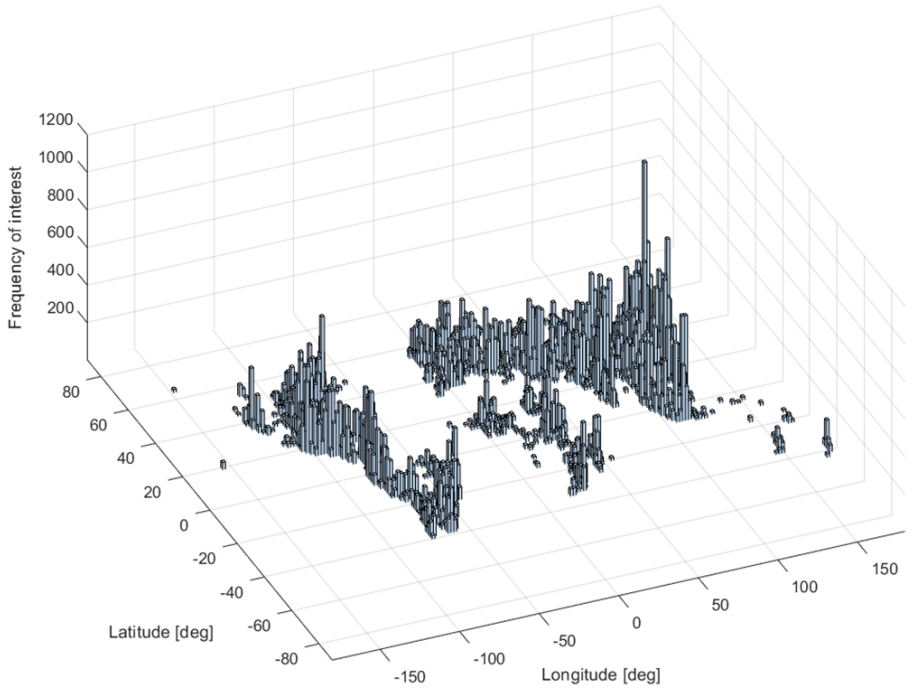
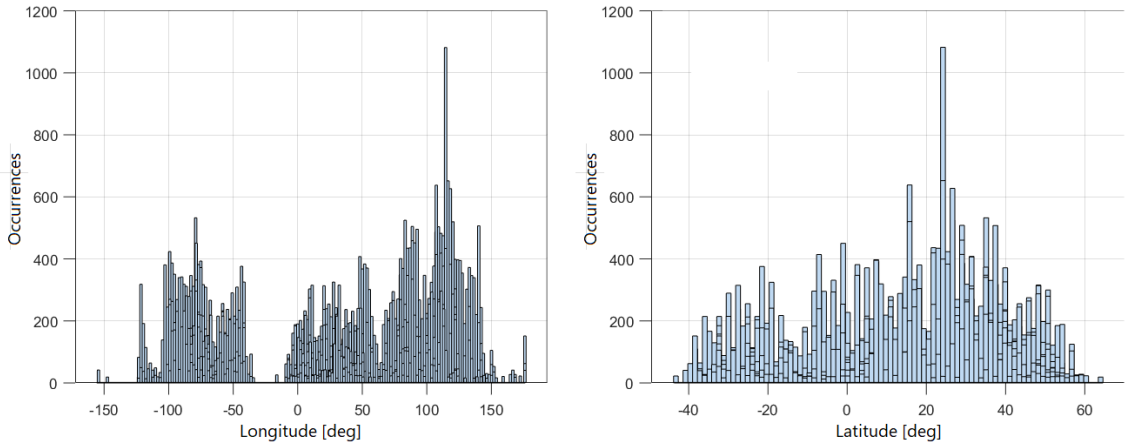


Figure 5-11: 3D histogram of natural disaster location data (weighted by economic impact) generated using Legge's original array of target decks

latitude and longitude data; parts of the landmasses on this diagram are recognizable as a world map (e.g. North and South America on the left side of the plot, and Asia in the top right) although some continents are under-represented compared to the geographical reality, either due to lower economic activity or lower instances of natural disaster occurrence in these regions. The background grid is removed from this figure to leave only more widely spaced grid lines, due to difficulty in distinguishing the details of the data when a higher resolution background grid is used.

This 3D histogram can be panned and rotated in its original interactive form on a computer, but due to the impossibility of presenting such a figure in a written document, an isometric view is selected for Figure 5-11 to present an overview of the information, as well as three orthographic views in the following Figures 5-12 and 5-13. The longitude-only and latitude-only axis views are shown in Figures 5-12a and 5-12b, although for clarity it must be observed that these are only single-axis orthographic views of the 3D histogram, where several vertical bins along the same



(a) Front-on view of the longitude axis

(b) End-on view of the latitude axis

Figure 5-12: Orthographic views of the 3D histogram shown previously in Figure 5-11, with a front-on view of the longitude axis shown on the left in Figure 5-12a and a end-on view of the latitude axis shown on the right in Figure 5-12b

axis may obscure one another. These views are not one-dimensional histograms of the latitude and longitude data, as the bins along the front-on axis in the view are simply lined up, rather than being combined into a single bin of data points as would occur in a one-dimensional histogram. Figure 5-13 shows a top-down orthographic view where the difference in disaster frequency is no longer visible, simply highlighting all locations with any occurrence whatsoever of natural disasters. Once again, the world map is partially recognizable in this figure, with the North and South American continents visible on the left of the plot, the Eurasian landmass visible in the top right, and parts of Africa visible in the center.

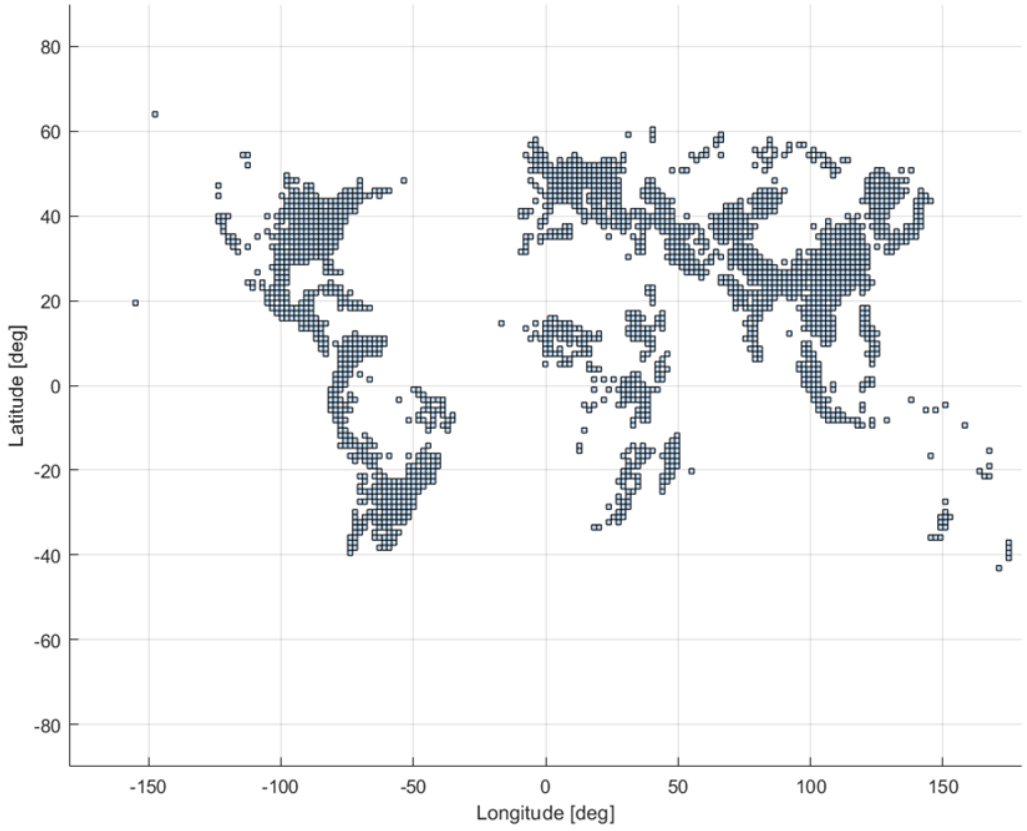


Figure 5-13: A top-down flattened orthographic view of the 3D histogram from Figure 5-11, showing all geographic locations with any level of natural disaster activity

In order to enable a more complete comparison between the use of histograms and KDEs as non-parametric methods of PDF estimation, a two-dimensional KDE was generated from the natural disaster location data. This was used to generate a three-dimensional surface plot, shown in Figure 5-14, with the vertical axis representing the probability density function (i.e. the total area under the plot equals 1). Although heatmap coloring is used in this surface plot, the high resolution of the plotted data means that this is hard to observe other than at the very highest peaks of the PDF (located in East Asia), shown on the right hand side of the plot. In the third orthographic view of this 2D KDE shown in Figure 5-16, magnified insets are used to highlight areas of this heatmap.

Similarly to the previous 3D histogram shown in Figure 5-11, the 3D surface plot was generated as an interactive graphic that can be panned or rotated on a computer,

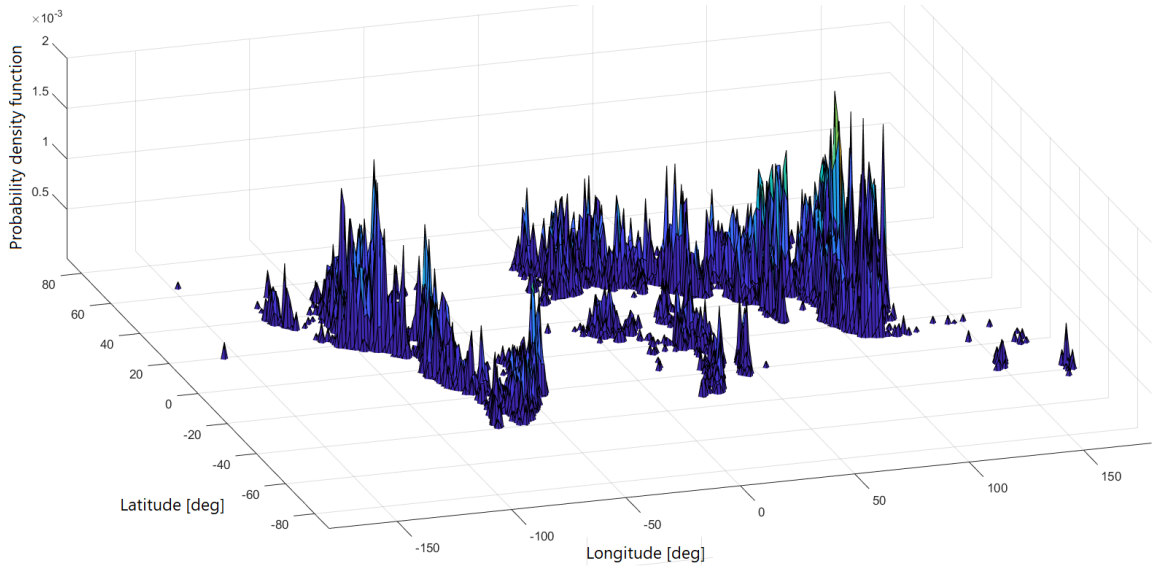
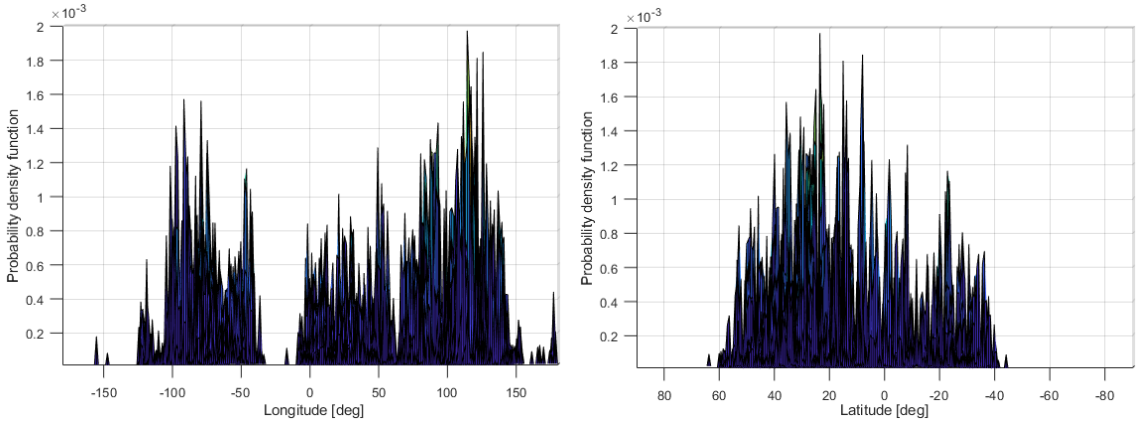


Figure 5-14: A 3D surface plot generated from a 2D KDE of natural disaster location data

but an isometric view was selected to provide the overview shown here in Figure 5-14. Once again, details of the continents are somewhat recognizable in this plot, with the American landmasses shown towards the left, parts of Africa shown in the center, and Eurasia in the upper right sector. Orthographic views are presented in the following figures, with Figure 5-15 showing longitude-only (on the left in Figure 5-15a) and latitude-only (on the right in Figure 5-15b) axis views. As was previously observed for the orthographic views of the 3D histogram, it must again be highlighted that these are only single-axis orthographic views of the KDE, where a series of spikes along the same latitude or longitude line may obscure one another. The plots shown in Figure 5-15 are not equivalent to one-dimensional KDEs, as the peaks of the estimated distribution are simply lined up along one axis, rather than being combined into a single curve as would occur in generating a one-dimensional KDE along the same axis. Figure 5-16 shows a top-down orthographic view with heatmap coloring designed to highlight the difference in disaster frequency across the map, although this is difficult to observe at the data resolution available in this written document, and therefore two inset boxes on the right-hand side of Figure 5-16 are used to highlight two geographic locations at a higher magnification level as examples, so that the color map can be



(a) Front-on view of the longitude axis (b) End-on view of the latitude axis

Figure 5-15: Orthographic views of the 3D surface plot generated from the 2D KDE shown above in Figure 5-14, with a front-on view of the longitude axis shown on the left in Figure 5-15a and a end-on view of the latitude axis shown on the right in Figure 5-15b

distinguished in finer detail. These example location insets are used to show the natural disaster economic impact distribution heatmap for a major part of Asia (top right inset box) and the Caribbean (lower right inset box).

Due to the large number of kernels required to generate an accurate two-dimensional KDE of the location data, discretization issues were experienced with the process of sampling from this PDF. Where a relatively small number of samples are desired (e.g. a mean of 20 locations for a standard 5-year target deck) compared to the number of kernel functions used to generate the estimate, large rounding errors must be introduced in order to arrive at the number of samples desired. This resulted in sampling from a two-dimensional KDE of the natural disaster location distribution failing to improve the accuracy level in modelling the latitude data. The objective of these sampling improvements was to improve the accuracy of the latitude PDF in order to improve the efficiency of the design optimization process, and therefore it was concluded from the implementation issues of the 2D KDE that this technique was not suited to making such an improvement in the optimization code.

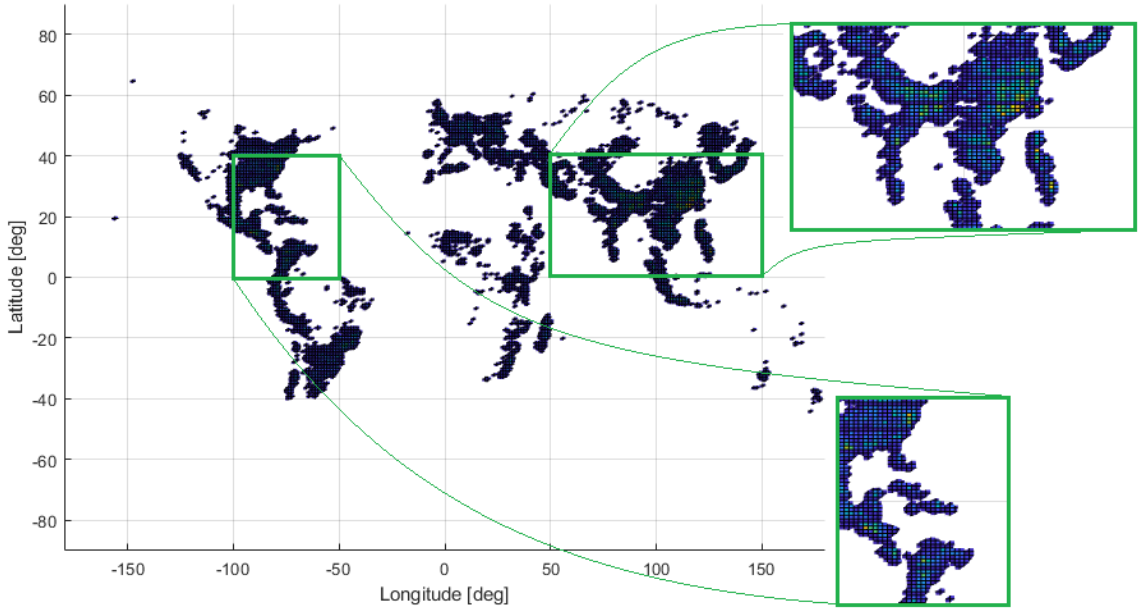


Figure 5-16: A top-down flattened orthographic view of the 3D surface plot generated from a 2D KDE that was shown in Figure 5-14, showing all geographic locations with any level of natural disaster activity. The inset boxes show magnified views of two locations in more detail, allowing for the detail of the heatmap color scheme to be observed at this enlarged scale

It was decided to prioritize the accuracy of locational sampling towards the latitude distribution, due to the previous conclusion drawn in Section 5.2.1 that the target latitude distribution has a strongly coupled effect on constellation optimization and resulting performance, while the target longitude distribution displays no such coupling. Modelling the longitude distribution as a uniform PDF and sampling it at random to generate longitude co-ordinates to pair with more accurately sampled latitude co-ordinates was concluded to be sufficient for avoiding pathological cases that would give highly inaccurate estimates for the constellation performance. Pairing a uniform longitude distribution with a one-dimensional KDE for the latitude distribution allows for sampling resources to be prioritized towards modelling latitude as this is the input parameter which results in the greatest effect on the simulation output. Due to the limitations imposed by the available computational resources and the intensive requirements of the ReCon model, making these changes to the sampling process for locations of interest allowed for the investigation of alternative sampling

methods and the comparison of confidence levels in the constellation performance estimates generated using different techniques. This created the possibility of achieving significant runtime improvements for the ReCon codebase while retaining the same level of confidence in the performance results predicted by the model.

5.4 Stratified sampling

Using simple Monte Carlo methods may require hundreds or thousands of functional evaluations of the computational model to achieve the desired level of accuracy in the estimated output (as described in Section 5.1). Using variance reduction techniques for Monte Carlo methods can lead to a significant reduction in the number of code executions required to arrive at a specified confidence level, when compared to the number required by the straightforward Monte Carlo method to reach the same level. Quasi-Monte Carlo techniques reduce the number of input configurations needed to gather adequate output data while more specifically directing input sampling to recreate representative properties of the full distribution. Some quasi-Monte Carlo techniques include the use of low-discrepancy sequences for sampling, where a deterministic set of quasi-random numbers are used to generate a more evenly distributed set of samples than would occur using uniformly random sampling. However this results in a derandomized algorithm which may not be suited to cases where the objective involves carrying out a less deterministic form of sampling.

Stratified sampling is one example of a variance reduction technique for Monte Carlo methods, which may be carried out across one dimension or several depending on the application. For two-dimensional distributions, Latin square sampling consists of dividing each axis into X equiprobable intervals, and then collecting X samples such that only one sample is taken from each row and column; this is essentially two-dimensional stratified sampling. Latin hypercube sampling is the extension of this technique for distributions with any desired number of dimensions, where each dimension is once again divided into a set of X equiprobable intervals, and X samples are

taken while ensuring that each equiprobable interval along each dimension of interest is represented only once in the sample space. Latin hypercube sampling methods are a way of ensuring that the sample set is representative of the actual variability of the underlying distribution, as opposed to random sampling which supplies only a sample set of random numbers with no guarantee of reflecting the actual variance of the distribution from which they were sampled.

Stratified sampling is a method of reducing variance in sampling; the precision is improved by reducing the amount of sampling error (where the selected samples are not representative of the entire population of data). After arriving at a PDF estimate for the variable of interest, this distribution must be 'stratified': the data is divided into subgroups (or 'strata') using one of the various techniques described in Section 5.4.1. These subgroups must usually be **collectively exhaustive**, to cover the entire range of the distribution of interest, and **mutually exclusive**, so that the same datapoint cannot be sampled from within multiple strata. Random sampling is then applied within each stratum, with the proportion of samples taken from each stratum determined by the stratification technique that was used. Stratified sampling is used to force the sampling distribution to cover the full range of the overall data distribution, reducing sampling error when compared to random sampling, which always has the chance of resulting in an unrepresentative sample that is poorly balanced compared to the underlying PDF of interest.

5.4.1 Stratified sampling techniques

There are a range of strategies for stratifying the overall population of data to be sampled. Several different techniques were considered for the process of stratification, and the details of these are described in the following sections. The benefits and limitations of each method are also considered, especially where these may result in problems with applying the technique to the specific application of latitude sampling in the ReCon model. These stratified sampling techniques are compared to Legge's original target decks, which were created using the PDF of natural disaster likelihood

weighted by economic impact published by the World Bank, which was then sampled using simple uniform random sampling to create individual five-year mission scenarios for assessment in the ReCon model.

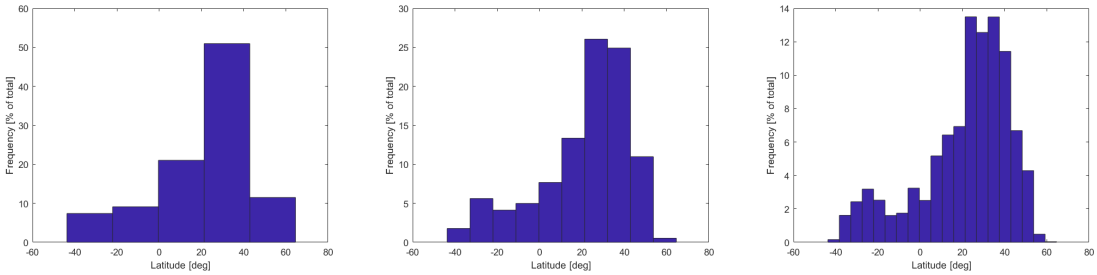
The sampling methods under consideration are summarized in the following list (with greater detail and examples given for each technique in the following subsections):

1. Legge's original target decks (random sampling, not stratified)
2. Proportionate allocation (equal sized bins, sampled proportionally to their share of the overall distribution)
3. Disproportionate allocation, or optimum allocation (equal sized bins, sampled proportionally to the standard deviation of each bin)
4. Proportionate definition (bins of equal probability, sampled equally)
5. Kernel function sampling (use kernel functions as data bins, sampled equally)

5.4.1.1 Proportionate allocation

In comparison to the original random sampling technique used by Legge, the first proposed method of stratified sampling is proportionate allocation, the most commonly used method of stratification.[115] For this method, the data is grouped into equally-sized strata, and then a sampling fraction is taken from each subgroup in proportion to its fraction of the overall distribution. First, the latitude data is divided into a number of strata, each of which is the same width in degrees of latitude and contains a different frequency of events according to the overall PDF. Then the desired number of samples is apportioned across each bin according to its frequency, and sampled randomly from within that stretch of the latitude range.

For example, considering the full latitude range used to generate target decks (minimum 43.72°S, maximum 64.61°N, full data range of 108.33° of latitude) and



(a) 5-bin latitude histogram (b) 10-bin latitude histogram (c) 20-bin latitude histogram

Figure 5-17: Histograms showing the PDF of natural disaster data by latitude, stratified into (a) 5 subgroups (b) 10 subgroups and (c) 20 subgroups

using proportionate allocation to select 20 samples (the number of events of interest in an average-sized five-year target deck):

- Dividing the data into 5 strata as shown in Figure 5-17a results in a width for each latitude bin of 21.67°. Moving left to right across these strata, the number of random samples taken from each bin is 2, 2, 4, 10 and 2 respectively.
- Dividing the data into 10 strata as shown in Figure 5-17b results in a width for each latitude bin of 10.83°. Moving left to right across these strata, the number of random samples taken from each bin is 0, 1, 1, 1, 2, 3, 5, 5, 2 and 0 respectively.
- Dividing the data into 20 strata as shown in Figure 5-17c results in a width for each latitude bin of 5.42°. Moving left to right across these strata, the number of random samples taken from each bin is 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 3, 3, 3, 2, 1, 1, 0 and 0 respectively.

Although the use of stratified sampling forces the sampling distribution to cover the full range of the overall data distribution, it can be seen from comparing the allocation of samples to each stratum that this technique suffers somewhat from discretization effects. Some rounding is frequently required to arrive at the exact number of samples desired. This also introduces a tendency for the lowest-probability latitude bins to be ignored entirely, meaning that no samples at all will be taken from certain latitude ranges. In cases where the selected numbers of strata and samples result in

bins from which zero samples are taken, this may introduce problems for the design optimization process.

For example, in the 20-strata case shown in Figure 5-17c, no samples are taken from 7 of the 20 latitude bins, resulting in zero sampling of more than a third of the latitude range. This includes zero sampling of 5 bins at the edge of the latitude range under consideration, artificially constraining event latitudes between a new minimum of 27.47°S and a new maximum of 53.78°N, thereby reducing the range of latitudes considered from 108.33° to only 81.25° and resulting in the loss of 25% of the original latitude span. Due to the effect of latitude selection on constellation inclination described in Section 5.2.1, this may result in the selection of an inclination at which the constellation cannot provide coverage of a large amount of the desired global range. Although proportionate allocation is the most commonly used method of stratification for this type of sampling, these discretization issues and the resulting effects on latitudinal sampling are significant enough to conclude that this technique is poorly suited to the present application, and after initial investigation into applying this method of stratification to the ReCon model, it was eliminated from consideration.

Optimum allocation (also known as 'disproportionate allocation') is another related method of stratification. In this technique, the strata for sampling are determined in the same way as for proportionate allocation, with subgroups defined of equal width (in this case, the same width in degrees of latitude). However, the sampling proportions are not determined by the proportion of the population that falls within each stratum, but instead by the standard deviation of the data in each subgroup. Higher proportions of samples are taken from the strata with the greatest variability, with the objective of minimizing the overall sampling variance even further than is achieved by proportionate allocation. However, upon further investigation, this technique was found to suffer from the same discretization issues as proportionate allocation, and so was not pursued as a viable method for use in this work.

5.4.1.2 Proportionate definition

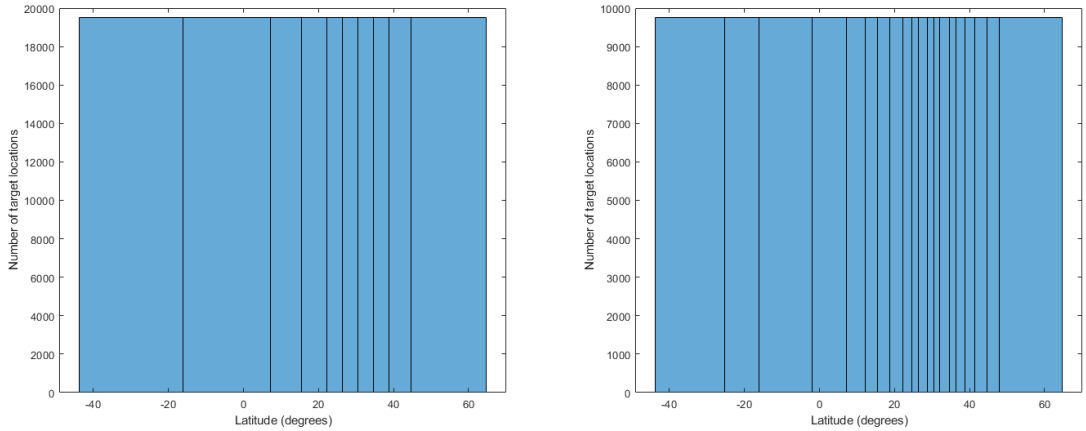
The next proposed method of stratified sampling is proportionate definition. For this technique, instead of creating strata of equal latitude width, the strata are defined using divisions of equal probability. These are known as equiprobable bins, which results in data groups with widely different boundary ranges, but which each contain an equal proportion of the PDF. This means that the sampling fraction taken from each bin will be equal.

For example, again considering the full latitude range used to generate target decks and using proportionate definition to select 20 samples for an average-sized target deck:

- Dividing the data into 10 equiprobable strata as shown in Figure 5-18a, it is straightforward to take 2 random samples from within the latitude range assigned to each bin. The bin widths vary from a minimum of 4.17° to a maximum of 27.5° (a factor of 6.6× between the most and least densely populated bins), but each stratum contains 10% of the overall latitude PDF.

- Dividing the data into 20 equiprobable strata as shown in Figure 5-18b, 1 random sample is taken from within the latitude range assigned to each bin. The bin widths vary even more widely, from a minimum of 1.67° to a maximum of 18.33° (a factor of 11× between the most and least densely populated bins), but each stratum contains 5% of the overall latitude PDF.

Using proportionate definition once again forces the sampling distribution to cover the full range of the latitude distribution, but in this case there are no lowest probability bins which end up unsampled (as occurred using the proportionate allocation technique), as each stratum is defined to contain the same proportion of the distribution. Concentration of the stratification around the most densely populated areas of the distribution results in definite sampling occurring within the highest probability ranges due to the narrow binning in these areas, whereas the least densely populated



(a) 10 equiprobable latitude bins (b) 20 equiprobable latitude bins

Figure 5-18: Histograms of natural disaster data by latitude divided into equiprobable bins, using (a) 10 bins each containing 10% of the distribution and (b) 20 bins each containing 5% of the distribution

latitude areas still have some probability of being sampled and no latitude ranges are entirely excluded, as occurred using proportionate allocation.

Discretization issues may still be introduced depending on the selected number of bins and samples, e.g. if 17 samples are required from 13 bins, there is no method to assign where the sampling should be carried out without rounding, which results in discretization. However, this situation can be avoided by modifying the proportionate definition technique to define the number of strata as equal to the desired number of samples. One caveat must be noted here, which is that if a very large number of samples is desired, using this modification of the technique will result in over-constraining the sampling. For examples, if hundreds of samples are desired and all of the bins are defined with extremely narrow ranges, the random element of the sampling will approach zero as such a high level of stratification will result in essentially one pre-defined latitude value for each bin.

5.4.1.3 Kernel function sampling

The third and final proposed method of stratified sampling is kernel function sampling. As described in more detail in Section 5.3.1, a kernel density estimation is a non-parametric estimation of the PDF of a random variable of interest. Although somewhat similar to histograms, rather than binning the data, a KDE is created by stacking kernel functions to arrive at a less discretized estimate of the PDF. For this technique, it is proposed to use these individual kernels as strata. The most obvious downside to such an approach is that it contravenes the usual rule for stratification highlighted at the start of Section 5.4: strata should be collectively exhaustive and **mutually exclusive**.

In this case, the kernel function strata are collectively exhaustive, because they are allocated across the entire variable range of interest according to probability density and stacked to arrive at the estimate of the full distribution. However, the standard normal density functions used as kernels are overlapped across the distribution, then summed to arrive at the KDE. Therefore, the strata would not be mutually exclusive, although the probability of samples being taken from the centers of the kernels is much higher than from the overlapping tails of the kernel functions.

The general concern expressed in literature for the idea of using overlapping strata is that individual datapoints which fall within the overlapping region would have a higher chance of being sampled compared to datapoints that only fall within a single region.[114, 115] This issue is solved by the structure of kernel sampling: samples are not taken by simple random sampling over the full width of the strata, but weighted by the shape of the Gaussian distribution in each kernel, meaning that datapoints falling within the tails of the kernel functions have a much lower chance of being selected from within the data range of that stratum. When the kernel functions are summed, an accurate estimation is made of the overall latitude PDF, illustrating how stacking multiple smaller probability curves does not result in inaccurate probabilities

of sampling the overlap. Instead, this cumulative probability weighting results in a sampling distribution that accurately reflects the underlying distribution, rather than providing a skewed likelihood of particular datapoints being selected, as would occur if random uniform sampling was carried out on these strata.

The argument made here to justify the inclusion of this sampling approach is based on how a KDE is created. Given that the estimate of the PDF under consideration was created by stacking multiple Gaussian distributions, the set of kernel functions must by definition represent the entire range of the data distribution, with each individual Gaussian PDF in turn representing an appropriate probability range for its area within the overall distribution. Therefore in this method, the existing kernel functions will be used to supply pre-defined strata for sampling, and upon reviewing literature, it was discovered that this technique is proposed elsewhere as a sampling method.[118, 119]

As shown in Figure 5-19, each kernel function is a small normal distribution PDF plotted along the latitude axis. For an example case where 20 samples are desired, a KDE is used made up of 20 kernel functions, to avoid discretization issues that occur where there is a mismatch between the number of strata and the number of stratified samples to be generated. A single sample is randomly selected from within each of these 20 strata kernel functions, and although the strata do overlap, the entire data range is covered. The centerpoints of each kernel distribution (highlighted with black arrows in Figure 5-19) are placed at the median latitude value for each 5% of the latitude range, and it can be seen how the kernel functions are unevenly spaced along the latitude axis to accurately approximate the probability density across the entire latitude distribution, meaning that sampling is still forced to cover the full range of latitudes by the use of these non-exclusive strata, with samples concentrated in the area of greatest frequency.

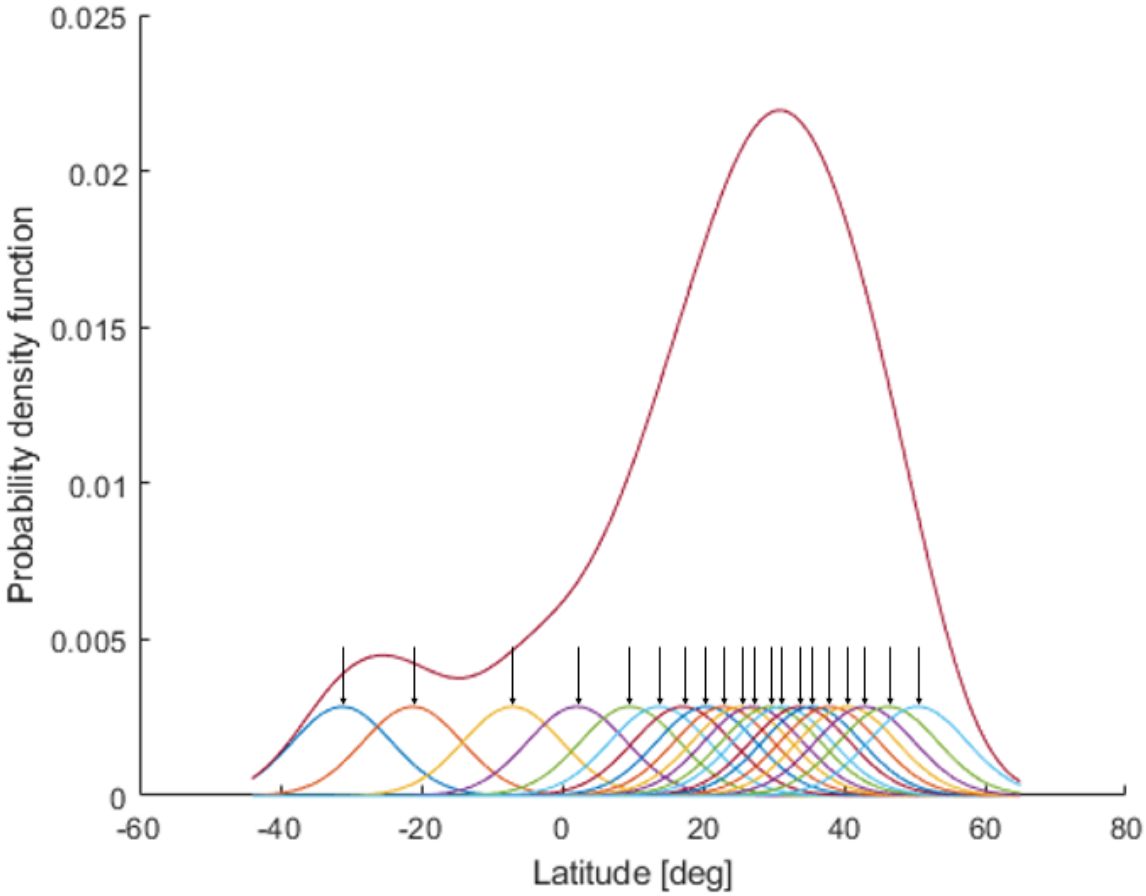


Figure 5-19: Natural disaster data by latitude KDE generated using a custom MATLAB function, showing the individual kernel functions (in a variety of colors) that were stacked to arrive at the PDF estimation, and highlighting the placements of each Gaussian kernel with black arrows at the centerpoints

5.4.2 Stratified sampling comparison and results

After initial investigation and testing within MATLAB, the discretization issues that were encountered with both the proportionate allocation and disproportionate allocation methods (and described in Section 5.4.1.1) led to the exclusion of these stratification techniques from implementation with the ReCon model. This left a comparison of Legge's original (unstratified) random sampling method with two methods of stratified sampling: proportionate definition and kernel function sampling.

The two new sampling methods were used to generate new target decks for use with the ReCon model. The evaluation code was repeatedly run using each of the

three target decks as inputs, in order to compare the effect on the performance data output from the model. Statistical analysis was carried out on the resulting performance estimates to assess the standard error of mean performance that occurred when using each of the three types of target decks.

The standard error of the mean is the standard deviation of the mean's sampling distribution, over a number of samples. It is a measure of how the mean value of a sampled parameter of interest is dispersed around the underlying population mean. The distribution of the sampled parameter mean is obtained by repeated sampling from the same population and calculation of the mean values from these samples. As the number of samples is increased, the sample means will move closer to the population mean. The exact standard error of the mean is calculated as shown in Equation 5.1:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \tag{5.1}$$

Where σ is the standard deviation of the population and n is the number of samples. However, the standard deviation of the underlying population is usually unknowable when only sampled data is available. In order to estimate the standard error of the mean, the population standard deviation can instead be replaced with the sample standard deviation, σ_x, as shown in Equation 5.2:

$$\sigma_{\bar{x}} \approx \frac{\sigma_x}{\sqrt{n}} \tag{5.2}$$

The accuracy of the estimated standard error varies depending on how many samples are used in the calculation. For a small number of samples, the standard error is typically underestimated, due to the sample standard deviation tending to give a lower value than the actual population standard deviation. However, statistical quality control literature has established a bias correction factor $c_4(n)$ that can be applied to give a more accurate estimation for the standard deviation.[120] This factor shows that using a very small sample size of 2 supplies a result that is underestimated by more than 25%, while a sample size of 5 gives an underestimate by about 6.4%, and

a sample size of 10 underestimates by about 2.8%. The sampling range of interest for the ReCon Monte Carlo simulation layer is around 20–24 target decks, for which the error is underestimated by around 1.0%, so a correction factor can be applied to give an unbiased estimate for the standard error, dividing by 0.987–0.989 (depending on the exact number of samples used).

Standard error of the mean performance was used to assess how the sample mean moved closer to the underlying population mean as the sample size increased. If the number of target decks used to evaluate constellation performance is increased, assessing the average performance against a greater number of potential mission scenarios, the standard error of the mean performance decreases, with the estimated mean performance for the constellation stabilizing as it is assessed against a higher number of target decks.

For an initial evaluation and comparison of the three sampling methods, a single optimized constellation design was selected with the parameters shown in Table 5.2. The decision was made to carry out a smaller initial comparison in order to validate the sampling approaches with a smaller amount of data, and troubleshoot the statistical analysis code on a single design before expanding the comparison to a larger area of the constellation design space.

Table 5.2: Design parameters for the sample constellation used for an initial comparison of the three different sampling methods under investigation

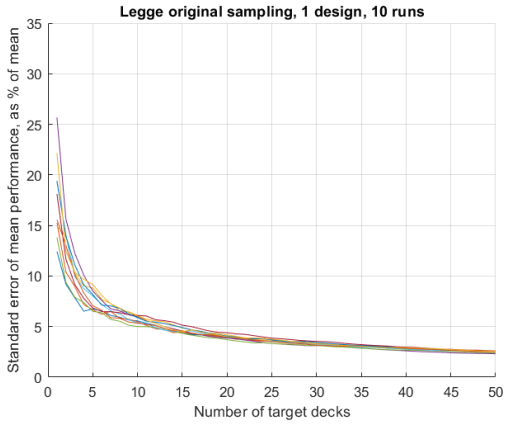
Design parameter	**Value**
Number of satellites	18
Number of orbital planes	18 (1 satellite per plane)
Orbital inclination	61.26°
Repeating ground track	15/1 (15 orbits per 1 day)
Reconfiguration ΔV lifetime budget	290.01 m/s
Aperture diameter	0.2945 m
Cost	$878M

The selected constellation design was used as the input for a series of evaluation

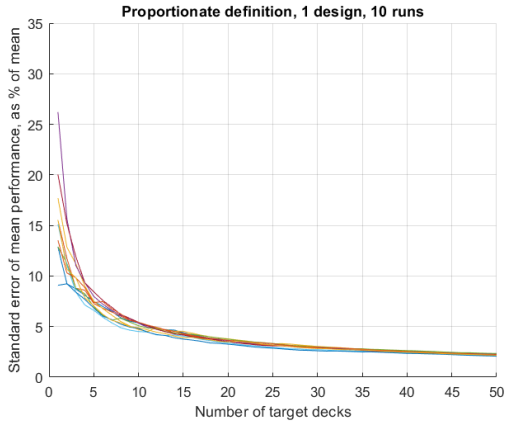
runs. These runs were carried out 10 times for each of the three sampling methods, using 50 different target decks for the performance evaluation each time. Standard error of the mean was calculated as a percentage of the mean performance value, and plotted against the number of target decks used to evaluate the constellation performance. The results for all three methods are shown in Figure 5-20, with the 10 evaluations using Legge's original method for target deck sampling shown in Figure 5-20a, the evaluations using decks generated using proportionate definition shown in Figure 5-20b, and the evaluations using decks generated using the kernel sampling technique shown in Figure 5-20c. Finally, Figure 5-20d shows a more direct comparison of the three methods, plotting the values for standard error averaged over all 10 evaluation runs for each of the three techniques, with the curve for Legge's original sampling shown in blue, proportionate definition shown in red, and kernel sampling shown in yellow.

The preliminary result shows a great deal of variation in standard error for individual evaluation runs, though this is adequately smoothed out by averaging over 10 evaluation runs as shown in Figure 5-20d. The comparison of sampling methods using a single example constellation design suggests that proportionate definition outperforms the other two methods by a small amount, at least for this individual satellite constellation design. As Legge conducted his calculations of confidence metrics using 20 non-dominated constellation designs to provide a broader basis upon which the results could be averaged, the decision was made to proceed to the next stage of expanding the evaluation and comparison from a single design to 20 optimized designs. This required an increase from 30 to 600 evaluation runs of the ReCon code, each carried out against 50 target decks for a total of 30,000 performance evaluations. This was carried out on the Supercloud cluster to enable parallelization of the evaluations against multiple target decks at a time.

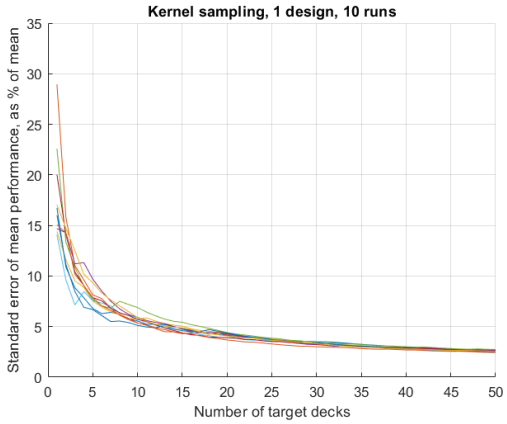
For the expanded comparison of sampling techniques, 20 optimized designs for reconfigurable constellation were selected, with a broad range of design variables



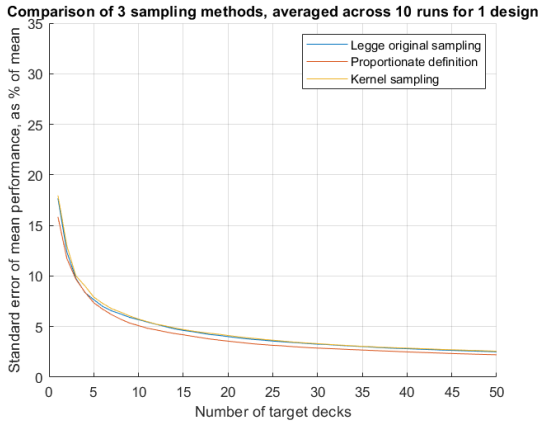
(a) Legge's original sampling



(b) Proportionate definition



(c) Kernel sampling

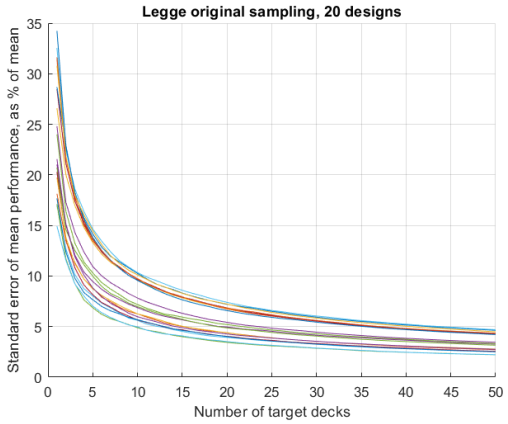


(d) Mean error for the three methods

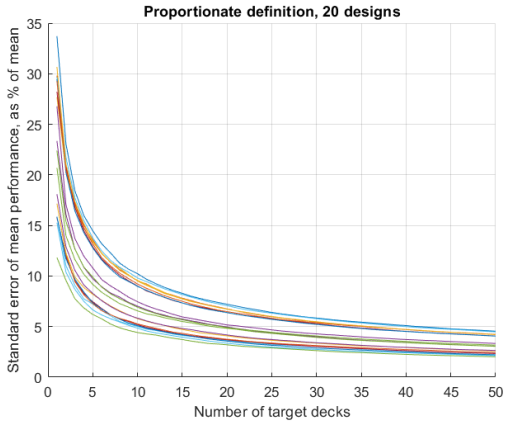
Figure 5-20: Comparison of standard error of mean performance calculated using three different methods of location sampling in the ReCon code for 10 optimization runs each, with (a) showing results using Legge's original sampling; (b) showing results using proportionate definition; (c) showing results using kernel sampling; and (d) showing a direct comparison of the standard error for the three techniques, averaged across the 10 runs that were carried out for each method

to represent a wide swath of the non-dominated design space. The size of these constellations ranged from 12 to 24 satellites, with variable numbers of satellites per plane, a range of repeating ground tracks with intervals between 13 orbits per 1 day and 31 orbits per 2 days, estimated constellation costs from \$500M–1500M, aperture diameters ranging from 0.10–0.62 m, and a range of inclinations clustered in two bands (all falling within $\pm 3^\circ$ of either 60° or 120°).

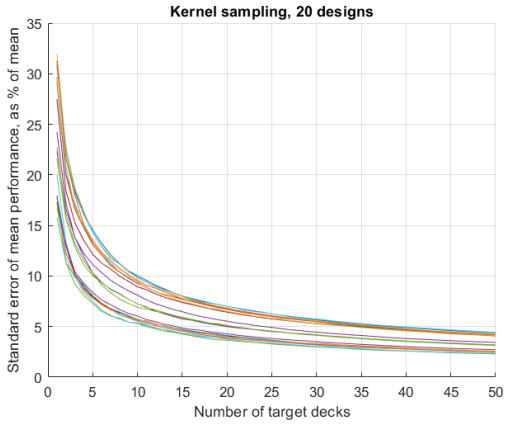
The results for this extended comparison of all three methods are shown in Figure



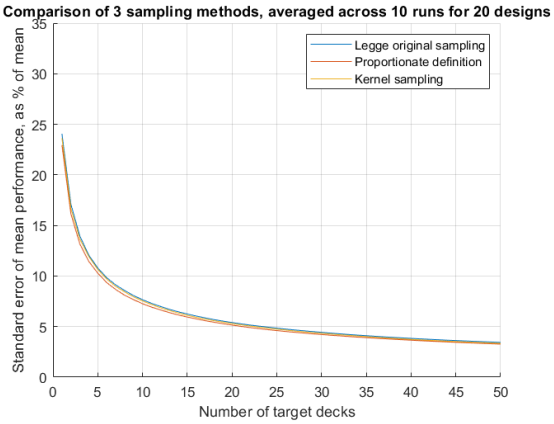
(a) Legge's original sampling



(b) Proportionate definition



(c) Kernel sampling



(d) Mean error for the three methods

Figure 5-21: Comparison of standard error of mean performance calculated using three different methods of location sampling in the ReCon code and averaged from 10 runs for each plotted line, with (a) showing results for 20 designs using Legge's original sampling; (b) showing results for 20 designs using proportionate definition; (c) showing results for 20 designs using kernel sampling; and (d) showing a direct comparison of the standard error for the three techniques, averaged across the 20 designs and 10 runs that were carried out for each method

5-21. Figure 5-21a shows the results for 20 designs evaluated using Legge's original method for target deck sampling, with the plotted result for each design averaged from 10 optimization runs. Figures 5-21b and 5-21c show results for the same 20 designs, evaluated using target decks generated with proportionate definition and kernel sampling respectively, and again averaged for each design from 10 optimization runs. Finally, Figure 5-21d shows a direct comparison of the three methods on the same axes, plotting the values for standard error averaged over all 20 constellation designs

and 10 evaluation runs (i.e. averaged from a total of 200 runs) for each of the three techniques. The standard error resulting from Legge's original sampling is shown in blue, the results using proportionate definition are shown in red, and kernel sampling is shown in yellow.

The expanded comparison in Figure 5-21 once again shows a large amount of variation in standard error, this time between the different designs. The error curves for each design in these figures are smoothed out by averaging each over 10 evaluations runs. The direct comparison of the three sampling methods in Figure 5-21d shows that when the results are averaged over a number of different satellite designs, kernel sampling now outperforms Legge's original sampling, which gives the highest error of the three techniques compared here. Proportionate definition outperforms both other sampling methods, giving the lowest standard error across the entire curve from 1–50 target decks.

Figure 5-22 shows a magnified version of Figure 5-21d where the separation of the three error curves can be seen more clearly. Legge's original selection of 24 target decks is taken as the baseline for standard error level, and this is highlighted on Figure 5-22 using black crosshairs. Averaged across 20 designs and 10 optimization runs for each design, the standard error of the mean performance for 24 target decks using Legge's sampling is 4.95% of the mean performance. In comparison, proportionate definition gives a standard error of 4.70% and kernel sampling gives a standard error of 4.85% using the same number of target decks. Standard error values for the sampling range of interest are highlighted in Table 5.3, in order to supply more specific figures for comparison of the three techniques.

Examining Figure 5-22 and Table 5.3, it can be seen that the same level of standard error established as the standard for the ReCon code in Legge's work can be maintained while using a lower number of target decks generated using the other techniques investigated here. Using kernel sampling would allow for the number of

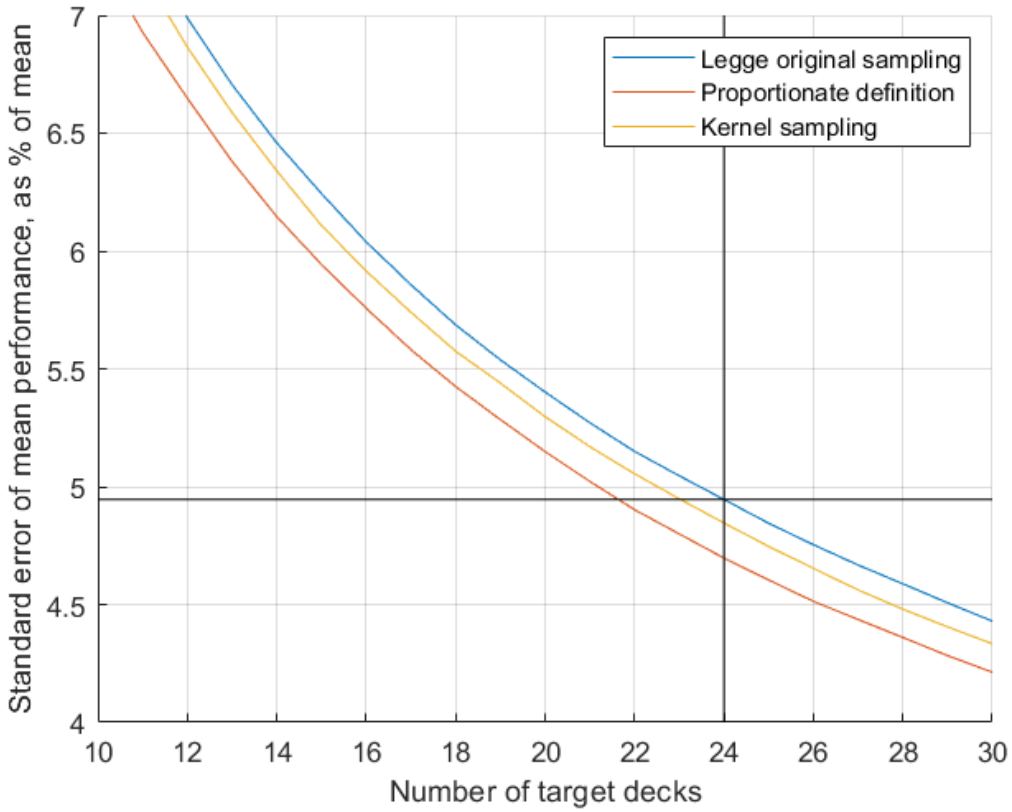


Figure 5-22: A zoomed-in version of Figure 5-21d, magnified to give a clearer view of the relative standard error obtained using the three sampling techniques. Black crosshairs are used to highlight Legge's original use of 24 decks and the associated standard error of mean performance at this sample size.

Table 5.3: Comparison of standard error of the mean performance values calculated using the three sampling techniques

Number of target decks	**Legge's sampling**	**Proportionate definition**	**Kernel sampling**
24	4.946%	4.697%	4.846%
23	5.046%	4.800%	4.948%
22	5.150%	4.902%	5.055%
21	5.271%	5.022%	5.170%

target decks to be reduced from 24 to 23 while retaining the same amount of standard error of mean performance. However, proportionate definition allows for even more of an improvement, with the ability to reduce the number of target decks used from 24 to 22 while still achieving a standard error that slightly outperforms the original standard (4.902% compared to 4.946%, a marginal 1% improvement).

The conclusion drawn from the extended comparison of sampling techniques is that proportionate definition is the superior method to employ for sampling the location PDF to generate representative target decks. Using this technique instead of Legge's original random sampling, the number of target decks used can be reduced from 24 to 22, a reduction of over 8% in the number of code iterations. This achieves a significant improvement in efficiency of the ReCon code. How this sampling improvement may translate into runtime savings is investigated in the following section.

5.4.3 Runtime results

An extensive archive of runtime data was collected throughout multiple years of running the ReCon model on the MIT Supercloud HPC cluster. This data was recorded for many different model settings as a variety of research questions were investigated over the course of this thesis work. Significant upgrades were made to the MIT Supercloud over time, and these must be factored in to any comparisons made between different lengths of ReCon optimization runs over any extended period of time.

Upgrades carried out include changes in the type of CPU and GPU nodes available to users, with Xeon-E5, Xeon-G6 and Xeon-P8 processors available as the default option for running jobs on the cluster at different points between 2018 and 2022. The user allocation limit for resources in simultaneous use (which may be employed to run a single job or several in parallel) was raised from 64 cores in 2018 through several intermediate levels during the following four years, increasing to 24 CPU nodes (a total of 1152 cores) and 8 GPU nodes (a total of 320 cores and 16 GPUs) at the present time.

The largest change made to the MIT Supercloud during the time period over which this research was conducted was the introduction of 'Triples Mode' in 2021. This provides a method of launching pMATLAB computing jobs with more flexibility and greater control over managing the specific amounts of memory and threads

allocated to a task, rather than using default parameters that may not be well-suited to a specific simulation model. The eponymous 'triples' are a set of three parameters used to specify the resources requested for the job at initiation: number of compute nodes, number of processes per node, and number of threads per process. These parameters must be calibrated to the specific application, requiring some initial time spent in assessing the optimal settings for the codebase, but after this step has been completed the expected efficiency improvement for most users is around double their previous standard using the cluster. This process was carried out for the ReCon code in early 2021, and Appendix A supplies more information on collected runtimes and the details of tuning each of these parameters.

Due to these significant changes over time to the MIT Supercloud, the runtime data has been divided into two sets, which will be referred to as Early Phase data and Triples Mode data from this point onwards. The introduction of Triples Mode led to significant drops in runtime, especially when paired with the increased allocation of cores per user, allowing for ReCon optimization runs to be carried out on a higher number of processors at once. Early Phase data is presented for a broader range of maximum constellation sizes (from 12 to 36 satellites) and numbers of target decks used to optimize the design (from 2 to 36 target decks). More recent Triples Mode results are presented using a set constellation maximum size of 18 satellites and a narrower range of target decks (from 2 to 24 target decks). These results focus on the potential improvements in runtime to be gained from reducing the number of target decks used for design optimization from 24 to 22 decks, as described previously in Section 5.4.2. For both sets of runtime data, possible improvements will be described in terms of percentage as well as the decrease in the time taken to reach the model termination criterion, to enable comparison across data sets.

Figure 5-23 shows runtime data collected during the Early Phase, collated from 35 optimization runs with the plotted model runtimes (shown as blue circles) averaged from 3–4 runs per number of target decks used. For this set of data, the maximum

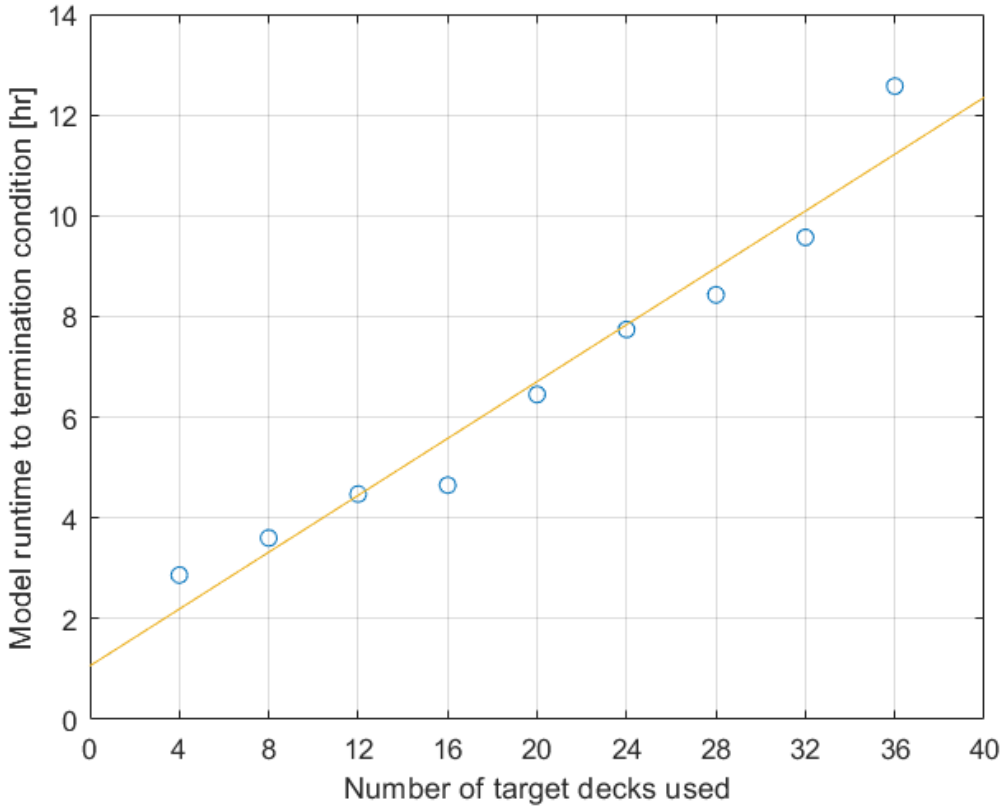


Figure 5-23: Early Phase runtime data shown in blue circles for optimization runs with a maximum of 18 satellites per constellation and a range of 4–36 target decks, with the line of best fit shown in yellow to highlight the trend of the data

constellation size was set to 18 satellites, with the number of target decks used for design optimization ranging from 4 to 36, increasing in increments of 4. Due to the use of the ϵ-NSGA-II algorithm to randomly mutate potential constellation designs during the optimization process in order to generate more candidate designs, individual optimization runtimes may only be described stochastically rather than deterministically. Excluding occasional anomalous cases where the optimization has stagnated and must be restarted, the runtime for each model run falls unpredictably but within an expected distribution. This can be seen in the amount of variation around the runtime trendline (shown in yellow on Figure 5-23), with the plotted points converging towards the linear line of best fit as the runtimes are averaged from increasing numbers of individual ReCon optimization runs.

The trendline in Figure 5-23 ranges from a predicted runtime of 2.19 hours using 4 target decks up to a predicted runtime of 11.22 hours using 36 target decks. It has a gradient of around 0.282, meaning that for every increase of 1 additional target deck used in the optimization, the runtime increases by around 0.282 hours, or 16.9 minutes. Each incremental increase of 4 additional target decks shown in the figure increases runtime by 1.13 hours, or 67.7 minutes. If the number of target decks required for an acceptable standard error value may be reduced from 24 to 22 as described in Section 5.4.2, this equates to a decrease in the predicted runtime from 7.84 hours to 7.27 hours. This reduction of 0.564 hours (or just under 34 minutes) is equivalent to a runtime decrease of 7.20%.

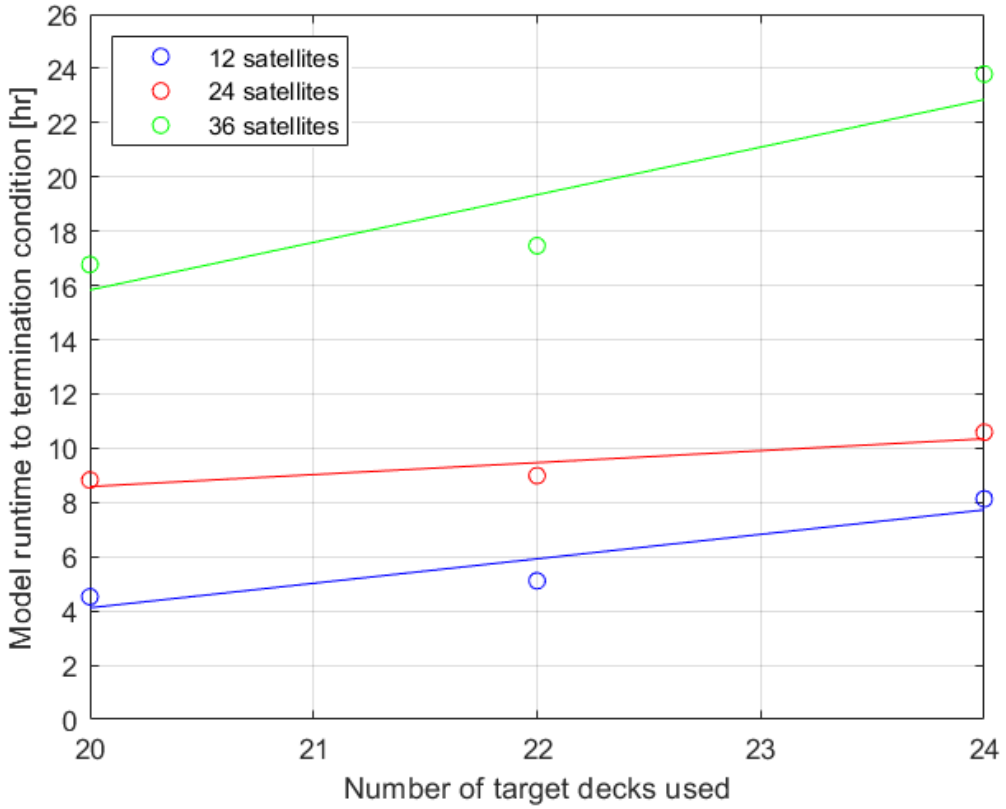


Figure 5-24: Early Phase runtime data shown for optimization runs with 3 different maximum constellation sizes (12/24/36 satellites) and a range of 20–24 target decks, with trendlines plotted in matching colors for each data set

Figure 5-24 shows additional Early Phase runtime data, collated from 27 opti-

mization runs with the plotted model runtimes each averaged from 3 runs. This data can be divided into 3 sets with different maximum constellation sizes, with optimization runs carried out for constellation designs up to 12 satellites (plotted in blue), 24 satellites (plotted in red) and 36 satellites (plotted in green). These runtimes were collected for comparison across different constellation sizes in order to verify that the general trends seen for constellation designs up to a maximum of 18 satellites (as shown in Figure 5-23) hold true for other maximum constellation sizes. The runtimes are only assessed for optimizations using a smaller range of 20–24 target decks, due to the conclusions drawn in Section 5.4.2 about the possibility of reducing the necessary number of target decks for design optimization to within this range.

The trendline for optimization runs using a 12-satellite maximum in Figure 5-24 (plotted in blue) varies from a predicted runtime of 4.12 hours using 20 target decks to 7.73 hours using 24 target decks. For each increase of 2 in the number of target decks used to optimize the design, the predicted runtime increases by 1.80 hours, or just over 108 minutes. If the number of target decks required for an acceptable standard error value may be reduced from 24 to 22 decks, this equates to a decrease in the predicted runtime from 7.73 hours to 5.92 hours. The reduction of 108.3 minutes is equivalent to a runtime decrease of 23.4%.

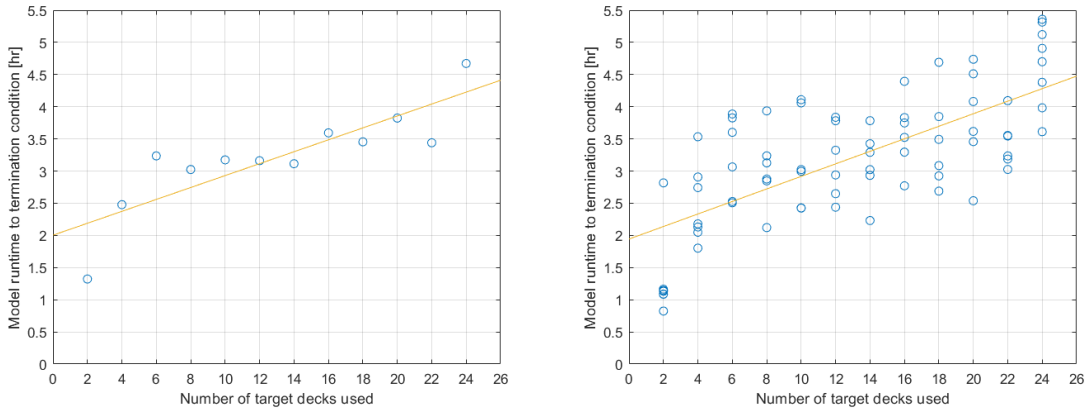
The trendline for optimization runs using a 24-satellite maximum in Figure 5-24 (plotted in red) varies from a predicted runtime of 8.59 hours using 20 target decks to 10.35 hours using 24 target decks. For each increase of 2 in the number of target decks used to optimize the design, the predicted runtime increases by 0.88 hours, or just under 53 minutes. If the number of target decks required for an acceptable standard error value may be reduced from 24 to 22 decks, this equates to a decrease in the predicted runtime from 10.35 hours to 9.47 hours. The reduction of 52.9 minutes is equivalent to a runtime decrease of 8.51%.

The trendline for optimization runs using a 36-satellite maximum in Figure 5-24

(plotted in green) varies from a predicted runtime of 15.83 hours using 20 target decks to 22.86 hours using 24 target decks. For each increase of 2 in the number of target decks used to optimize the design, the predicted runtime increases by 3.51 hours, or over 210 minutes. If the number of target decks required for an acceptable standard error value may be reduced from 24 to 22 decks, this equates to a decrease in the predicted runtime from 22.86 hours to 19.34 hours. The reduction of 210.6 minutes is equivalent to a runtime decrease of 15.4%.

Although the runtime data for all three maximum constellation sizes shows the same general trend, there is a large variation in the percentage of runtime saved by reducing the number of target decks used to optimize constellation designs from 24 to 22 decks. Each point in Figure 5-24 was averaged from the runtimes recorded for 3 optimization runs using these settings, but it may be the case that larger numbers of model runs are needed to adequately average this data. This is due to the large variation seen between individual optimization runs in the amount of runtime needed to reach the optimization termination criterion, and the stochastic nature of the ϵ-NSGA-II algorithm used to generate new potential constellation designs. However, due to upgrades to the MIT Supercloud, it is no longer possible to collect additional Early Phase data to increase the size of this data set, and therefore these runtime figures are presented as-is, with some preliminary conclusions drawn here before moving on to consider more recent Triples Mode data.

Figure 5-25 shows runtime data collected during the more recent Triples Mode phase of Supercloud operations, collated from 76 optimizations. The plotted model runtimes in Figure 5-25a (shown as blue circles) are each averaged from 6–8 runs carried out for each number of target decks used for design optimization, with a full scatter plot of all 76 of these optimization runtimes shown on the right hand side in Figure 5-25b. All of this runtime data was collected with the maximum constellation size set to 18 satellites, and the number of target decks used ranges from 2 to 24, increasing in increments of 2.



(a) Mean runtime data

(b) All runtime data

Figure 5-25: Triples Mode runtime data shown in blue circles for optimization runs with a maximum of 18 satellites per constellation and a range of 2–24 target decks, with the line of best fit shown in yellow to highlight the trend of the data. Figure 5-25a shows mean runtime data for each target deck value; and Figure 5-25b shows a scatter plot of runtime for all 76 optimization runs to illustrate the amount of variation around the trendline

The trendline in Figure 5-25 ranges from a predicted runtime of 2.19 hours using 2 target decks up to a predicted runtime of 4.23 hours using 24 target decks. It has a gradient of around 0.0927, meaning that for every increase of 1 additional target decks used in the optimization, the runtime increases by 0.0927 hours, or 5.56 minutes. Each incremental increase of 2 additional target decks shown in the figure increases runtime by 0.185 hours, or 11.1 minutes. If the number of target decks required for an acceptable standard error value may be reduced from 24 to 22 as described in Section 5.4.2, this equates to a decrease in the predicted runtime from 4.23 hours to 4.04 hours. This reduction of 0.185 hours is equivalent to a runtime decrease of 4.38%.

The significant decrease in overall model runtime is immediately clear when comparing Figure 5-25 to Figure 5-23, corresponding to the aforementioned upgrades to the MIT Supercloud and increased model efficiency due to the adoption of Triples Mode resource allocation. Both figures are created from runtime data for a maximum constellation size of 18 satellites, but where the predicted optimization runtime

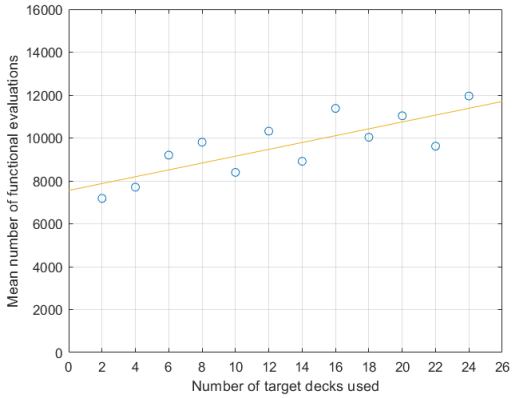


Figure 5-26: Mean number of functional evaluations required to reach the termination criterion for optimization runs using 2–24 target decks

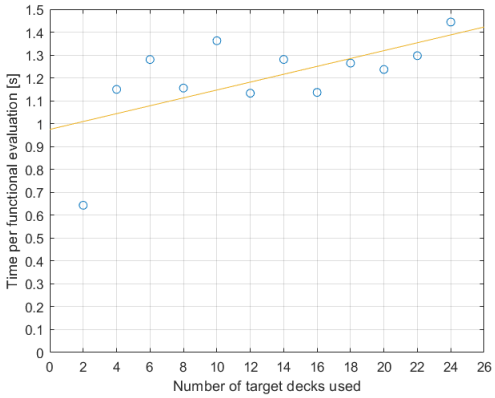


Figure 5-27: Mean time per functional evaluation for optimization runs using 2–24 target decks

in Figure 5-23 (Early Phase runtime data) varies from 2.19 hours using 4 target decks or 7.84 hours using 24 target deck, in Figure 5-25 (Triples Mode runtime data) the predicted runtime using 4 target decks is 2.37 hours and using 24 target decks is 4.23 hours. It must be observed that predicted runtimes for small numbers of target decks (1–4) using Triples Mode data are slightly higher than those predicted by the Early Phase data, which may reflect some inefficiency in the selected Triples Mode parameters when applied to much smaller ReCon optimization runs than those used in the Triples Mode tuning process on the MIT Supercloud. However, for any optimization run using 5 or more target decks, the Triples Mode runtimes are lower, with increasing improvements in runtime observed as higher numbers of target decks are used to optimize the designs.

Figure 5-26 shows how the mean number of functional evaluations needed to reach the optimization termination criterion has a positive correlation with the number of target decks used to optimize constellation designs. As the performance of each potential design is assessed against an increasing number of possible mission scenarios, the number of evaluations required to arrive at an optimized set of non-dominated constellation designs also increases, reflecting the increased complexity of optimizing designs to perform consistently well under a broad range of potential operating con-

ditions. Figure 5-27 shows how the time to carry out each functional evaluation also correlates positively with the number of target decks used to optimize constellation designs. This result is expected, as one functional evaluation is defined as the assessment of a single constellation design against all target decks used (as described in Section 5.1); by default, one functional evaluation of a constellation design will consist of 24 code iterations against 24 different target decks. It follows logically that larger functional evaluations carried out against a higher number of target decks will take a longer time to reach completion, due to the number of code iterations within this functional evaluation being directly equivalent to the number of target decks used in the optimization run.

The combination of stochasticity in both the number of functional evaluations required to reach the termination condition for the design optimization (shown in Figure 5-26) and in the time taken per functional evaluation (shown in Figure 5-27) both account for part of the large variation in overall model runtime (shown in Figure 5-25). The runtime for any individual optimization run varies significantly as shown in Figure 5-25b, but as mean runtimes are averaged from increasing numbers of individual model runs, the line of best fit across a range of different optimization sizes smooths out towards a linear trend, as shown in Figure 5-25a.

5.5 Conclusions

The nature of applying Monte Carlo based model propagation to optimize satellite constellation designs for uncertain operating circumstances means that a range of different scenarios and system responses must be simulated. Due to the variety of maneuvers and drift orbits that may be selected as the optimal response under different operational conditions, computational runtime would vary even if identical constellation designs were modelled each time the simulation was run. As the ϵ-NSGA-II algorithm is used to randomly mutate constellation design variables during each optimization run in order to generate new candidate designs for assessment, the

designs under investigation will also vary with each model run, and different runtimes will be required to reach the optimization termination criterion according to how the population of designs progresses over several generations of the optimization process.

The choice was made to simplify the modelling of the longitude distribution in order to focus sampling resources on the latitude distribution, due to the strong coupling between achievable latitude coverage and constellation orbital inclination and the lack of coupling between longitude coverage and constellation performance. Prioritizing the latitude modelling in this fashion allowed for the investigation of two new methods of sampling for generating target decks: proportionate definition and kernel function sampling. Results from these techniques are covered in Section 5.4.2, and after comparison with standard error of mean performance values from Legge's original sampling method for the ReCon codebase, it was concluded that the same baseline level of standard error could be maintained or improved upon while reducing the number of target decks used to optimize potential constellation designs from 24 to 22. This would result in a reduction in the number of code iterations of more than 8%.

Early Phase runtime data gathered before upgrades were made to the MIT Supercloud cluster showed that reducing the number of target decks from 24 to 22 for an optimization run with a maximum constellation size of 18 satellites led to a decrease in the predicted runtime from 7.84 hours to 7.27 hours, a saving of 7.20%. When this was expanded to different maximum constellation sizes of 12, 24, or 36 satellites, mean runtime reductions of between 8.51% and 23.4% were observed, although these were based on smaller numbers of optimization runs and therefore mean values that may be less reliable. Due to setup changes carried out on the MIT Supercloud cluster, it was not possible to gather additional runtimes using the same settings to expand this data set and attempt to stabilize these mean values further.

After Triples Mode upgrades to the MIT Supercloud, there was a mean runtime reduction of 31.1% in model runtime for the ReCon codebase after tuning of the three

resource allocation parameters was carried out, with improvements as high as 46.1% for the largest numbers of target decks used. The greatest efficiency improvements were seen for optimization runs using 24 target decks, as might be expected after the tuning process was carried out on the cluster while using this number of target decks. Control of these parameters for parallelized code runs on the cluster resulted in greater runtime improvements than any of the sampling changes made here, and the tuning process for these is described in Appendix A.

Applying the latitude sampling improvements to generate new target decks and using these to carry out Triples Mode runs of the ReCon code resulted in smaller but still significant runtime improvements when the number of target decks used in the optimization process was reduced while retaining the same level of standard error of mean performance. Using the proportionate definition method of stratified sampling to model the latitude PDF was found to result in the greatest improvement, reducing the number of target decks from 24 to 22 decks while improving upon the baseline standard error value of 4.95% set by Robert Legge's work, achieving a level of 4.90% using the improved sampling technique. This resulted in mean runtime improvements of 4.38%, equating to a decrease in the predicted runtime from 4.23 hours to 4.04 hours. A smaller improvement was seen when kernel function sampling was applied to the model, allowing for the number of target decks to be reduced from 24 to 23 while retaining the same level of standard error (4.95% in both cases), resulting in a runtime decrease from 4.23 hours to 4.13 hours (a saving of 2.19%).

The use of variance reduction techniques was adopted with input distributions used to model uncertain target locations for future coverage by reconfigurable constellations. Input modelling was prioritized for variables to which the system output was more sensitive, with the objective of reducing the standard error of mean performance for output designs. Modified approaches to stratified sampling were compared and two methods were implemented with the aim of achieving a reduction in error while using the same number of scenarios to optimize design candidates. This objec-

tive was achieved, reducing standard error from 4.95% using Legge's original location sampling to 4.85% for kernel function sampling and 4.70% for proportionate definition.

These reductions in standard error may be applied to reduce the runtime requirements for the ReCon codebase. The number of target decks used for design optimization may be reduced while maintaining the original benchmark set for standard error. Legge's original sampling achieved 4.95% standard error of mean performance with the use of 24 target decks, while kernel function sampling achieves 4.95% standard error at 23 target decks and proportionate definition achieves 4.90% standard error at 22 target decks. Reducing the number of target decks used from 24 to 22 results in a reduction of 8.3% in the number of code iterations required to evaluate candidate designs. This reduction corresponded to runtime savings of between 7.20–23.4% on the Supercloud with the settings used prior to 2021, and mean runtime improvements of 4.4% on the Supercloud with upgraded 'Triples Mode' settings.

Triples Mode upgrades to the MIT Supercloud required code tuning to be carried out (as described in Appendix A), but following this process mean runtime reductions of 31.1% were achieved for the ReCon code, with improvements as high as 46.1% observed for the largest optimization sizes carried out. This improved efficiency due to greater control over resource allocation parameters on the Supercloud cluster is theorized to be part of the reason for the reduced runtime improvements observed after these upgrades were made. These combined runtime improvements are significant, and may be used to enable the investigation of larger or more complex design optimizations in future work.

Chapter 6

Operational decision options for responsive maneuvering

Past work on reconfigurable satellite constellations has demonstrated the Value of Reconfigurability (VoR) when compared to traditional static constellations. Where constellation designs are considered on an iso-performance basis, reconfigurable designs are 20–70% cheaper than their identically-performing static equivalents. Where designs are considered on an iso-cost basis and compared on the metric of ΔP, reconfigurable designs outperform static equivalents by average increases in performance of up to 98%. Over 74% of iso-cost design pairs were found to have a reconfigurable performance that exceeded static design performance by a margin of 10% or higher. Even the worst-performing reconfigurable designs were found to provide comparable performance to iso-cost static designs. However, no reconfigurable constellations have been launched to date, despite the fact that many operational constellations already possess some minimal maneuvering ability used for station-keeping, drag makeup, and even collision avoidance.

Previous research on reconfigurable constellations has focused primarily on aspects of designing the constellations and satellites to perform well under unpredictable circumstances. Less work has been devoted to operational decisions once such a constellation is launched, other than the obvious necessity of simulating reconfiguration

maneuvers. Investigating decision options during the operational phase of the mission requires simulation of the constellation's activities over time, due to the lack of ability to validate such options against actual operating data in the absence of existing reconfigurable constellations. Robert Legge conducted research in this area on satellite assignment options, considering how human decision makers might use the ReCon model output to decide how many satellites should be reconfigured in response to an event of interest, and how this could be implemented as a computer decision model that balances the competing objectives of obtaining coverage over an area of interest as soon as possible and conserving propellant for responses to later events of interest. A penalty function was also incorporated into the model to motivate the conservation of propellant in satellites which possess a less-than-average quantity of remaining fuel, balancing ΔV use across the constellation as a whole to ensure its continued functioning until the end of the mission lifetime.

The effects of propulsion and propellant system constraints on the performance of a reconfigurable constellation are considered in Section 6.1. Depending on the ΔV budget available for reconfiguration maneuvers, more drastic altitude changes may become an option for satellite operators. Where extremely high-value events of interest occur, the possibility arises of maneuvering not only between GOM and ROM into a single RGT orbit, but maneuvering between multiple RGT orbit options. The feasibility of introducing reconfiguration between different RGTs (depending on available quantities of ΔV) and the potential benefits of doing so are considered in Section 6.2.

Other areas of interest in how a reconfigurable constellation might be operated include the possibility of delaying reconfiguration maneuvers where the time to achieve the first satellite coverage is less urgent, and this is briefly considered in Section 6.3. Under circumstances where persistent coverage over multiple days or weeks is desired but a delay of a few hours to the first pass time is inconsequential, it may be possible to make significant propellant savings by delaying the start of reconfiguration maneu-

vers until the most favorable drift orbit options are available to enter the desired RGT.

6.1 Effect of propulsion constraints

There are two interrelated constraints of interest placed on the propulsion system and its capabilities within the ReCon model. These are the upper limits imposed on the propulsion system mass fraction (PSMF) and the added propulsion module ΔV. Propulsion system mass fraction is defined as the total propulsion system mass including propellant, divided by the total spacecraft wet mass. Added propulsion module ΔV is also known as ΔV_{recon}, and is the amount of ΔV budgeted specifically for reconfiguration maneuvers. ΔV_{recon} makes up the largest fraction of ΔV_T, the total ΔV budget for the spacecraft, which also includes quantities of ΔV that are assigned for initial deployment from the launch vehicle into the desired orbit, correction of any errors in the launch vehicle orbital injection, stationkeeping and drag makeup over the mission lifetime, and deorbiting at the end of the mission by reducing the perigee of the satellites' orbits to 50 km.

The choice of the total ΔV budget, ΔV_T, and specific impulse, I_{sp}, both have an effect on the overall size of the propulsion system. Legge's past work on the ReCon model used only chemical propulsion systems with monopropellant, setting an I_{sp} range between 220 s for hydrazine and 260 s for novel green monopropellants, and using 240 s as a typical I_{sp} for this type of system. A constraint was also imposed on the propulsion system mass fraction, limiting this value to a maximum of 0.42 in order to ensure that only realistic constellation designs are generated by the ReCon codebase. Consulting literature, a propulsion system mass fraction of approximately 24% was cited as an average value for a LEO spacecraft with propulsion, though this was not specific to monopropellant systems.[86] Other sources suggested a typical propulsion system mass fraction of ≤ 0.3 while confirming the primacy of monopropellant hydrazine systems for use in this region of the design space, due to meeting

ΔV requirements with high reliability and low cost.[121]

Legge and other sources show that propulsion system mass fraction is a function of total ΔV and I_{sp}, which may be plotted for different propulsion system options.[8, 121] Legge shows that the specific impulse range under consideration equates to a range for ΔV_T of 930 – 1100 m/s.[8] Other research covering this area confirms that a hydrazine monopropellant system with a nominal specific impulse of 220 s and a ΔV budget of 1000 m/s will require a propulsion system mass fraction of just over 0.4.[121]

Legge elected to limit the ΔV_{recon} budget to a maximum of 1000 m/s. Non-reconfiguration ΔV allocations for reconfigurable designs consistently add up to at least 150 m/s, and in some cases as much as 300 m/s. Combined with the ΔV_{recon} budget, this results in maximum ΔV_T values as high as 1200–1300 m/s for some designs generated by the optimization.

Reducing the ΔV_{recon} budget reduces the ΔV_T, meaning that a smaller propulsion system mass fraction is needed for the satellite design. Alternatively, if the allowable propulsion system mass fraction is increased or decreased, the total ΔV budget also increases or decreases accordingly. Optimization runs were carried out for a range of altered values for these constraints, to assess the effects on the resulting designs for reconfigurable constellations. The upper constraint upon the propulsion system mass fraction was varied between 0.1 and 0.6, and the maximum ΔV budget assigned for reconfiguration was set to values of 150, 250 and 500 m/s, as well as the original limit of 1000 m/s.

Figure 6-1 shows Pareto curves depicting the sets of non-dominated reconfigurable designs generated by a series of optimization runs using different values for the upper constraints placed upon propulsion system mass fraction and ΔV_{recon}. Each set of designs is represented by the constellation cost on the x-axis and the normalized performance score on the y-axis.

Figure 6-1a shows a series of curves depicting six different maximum propulsion system mass fractions between 0.1 and 0.6, and a ΔV_{recon} budget of 1000 m/s. The propulsion system mass fractions of 0.4–0.6 result in very similar design sets, depicted by green, yellow and orange curves. The mass fraction of 0.3 (shown in aqua) reaches a lower knee in the curve at a constellation cost of approximately \$950M, and performance starts to plateau from this point. The mass fraction of 0.2 (shown in blue) follows a more diagonal shape than the other curves with lower performance for the same costs compared to the higher mass fractions. The mass fraction of 0.1 (shown in purple) results in an extremely limited design set with much lower performance, with the optimization stalling out and generating no designs above a constellation cost of \$950M.

Figure 6-1b shows a series of curves depicting six different maximum propulsion system mass fractions between 0.1 and 0.6, and a ΔV_{recon} budget of 500 m/s. The Pareto fronts generated for each propulsion system mass fraction follow very similar curve shapes to Figure 6-1a, although with fewer designs generated in the high-cost region of the design area.

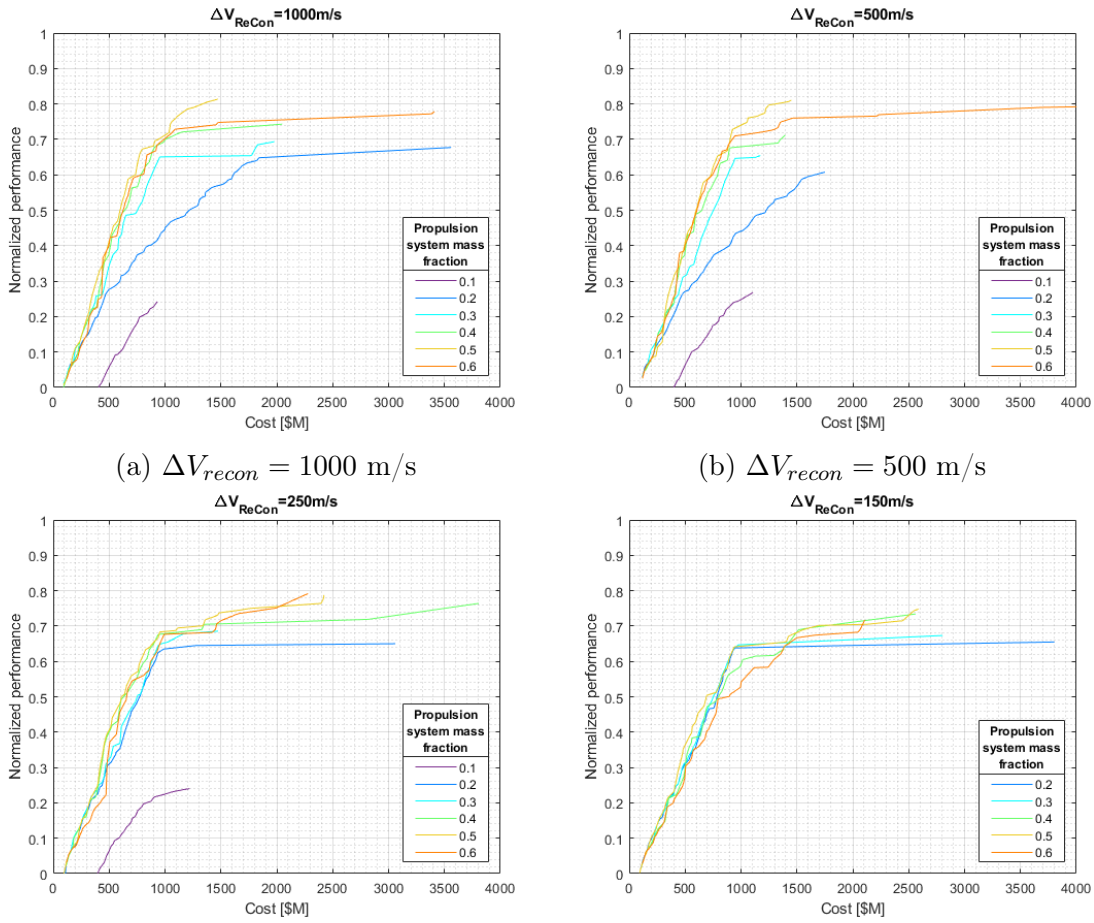
Figure 6-1c shows a series of curves depicting six different maximum propulsion system mass fractions between 0.1 and 0.6, and a ΔV_{recon} budget of 250 m/s. The knee in the curve for the highest mass fractions between 0.4 and 0.6 drops in performance when compared to the higher ΔV_{recon} values of 500 or 1000 m/s, dropping in performance towards the curve for a mass fraction of 0.3, though remaining slightly above it. The 0.2 mass fraction design front also moves up in performance compared to the previous plots, resulting in a curve that more closely follows the 0.3 mass fraction plot. The lowest 0.1 mass fraction curve also plateaus at a slightly lower performance value for this reduced ΔV_{recon} budget.

Figure 6-1d shows a series of curves depicting six different maximum propulsion

system mass fractions between 0.2 and 0.6, and a ΔV_{recon} budget of 150 m/s. The propulsion system mass fraction of 0.1 is not included in this plot, as the combination of an extremely constrained propulsion system size and an extremely constrained ΔV_{recon} budget resulted in a failure to generate any designs within the specified limits. The 0.2 mass fraction plot retains the same position of the knee in the curve, and the 0.3 curve drops in performance to follow the same abrupt plateau. The curves for the higher mass fractions between 0.4 and 0.6 still outperform the lower mass fractions in the highest and lowest cost regions, but drop below the knee in the curve for the central region, exhibiting much less of a knee and much more of a gradual slope. It is theorized that this result is due to the interaction of the two constraints; the lower propulsion system mass fraction designs are primarily constrained by the size of the propulsion system and so exhibit more consistent performance as the ΔV_{recon} budget is reduced, while the higher propulsion system mass fraction designs are able to size up their propulsion systems, but can make little use of a larger system for generating higher performance once the ΔV_{recon} budget is reduced this far.

Figure 6-2 shows Pareto curves depicting the sets of non-dominated reconfigurable designs generated by a series of optimization runs using values of 0.1 to 0.6 for the upper constraints placed upon propulsion system mass fraction, and values of 150–1000 m/s for the ΔV_{recon} budget. Each set of designs is represented by the constellation cost on the x-axis and the normalized performance score on the y-axis. This data is the same output data that was presented in Figure 6-1, but rearranged to focus on the effect of ΔV_{recon} in each subplot as opposed to propulsion system mass fraction as was previously presented.

Figure 6-2a shows three very low, flat Pareto curves, depicting the non-dominated designs generated under constraints of 250, 500 and 1000 m/s maximum ΔV_{recon} budget for a maximum propulsion system mass fraction of 0.1. The curves for a maximum of 500 and 1000 m/s ΔV_{recon} slightly outperform the 250 m/s curve, but all of the designs in this subplot are extremely constrained by the stringent propulsion system



(a) $\Delta V_{recon} = 1000$ m/s

(b) $\Delta V_{recon} = 500$ m/s

(c) $\Delta V_{recon} = 250$ m/s

(d) $\Delta V_{recon} = 150$ m/s

Figure 6-1: Pareto curves of constellation cost against normalized performance, showing non-dominated design fronts for six different upper limits placed on propulsion system mass fraction, for ΔV_{recon} budgets of a) 1000 m/s b) 500 m/s c) 250 m/s and d) 150 m/s

size limit, and the highest-performing designs achieve less than half of the normalized performance values seen in all the other subplots of Figure 6-2 for larger propulsion system limits.

Figure 6-2b shows four curves representing designs with maximum ΔV_{recon} budgets of 150, 250, 500 and 1000 m/s for a maximum propulsion system mass fraction of 0.2. All of the designs shown here easily outperform the designs from Figure 6-2a at the same cost. The two highest curves shown (in red and green) are for the 150 m/s and 250 m/s ΔV_{recon} budgets. The 500 m/s and 1000 m/s ΔV_{recon} budget designs

show lower normalized performance scores for the same constellation costs, suggesting that the propellant budget may be taking up an impractical proportion of the total propulsion system wet mass for these systems, resulting in inadequate mass devoted to sizing the propulsion system itself.

Figure 6-2c shows four curves representing designs with maximum ΔV_{recon} budgets of 150, 250, 500 and 1000 m/s for a maximum propulsion system mass fraction of 0.3. The knee in the Pareto front curve increases slightly from Figure 6-2b, and all four ΔV_{recon} outputs follow essentially the same curve. It is theorized that this is due to the propulsion system mass fraction consistently presenting the largest constraint upon all the design sets in this subplot, regardless of the allocated ΔV_{recon} maximum value.

Figure 6-2d shows four curves representing designs with maximum ΔV_{recon} budgets of 150, 250, 500 and 1000 m/s for a maximum propulsion system mass fraction of 0.4. The knee in the Pareto front curve increases slightly from Figure 6-2c for ΔV_{recon} budgets of 250, 500 and 1000 m/s, but the 150 m/s ΔV_{recon} budget is now outperformed, showing that the propellant budget now imposes more of a significant constraint upon this curve (shown in red) compared to the other three. The designs for 250 and 500 m/s ΔV_{recon} budgets follow very similar lines, with the 1000 m/s curve slightly outperforming these and plateauing at a higher performance by a small margin.

Figure 6-2e shows four curves representing designs with maximum ΔV_{recon} budgets of 150, 250, 500 and 1000 m/s for a maximum propulsion system mass fraction of 0.5. The 150 m/s curve follows essentially the same line as in the previous subplot, showing that the propellant budget continues to impose the major constraint upon this optimization run, which gains no additional performance from the increase in the limit upon the propulsion system mass fraction. The 250 m/s curve increases in performance by a small amount compared to Figure 6-2d, but plateaus relatively close

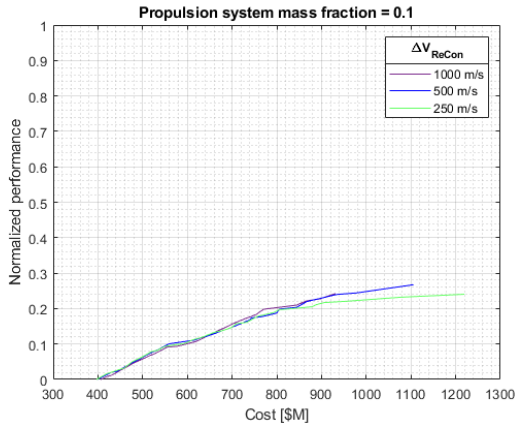
to the 150 m/s curve. The 500 and 1000 m/s curves increase further in performance, gaining additional utility from the possibility of increasing the size of the propulsion system with the increased maximum limit.

Figure 6-2f shows four curves representing designs with maximum ΔV_{recon} budgets of 150, 250, 500 and 1000 m/s for a maximum propulsion system mass fraction of 0.6. The curves follow similar lines to the previous subplots, suggesting that at this point, other constraints are taking effect upon the system. No real gains in performance are observed from further increasing the propulsion system mass fraction.

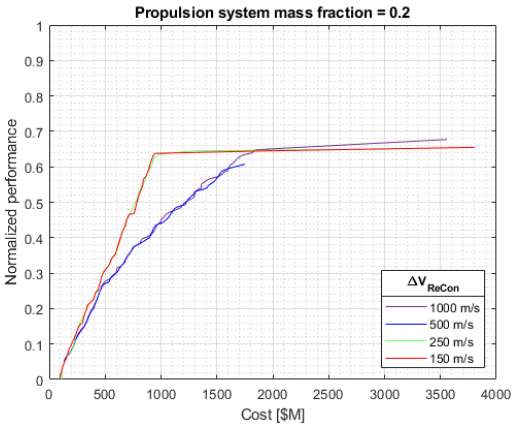
Figures 6-1 and 6-2 show the interrelation of the two propulsion constraints in the ReCon model. The maximum ΔV budget for reconfiguration can be varied within a range of 0–1000 m/s, where 0 is equivalent to a static constellation design. Imposing a maximum propulsion system mass fraction of 0.42 results in a ΔV_T limit of around 1100 m/s, depending on the specific impulse of the selected propellant. Reducing the ΔV_{recon} limit from 1000 to 500 m/s was observed to have little effect on constellation performance. Reducing the limit below 500 m/s lowers the cost/performance curve for the optimal set of designs generated by the ReCon model; lower performance is achieved for the same cost as the ΔV_{recon} budget is decreased. The threshold of this effect appears to be in the range of 300–400 m/s ΔV assigned for reconfiguration.

The limits on propulsion system mass fraction may also be varied within a 0–1 range, where 0 is equivalent to no propulsion system on the spacecraft and 1 is equivalent to a spacecraft that consists of a propulsion system and nothing else. Legge assigned 0.42 as a realistic maximum value, and this is borne out by results in other literature. A reasonable mass fraction must be selected to ensure that designs generated by the optimization model are constrained within the values that are achievable by real satellites.

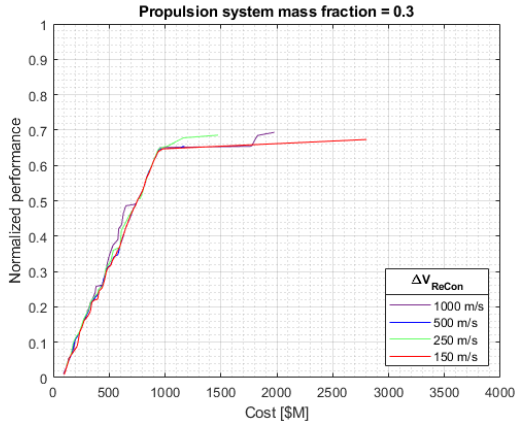
As the mass fraction limit is reduced, the cost/performance curves shown in Fig-



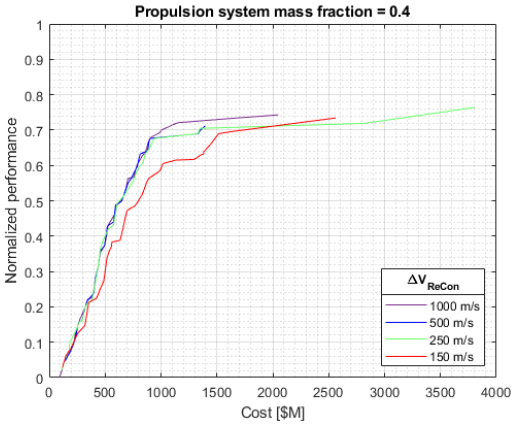
(a) Maximum PSMF=0.1



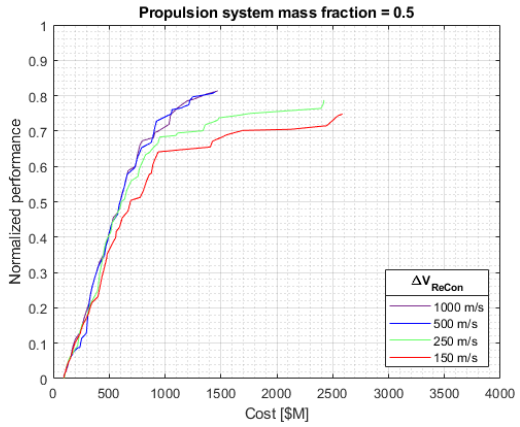
(b) Maximum PSMF=0.2



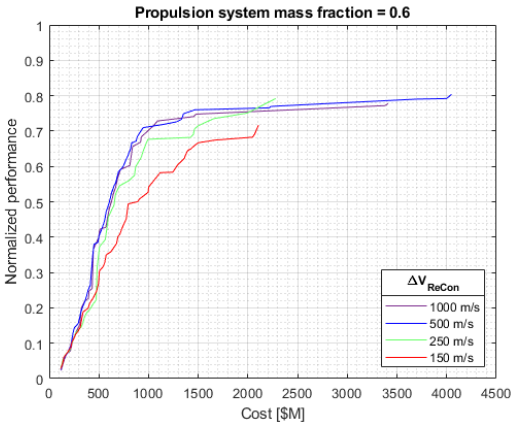
(c) Maximum PSMF=0.3



(d) Maximum PSMF=0.4



(e) Maximum PSMF=0.5



(f) Maximum PSMF=0.6

Figure 6-2: Pareto curves of constellation cost against normalized performance, showing non-dominated design fronts for four different upper limits placed on the ΔV budget for reconfiguration, for maximum propulsion system mass fractions of a) 0.1 b) 0.2 c) 0.3 d) 0.4 e) 0.5 and f) 0.6

ures 6-1 and 6-2 are lowered, meaning that lower performance is achieved for the same constellation cost, due to the more constrained design space. This effect is less pronounced in the low cost/low performance region of the design space, which are already somewhat constrained by budget, and more pronounced for higher cost/higher performance constellation designs. The areas within which the two constraints overlap in their effect upon the resulting designs can be observed in Figures 6-1 and 6-2. For example, at a ΔV_{recon} budget of 500 m/s, there is a significant drop in performance when the propulsion system mass fraction is reduced from 0.3 to 0.2. However, for a ΔV_{recon} budget of 250 m/s, the performance for the mass fractions of 0.3 and 0.2 is much more similar, where the ΔV constraint begins to have the greater impact of the two limits placed upon the design space.

This conclusion is corroborated by Figure 6-3, showing the breakdown of ΔV used for reconfiguration maneuvers over a 5-year mission lifetime. This data is shown for a sample 18-satellite constellation evaluated against a target deck of 19 locations of interest, randomly distributed in both temporal and regional spacing. Although this constellation design was generated using the original constraints of a maximum propulsion system mass fraction of 0.42 and maximum ΔV_{recon} of 1000 m/s, no individual satellite uses more than 300 m/s ΔV for responsive maneuvering. No individual event of interest uses more than 30 m/s ΔV from any satellite for reconfiguration into a RGT that provides persistent coverage.

The ΔV_{recon} and propulsion system mass fraction constraints place significant limitations on constellation performance once reduced below certain thresholds, though the interplay between the two limits must also be considered. Examining the performance of non-dominated designs for reconfigurable satellite constellations against a range of randomized target decks, it was observed that the majority of systems fall within a maximum of around 350 m/s ΔV used for reconfiguration maneuvers. Even where constellations use 150–250 m/s of additional ΔV for non-reconfiguration maneuvering (such as initial deployment and orbital injection corrections, drag makeup

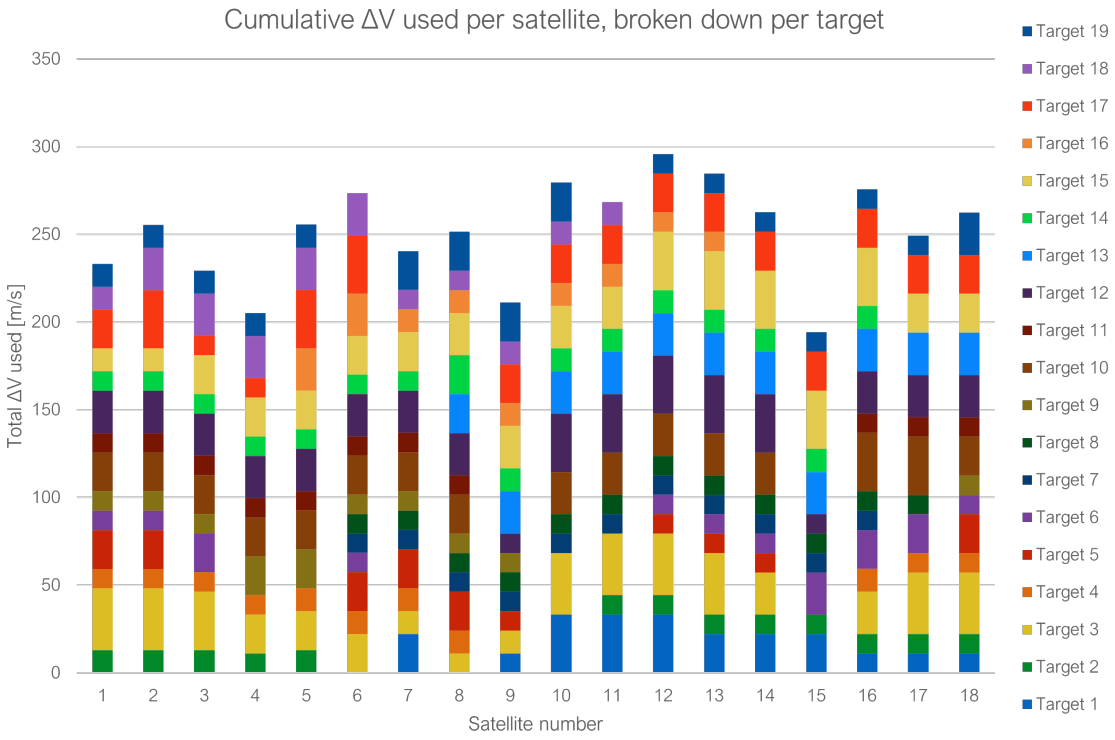


Figure 6-3: Breakdown of ΔV used for reconfiguration by an 18-satellite constellation over a sample 5-year mission consisting of 19 target locations of interest

and stationkeeping, and ΔV budgeted for end-of-life deorbiting maneuvers), this results in total ΔV usage that rarely exceeds 600 m/s.

Other design variables were examined for noticeable effects of the variations in imposed constraints. Decreasing PSMF from 0.4 to 0.3 resulted in a reduction in the number of satellite planes and a lower ΔV budget, theorized to be due to the propulsion system mass allocation having less capacity for propellant under the increased constraint. Decreasing PSMF again to 0.2 resulted in another drop in satellite planes and ΔV budget, confirming the trend observed for the previous constraint. For PSMF= 0.2, an increase was also observed in the selected GOM altitude of designs, and a corresponding increase in aperture diameter for imaging. This was theorized to be motivated by the increased coverage footprint from a higher orbit while less reconfiguration was possible due to the constrained ΔV budget.

The discrepancy between the amount of ΔV actually used and the limits imposed upon the design optimization suggest that the best-performing reconfigurable constellations are frequently found to be those with ΔV budgets that remain somewhat lower than the maximum that they could be allocated within the design constraints. This may be due to cost savings from reducing the size of the propulsion system or mass allocated for propellant. Many constellations also have remaining propellant in their reconfiguration budget at the end of the mission lifetime, due to the optimization process rewarding a cautious approach to propellant budgeting, to ensure that early obsolescence does not occur due to a loss of maneuverability. This could potentially allow for small extensions to be made to the mission lifetime.

Satellite operators could make use of a comparison between the expected use rate of propellant and the recorded use rate throughout the constellation lifetime. This would allow for assessment of when to adopt more conservative maneuvering strategies, and when faster reconfiguration strategies might be pursued, prioritizing earlier coverage over the conservation of propellant in situations where a large margin of ΔV is available. Other uses could be made of a bigger reconfiguration budget or large surplus of unused ΔV, including making more extensive changes to the orbital altitude in order to achieve greater improvements in coverage performance, and this subject is covered more extensively in the following section.

6.2 Crossover latitudes

The constellation designs generated for reconfigurable and static constellations as part of this work are constrained to certain options for their choice of orbit. Only circular orbits and symmetric constellation designs are considered, and the RGT selection for ROM is made from a list of six candidate orbits. The ReCon model optimizes constellation designs based on these six orbits, which range between mean altitudes of approximately 409.7 and 1253.7 km (with considerable variation around these numbers depending on the selected orbital inclination). Some of the parameters of these

six orbits are shown in Table 6.1, including mean, minimum and maximum values for the range of orbital altitudes associated with each candidate orbit.

Three of these candidate RGT orbits are $n/1$ orbits, meaning that they return to pass over the same point on the ground after 1 day and n orbits of the Earth. For example, a 13/1 orbit completes 13 orbits of the Earth in 1 day before passing back over the same point on the ground and continuing back along the same repeating ground track. The other three orbits are $n/2$ orbits, meaning that they do not pass back over the same point on the ground until two full days and n orbits have passed. For example, a 31/2 orbit completes 31 orbits of the Earth in 2 days before passing back over the same point on the ground.

Table 6.1: Orbital parameters for the six candidate RGT orbits used in the ReCon model

Orbits/days for selected RGT	**Period [s]**	**Mean altitude [km]**	**Minimum altitude (at i=0°) [km]**	**Maximum altitude (at i=180°) [km]**
31/2	5559.0	409.75	320.89	520.00
15/1	5744.3	559.71	476.04	664.61
29/2	5942.3	718.21	639.60	817.86
14/1	6154.6	886.07	812.38	980.57
27/2	6382.5	1064.2	995.30	1153.7
13/1	6628.0	1253.7	1189.4	1338.2

The large difference between minimum and maximum altitude figures to achieve each RGT is due to the need to account for the oblateness of the Earth and the perturbing rotation it induces in the desired orbit, creating rotation in the right ascension of the ascending node, argument of perigee and mean anomaly. Due to the fact that the orbit perturbations caused by oblateness are a function of orbital altitude, calculating these altitudes for each RGT requires solving the following set of equations iteratively until an appropriate estimate for the RGT altitude is found.[86] This was carried out in MATLAB using the 'fminbnd' function, an algorithm based on golden section search and parabolic interpolation.[122]

First, a specific RGT orbit is selected, with an integer number of orbits, j, taking place in an integer number of days, k. For example, when considering a 31/2 RGT orbit, j=31 and k=2. An initial estimate is made for the orbital altitude, H_0, by calculating the repeating ground track altitude without any oblateness effects, using the following equation:

$$H_0 = \mu^{1/3} \left(\frac{2\pi j}{D^* k} \right)^{-2/3} - R_E \tag{6.1}$$

where μ is the Earth's gravitational constant (398600.44 km^3/s^2), D^* is the length of a sidereal day (86164.1 seconds), and R_E is the Earth's radius (the equatorial radius of 6378.1 km is used for the initial estimation, with the assumption of a spherical Earth).

Equation 6.1 may be further simplified with the use of the following constant:

$$k_1 = \mu^{1/3} \left(\frac{2\pi}{D^*} \right)^{-2/3} \approx 42164.2 \text{ km} \tag{6.2}$$

Combining Equations 6.1 and 6.2 gives:

$$H_0 = k_1 \left(\frac{j}{k} \right)^{-2/3} - R_E \tag{6.3}$$

Another useful constant k_2 is defined as follows:

$$k_2 = 0.75 \left(\frac{360}{2\pi} \right) \mu^{1/2} R_E^2 J_2 \approx 1.02955 \times 10^{14} \text{ km}^{3.5} \text{ deg/sidereal day} \tag{6.4}$$

where J_2 is the factor representing the Earth's oblateness. Defining the rotation rate of the Earth, $\dot{L}$, as 360 deg/sidereal day, Equation 6.4 is used to determine the rate of change of the ascending node, $\dot{\Omega}$, the rate of change of the perigee, $\dot{\omega}$, and the rate of change of the mean anomaly, $\dot{M}$, which are all calculated in units of deg/sidereal day using the following equations:

$$\dot{\Omega} = -2k_2 a^{-7/2} \cos i (1 - e^2)^{-2} \tag{6.5}$$

$$\dot{\omega} = k_2 a^{-7/2}(5\cos^2 i - 1)(1 - e^2)^{-2} \tag{6.6}$$

$$\dot{M} = k_2 a^{-7/2}(3\cos^2 i - 1)(1 - e^2)^{-3/2} \tag{6.7}$$

where e is the eccentricity of the orbit, i is the orbital inclination, and a is the estimated semi-major axis ($R_E + H_0$). The mean angular motion for the repeating ground track, n, is estimated using:

$$n = \frac{j}{k}(\dot{L} - \dot{\Omega}) - (\dot{\omega} + \dot{M}) \tag{6.8}$$

and finally, the estimated repeating ground track altitude, H, is revised using the following equation:

$$H = \mu^{1/3}\left(\frac{180}{n\pi}\right)^{2/3} - R_E \tag{6.9}$$

The estimated semi-major axis, a, may be recomputed using $a = R_E + H$, and used to iterate through Equations 6.5–6.9 until an acceptable estimate of the RGT altitude is reached.

Figure 6-4 shows the range of orbital altitudes associated with inclinations between 0 and 180°; as the inclination is increased from an equatorial orbit at 0°, the orbital altitude must be increased in order to maintain the desired repeating ground track of the same ground pass cadence. The 31/2 candidate orbit has the lowest orbital altitude; to achieve the desired cadence of 31 orbits in 2 days before passing over the same ground location and beginning to repeat its ground track, its altitude ranges from a minimum of 320.89 km (at 0° inclination) up to a maximum of 520.00 km (to achieve the same cadence at 180° inclination). The highest altitude orbit considered as a candidate is the 13/1 RGT orbit; to achieve the desired cadence of 13 orbits in 1 day before passing over the same point and repeating the same ground track, its altitude ranges from a minimum of 1189.4 km altitude (at 0° inclination) up to a maximum of 1338.2 km (at 180° inclination).

Orbits with an inclination between 0 and 90° are classed as prograde orbits, meaning they are orbiting in the same direction as the rotation of the Earth. Orbits with

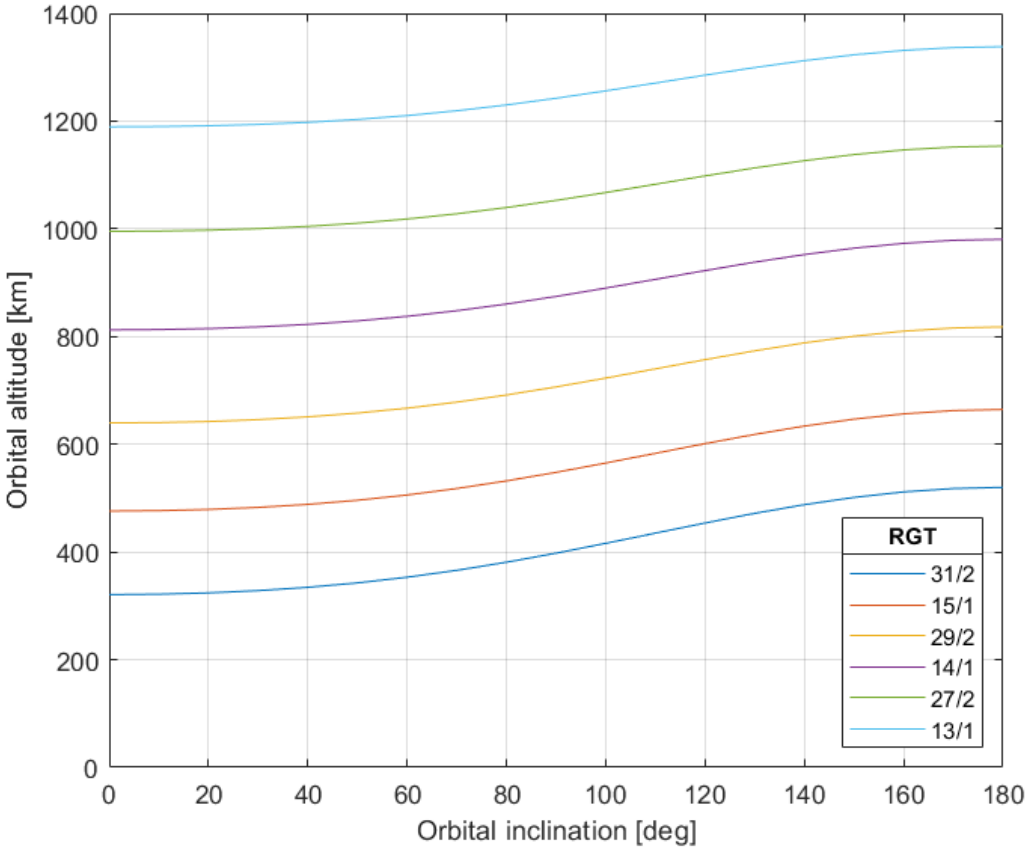


Figure 6-4: Impact of inclination on orbital altitudes for 6 different candidate RGT orbits

an inclination between 90 and 180° are classed as retrograde orbits, meaning they are travelling in a direction opposite to the rotation of the Earth. The majority of Earth-orbiting satellites are launched in prograde orbits, as less propellant is required to reach orbit due to moving with the Earth's rotation rather than against it. A polar orbit with an inclination of exactly 90° is classed as a perpendicular orbit, meaning it is neither prograde nor retrograde, but perpendicular to the direction of the Earth's rotation. A prograde equatorial orbit (with the direction of the planet's rotation) has an inclination of 0°, and a retrograde equatorial orbit (against the direction of the planet's rotation) has an inclination of 180°.

When satellite ground tracks are plotted on a flat latitude/longitude map pro-

jection for the candidate RGT orbits, similar sinusoidal patterns are seen for each prograde inclined orbit between 0 and 90°, with the upper and lower limits of latitude coverage dictated by the choice of orbital inclination. Figure 6-5 shows ground tracks plotted for a single orbit of a 15/1 RGT and how the shape of this ground track changes as the orbital inclination is increased from 10° up to 80°. These ground tracks change shape from a shallow curve at the lowest inclination up to a more elongated curve at the highest inclination, with an appearance closer to a square wave at the highest and lowest latitudes on the plot due to the fact that the rectangular map projection does not account for the closer spacing of the longitude lines at higher latitudes.

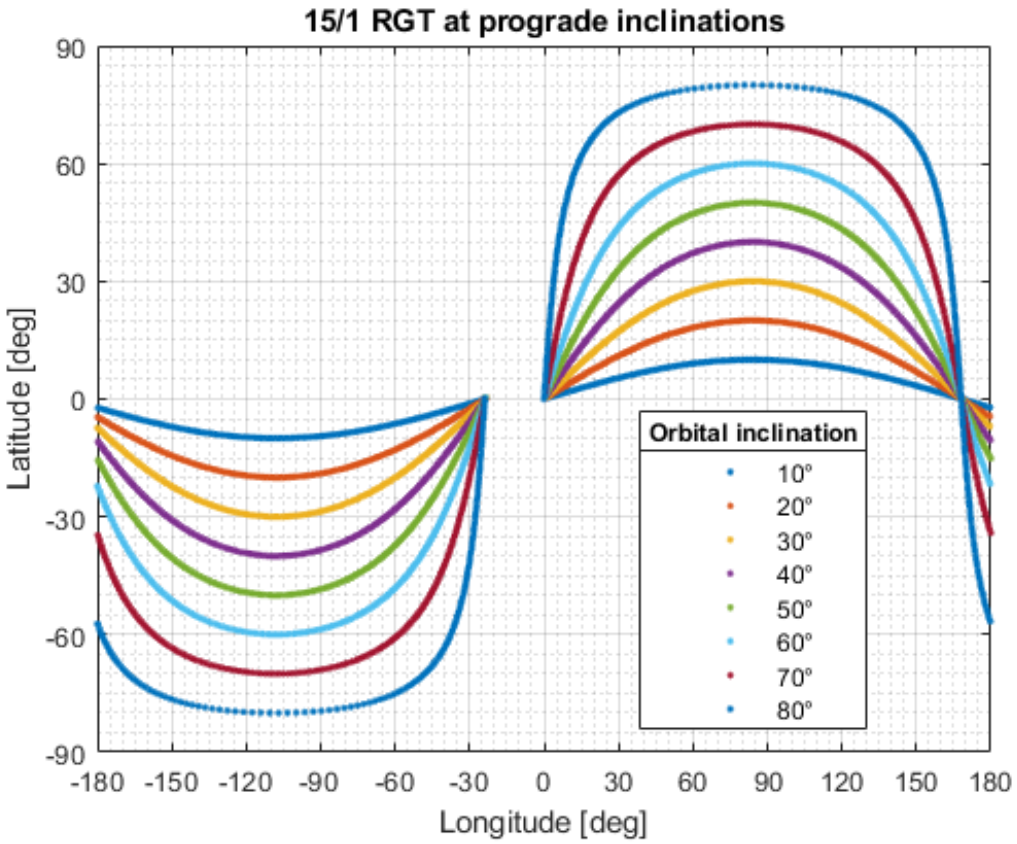


Figure 6-5: Ground tracks plotted in latitude/longitude for a single prograde orbit of a 15/1 RGT, showing the variation in shape that occurs for a range of orbital inclinations between 10–80°

For a prograde RGT orbit, each full orbit around the Earth (from ascending node

to ascending node) is a little less than 360° in longitude, due to the amount of the Earth's rotation that occurs in the same direction during the time period of the orbit. The Earth will complete one full rotation in one day, so for a 15/1 RGT orbit, each satellite will complete 14 full circumnavigations of the Earth in 15 orbits. The 15/1 RGT satellite ground track completes an ascending pass over the equator every 336° of longitude, which is 14/15 of the Earth's circumference around the equator, and this westward drift can be seen in the prograde orbits plotted in Figure 6-5.

Figure 6-6 shows satellite ground tracks plotted for the same 15/1 RGT orbit as in Figure 6-5, but this time featuring retrograde inclined orbits between 100° and 170°, which demonstrate similar sinusoidal patterns to the prograde orbits shown in the previous figure. The upper and lower limits of latitude coverage are similarly dictated by the choice of orbital inclination, but with the extent of coverage inversely correlated with the increasing inclination values in this case. A retrograde orbit with 100° inclination provides comparable coverage limits (at around ±80° of latitude) to a prograde orbit with 80° inclination, a retrograde orbit with 120° inclination provides comparable limits (at around ±60° of latitude) to a prograde orbit of 60° inclination, and so forth.

For a retrograde RGT orbit, each full orbit around the Earth (from ascending node to ascending node) is a little greater than 360° in longitude, due to the amount of the Earth's rotation that occurs in the opposite direction during the time period of the orbit. The Earth will complete one full rotation in one day, so for a 15/1 RGT orbit, each satellite will complete 16 full circumnavigations of the Earth in 15 orbits. The 15/1 RGT satellite ground track completes an ascending pass over the equator every 384° of longitude, which is 16/15 of the Earth's circumference around the equator, and this westward precession can be seen in the prograde orbits plotted in Figure 6-6.

As discussed in Section 5.2.1, the choice of inclination for a constellation design

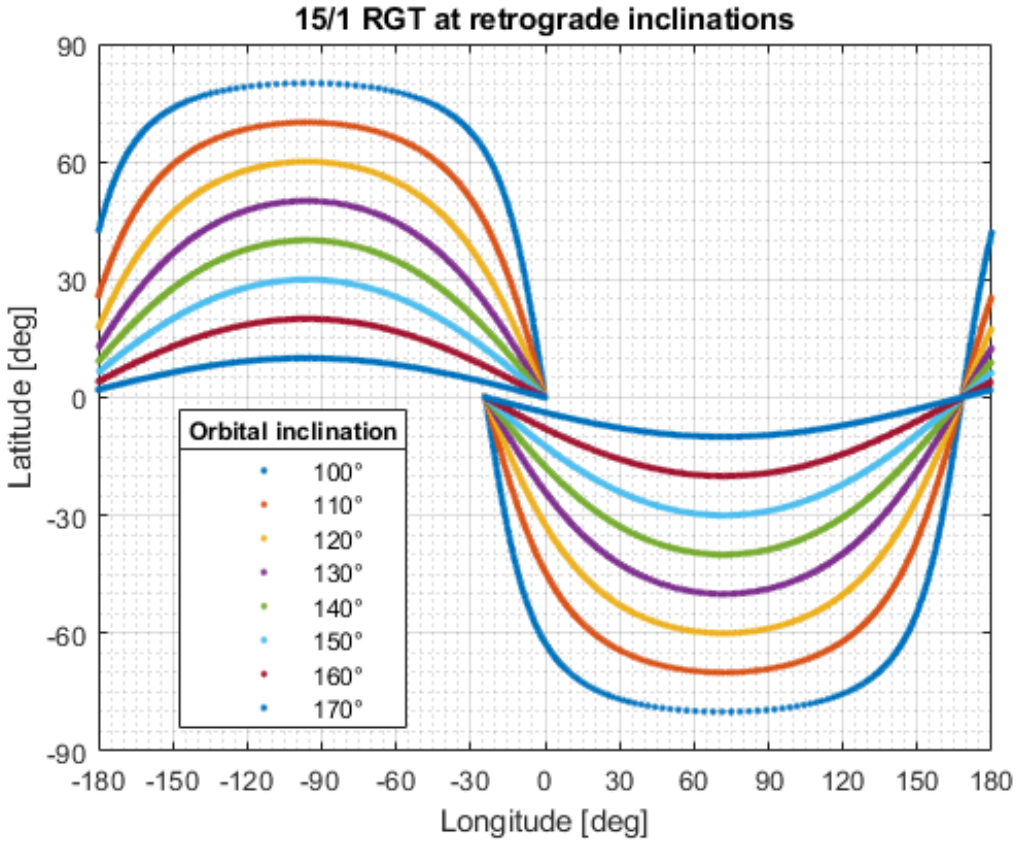


Figure 6-6: Ground tracks plotted in latitude/longitude for a single retrograde orbit of a 15/1 RGT, showing the variation in shape that occurs for a range of orbital inclinations between 100–170°

dictates the upper and lower limits of latitude coverage that can be achieved by the constellation's satellite passes. The exact limits of coverage for a particular choice of inclination will be dependent upon the minimum elevation angle requirements for usable coverage for the desired application, which will determine the swath width of satellite coverage for a given pass. This swath width determines the off-ground-track angle at which coverage can be achieved on either side of the ground track dictated by the selected inclination. Due to the extremely high ΔV costs associated with changing orbital inclination once the constellation is launched, it is important to select an appropriate inclination at the design phase to ensure that all potential latitudes of interest will be included within the coverage range.

Figure 6-7 shows a histogram of the selected constellation inclinations for a set of

5049 non-dominated designs. This set was generated by combining the output of 36 optimization runs of the ReCon code, which were carried out on the MIT Supercloud over the course of several weeks. The histogram is normalized so that the y-axis values are presented by probability, meaning that the sum of the bar heights is equal to 1. By default, the ReCon code implements constraints on the possible orbital inclinations selected by the model, limiting these to a range of 50° to 130°, which is reflected in the distribution shown here. This ensures that the constellation designs generated by the ReCon code will always supply coverage over a minimum latitude range of 50°N to 50°S, which accounts for the latitudes of around 96.9% of the generated target locations used to optimize and evaluate designs.

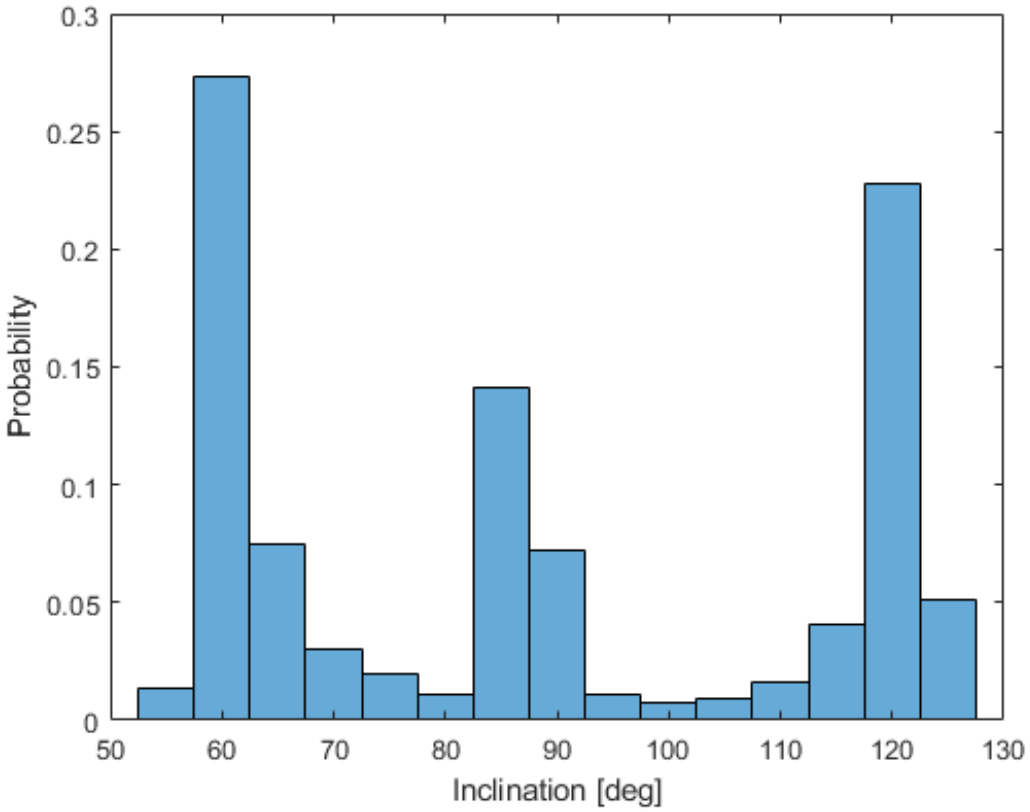


Figure 6-7: Histogram of the selected orbital inclinations for 5049 non-dominated constellation designs collated from 36 optimization runs of the ReCon code

It can be seen from Figure 6-7 that the selected orbital inclinations of this set

of 5049 optimized constellation designs fall in three clusters. The largest cluster is around an inclination of 60°, meaning that these constellations are in prograde orbits supplying ground coverage between latitude limits of approximately 60°N and 60°S. As the northernmost target in the decks used to optimize constellation designs is located at 60.44°N and the southernmost target is located at 43.72°S, these orbits are expected to provide full coverage of the latitude range of interest. The second largest cluster of designs is around an inclination of 120°, meaning that these constellations are in retrograde orbits, also supplying ground coverage between latitude limits of approximately 60°N and 60°S and therefore covering the full latitude range of interest. Although these two clusters supply similar ranges of ground coverage, it is unsurprising that the 120° inclination is selected less frequently out of the two options, due to the greater ΔV costs of launching into a retrograde orbit. The third cluster of designs is around 85–90° inclination, meaning that these are near-polar orbits which will supply coverage of all or almost all latitudes. Although this might be beneficial if unexpected events occurred outside of the latitude range for which the constellation coverage was designed, if all events occur within the expected latitude range, this means that some time is wasted in orbiting near the poles over regions for which no coverage is desired.

Figure 6-8 shows the orbital altitudes for the set of six candidate RGT orbits, plotted for 60° and 120° inclinations: the two most common clusters of non-dominated designs from Figure 6-7. The $n/2$ ground tracks are plotted by the number of orbits per day in this figure, so a 27/2 RGT is equivalent to 13.5 orbits per day, a 29/2 RGT is equivalent to 14.5 orbits per day, and a 31/2 RGT is equivalent to 15.5 orbits per day. Figure 6-8 illustrates how the number of orbits per day decreases as the semi-major axis of the orbit (and therefore the orbital altitude) is increased. To achieve a higher number of orbits per day, a lower orbital altitude must be selected.

Examining specific numbers from Figure 6-8, it can be observed that 13 orbits per day occurs for an orbit at an altitude of 1210 km for 60° inclination and 1285 km

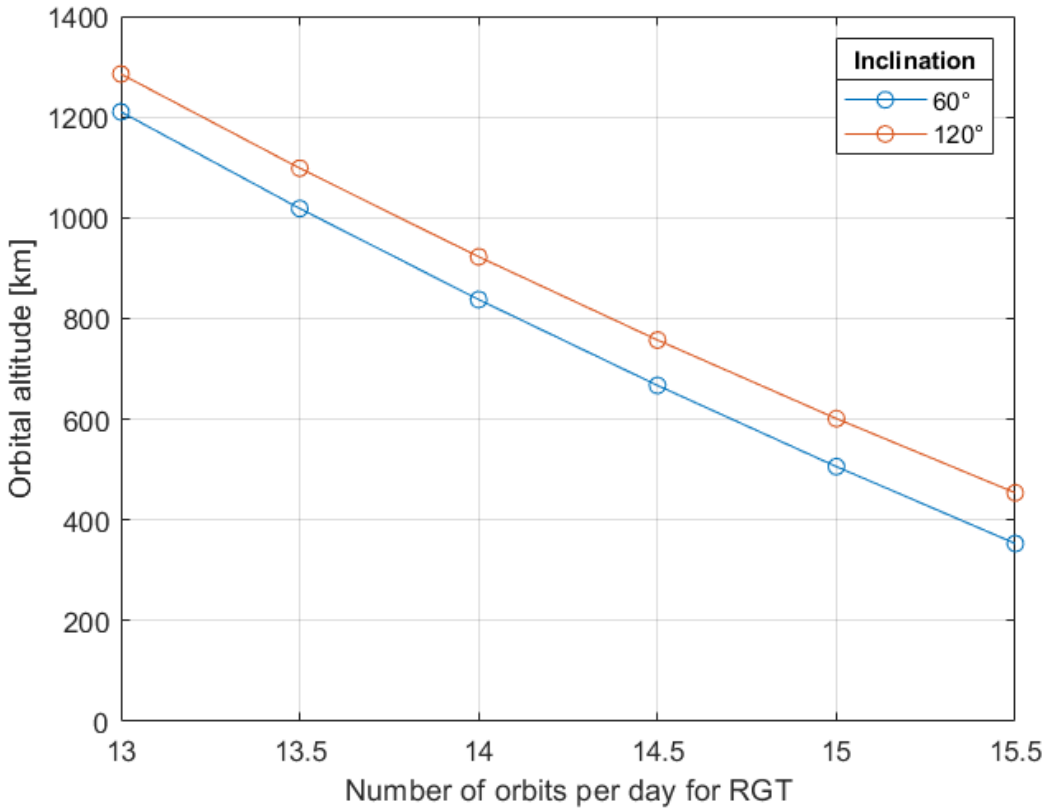


Figure 6-8: Six candidate RGT orbits plotted at a prograde inclination of 60° (in blue) and a retrograde inclination of 120° (in red), showing the increase in orbits per day with decreasing orbital altitude

for 120° inclination. To move to the next RGT and increase the number of orbits to 13.5 per day (a 27/2 RGT) would require a decrease in altitude to 1018 km for 60° inclination and 1098 km for 120° inclination. To increase the number of orbits all the way to 15.5 per day (a 31/2 RGT, at the other extreme of the RGT candidate list) would require a decrease in altitude to 353 km for 60° inclination and 454 km for 120° inclination.

To evaluate how the performance of reconfigurable constellation designs varies against a range of events of interest, an example constellation design was selected from the non-dominated set of optimized designs. This design consisted of 18 satellites in 18 orbital planes, with an inclination of 61.26°, a GOM altitude of 505.17 km and a RGT of 15/1. These parameters were selected as broadly representative

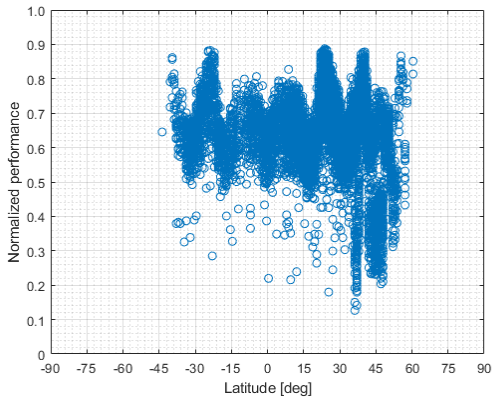


Figure 6-9: A scatter plot showing the normalized performance values achieved by a sample constellation against 9765 events of interest from 500 target decks

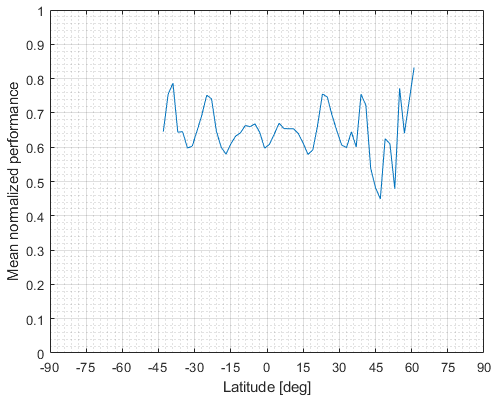


Figure 6-10: An averaged plot of the values from Figure 6-9, showing the mean normalized performance by latitude for the sample constellation

of the largest cluster of designs around 60° inclination. In examining the set of 5049 non-dominated optimized reconfigurable constellation designs collated from multiple optimization runs on the MIT Supercloud, it was discovered that the 15/1 RGT was selected in around 81% of cases, and so this was also considered to be the most representative RGT orbit when making the selection of an example design for initial evaluation.

The constellation was evaluated against 500 target decks consisting of 9765 individual events of interest. The resulting normalized performance values (where 0 = no utility gained and 1 = coverage that perfectly matches the desired revisit cadence and resolution for the entire period of interest) are shown as a scatter plot in Figure 6-9.

Examining Figure 6-9, it can be seen that the performance values appear to follow a roughly symmetric distribution about 0° of latitude. Fewer events of interest occur in the southern hemisphere (due to the lower population of this hemisphere, there is a proportional lower economic impact of disasters), but symmetric peaks in performance can be seen at approximately ±7°, ±25° and ±35° before the southern latitude distribution peters out. After detecting this pattern in Figure 6-9, the performance

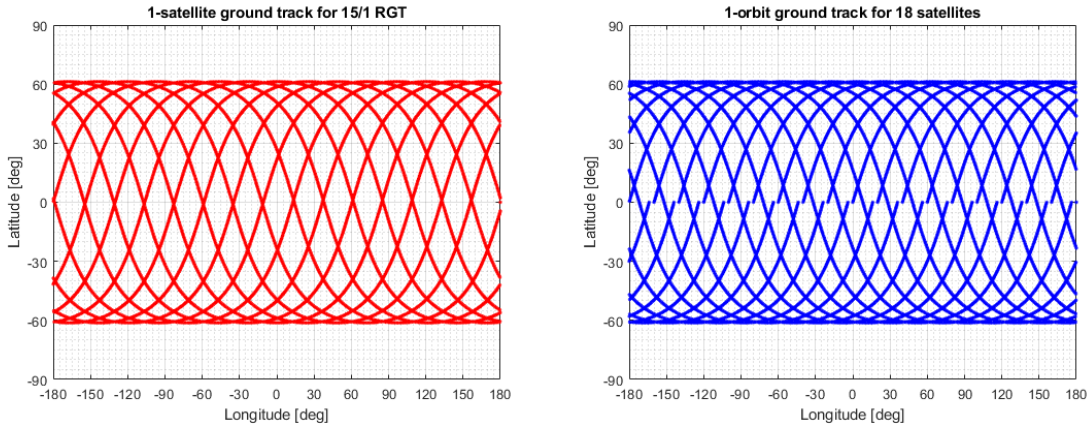
values were averaged by latitude to arrive at the distribution of the mean normalized values shown in Figure 6-10. The distribution of the performance by latitude is much more apparent in this figure, and the symmetric peaks and troughs in performance values are clearly visible. The mean normalized performance also increases to its highest point at the right-hand side of Figure 6-10 as the target latitudes occur in closer proximity to the orbital inclination, following the predicted result from the geometry of the orbit.

Upon considering the symmetric nature of the peaks seen in the normalized performance when it was plotted against latitude, it was hypothesized that this might relate in some way to the positioning of the satellite ground tracks of the constellation. Ground tracks of the subsatellite point (the ground location directly below the satellite's position) are shown in Figure 6-11, for the example design using a 15/1 RGT orbit at 61.26° inclination.

Figure 6-11a shows a plot (in red) of the ground track for a single satellite in the 18-satellite constellation, completing 15 orbits over the course of one day. This figure shows the pattern of where an individual satellite crosses its own previous ground track traces, over the course of one day.

Figure 6-11b shows a plot (in blue) of the ground track for a single orbit of all 18 satellites in the constellation, occurring over a time period of around 95.7 minutes. This figure shows the pattern of where each satellite crosses the ground track traces of the other satellites in the constellation, over the course of a single orbit.

Both Figures 6-11a and 6-11b show a characteristic 'mesh' ground track pattern, where the geographical area over which the satellites pass is divided by the ground track lines into a series of approximately diamond-shaped areas. Each corner of these diamonds is defined by 'crossover' points, where two adjacent ground tracks cross one another's path. Figure 6-12 shows a magnified view from Figure 6-11a, labelling an



(a) One-day ground track of a single satellite (b) One-orbit ground track of 18 satellites

Figure 6-11: Satellite ground tracks for a 18-satellite constellation in a 15/1 RGT orbit at 61.26° inclination

ascending pass (where the satellite is passing from south to north) and an adjacent descending pass (where the satellite is passing from north to south) and the crossover point where these two paths intersect.

These crossover points are seen both where a satellite crosses its own ground track (which will be referred to as an 'individual satellite crossover point' from here onwards) and where a satellite crosses the ground tracks of other satellites in the constellation (which will be referred to as a 'multi-satellite crossover point' from here onwards), and these both occur at symmetric latitudes north and south of the equator. The geometry of these ground track plots and the symmetry of the latitudes at which crossovers occur led to the theory that the improved mean performance values seen in Figure 6-10 might be found at latitudes where both ascending and descending passes cross above the same locations. The improvement in coverage performance would be explained by the opportunity of acquiring twice as many satellite passes of targets of interest that are located at these latitudes

In order to test the theory that the peaks in normalized performance are occurring at crossover latitudes, Figure 6-13 was created to show an overlay of the crossover latitudes of the ground tracks of the 18-satellite constellation in a 15/1 RGT (shown

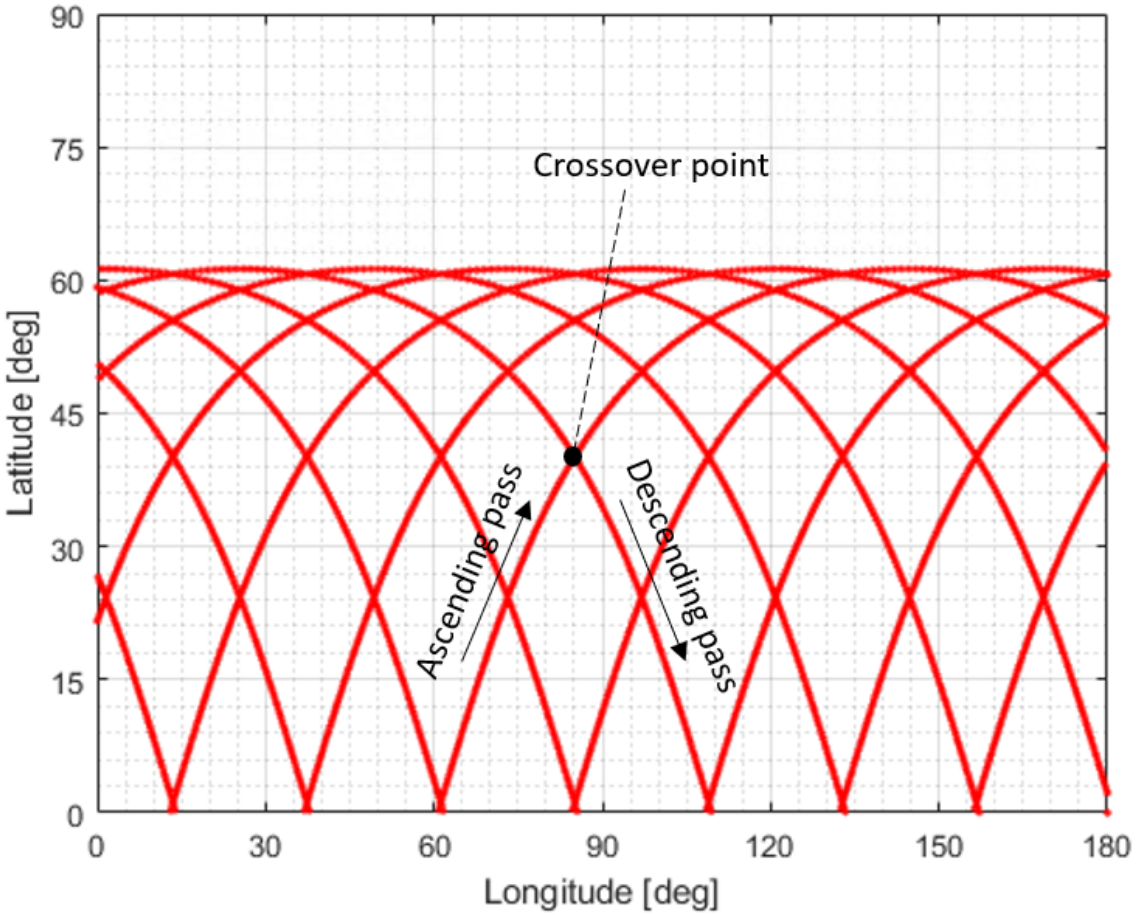
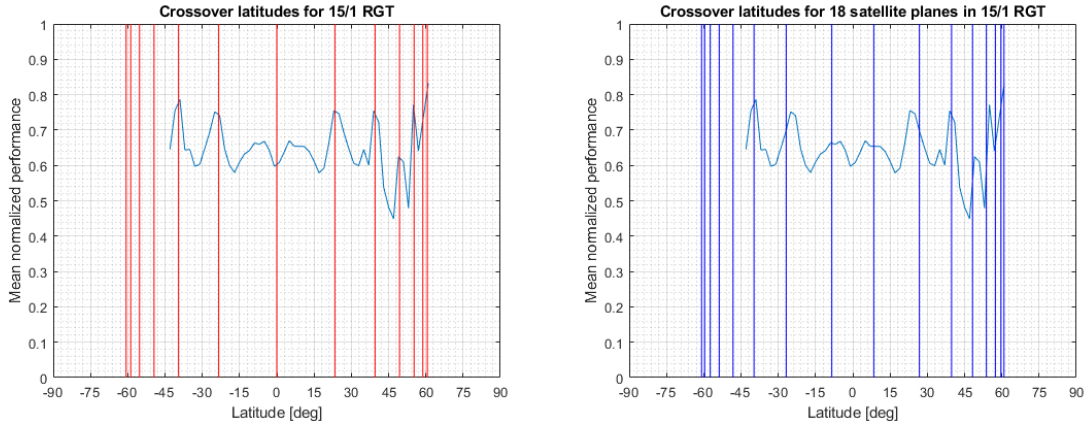


Figure 6-12: Zoomed-in view of ground tracks from the upper-right quadrant of Figure 6-11a, highlighting a 'crossover' point over which both ascending and descending satellite passes occur

in Figures 6-11a and 6-11b) onto the plot showing normalized performance against latitude (from Figure 6-10). Examining the performance peaks in Figure 6-13a, it can be observed that the individual satellite crossover latitudes (shown in red) where a satellite passes over its own ground track traces over the course of a day intersect with almost every peak on the plot, except for the broadest lowest pair of peaks seen at approximately ±6–10° of latitude. However, the peaks seen at around ±24°, ±40°, +49°, +55° and +60° all correlate strongly with the positions of crossover latitudes from Figure 6-11a. Examining the performance of targets located within ±1° of the individual satellite crossover latitudes plotted in red, it was found that these areas have a mean performance that is 15.1% higher than that achieved across all other

latitudes.



(a) Individual satellite crossover latitudes (b) Multi-satellite crossover latitudes

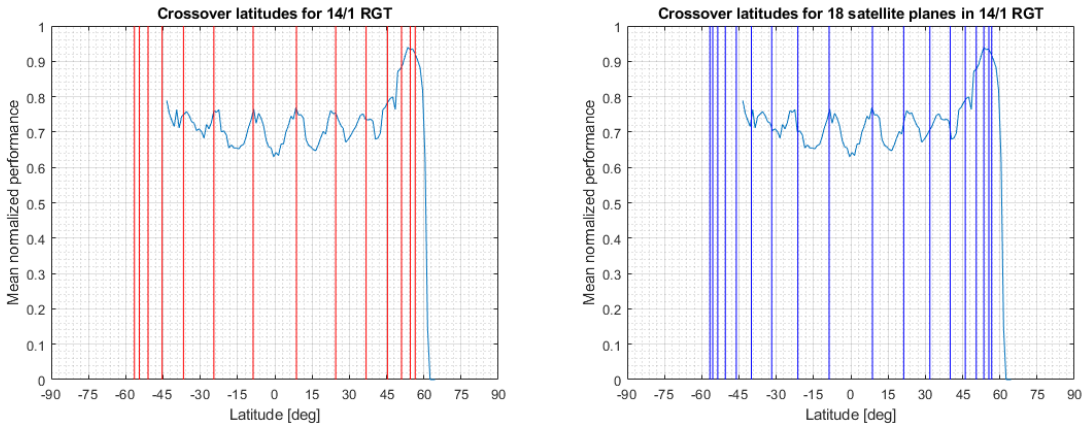
Figure 6-13: The latitudes of the crossover points shown in Figure 6-11 are overlaid onto the mean normalized performance by latitude shown in Figure 6-10

Examining the performance peaks in Figure 6-13b, it can be observed that the multi-satellite crossover latitudes (shown in blue) where a satellite crosses the ground track traces of other satellites in the constellation over the course of a single orbit intersect with some but not all of the peaks on the plot. Unlike Figure 6-13a, in Figure 6-13b there is a crossover within the broadest lowest peak around $\pm 8^{\circ}$, and there is also a reasonable correlation between performance peaks and crossover latitudes at around $\pm 40^{\circ}$ and $+60^{\circ}$. However, the crossover points in Figure 6-13b do not correlate well with the performance peaks seen at approximately $\pm 24^{\circ}$, $+49^{\circ}$ and $+55^{\circ}$. Examining the performance of targets located within $\pm 1^{\circ}$ of the multi-satellite crossover latitudes plotted in blue, it was found that these areas have a mean performance that is 7.98% higher than that achieved across all other latitudes.

The conclusion drawn from the two plots shown in Figures 6-13a and 6-13b is that the combination of both types of crossover points can explain the improved performance, but neither type of crossover can explain the discrepancy alone. In order to verify the existence of this correlation with an additional case, an alternative constellation design was selected for assessment. This design also consists of an 18-satellite

constellation with 18 orbital planes, but differs from the previous example design with a retrograde inclination of 122.89°, a GOM altitude of 912.85 km and a RGT of 14/1. These design parameters were chosen to provide a different set of both individual and multi-satellite crossover points, and the performance of this constellation design was assessed against a different set of 500 target decks (containing 9773 individual events), in order to assess whether the same improved mean performance is seen with a new and distinct set of both types of crossover latitudes.

Figure 6-14 was created to show an overlay of the crossover latitudes of the ground tracks of the selected 18-satellite constellation design in a 14/1 RGT and at a 122.89° retrograde inclination onto the mean normalized performance of this constellation by latitude. This provides a comparison to the plots shown in Figure 6-13 for the previously-assessed 15/1 RGT constellation design at a 61.26° prograde inclination.



(a) Individual satellite crossover latitudes (b) Multi-satellite crossover latitudes

Figure 6-14: The latitudes of 14/1 RGT crossover points overlaid onto the mean normalized performance by latitude achieved by the second sample constellation

Examining the performance peaks in Figure 6-14a, it can be observed that the individual satellite crossover latitudes where a satellite passes over its own ground track (shown in red) intersect with almost every peak on the plot, except for one minor peak seen at around -40° latitude. The peaks seen at around ±9°, ±25°, ±37°, +45° and +51–57° all correlate strongly with the positions of crossover latitudes shown.

Examining the performance of targets located within ±1° of the individual satellite crossover latitudes plotted in red, it was found that these areas have a mean performance that is 10.2% higher than that achieved across all other latitudes.

Examining the performance peaks in Figure 6-14b, it can be observed that the multi-satellite crossover latitudes where a satellite crosses the ground tracks of other satellites in the constellation (shown in blue) also intersect with almost every peak on the plot, except for the peaks seen at around ±37° latitude. The peaks seen at around ±9°, ±25°, -40°, +46° and +51–57° all correlate strongly with the positions of crossover latitudes shown. In some cases the crossover latitudes shown in Figure 6-14b fall extremely close to the crossover points highlighted in Figure 6-14a, which may account for the breadth of some of the performance peaks, where the actual tip of the peak falls between two different lines. Examining the performance of targets located within ±1° of the multi-satellite crossover latitudes plotted in blue, it was found that these areas have a mean performance that is 6.91% higher than that achieved across all other latitudes.

The conclusion drawn from the two plots shown in Figures 6-14a and 6-14b is that once again, the combination of both types of crossover points (a satellite crossing its own ground track and a satellite crossing the ground tracks of other satellites in the constellation) occurring at certain latitudes correlates strongly with the position of all of the peaks where higher performance is observed. However, as was also seen in Figures 6-13a and 6-13b, neither type of crossover can account for the full set of performance peaks. Both crossover types must be considered together to account for the correlation between higher performance and the latitudes at which ascending and descending passes occur over the same ground location.

The same conclusions were drawn from the evaluation of a 15/1 RGT orbit with a prograde inclination of 61.26° and a 14/1 RGT orbit with a retrograde inclination of 122.89°. When assessing the individual satellite crossover latitudes, a higher perfor-

mance improvement was seen (15.1% improvement for the 15/1 prograde RGT and 10.2% for the 14/1 retrograde RGT) compared to the multi-satellite crossover latitudes (7.98% for the 15/1 prograde RGT and 6.91% for the 14/1 retrograde RGT). The need to account for both types of crossover latitudes at which the improved performance is observed means that the number and location of crossover points will be determined by a range of different design parameters for the constellation under consideration. Although fixed values for all or most of these parameters are likely to be selected at the point where the constellation design is finalized prior to launch, the costs and ability to alter each value during the operational phase of the mission will vary widely according to the nature of the variable, but any capability of deliberately altering these latitudes could be extremely beneficial to constellation performance.

6.2.1 Adjustment of crossover latitudes

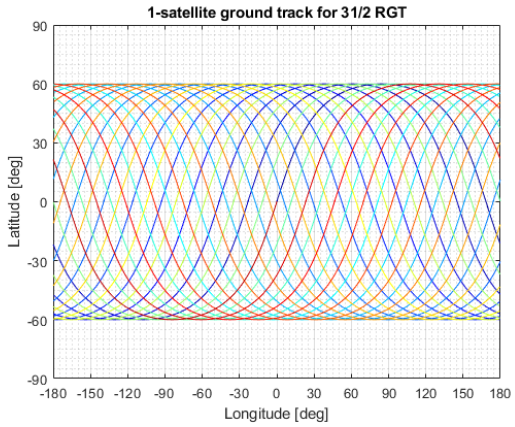
The locations at which crossover points occur are defined by a number of the constellation design parameters. These include the orbital inclination (as shown in Figures 6-5 and 6-6), which affects the upper and lower limits of latitude coverage as previously described, and also whether it is a prograde or retrograde inclination, which affects the width (in terms of longitude) of the ground track for each orbit depending on whether the orbit is moving in the same direction as the Earth's rotation or against it. The total number of satellites in the constellation, how many orbital planes they are launched into, and the spacing in RAAN between these planes are additional factors that affect these latitudes, determining how many other ground traces each satellite will cross during each orbit and how far apart they are spaced, primarily affecting the multi-satellite crossover points (as shown in Figure 6-11b). The choice of RGT orbit is another important factor, determining how many orbits per day each satellite will complete before beginning to repeat the same ground track, and therefore how many times per day each satellite will cross its own ground trace and with what spacing, primarily affecting the individual satellite crossover points (as shown in Figure 6-11a).

In past work using the ReCon codebase, all of these parameters are set during

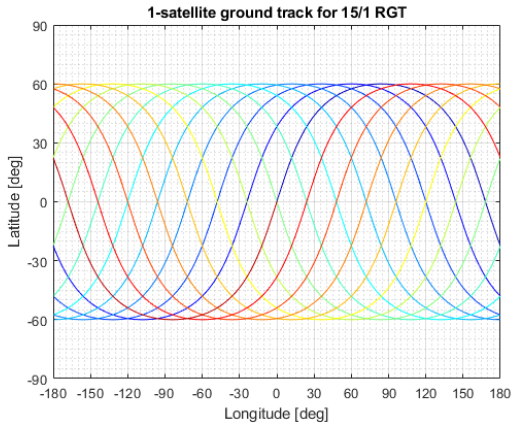
the design phase of the mission, and none of the values can be changed during the operational lifetime of the satellites. Launching new satellites to augment the constellation would be possible after the start of the mission, but would entail extremely high monetary costs to build and launch additional satellites, especially if this was not a planned option factored in during the initial manufacturing of the constellation. If augmenting the size of the constellation entailed changing the spacing of the orbital planes, this would also require additional ΔV usage, curtailing the budget available for reconfiguration and likely diminishing future performance due to the reduction in maneuvering options. Changing the orbital inclination during operations is also theoretically possible but would result in extremely high ΔV costs, likely consuming all or most of the ΔV budget assigned for satellite reconfiguration in response to events. The most feasible parameter to change during the operational lifetime of a reconfigurable constellation is the choice of RGT orbit used to achieve persistent coverage over areas of interest. Changing RGT would require a change in orbital altitude, making this another parameter change which would necessitate use of the ΔV budget, but this would be less costly than other maneuvers that involve the need to change orbital planes. Changing the orbital altitude will also have the benefit of primarily affecting the individual satellite crossover latitudes, which showed greater performance improvements than the multi-satellite crossover latitudes in the two cases evaluated in Section 6.3 and plotted in Figures 6-13 and 6-14.

To examine the feasibility of changing between different RGT options after the launch of the constellation, the six RGT orbits used in the existing ReCon codebase were adopted as the candidate orbits between which maneuvers could be carried out during the operational phase of the mission. These orbits were previously outlined in Table 6.1, consisting of three $n/1$ orbits that orbit n times in a single day before beginning to repeat the same ground track (15/1, 14/1, and 13/1 RGTs), and three $n/2$ orbits that orbit n times over 2 days before repeating the same ground track (31/2, 29/2, and 27/2 RGTs). This set of orbits limits the range of ROM altitudes to between 320.89 km (for a 31/2 orbit at 0° inclination) and 1338.2 km (for a 13/1

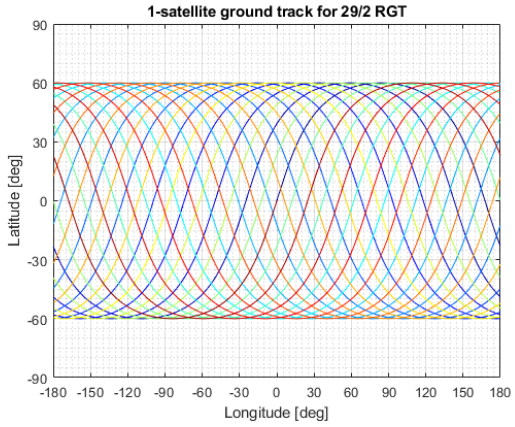
orbit at 180° inclination), providing candidates across the spectrum of LEO options.



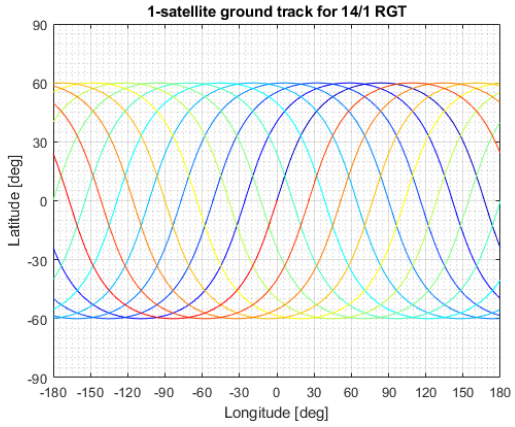
(a) 31 orbits in 2 days



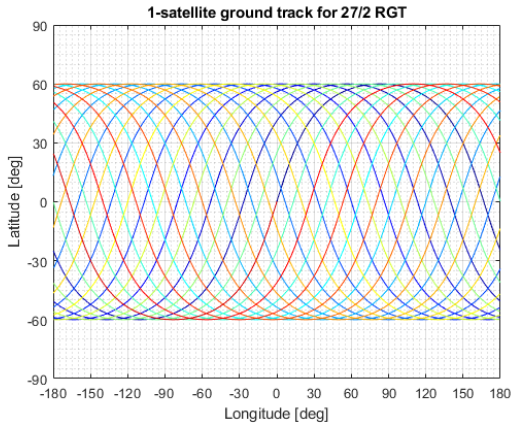
(b) 15 orbits in 1 day



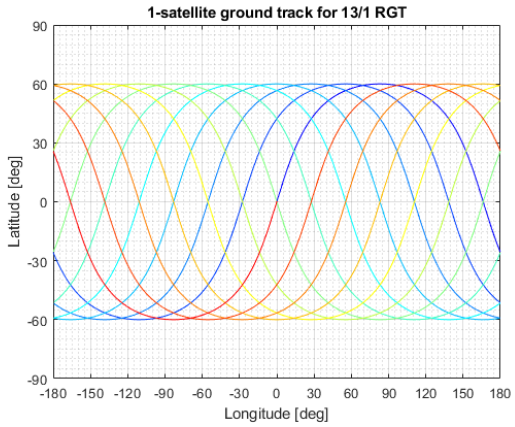
(c) 29 orbits in 2 days



(d) 14 orbits in 1 day



(e) 27 orbits in 2 days



(f) 13 orbits in 1 day

Figure 6-15: Repeating ground tracks plotted for a single satellite in each of the six RGT candidate orbits, assuming a 60° prograde orbital inclination

Figure 6-15 shows plots of the repeating ground tracks for an individual satellite in each of these six orbits, in order of altitude from lowest (31/2, shown in Figure 6-15a) to highest (13/1, shown in Figure 6-15f). A prograde orbital inclination of 60° was selected for the creation of these plots, due to the prevalence of inclination values around this number in non-dominated constellation designs, and the fact that this inclination provides coverage of essentially the full range of target latitudes of interest. Figures 6-15a, 6-15c and 6-15e show the $n/2$ RGTs of 31/2, 29/2 and 27/2 respectively, and are plotted over a period of 2 days to show the full ground track. Figures 6-15b, 6-15d and 6-15f show the $n/1$ RGTs of 15/1, 14/1 and 13/1 respectively, and are plotted over a period of 1 day, again showing the full extent of the repeating ground track.

It is immediately clear from examining Figure 6-15 that $n/2$ RGT orbits feature a greater number of crossover points than the $n/1$ RGT orbits. This is due to the difference in ground tracks between the first and second day for the $n/2$ RGT orbits; although there are a greater number of crossover points, each of these points is only revisited every two days, rather than every day as in the $n/1$ RGTs. The number of crossover points declines as the orbital altitude increases, with 28, 26 and 24 crossover latitudes seen for the 31/2, 29/2 and 27/2 orbits respectively, and 13, 12 and 11 crossover latitudes seen for the 15/1, 14/1 and 13/1 orbits respectively, for the set of prograde orbits (at 60° inclination) plotted in Figure 6-15.

After comparing ground tracks for different orbital inclinations, it was discovered that the number of crossover latitudes for retrograde orbits is slightly higher than the number achieved during prograde orbits, due to the additional circumnavigation of the Earth achieved each day. A comparison of the number of crossover latitudes that occur during each RGT period and per day for each RGT orbit is presented in Table 6.2, showing values for both prograde and retrograde inclinations.

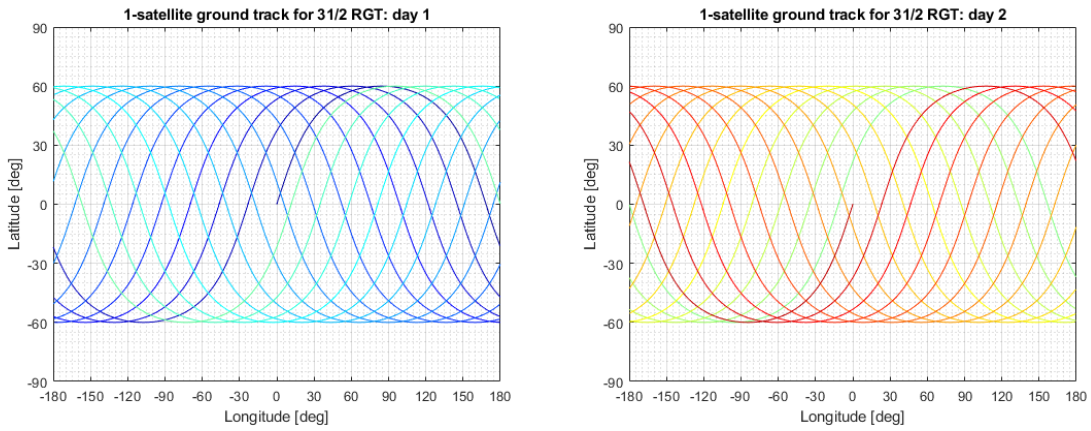
As mentioned in the discussion of Figures 6-5 and 6-6, prograde and retrograde

Table 6.2: Number of individual satellite crossover latitudes that occur per RGT period and per day, for prograde vs retrograde orbits

RGT	**Prograde crossovers during RGT period**	**Prograde crossovers per day**	**Retrograde crossovers during RGT period**	**Retrograde crossovers per day**
31/2	28	14	32	16
15/1	13	13	15	15
29/2	26	13	30	15
14/1	12	12	14	14
27/2	24	12	28	14
13/1	11	11	13	13

orbits complete a different number of circumnavigations compared to the number of orbits per day. For prograde orbits, as the Earth completes one full rotation per day in the same direction as the orbits, each satellite completes one less circumnavigation per day compared to its number of orbits during that time. For example, a satellite in a prograde 15/1 RGT will complete 14 full circumnavigations of the Earth during 15 orbits over the course of one day, as each complete orbit passes over only 336° of longitude. For retrograde orbits, as the Earth is rotating in the opposite direction to the one in which the orbits are travelling, each satellite completes one more circumnavigation per day compared to its number of orbits during that time. For example, a satellite in a retrograde 15/1 orbit will complete 16 full circumnavigations of the Earth during 15 orbits over the course of one day, as each complete orbit passes over 384° of longitude.

Due to this additional circumnavigation, the number of crossover latitudes for retrograde orbits was found to be slightly higher compared to the number observed for prograde orbits. This increase can be seen by comparing the listed number of crossovers for each RGT in Table 6.2. The number of crossover latitudes seen for retrograde $n/2$ orbits increased to 32, 30 and 28 for the 31/2, 29/2 and 27/2 RGTs respectively, and to 15, 14 and 13 for the 15/1, 14/1 and 13/1 orbits respectively: a consistent increase for all six RGTs of 2 additional crossover latitudes per day for retrograde vs prograde orbits.



(a) Day 1 of 2-day RGT

(b) Day 2 of 2-day RGT

Figure 6-16: Ground tracks for a 60° inclination 31/2 RGT orbit, divided into the first and second days of the 2-day repeating ground track

Figure 6-16 shows the ground tracks for the 31/2 RGT (as shown in Figure 6-15a), separated out into the first day of the orbit in Figure 6-16a and the second day of the orbit in Figure 6-16b. This illustrates how the frequency of crossover points is very similar to the $n/1$ plots when considered on a daily basis. It can also be observed that the crossover points are not symmetric in latitude or longitude for the $n/2$ RGT when separated out by individual days, though they are approximately rotationally symmetric. Although there is an overall higher number of crossover latitudes for $n/2$ orbits, the lower revisit cadence for this orbit may not lead to performance gains that are as consistent as the $n/1$ orbits in achieving coverage with a high level of persistence over a specific ground location.

Figure 6-17 separates out the individual satellite crossover latitudes from the full repeating ground tracks shown in Figure 6-15, illustrating the number and distribution of these crossover points for each RGT orbit by orbital altitude (at a 60° inclination). This illustrates which regions possess the possibility of improved crossover latitude performance for each RGT candidate, and how these locations vary between the different orbit options. For example, the 15/1 (plotted in orange) and 13/1 (plotted in pale blue) RGTs both have a crossover latitude at the equator, but the 14/1

(plotted in purple) RGT has the largest gap between crossover points around this area, with the nearest pair of crossovers occurring at ±12.75° latitude. A satellite operator who wanted to ensure improved crossover coverage over equatorial locations might take these crossover points into consideration when making the initial selection of which RGT to use for the constellation. This principle can be applied to any crossover latitude of potential higher value to the operator. Although a reconfigurable constellation can maneuver to achieve persistent coverage over any desired location (within the latitude limits determined by the orbital inclination) by aligning its RGT over the point in question, if a constellation operator wants to ensure the availability of improved crossover coverage over specific latitudes of particular interest, this could motivate the selection of certain RGT candidates over others, as long as there is a crossover available in the desired region.

The difference in altitude between RGT candidates must be considered to assess the feasibility of maneuvering between different RGT orbits during a constellation's operational phase. Figure 6-17 shows the crossover latitudes for each RGT on the y-axis, with the orbital altitudes for each candidate plotted along the x-axis, giving an initial idea of the approximate altitude changes required to switch between these six orbits at a 60° inclination.

Table 6.3 shows the required altitude changes in more detail, listing the altitude differences between all of the six candidate RGT orbits (given in this table as the mean altitude change averaged across all orbital inclinations from 0–180°). Altitude changes between adjacent pairs of orbits vary from a minimum of 149.96 km difference between the 31/2 and 15/1 RGTs (the lowest two RGT candidates in terms of altitude) up to 189.47 km between the 27/2 and 13/1 RGTs (the highest altitude RGT candidates considered in this work). For the sake of completeness, the altitude changes required to move between any of the six orbits are included in this table, up to a difference of 842.92 km that would be needed to move from the lowest 31/2 RGT all the way up to the highest 13/1 RGT orbit. These larger altitude changes are not expected to

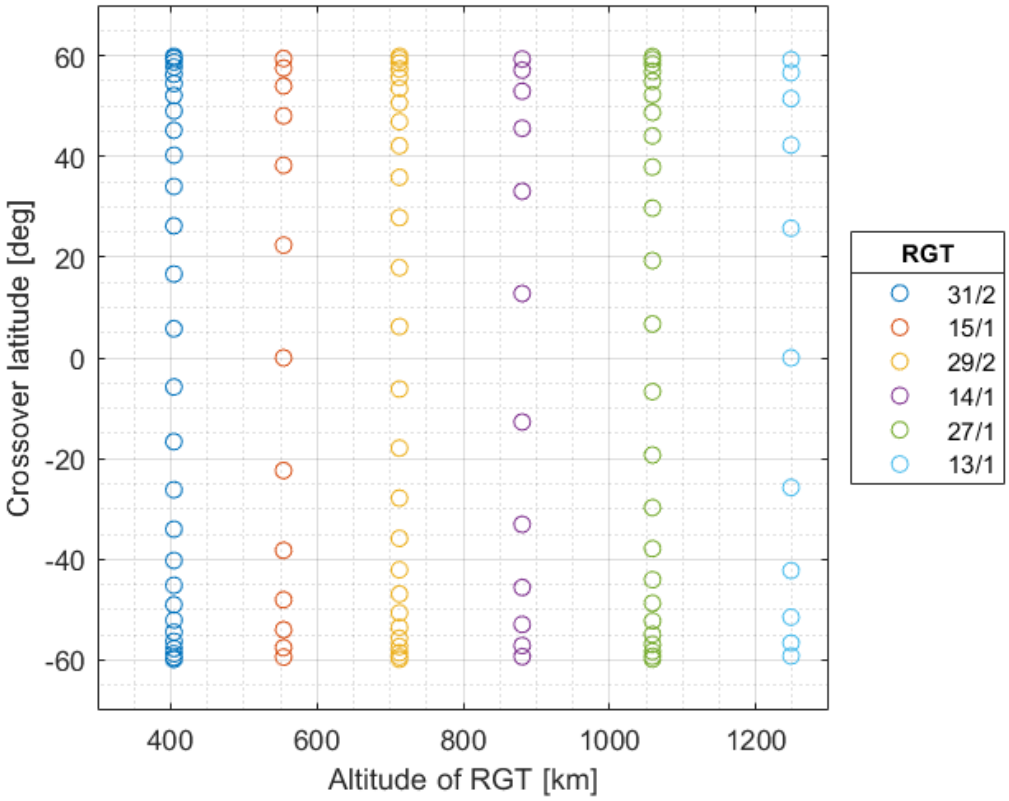


Figure 6-17: Individual satellite crossover latitudes plotted by orbital altitude for all six candidate RGT orbits at 60° inclination

be feasible within the ΔV budget assigned to any of the reconfigurable constellation designs generated by the present version of the ReCon code, although future work could consider some of these more extensive transfers if less strict constraints on the design are found to be a realistic possibility.

Table 6.3: Mean altitude changes that would be required to move between any of the six candidate RGT orbits

Δaltitude [km]		Target RGT (to move to)					
		31/2	**15/1**	**29/2**	**14/1**	**27/2**	**13/1**
Originating RGT (to move from)	**31/2**		149.96	308.46	476.32	654.45	843.92
	15/1	149.96		158.50	326.36	504.50	693.96
	29/2	308.46	158.50		167.86	345.99	535.46
	14/1	476.32	326.36	167.86		178.14	367.60
	27/2	654.45	504.50	345.99	178.14		189.47
	13/1	843.92	693.96	535.46	367.60	189.47	

Table 6.4 shows the ΔV amounts that would be required to achieve the altitude changes needed to move between different RGT orbits. These ΔV values are calculated for the average altitude changes shown in Table 6.3, and will vary by up to $\pm 5\%$ depending on the selected orbital inclination. ΔV amounts to move between adjacent pairs of orbits vary from a minimum of 83.32 m/s to move between the 31/2 and 15/1 RGTs up to a maximum of 91.44 m/s to move between the 27/2 and 13/1 RGTs.

For the sake of completeness, ΔV amounts to move between any of the six RGT orbits are included in Table 6.4, but it is clear from the values shown that larger transfers across several orbits will not be feasible within the current ΔV budget allocated for constellation reconfiguration. In continuing to investigate the opportunity presented by the improved performance seen at crossover latitudes, the option of maneuvering will be limited to considering transfers only between adjacent RGT orbits.

Table 6.4: Mean ΔV changes that would be required to move between any of the six candidate RGT orbits

ΔV [m/s]		Target RGT (to move to)					
		31/2	**15/1**	**29/2**	**14/1**	**27/2**	**13/1**
Originating RGT (to move from)	**31/2**		83.32	168.47	255.55	344.66	435.91
	15/1	83.32		85.17	172.28	261.45	352.77
	29/2	168.47	85.17		87.13	176.33	267.71
	14/1	255.55	172.28	87.13		89.22	180.64
	27/2	344.66	261.45	176.33	89.22		91.44
	13/1	435.91	352.77	267.71	180.64	91.44	

The upper limit on the ΔV budget that can be assigned for reconfiguration in the ReCon code is 1000 m/s, which would mean that moving between two adjacent RGTs uses a minimum of 8–9% of this resource for a single transfer. Many non-dominated constellation designs are optimized with even lower values than this limit, due to potential cost savings in allotting a lower ΔV budget to the satellites. The mean number of events expected over a 5-year mission lifetime is around 20, so using 8–9% or more of the ΔV budget to respond to a single event will not be possible for every location of interest. However, maneuvering between RGTs may still present a useful

option for satellite operators to appraise in cases where achieving coverage of certain targets is considered to be of extremely elevated value above the baseline level of desired coverage.

Table 6.5 shows mean altitude changes (averaged across all inclinations) to the adjacent orbits above and below each candidate. This table combines data from Tables 6.3 and 6.4 to highlight the two maneuvering options left under consideration for each RGT, either moving up or down the list by one candidate.

Table 6.5: Mean altitude changes and ΔV required to change between adjacent RGT orbits

Orbits/days for selected RGT	Mean orbital altitude [km]	Mean altitude change to next lower RGT [km]	Mean ΔV to change to next lower RGT [m/s]	Mean altitude change to next higher RGT [km]	Mean ΔV to change to next higher RGT [m/s]
31/2	409.75	n/a	n/a	149.96	83.32
15/1	559.71	149.96	83.32	158.50	85.17
29/2	718.21	158.50	85.17	167.86	87.13
14/1	886.07	167.86	87.13	178.14	89.22
27/2	1064.2	178.14	89.22	189.47	91.44
13/1	1253.7	189.47	91.44	n/a	n/a

For the lowest-altitude orbit of a 31/2 RGT, the only option presented in Table 6.5 is to increase altitude to a 15/1 RGT, to avoid adding additional candidate orbits to the list at this stage. In addition, the lower orbit altitude required to enter a 16/1 RGT would potentially (depending on the orbital inclination for the constellation in question) contravene the minimum altitude constraint of 300 km imposed in the Re-Con codebase. This means that even if extra RGT candidate orbits are considered for investigation in future work to expand the options available to satellite operators, the 31/2 RGT is the lowest-altitude option that does not violate the design constraints at some of the possible orbital inclination values, and so expansion to higher-altitude orbits should instead be considered, or investigation of other RGT cadences such as $n/3$ orbits.

For the highest-altitude orbit of a 13/1 RGT, the only option presented in Table 6.5 is to decrease altitude to a 27/2 RGT, again avoiding the addition of extra candidates at this stage, and recognizing that a cutoff must be imposed at some point in the list of maneuvering options. It can also be observed from Table 6.5 that the ΔV required to change to an adjacent RGT increases as the altitudes of the orbits increase, and so adding additional higher-altitude RGT options would require higher quantities of ΔV to maneuver into these orbits. As the listed ΔV quantities of 83.32–91.44 m/s are already impractically high for universal application in response to every event of interest while staying within the available ΔV budget for constellation reconfiguration, this set of six orbits is considered to provide a reasonable feasibility study on the option of maneuvering between RGTs for a high-value subset of targets during the operational phase of the mission. If changing RGTs proves to be of sufficient benefit for adoption as a decision option for satellite operators, future work may choose to expand this list of options as far as the ΔV requirements may be deemed attainable.

Figure 6-18 shows the effect of different orbital inclinations on the change in altitude that is required to transition between adjacent pairs of RGT orbits. As shown in the legend of this figure, the differences in altitude are unaffected by the direction of the transfer: moving from a 31/2 to a 15/1 RGT requires the same altitude change as moving from a 15/1 to a 31/2 RGT. The largest altitude changes are required at the lowest orbital inclination; for example, moving from a 31/2 RGT to a 15/1 RGT at 0° inclination requires an altitude change of 155.2 km, but transferring between the same two orbits at 180° inclination requires an altitude change of only 144.6 km.

Figure 6-19 shows the effect of different orbital inclinations on the amount of ΔV required to transition between adjacent pairs of RGT orbits. These ΔV values are based on carrying out a Hohmann transfer between the two circular orbits, which will provide a lower bound on the amount of ΔV required. If a drift orbit is required as part of the transfer to change the phasing of the RGT over a specific ground location of interest, the ΔV required will be higher. As shown in the legend of Figure 6-19, the

amount of ΔV required for a straightforward transfer between two RGTs is unaffected by the direction of the transfer: a Hohmann transfer from a 31/2 to a 15/1 RGT uses the same amount of ΔV as a transfer from a 15/1 to a 31/2 RGT.

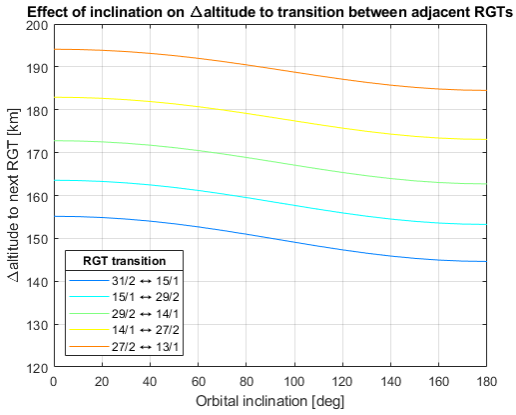


Figure 6-18: Effect of inclination on altitude change required to transition between adjacent pairs of RGT orbits

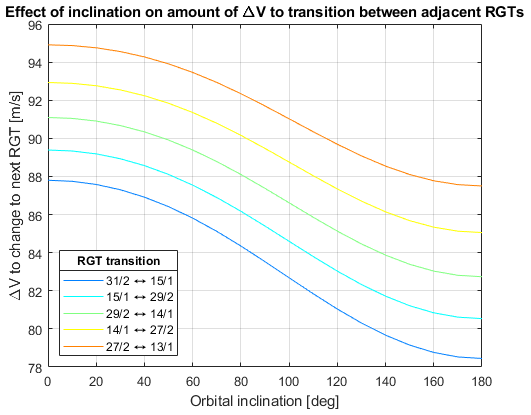


Figure 6-19: Effect of inclination on ΔV required to transition between adjacent pairs of RGT orbits

The largest ΔV requirements to change RGTs occur at the lowest orbital inclinations, where the largest altitude changes are required to move between adjacent candidate orbits. For example, moving from a 31/2 RGT to a 15/1 RGT at 0° inclination (requiring an altitude change of 155.2 km) takes a ΔV of 87.8 m/s, but a transfer between the same two orbits at 180° inclination (with an altitude change of 144.6 km) takes a ΔV of 78.4 m/s. The impact of inclination is slightly greater for lower orbits, with a larger ΔV difference of 9.4 m/s observed between the highest and lowest ΔV requirements for the 31/2 $\leftrightarrow$ 15/1 transfer (plotted in dark blue in Figure 6-19), and a smaller ΔV difference of 7.4 m/s observed between the highest and lowest ΔV requirements for the 27/2 $\leftrightarrow$ 13/1 transfer (plotted in orange in Figure 6-19).

In summary, the location of crossover latitudes is affected by multiple interacting design variables, including the total number of satellites in the constellation, the number of orbital planes and the RAAN spacing between them, the orbital inclination

(and whether it is a prograde or retrograde orbit), and the choice of RGT orbit. Of these, the most feasible variable to adjust once the constellation is already launched and operational is the choice of RGT orbit.

With the current constraints imposed for the optimization of reconfigurable constellation designs, the largest possible realistic ΔV budget for reconfiguration is 1000 m/s. To stay within this maneuvering budget, the achievable RGT changes are limited to switching between adjacent pairs of RGT orbits. The mean ΔV (averaged across all orbital inclinations) for these transfers between neighboring RGTs varies between 83.32 and 91.44 m/s, with the higher-cost transfers occurring at the higher-altitude pairs of orbits. Considering ΔV costs across all orbital inclinations, the lowest-ΔV transfer occurs for the highest-inclination lowest-altitude pairing, requiring 78.44 m/s to transfer between 31/2 and 15/1 RGTs at 180° inclination. The highest-ΔV transfer between adjacent RGTs occurs for the lowest-inclination highest-altitude pairing, requiring 94.91 m/s to transfer between 27/2 and 13/1 RGTs at 0° inclination.

Even with the highest available maneuvering budget of 1000 m/s, each transfer between adjacent RGTs will use up at least 7.8–9.5% of the constellation's ΔV. For constellation with smaller amounts of ΔV allocated for reconfiguration, this percentage will be even higher. The mean number of events of interest over a 5-year mission lifetime is 20, with a range of 14–26 events modelled within the array of 5-year target decks generated for use in optimizing constellation designs and assessing performance against a range of possible scenarios. This makes it clear that maneuvering between RGTs is not a feasible option in response to every event of interest during the mission lifetime. However, the cost of such a transfer is not so high as to entirely eliminate it from consideration for use in a subset of targets. Adding this maneuver as an available decision option for satellite operators in the case of occasional high-value targets provides the possibility of increasing constellation performance by a significant amount in extraordinary circumstances. This would allow for constellation operators to assess the desired prioritization between the additional ΔV costs as a proportion

of the remaining reconfiguration budget and the expected performance improvement of 10–15% in situations where exceptional coverage is required.

6.2.2 Effect of elevation angle on crossover latitude coverage

The selection of the minimum elevation angle ϵ_{min} has a significant effect on the ground coverage achieved by a constellation. The spacecraft elevation angle ϵ is the angle between the local horizontal plane at a ground location and the elevation at which the spacecraft is observed in the sky, where 90° would be a direct overhead pass. For optical imaging applications, a minimum elevation angle of 45–60° is generally required to acquire clear ground imagery that is not obscured by topographic features or heavily built-up areas.

Figure 6-20 illustrates the geometry of the angular relationships between a satellite, the subsatellite point (SSP) on the Earth's surface, a target location of interest on the Earth's surface, and the Earth's center. The satellite is shown at the right of the figure, with an orbital altitude of h above the SSP, and maximum slant range of D_{max} to the outer or true horizon. A target location is shown on the Earth's surface, with a slant range D to the satellite, and an elevation angle of ϵ measured at the target between the local horizontal plane and the satellite. R_E is the radius of the Earth, and the Earth central angle λ is measured at the Earth's center between the SSP and the target. The nadir angle η is measured at the satellite between the SSP (in the nadir direction from the spacecraft) and the target. The maximum Earth central angle λ_{max} is also shown, measured at the Earth's center between the SSP and the outer horizon, and the angular radius of the Earth ρ, measured at the satellite between the Earth's center and the outer horizon.

The angular radius of the Earth, ρ, and the maximum Earth central angle, λ_{max}, can be calculated from the Earth's radius R_E and the satellite altitude h as follows:

$$\sin \rho = \cos \lambda_{max} = \frac{R_E}{R_E + h} \tag{6.10}$$

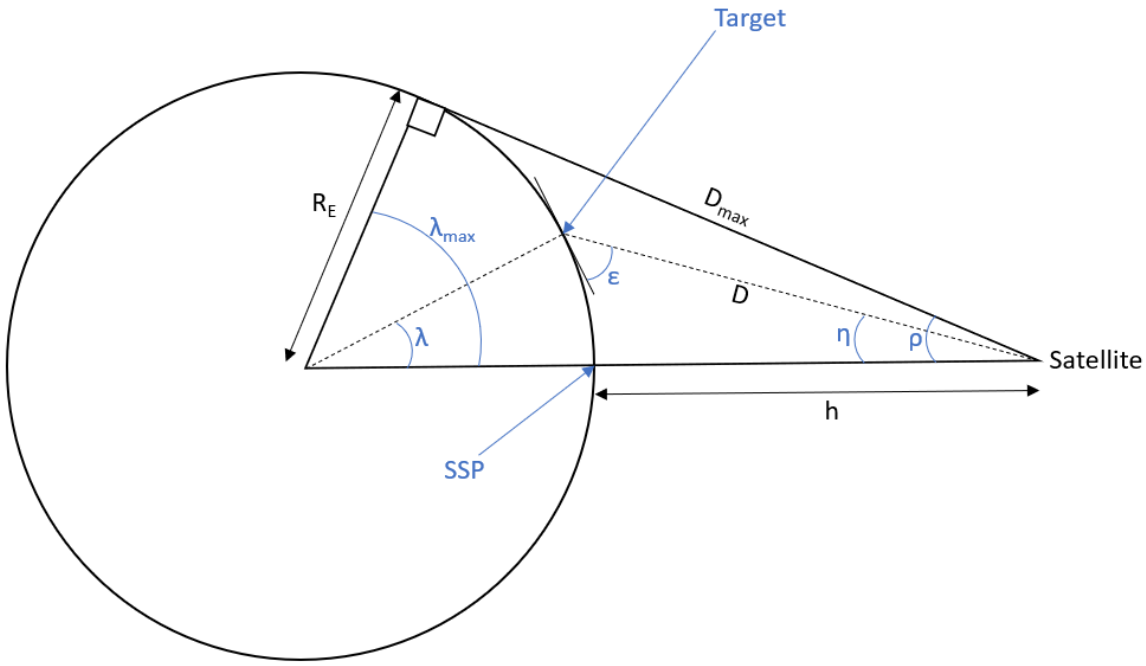


Figure 6-20: Diagram of the angular relationships between a satellite, a target location on the Earth's surface, and the Earth's center

The slant range distance to the outer horizon, D_{max}, may then be calculated using R_E and the value calculated for ρ:

$$D_{max} = \frac{R_E}{\tan \rho} \tag{6.11}$$

Depending on whether the Earth central angle λ, the elevation angle ϵ or the nadir angle η is known, the unknown angles may be calculated using a combination of the following equations:

$$\text{If } \eta \text{ is known, find } \epsilon \text{ using:} \quad \cos \epsilon = \frac{\sin \eta}{\sin \rho} \tag{6.12}$$

$$\text{If } \epsilon \text{ is known, find } \eta \text{ using:} \quad \sin \eta = \cos \epsilon \sin \rho \tag{6.13}$$

$$\text{If } \lambda \text{ is known, find } \eta \text{ using:} \quad \tan \eta = \frac{\sin \rho \sin \lambda}{1 - \sin \rho \cos \lambda} \tag{6.14}$$

$$\text{The remaining angle may be calculated using:} \quad \eta + \epsilon + \lambda = 90^\circ \tag{6.15}$$

Once the Earth central angle λ and the nadir angle η are both known, the slant range

D between the satellite and the target can be calculated as follows:

$$D = R_E \frac{\sin \lambda}{\sin \eta} \tag{6.16}$$

If a particular minimum elevation angle ϵ_{min} is selected for the desired satellite coverage and used to find the Earth central angle λ, this can be used to estimate the swath width w_{swath} of satellite coverage (the distance between the SSP and target location) that will be achieved with the specified parameters as follows:

$$w_{swath} = 2\pi R_E \frac{\lambda}{360} \tag{6.17}$$

Equation 6.17 assumes that λ is calculated in degrees; if it calculated in radians, the swath width may be calculated even more simply by multiplying λ by R_E.

Legge compared the Value of Reconfigurability (VoR, which he defined as the saving in system cost for a reconfigurable constellation compared to an iso-performance static constellation) at a minimum elevation angle of 45° and 60°, finding that VoR was around twice as high for an increased ϵ_{min} value of 60°. Reconfigurability was found to provide a significantly higher benefit when a higher ϵ_{min} constraint is applied, as this reduces the usable ground footprint of each satellite, worsening the coverage that can be provided by static constellation designs. Legge carried out a sensitivity analysis of eight design parameters and constraints and found that the VoR was most sensitive to changes in the minimum elevation angle. He adopted a minimum elevation angle of 60° with the justification that this would ensure high-quality satellite coverage even in areas with tall buildings or drastic variations in terrain elevation, but hypothesized that further increasing ϵ_{min} above 60° would lead to even greater increases in VoR.

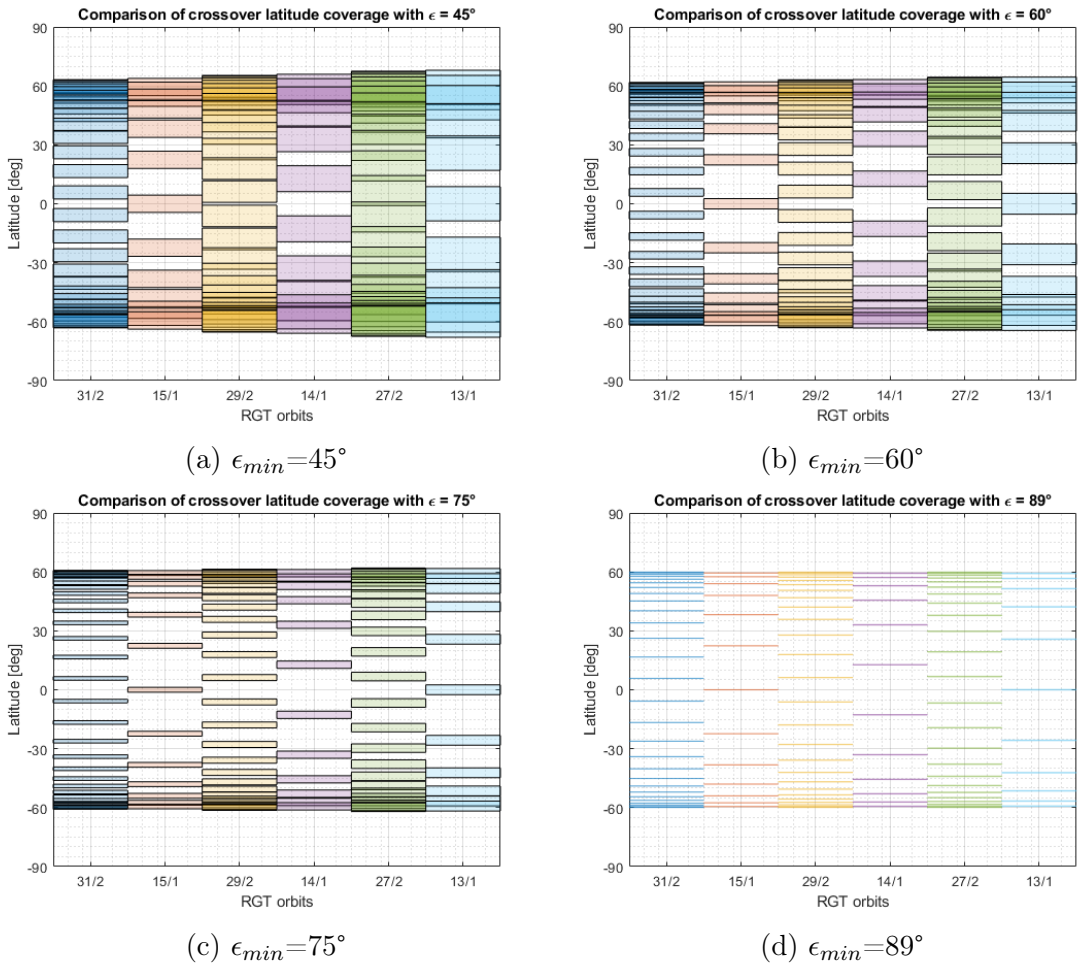
Decreases in the requirement for ϵ_{min} lead to significant increases in the satellite ground footprint within which usable coverage is achieved. This leads to substantial improvements in coverage and therefore constellation performance, especially for

static designs, which have no way of maneuvering to improve coverage outside of the static ground footprint for the chosen orbit. Reconfigurable constellation designs are less affected by the choice of ϵ_{min}, due to the fact that most of the satellite passes over areas of interest are achieved by maneuvering into RGTs, which ensure that the SSP intersects directly with the target location (where ϵ is 90°, and λ and η are both 0°). However, reconfigurable constellation performance still sees some improvement when minimum elevation angle is reduced, as some coverage is achieved for each event by satellites which do not reconfigure into RGTs and remain in GOM throughout the event period of interest, and the performance of these satellites will be improved by increasing the footprint of usable coverage.

Considering the possibility of maneuvering between adjacent RGT orbits as described in Section 6.3.1, it was decided to conduct an evaluation of the extent of crossover latitude coverage that can be achieved when different levels of constraint are placed upon the minimum elevation angle for usable satellite coverage. A 60° orbital inclination was assumed for all RGT orbits, using the set of six ground tracks shown in Figure 6-15. The method of swath width estimation presented in Equation 6.17 was incorporated into approximating the extent of the latitudinal coverage that can be achieved at each crossover point. Coverage was assessed for latitudes between 60°N and 60°S; these were chosen as appropriate limits based on the latitudinal distribution used to generate target decks. The limits were also validated by assessment of the distribution of global population in order to evaluate likely disaster impact on populated areas; 99.54% of the world population lives within this latitude range, with 0.46% of the population located north of 60°N, and 0% below 60°S.[123]

Figure 6-21 shows a comparison of the extent of crossover latitude coverage that can be achieved for each of the six RGT candidates. The swath width for each RGT is estimated based on the Earth central angle λ for each of four possible minimum elevation angle requirements: 45° (shown in Figure 6-21a), 60° (shown in Figure 6-21b), 75° (shown in Figure 6-21c) and 89° (shown in Figure 6-21d, and used as an

approximate stand-in for direct overhead coverage).



(a) ϵ_{min}=45°

(b) ϵ_{min}=60°

(c) ϵ_{min}=75°

(d) ϵ_{min}=89°

Figure 6-21: Comparison of the extent of possible crossover latitude coverage (shown as shaded areas of color) achievable for each of the six RGTs at four different minimum elevation angle requirements

With a minimum elevation angle of 45° as shown in Figure 6-21a, crossover latitude coverage can be achieved for between 64% (for the 15/1 RGT) and 100% (for the 27/2 RGT) of the latitude range between 60°N and 60°S. These crossovers will still occur at distinct points spaced in longitude, so this is not equivalent to improved performance at 64–100% of all locations, as the crossovers may occur at the desired latitude but some distance away in longitude. For a reconfigurable constellation where drift orbits can be used to maneuver a RGT to the desired coordinates in longitude, this does enable the possibility of maneuvering the crossover points over any desired

location. For a static RGT, this means that improved coverage incorporating both ascending and descending passes is possible over the entire latitude range of interest for a 27/2 RGT, but it is unlikely to also occur at the exact longitudinal coordinates of interest, and maneuvering the orbit to intersect with this point is not possible without reconfigurability. However, the drastic improvements in swath width due to a lower minimum elevation angle lead to a significant improvement in usable satellite footprint even without maneuverability, and static constellations see a greater performance improvement than reconfigurable designs with the lowered constraint on achievable footage. The swath width increases with increased orbital altitude, ranging from a minimum of ±3.3° of the latitude for the lowest-altitude 31/2 RGT up to a maximum of ±8.7° of latitude for the highest-altitude 13/1 RGT.

With a minimum elevation angle of 60° as shown in Figure 6-21b, crossover latitude coverage can be achieved for between 45% (for the 15/1 RGT) and 89% (for the 27/2 RGT) of the latitude range between 60°N and 60°S. As the number of overlaps in crossover coverage decrease compared to Figure 6-21a, the correlation between increased achievable swath width and increased orbital altitude (moving left to right across the plot from lower to higher altitude orbits) can be seen more clearly. The narrowest swath width of ±2.0° of latitude is seen for the 31/2 RGT coverage with the lowest altitude (plotted in dark blue on the left side of the figure), and the widest swath width of ±5.3° of latitude is seen for the 13/1 RGT coverage with the highest altitude (plotted in pale blue on the right side of the figure).

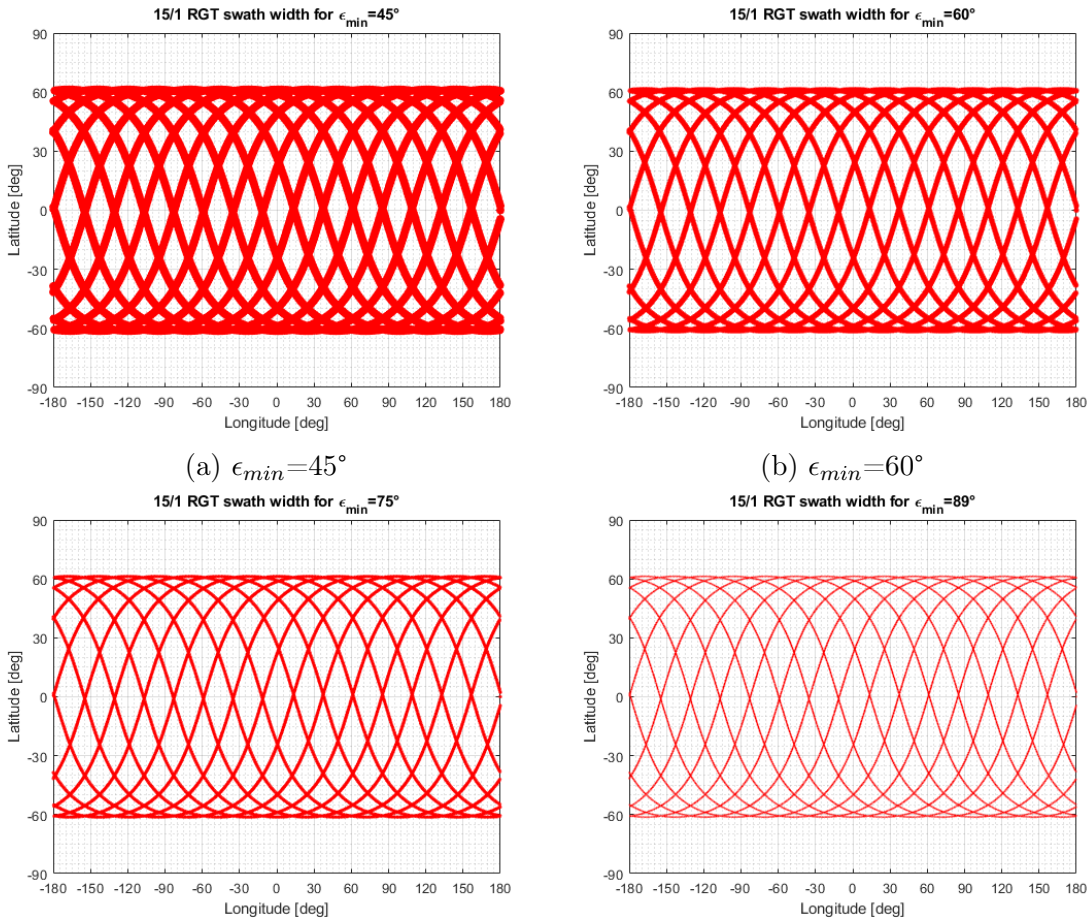
With a minimum elevation angle of 75° as shown in Figure 6-21c, crossover latitude coverage can be achieved for between 25% (for the 15/1 RGT) and 59% (for the 27/2 RGT) of the latitude range between 60°N and 60°S. Most of the coverage swaths are well-spaced for this minimum elevation angle, with the only remaining overlaps between coverage boxes occurring towards the limits of coverage, outside the coordinates of ±50° latitude. The crossover coverage swath widths continue to narrow with the increasingly stringent elevation requirements, ranging from a minimum of ±0.91°

of latitude for the lowest-altitude 31/2 RGT up to a maximum of ±2.5° of latitude for the highest-altitude 13/1 RGT.

A minimum elevation angle of 89° is shown in Figure 6-21d, and this is used as the closest available approximation for direct overhead coverage. A direct nadir pass would have an elevation angle of 90°, but using this angle resulted in coverage calculations of zero width over a point location, so an offset of 1° was implemented to give a narrow acceptable swath width. The areas of crossover latitude coverage now appear as lines of color in the plot rather than outlined boxes, due to the much narrower swath widths achievable with such a high minimum elevation requirement. Crossover latitude coverage can be achieved for between 1.7% (for the 15/1 RGT) and 5/7% (for the 27/2 RGT) of the latitude range between 60°N and 60°S. These swath widths vary from a minimum of ±0.06° of latitude for the lowest-altitude 31/2 RGT (resulting in a coverage band of approximately ±6.6 km from the SSP) up to a maximum of ±0.16° of latitude for the highest-altitude 13/1 RGT (resulting in a coverage band of approximately ±18.2 km from the SSP).

As expected, Figure 6-21 illustrates that imposing stricter minimum elevation angle constraints on the constellation results in more limited coverage, due to the reduction in the size of the usable ground footprint for each satellite. Figure 6-22 shows a comparison of the usable ground coverage for the same four minimum elevation angles shown in Figure 6-21, illustrating the effect of elevation angle on swath width using plots showing a one-day ground track for a 15/1 RGT. These ground tracks correspond to the orange coverage boxes shown in Figure 6-21 (plotted second from the left in each subplot), with the four coverage subplots i–iv in Figure 6-21 corresponding to the same order of elevation angle for the four ground track subplots i–iv shown in Figure 6-22.

The largest estimated swath width occurs for the lowest minimum elevation angle of 45° (shown in Figure 6-22a), resulting in a usable coverage band of ±492 km on



(a) ϵ_{min}=45°

(b) ϵ_{min}=60°

(c) ϵ_{min}=75°

(d) ϵ_{min}=89°

Figure 6-22: Comparison of usable ground coverage for a 15/1 RGT (plotted for a single satellite over one day) based on four different minimum elevation angle requirements

either side of the SSP. This corresponds to achievable crossover coverage in 64.1% of the latitude range between 60°N and 60°S. Examining Figure 6-22a illustrates how this does not equate to crossover coverage over 64.1% of all locations, but shows that latitude bands where crossovers exist occur in horizontal stripes around 0°, ±22°, ±38°, and so on in increasingly close bands approaching the outer limit of latitudinal coverage around ±60°. For reconfigurable constellations, the crossover points within these bands of potential coverage can be moved in longitude using drift orbits. For static constellations, Figure 6-22a shows the extent of achievable coverage for a 15/1 RGT at 60° inclination with 45° minimum elevation angle required.

For the 15/1 repeating ground track using a 60° minimum elevation angle (shown in Figure 6-22b), crossover coverage is achievable over 44.9% of the latitude range. Using this value of ϵ_{min}, the width of the ground coverage band is reduced to ±291 km.

For the 15/1 RGT using a 75° minimum elevation angle (shown in Figure 6-22c), crossover coverage is achievable over 24.5% of the latitude range. Using this value of ϵ_{min}, the width of the ground coverage band is reduced to ±136 km.

For the 15/1 RGT using a 89° minimum elevation angle to approximate a requirement for direct nadir passes (shown in Figure 6-22d), crossover coverage is achievable over 1.73% of the latitude range. Using this value of ϵ_{min}, the width of the ground coverage band is reduced to ±8.9 km.

The subplots shown in Figure 6-22 illustrate the drastic reduction in usable ground footprint for each satellite as the minimum elevation angle requirements are increased from 45°, through 60° and 75°, and up to a maximum of 89°. For static constellation designs, this results in considerably worse coverage throughout the mission lifetime, due to the small amount of coverage that can be supplied along an unchanging ground track. For reconfigurable constellations, the satellites are able to maneuver to achieve coverage over any desired ground location (within the latitude limits imposed by the choice of orbital inclination), and so the impact of elevation angle on performance is much less significant.

It is theorized that additional performance benefits could be gained for reconfigurable architectures by adding a decision option for constellation operators to maneuver between adjacent RGTs. This would be advantageous in circumstances where changing the RGT would allow for crossover points (where both an ascending and descending pass occur over the same ground location) to be maneuvered to intersect with a high-value target location of interest. The increased ΔV cost of such a maneuver would need to be balanced against the remaining ΔV budget available for

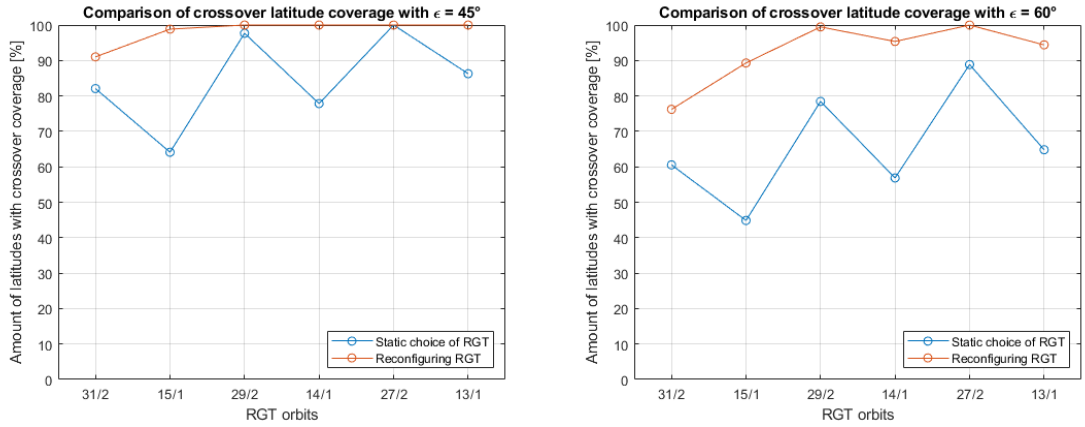
constellation reconfiguration, as it is not feasible to change RGTs in response to every single event of interest over the course of the mission lifetime with the current maneuvering budget. This option is therefore only proposed for use in a subset of especially high-value targets, at the discretion of the satellite operator and their relative level of priority for achieving coverage of different locations.

Figure 6-23 shows the percentage of the latitude range (again assessed between limits of 60°N and 60°S) where crossover coverage can be achieved for each of the six RGT orbits, comparing the extent of potential crossover point coverage for a single unchanging choice of RGT (plotted in blue) and the coverage that can be achieved if maneuvering between pairs of adjacent RGTs is an option (plotted in orange).

Figure 6-23a shows the potential crossover coverage at a minimum elevation angle of 45°. The achievable crossover coverage for a single unchanged RGT varies between 64% (for a 15/1 RGT) and 100% (for a 27/2 RGT) of the latitude range, but this increases to a range of 91% (for a 31/2 RGT) to 100% (for a 29/2, 14/1, 27/2 or 13/1 RGT) when maneuvering to adjacent RGTs is allowed. The percentage improvement in available crossover coverage varies from 0% for the 27/2 RGT (as this orbit already achieved 100% crossover coverage for the latitude range) to 54% for the 15/1 RGT.

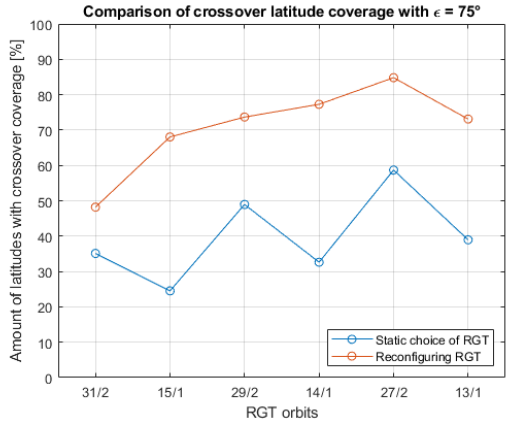
Figure 6-23b shows the potential crossover coverage at a minimum elevation angle of 60°. The achievable crossover coverage for a single unchanged RGT varies between 45% (for a 15/1 RGT) and 89% (for a 27/2 RGT) of the latitude range. This increases to a range of 76% (for a 31/2 RGT) to 100% (for a 27/2 RGT) when maneuvering to adjacent RGTs is allowed. The percentage improvement in available crossover coverage varies from 13% for the 27/2 RGT to 99% for the 15/1 RGT.

Figure 6-23c shows the potential crossover coverage at a minimum elevation angle of 75°. The achievable crossover coverage for a single unchanged RGT varies between 25% (for a 15/1 RGT) and 59% (for a 27/2 RGT) of the latitude range. This increases

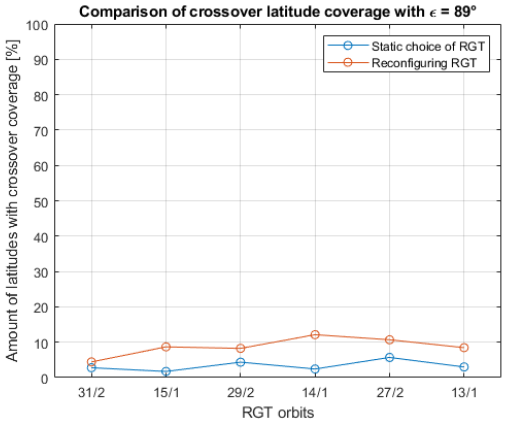


(a) ϵ_{min}=45°

(b) ϵ_{min}=60°



(c) ϵ_{min}=75°



(d) ϵ_{min}=89°

Figure 6-23: Percentage of latitude range where crossover coverage is achievable for each RGT orbit, compared between static choice of a single RGT (in blue) vs reconfiguring between adjacent RGTs (in orange), compared for four different minimum elevation angle requirements

to a range of 48% (for a 31/2 RGT) to 85% (for a 27/2 RGT) when maneuvering to adjacent RGTs is allowed. The percentage improvement in available crossover coverage varies from 37% for the 31/2 RGT to 177% for the 15/1 RGT.

Figure 6-23d shows the potential crossover coverage at a minimum elevation angle of 89°. The achievable crossover coverage for a single unchanged RGT varies between 1.7% (for a 15/1 RGT) and 5.7% (for a 27/2 RGT) of the latitude range. This increases to a range of 4.4% (for a 31/2 RGT) to 12.1% (for a 14/1 RGT) when maneuvering to adjacent RGTs is allowed. The percentage improvement in available

crossover coverage varies from 59% for the 31/2 RGT to 401% for the 15/1 RGT.

Figure 6-24 summarizes the improvements in crossover coverage seen in Figure 6-23 when maneuvering between adjacent RGTs is added as a reconfiguration option for the constellation. The largest improvements in the amount of crossover latitude coverage are seen for $n/1$ orbits, as the available maneuvering options for these orbits are to move up or down by one RGT to an $n/2$ orbit. This introduces a higher number of crossover points, though these are each revisited at a lower frequency. Smaller improvements are seen for the $n/2$ orbits, which have the option to maneuver up or down by one RGT to an $n/1$ orbit.

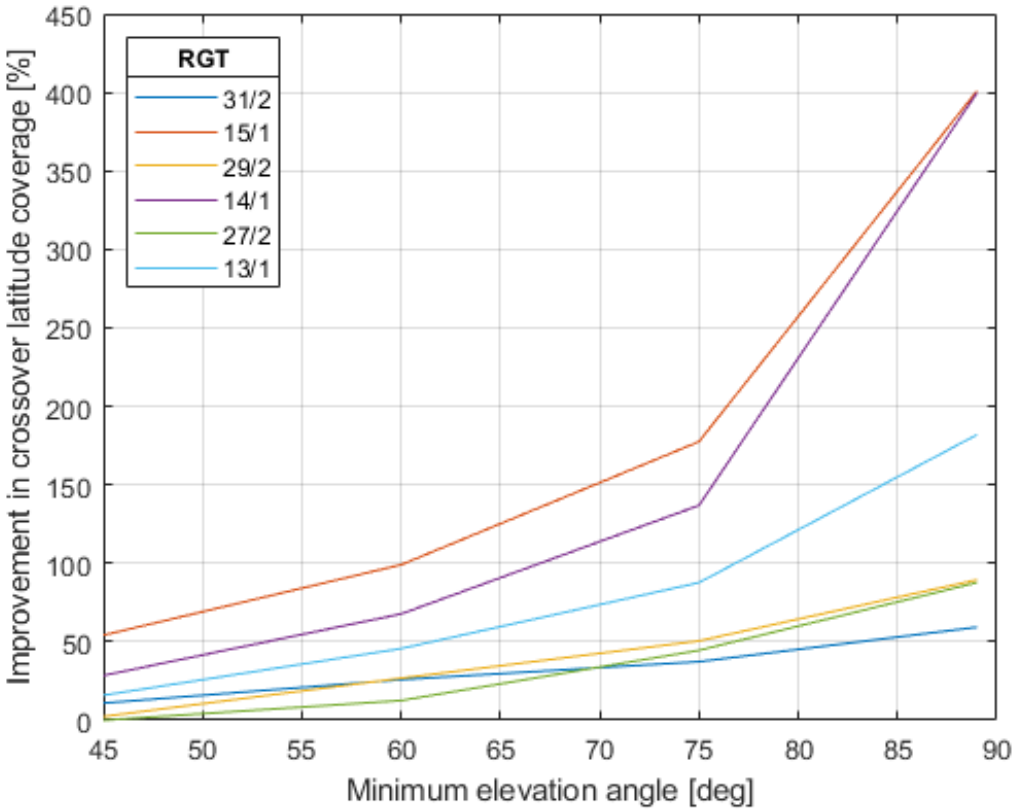


Figure 6-24: The effect of minimum elevation angle requirements on percentage improvement in the amount of achievable crossover latitude coverage when maneuvering between adjacent RGTs is implemented

The greatest improvements in availability of crossover coverage from maneuvering

between RGTs are seen with the most stringent requirements for minimum elevation angle. At ϵ_{min}=45°, switching RGTs offers 0–54% improvement depending on the initial RGT selection, and this minimal improvement can be attributed to the wide satellite footprint already available at this elevation angle without the need for changing orbits. At ϵ_{min}=60°, the improvement in possible crossover latitudes when maneuvering is implemented increases to 13–99%, suggesting that at Legge's original elevation angle requirements, changing RGTs offers a useful option for satellite operators to improve performance. At ϵ_{min}=75°, changing RGTs offers 37–177% improvement in the latitudes where crossovers can be achieved, and at ϵ_{min}=89°, 59–401% improvement in crossover range is seen, reflecting the extremely narrow coverage bands available when direct nadir passes are required.

Maneuvering between RGTs is an expensive option for satellite operators, using at least 7.8–9.5% of the constellation's lifetime ΔV budget for a single transfer, and potentially significantly higher percentages for some constellation designs where a lower ΔV quantity is allocated for maneuvering. These high costs make it clear that changing RGTs in response to every event of interest over the course of the mission lifetime is not a feasible reconfiguration option. However, the ΔV costs are within a range where this maneuver is considered to be practical in occasional extraordinary circumstances, given sufficient benefits to be gained in increased performance.

Reconfigurable constellations already make use of drift orbits to precess repeating ground tracks in longitude and align satellite coverage with the ground location of interest. However, if crossover points (where ascending and descending satellite passes intersect over the same ground location) do not occur at the desired latitude, there is currently no way to align these points with the target location. Changing between adjacent RGT orbits supplies the additional option of altering the crossover locations in latitude, providing the possibility of increasing constellation performance by an average of 10–15%. Adding such a maneuver as an option for constellation operators to employ in response to extremely high-value targets allows for responsiveness in the

prioritization of constellation resources. The opportunity costs of using additional ΔV and reducing the remaining budget for future reconfiguration maneuvers must be weighed against the improved performance to be gained when crossover points are aligned over a ground location of interest to give exceptional coverage using both ascending and descending satellite passes.

The extent to which these crossover points can be aligned as desired depends heavily on the initial choice of RGT and the minimum elevation angle requirements for coverage, and so the specific decision options under consideration should be modelled based on a finalized constellation architecture. When satellite coverage is subject to stricter constraints on elevation angle, the usable ground footprint is decreased, with the most significant reduction in performance seen for static constellations, due to the lack of maneuverability from a narrow ground track. Reconfigurable constellation designs gain the most benefit from their reconfigurability under these circumstances, with the largest improvement in achievable coverage seen for extremely high elevation angle requirements, due to the low probability of being able to achieve such constrained coverage from a static orbit.

6.3 Delaying reconfiguration maneuvers

This section considers possible effects of delaying reconfiguration, and how satellite operators must assign relative weighting to the competing priorities of conserving propellant for future reconfigurations and achieving the fastest coverage of a location of interest. This section is primarily left as an area for future work, due to the need for additional design optimization runs to establish whether the preliminary results demonstrate a genuine effect or a model discretization error.

Tradeoffs must be made between time taken to reconfigure versus the amount of ΔV used; this is not a linear relation, but different maneuvering options are better suited to one objective or the other. Choosing between different reconfiguration op-

tions at different delay points offers the possibility of saving either time or fuel during the maneuver, by waiting to initiate reconfiguration at a different starting point. This also alters the availability of which drift orbits are feasible options at a given time.

The existing ReCon model carries out an optimization process to select from the possible options for responsive maneuvering to achieve coverage of a given target event. This means that the maneuvers selected at the time an event of interest is flagged for coverage should provide the best available option for reconfiguration. However, when delays of 1–24 hours were imposed before the constellation could initiate maneuvering, cases were found where savings in ΔV appear to be made. Further work is required to establish whether this is an accurate result or an artifact of the model structure, but preliminary results are presented here to inform future work in this area.

Figure 6-25 shows the variation in the number of satellites that are reconfigured in response to a single event of interest. The variable of design number shown on the x-axis corresponds to simple indices for the entire set of non-dominated designs generated by one optimization run, ordered by constellation cost. It may be observed from this figure that there is a large amount of variation in the number of reconfigured satellites between design numbers 55 and 80, suggesting some distinguishing factor in the architecture of these designs. Additional research is needed to determine if a specific design variable or family of architectures is correlated with this result, as initial analysis was unable to determine any particular factor causing this effect.

Figure 6-26 shows the variation in the total amount of ΔV used across the whole constellation in response to a single event of interest. This figure shows a much greater amount of variation throughout its range compared to Figure 6-25. This is explained by the slightly different drift orbits and maneuver starting points that result from the different delay lengths imposed upon beginning to reconfigure the constellation.

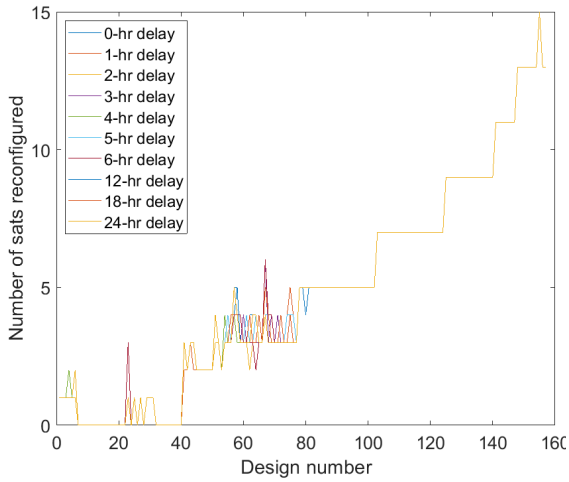


Figure 6-25: Variation of the number of satellites that are reconfigured in response to one event, for different constellation designs and delay lengths

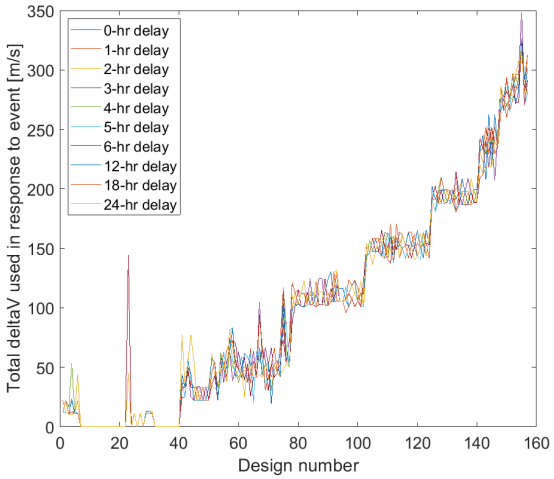


Figure 6-26: Variation of the total ΔV used across the whole constellation to respond to one event, for different constellation designs and delay lengths

Figure 6-27 shows the variation in the number of satellites that are reconfigured in response to a single event of interest for a second separate set of non-dominated designs, generated by a different ReCon optimization run. This figure shows a much larger amount of variation in the number of reconfigured satellites for almost the entire design space. This suggests specific design variables or architectures must indeed be linked to the amount of variation in this results, but again further analysis is needed to determine what these might be.

Figure 6-28 shows the variation in the total amount of ΔV used across the whole constellation in response to a single event of interest, for the second set of designs shown in Figure 6-27. This figure also shows a much greater amount of variation in the results compared to the first set of designs shown in Figure 6-26, which is explained by the total ΔV used being drastically different when a significantly different number of satellites are maneuvered.

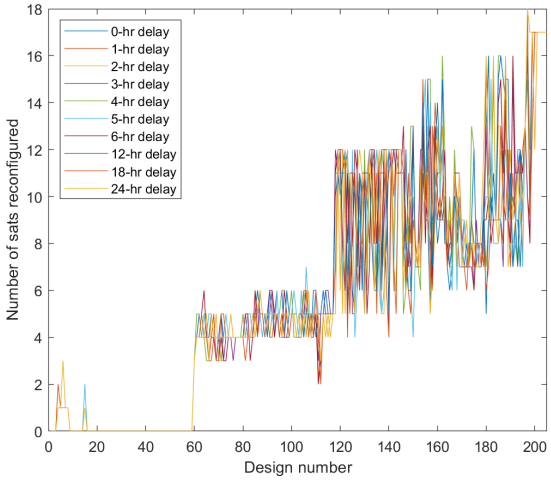


Figure 6-27: Variation of the number of satellites that are reconfigured in response to one event, for different constellation designs and delay lengths and a second set of designs

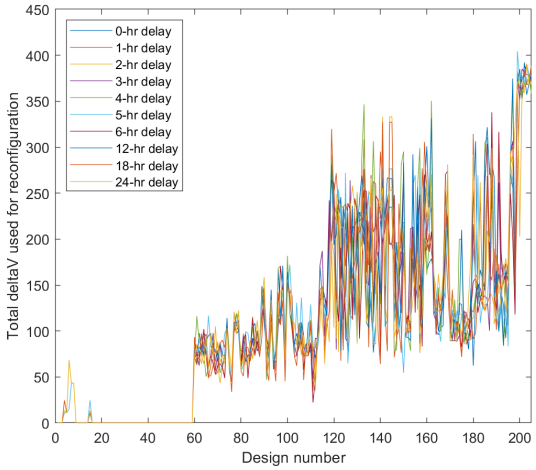


Figure 6-28: Variation of the total ΔV used across the whole constellation to respond to one event, for different constellation designs and delay lengths and a second set of designs

A magnified version of the total ΔV used by a small subset of designs from Figure 6-28 is shown in Figure 6-29, limiting the delay duration to a maximum of 6 hours. As designs that are very close in design number tend to also be very closely related in terms of design variables, this figure highlights the significant differences that are found between results for constellation designs that are extremely similar. For example, design 87 shows almost no variation in the amount of ΔV used for delay lengths between 0 (i.e. no delay at all) and 6 hours, while design 85 shows ΔV usage between 73 and 117 m/s depending on delay length.

Figure 6-30 shows the effect of different delay lengths on the total ΔV used by a small subset of 6 designs. This highlights how different the effect of delaying reconfiguration can be on different constellation designs. Designs 4 and 6 show much greater variation in the total ΔV used, varying between usage of 12–45 m/s and 11–54 m/s respectively depending on delay length. The other four designs show much less extensive variation, generally within ΔV usage of 11–23 m/s.

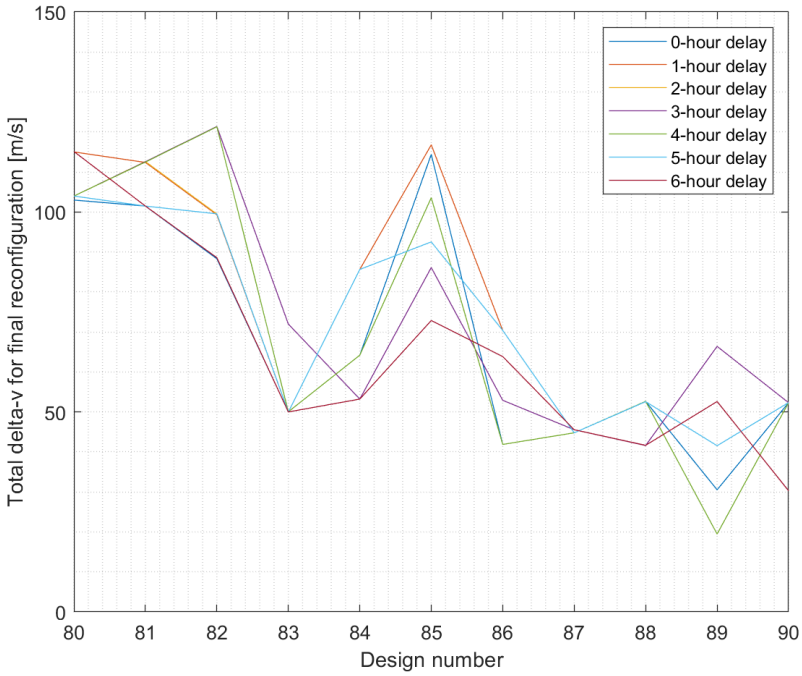


Figure 6-29: Zoomed-in view of total ΔV used for designs 80–90 shown in Figure 6-28

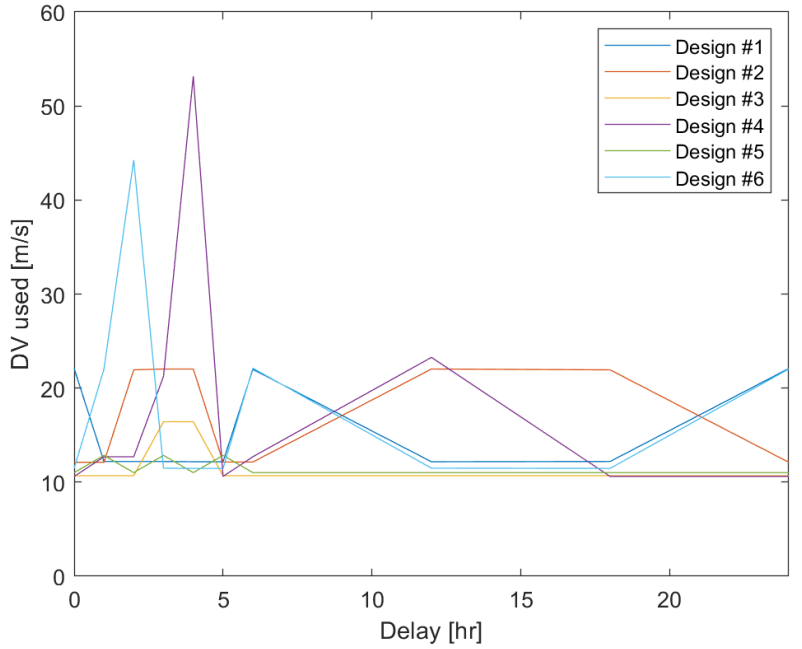


Figure 6-30: Impact of delay length of total ΔV used for six sample constellation designs

Preliminary results on the effect of delays to reconfiguration maneuvers are included here to present information for future researchers. Although the amount of variability in number of satellites maneuvered and total ΔV used could not be linked to specific design variables or families of constellation architectures, it is theorized that such a link exists and could usefully be identified within future work. This may lead to the development of useful future decision options, or modifications to the maneuver selection process for reconfiguration.

6.4 Conclusions

The effects of different constraints upon the ΔV budget or propulsion system mass fraction were considered. Limits of 150–1000 m/s were applied to the ΔV budget, and limits of 0.1–0.6 were applied to the propulsion system mass fraction. The combination of these two constraints was used to explore interactions between the two limits, and the resulting effect on performance and selected design variables.

Constraining propulsion system mass fraction to lower values was found to result in a reduction in the number of satellite planes and the ΔV budget, which is theorized to occur due to the propulsion system having less capacity for propellant within a more constrained mass budget. Reductions in the ΔV budget for reconfiguration from 1000 to 500 m/s were found to have less effect on system performance, but these began to affect constellation performance when constrained to the level of 250 m/s.

The ΔV budget for reconfiguration allocated for non-dominated constellation designs was found to generally lie within a maximum value of 350 m/s, significantly below the default maximum value of 1000 m/s. This could be explained by cost savings from selecting a smaller propulsion system for a constellation, or by launching a lower overall satellite mass due to reduced propellant mass. When constellations are evaluated against different 5-year operating scenarios, they are frequently found to retain some margin of ΔV in the reconfiguration budget at the end of the mis-

sion lifetime. This is theorized to be due to the maneuver selection process favoring a conservative approach to propellant usage, ensuring that it is balanced across the constellation and that the full mission duration will be achieved, without loss of reconfigurability due to premature usage of the entire ΔV budget. This maneuver selection is automated in the ReCon codebase due to the lack of an actual satellite operator to make decisions about the relative priority of rapid reconfiguration and conservative fuel use. Future satellite operators may take a more active role in managing maneuver prioritization based on target value, adopting more aggressive approaches to achieving coverage for a high-value target of interest or when ΔV reserves are high, and adjusting to a more cautious approach when ΔV reserves are depleted or achieving rapid coverage is not a high priority.

Any extra margin of ΔV may be used to gain other benefits, such as maneuvering between RGT orbits to gain improvements in performance. Higher performance scores were observed at specific latitudes and these were linked to the locations of crossover latitudes, where both ascending and descending satellite passes occur over the same ground location. Individual satellite crossover points are defined as the points at which a satellite crosses its own ground trace, and multi-satellite crossover points are defined as the locations where a satellite crosses the ground traces of other satellites in the constellation. Improvements of 10.2–15.1% in performance were observed for individual satellite crossovers, and improvements of 6.91–7.98% in performance were observed for multi-satellite crossovers.

These crossover points naturally shift in longitude if the constellation is not in a RGT orbit, as the ground tracks precess over time. They may also be shifted in longitude by using intermediate drift orbits for phasing, and this is the current approach to changing the coverage locations for ReCon. However, adjustments in latitude do not occur naturally and must be carried out deliberately, requiring significant ΔV usage to achieve. The amount of ΔV necessary to move between adjacent RGTs and adjust the latitude was calculated, varying between 83.3 m/s for the lowest RGT pair

and 91.4 m/s for the highest pair.

The upper limit assigned for the ΔV budget for reconfiguration is 1000 m/s for a five-year mission lifetime, meaning that carrying out such a maneuver would use a minimum of 7.8–9.5% of the ΔV_{recon} budget for a single transfer. This proportion would be even higher for many constellation designs with lower ΔV budgets. It was concluded that this maneuvering strategy is evidently not feasible for use in responding to every single target location of interest. However, maneuvering between RGTs presents a useful option for satellite operators to consider in cases where targets have an extraordinarily elevated value beyond the standard level of utility gained by achieving desired coverage for locations of interest.

The initial selection of RGT orbit for the constellation design was demonstrated to have a significant effect upon the extent to which crossover locations can be maneuvered to a desired location. The specificity of this type of maneuvering render it best-suited to modelling based on a finalized constellation architecture, rather than attempting to extrapolate the best decision options for every possible optimized design. The minimum elevation angle requirements that are assigned for usable ground coverage also affect the improvements in performance that may be achieved with this maneuvering strategy. The higher the constraint placed upon elevation angle, the larger the performance reduction seen for static constellations, due to the inability to move from a predetermined ground track to improve coverage of the desired area. The greatest benefits can be demonstrated for reconfigurability under these conditions, where the largest percentage improvements in coverage may be made relative to extremely constrained coverage angles. This is due to the low probability of static architectures achieving the desired coverage from an extremely narrow and fixed ground track.

Chapter 7

Conclusions and future work

7.1 Thesis summary

This work is motivated by the benefits that reconfigurable satellite constellations can offer in achieving persistent satellite coverage of unfolding events at unpredictable locations. Reconfigurability reduces the number of satellites needed to achieve a desired persistence level, by improving the utilization of each individual satellite when compared to a traditional static constellation architecture. Achieving comparable performance with fewer satellites reduces constellation costs, and presents an alternate option to the frequent proposal of achieving extensive global coverage by deploying mega-constellations in an increasingly-congested space environment.

Chapter 4 of this thesis presents an uncertainty-based design optimization approach for reconfigurable satellite constellations. Reliability-based design optimization is used to inform the development of a probabilistic performance metric, comparing ReCon design candidates on the probability of success in outperforming iso-cost static constellations over a range of operating scenarios, rather than endeavoring simply to achieve the highest mean performance at any cost point. The margin by which reconfigurable designs outperform iso-cost static costs is also quantified probabilistically, and both the mean and minimum ΔP cases are examined.

In Chapter 5, variance reduction techniques are applied to the ReCon code model to reduce the standard error of mean performance estimated by the simulation. This allows for a reduction in the number of scenarios that are used to optimize constellation designs while maintaining the existing benchmark for error in the results. This decrease in the number of code executions carried out for each functional evaluation of a candidate constellation design achieves runtime savings when simulations are carried out using the same maximum constellation size. The reduction in runtime will also enable the design optimization to be carried out for larger constellations or designs with increased complexity, due to the improvement in computational tractability additionally reducing runtimes for cases that may previously have been too computationally intensive to run.

Many of the research areas related to reconfigurable constellations (in both this and other work) are related to the design phase of such systems. Justifications for the creation and launch of reconfigurable satellite constellations cite the improved performance, the reduced cost, the ability to flexibly respond to a changing operational environment, or the sustainability gained when comparable performance may be achieved by launching fewer satellites and contributing less to space traffic management issues. However, if reconfigurable constellations are to escape the drawing board and make their way into widespread use alongside existing static constellations, future work must consider how such systems may actually be operated by decision-makers. Chapter 6 of this work considers possible effects of delaying reconfiguration, and how satellite operators must assign relative weighting to the competing priorities of conserving propellant for future reconfigurations and achieving the fastest coverage of a location of interest. The effects of different constraints placed upon the ΔV budget or propulsion system mass fraction are considered, and linked with the ability to conduct occasional large orbital adjustments at increased ΔV cost, in situations where such a cost is deemed worthwhile in order to gain significant improvements in coverage of especially high-value targets.

7.2 Thesis contributions

This thesis:

1. Develops a probabilistic figure of merit which evaluates the level of confidence that one performance distribution outperforms a second distribution (and by what margin of performance). This acknowledges the role of the statistical distribution of performance for a system designed to respond flexibly to a range of uncertain operating conditions, and avoids the reliability issues created by using the non-fixed metric of average performance as the fixed axis upon which to compare 'iso-performance' pairs of designs on a cost basis.

2. Applies this probabilistic figure of merit to reconfigurable constellation performance to evaluate the level of confidence that ReCon outperforms iso-cost static designs (and by what margin of performance) in a range of possible mission scenarios.

3. Finds that 74.2% of reconfigurable designs outperform iso-cost static designs with a confidence level of 90% or higher, and that all reconfigurable designs above a constellation cost of $223M surpass iso-cost static performance with a confidence level of 93% or higher. Finds that 19.4% of reconfigurable designs outperform iso-cost static designs by a margin of at least 50% higher performance, with a confidence level of 90% or higher.

4. Adapts variance reduction techniques for use with input distributions modelling uncertain parameters for the ReCon codebase, and prioritizes input modelling for variables to which the system output has greater sensitivity, resulting in a reduction in standard error in the model output that improves computational efficiency of the design optimization.

5. Demonstrates the feasibility of switching between adjacent RGT orbits to alter crossover latitude locations, and presents this as an available decision option for satellite operators to prioritize exceptional coverage over ΔV conservation for high-value targets.

7.3 Conclusions

7.3.1 Probabilistic metrics for system performance

This thesis presents a probabilistic approach to performance evaluation for reconfigurable satellite constellations, comparing this metric to past work by Robert Legge using the Value of Reconfigurability metric. Legge found that reconfigurable constellations cost 20–70% less than iso-performance static constellations evaluated against the same set of mission scenarios, but treated this mean performance as a point metric, rather than a statistical distribution of coverage.

Uncertainty-based multidisciplinary design optimization is reviewed, falling into two main approaches of robust design optimization and reliability-based design optimization. Robust design optimization seeks to increase design robustness by decreasing sensitivity to input variations, resulting in improved stability of performance under uncertain conditions. Reliability-based design optimization seeks to improve design reliability by reducing the probability of failure conditions; in this case, the failure condition is assigned as situations for which static designs outperform iso-cost reconfigurable designs, and the design evaluation is carried out with the objective of reducing the probability of this state as far as possible. Recognizing the statistical nature of performance achieved by ReCon against unknown and unpredictable operating scenarios requires the use of a probabilistic metric, which can quantify the likelihood of success; in this case, outperforming an iso-cost static constellation design.

Re-evaluating the performance score of optimized designs against alternate sets of target decks shows that the optimization model overpredicts future performance under uncertain conditions. This effect becomes more pronounced the smaller the optimization size is. If designs are optimized against only 2 possible target decks of scenarios, the performance is overestimated by approximately 4% for static designs and 6% for reconfigurable designs. Optimizing designs against 10 target decks gives an overestimate of approximately 2% for static designs and 3% for reconfigurable de-

signs, and optimizing against 20 target decks gives an overestimate of approximately 0.75% for static designs and 1.75% for reconfigurable designs. From this result, it is concluded that the minimum design optimization size should not fall below 20 target decks, to maintain the level of performance overestimation error within 1% for static designs and 2% for reconfigurable designs.

The VoR metric is an iso-performance metric, meaning that the cost scores of candidate designs are compared on an iso-performance basis. However, in the output from the ReCon model, performance scores vary from scenario to scenario, while cost scores stay constant once a design is fixed, as operational costs (that might vary) are not included in the cost modelling. Evaluating on a static axis results in a more reliable basis of comparison for paired designs, and so the decision is made to switch from an iso-performance to an iso-cost figure of merit for comparison. Even if operations costs are added to the model as a useful area of future work, resulting in variations in constellation cost between different mission scenarios, assessing designs on an iso-cost basis will still be a more reliable axis for comparisons as long as the maximum amount of cost variation remains within 25%.

ΔP is developed as a figure of merit that provides a directly equivalent alternative to Legge's Value of Reconfigurability as a single-axis comparison. Instead of describing the cost savings to be gained by adopting reconfigurability at the same performance level, this metric focuses on the performance gains to be made by adopting reconfigurability at the same constellation cost. By acknowledging variations in performance as part of the creation of this metric, ΔP allows for statistical comparisons to be drawn between the distributions of performance observed against a range of possible scenarios.

Chapter 4 shows 95% confidence intervals calculated for the mean and median performance values from a re-evaluation of optimized designs against 500 different scenarios, as well as the same confidence intervals for 25th and 75th percentiles of

the full distribution of performance scores. Almost complete overlap is observed in the performance distributions for designs in the low-cost low-performance region of the design space, up to a constellation cost of approximately $220M. There is a dwindling amount of overlap in the tails of these distributions up to a constellation cost of around $380M.

Confidence that reconfigurable designs outperform iso-cost static designs is calculated, comparing five different methods to assess the level of error introduced by different approaches, and determining that only the independent hypothesis testing method resulted in an unacceptable level of error in the results. The confidence that ReCon outperforms static is at least 97% for all iso-cost design pairs above a constellation cost of $223M, with the exception of one design with a confidence level of 93%. In addition, 4 designs in the low-cost region of $113M–124M outperformed static alternatives with a confidence level of 92.3% or higher. However, designs in the $116M–223M region performed less consistently, with a 50-50 split in this area as to whether reconfigurable or static designs performed best.

Evaluating the performance margins by which ReCon outperforms static, it is shown using mean ΔP that 85.5% of ReCon designs outperform static by any amount, 74.2% outperform static by a margin of at least 10% improvement in performance, and 64.5% outperform static by a mean margin of 25% or higher. The highest mean ΔP is observed in the $750M–800M region, where reconfigurable designs achieve a 95–100% improvement over iso-cost static constellations. The minimum ΔP observed for the worst-case scenario in this region shows a 45–60% improvement over the static alternative.

Evaluating the minimum ΔP worst-case scenarios across the design space, the minimum value is above zero in 61.3% of cases, meaning that performance still surpasses the static case for these scenarios. For the cost region of $500M–1250M, reconfigurable designs outperform static by at least a 20% margin with 95% confi-

dence in the worst-case value. For all designs above a cost of $300M, and the region between $223M–260M, minimum ΔP is above zero with 95% confidence, meaning that reconfigurable designs are still showing higher performance. However, the cost regions below $223M and between $260M–300M predict a negative ΔP for the worst-case scenario, meaning that static designs demonstrate superior performance in these cases.

The mean ΔP in the cost region between $223M–300M varies between 0–38%, meaning that on average the reconfigurable designs will meet or surpass the static performance. However, the minimum ΔP worst-case scenarios in this region vary between -18% and +18%, highlighting the need to consider the distribution of potential performances in planning for unknown future operating contexts. This region fails on the probabilistic metric of success, meaning that reconfigurable designs at these cost levels cannot outperform iso-cost static designs with a 95% level of confidence. However, the worst-performing designs in this region have confidence levels of 56–82% that static designs outperform ReCon, and these all fall within the cost region of $130M–214M, where mean ΔP is $\pm 5\%$ of static performance. Although reconfigurable designs cannot be shown to reliably outperform static designs in this cost region of the design space, when considered on average performance terms, they are at least shown to perform on par with static alternatives.

7.3.2 Uncertainty analysis and variance reduction

This work aims to optimize the designs of reconfigurable constellations to perform consistently well under uncertain operating conditions. A range of different operating scenarios must be simulated in order to design for uncertainty, and this is carried out using Monte Carlo based model propagation, resulting in a high number of code executions to evaluate candidate designs against many different mission scenarios. Combined with the complexity of the possible solution space, the computational resources needed to complete the optimization are intensive for constellations of any significant size.

Due to the significant sensitivity of the design to the latitude distribution of the input scenarios, and the lack of coupling of the longitude distribution to particular design effects, the decision was made to simplify the longitudinal distribution as far as possible and devote sampling resources to the latitudinal distribution. This allows for the investigation of two new methods of sampling for generating input target decks for use with the ReCon model. In Chapter 5, Legge's original sampling is found to result in a standard error of mean performance of 4.95% using 24 target decks, and the two new sampling methods are compared to this standard. It is found that using kernel function sampling allows for the number of target decks to be reduced to 23 while maintaining a standard error of 4.95%, and using proportionate definition allows for the number of target decks to be reduced to 22 while slightly reducing the standard error to 4.90%.

This reduction in the number of target decks reduces the number of code iterations by more than 8% for each optimization. Early Phase runtime data (gathered before upgrades were made to the MIT Supercloud in 2021) shows that such a decrease reduces predicted runtime by amounts between 7.20–23.4%, depending on maximum constellation size. More recent Triples Mode runtime data for the ReCon code on the MIT Supercloud results in a predicted runtime decrease of 4.4%. In addition to directly saving computational time for optimization runs of the sizes that may already be carried out, these runtime savings may be used to enable the investigation of larger or more complex design optimizations.

7.3.3 Operational decision options

Although past work on reconfigurable constellations has explored aspects of designing constellations and satellites to perform well under unpredictable circumstances, less research has considered operational decisions during the lifetime of such a constellation, aside from the simulation of reconfiguration maneuvers.

The effects of different constraints upon the ΔV budget or propulsion system mass fraction are considered in Chapter 6, showing interactions between the two and their effect on design variables. Reducing propulsion system mass fraction is found to result in a reduction in the number of satellite planes and the ΔV budget, which is theorized to be due to the propulsion system having less capacity for propellant within its allocated mass budget. Reductions in the ΔV budget for reconfiguration from 1000 to 500 m/s are found to have less effect, but start to affect constellation performance when constrained to the level of 250 m/s.

Non-dominated constellation designs are found to generally allocate a maximum of 350 m/s for the ΔV budget for reconfiguration: considerably lower than the default constraint of 1000 m/s. This is theorized to be due to cost savings from a smaller propulsion system or from launching a lower mass of propellant. In addition, when constellations are evaluated against different operating scenarios, they are often found to have remaining propellant in their reconfiguration budget at the end of the mission lifetime. This is explained by the optimization process used to select maneuvers, which favors a cautious approach to propellant budgeting to ensure that the constellation does not fail prematurely due to a sudden loss of maneuverability. Satellite operators could take a more active role in managing constellation maneuvers, electing to adopt more aggressive maneuvering strategies to achieve faster coverage when fuel reserves are high and adjusting to a more conservative approach when reserves are lower or targets are considered to be of lower priority.

Other benefits could be gained by making use of surplus ΔV, such as maneuvering between adjacent RGT orbits. Improved performance is observed at specific latitudes, and after evaluation of the parameters of the constellation RGT, these are linked to the locations of crossover latitudes. Individual satellite crossover points are defined as the points at which a satellite crosses its own ground trace, and multi-satellite crossover points are defined as the locations where a satellite crosses the ground traces of other satellites in the constellation. The increased performance observed at these locations

is explained by the presence of both ascending and descending satellite passes over the same point. Improvements of 10.2–15.1% in performance are observed for individual satellite crossovers, and improvements of 6.91–7.98% in performance are observed for multi-satellite crossovers.

These crossover points may easily be shifted in longitude by using drift orbits, or by simply waiting if the constellation is not in a RGT orbit and ground tracks are naturally precessing. However, adjustments in latitude do not occur naturally and must be carried out deliberately. The amount of ΔV necessary to move between adjacent RGTs and adjust the latitude is calculated, varying between 83.3 m/s for the lowest RGT pair and 91.4 m/s for the highest pair.

As the upper limit assigned for the ΔV budget for reconfiguration is 1000 m/s for a five-year mission lifetime, such a maneuver would use a minimum of 7.8–9.5% of the total for a single transfer, and a much higher proportion for many constellations with lower ΔV budgets. It is clear that such a maneuvering strategy is not feasible for use in responding to every target location of interest. However, such a maneuver may present a useful option for satellite operators to consider in cases where targets have an extremely elevated value beyond that assigned to the standard level of desired coverage for locations of interest.

The extent to which crossover locations can be maneuvered to a desired location is shown to depend strongly on the initial choice of RGT orbit for the constellation and the minimum elevation angle requirements for useful ground coverage. The specificity of this strategy means that it is best-suited for modelling based on a finalized constellation architecture. Imposing more stringent elevation angle constraints on coverage decreases the usable ground footprint of the satellite, resulting in a larger performance reduction for static constellations compared to ReCon, due to their lack of maneuverability from a pre-defined ground track. The most benefit can be gained by the use of reconfigurability in these circumstances, with the largest improvements

in coverage availability seen relative to extremely high elevation angle requirements, due to the low probability of achieving such coverage from a static orbit with an extremely narrow and fixed ground track.

7.4 Future work

After developing a probabilistic performance metric to evaluate reconfigurable designs against iso-cost static designs, an interesting future application of this metric would be to incorporate it into the design optimization process. This would allow for reconfigurable designs to be optimized with the objective of achieving the highest confidence in outperforming static options at the same total constellation cost, rather than simply optimizing designs with the goal of achieving the highest mean performance at each cost point. Using a probabilistic figure of merit means that designs can be optimized to give the most reliable performance, selecting design variables that achieve the most consistent probability of mission chance, rather than relying on an averaged metric that can be skewed by outlying values and which gives very little information about the nature of the statistical distribution of performance.

The process of implementing a probabilistic metric as part of the design optimization would add complexity to the codebase, due to the fact that the ΔP metric is calculated from a comparison of reconfigurable and static design performance at the same cost point. One suggested approach for this area would be to conduct an initial optimization run to generate a set of non-dominated static constellation designs, and then use these as part of the input for a second optimization run used to generate a set of reconfigurable constellation designs. The design optimization objective for this second run would be targeted to select those design candidates which show the greatest percentage improvement over the static performance score at each cost point. Target decks would also need to be selected in a replicable fashion rather than chosen randomly for each set of optimizations, in order to ensure that ΔP is being calculated as the difference in performance achieved against the same scenario by the paired re-

configurable and static designs.

An interesting expansion for future research would be to expand the uncertainty in the target deck scenarios to the temporal distribution of events. In the present version of these five-year mission scenarios, all events are generated with a minimum separation in the time of occurrence, which is used to avoid overlapping or simultaneous events of interest. Generating temporal overlaps between the end of one event of interest and the beginning of another or creating entirely simultaneous events in different locations would introduce an entire new range of research questions as to how the constellation resources should be divided up for allocation between multiple targets. This could provide a topic for future work on generating useful decision options and questions of resource prioritization to be offered for consideration by the satellite operators of a reconfigurable constellation.

Following on from Straub's work in evaluating the use of slewing to achieve broader coverage across a region that lies within a reconfigurable constellation's field of regard,[48] an additional area of expansion for uncertainty in the regions of interest would be to incorporate maneuvering options to achieve coverage of larger regions. This could include areas of interest even larger than an individual satellite's field of regard, meaning that even when slewing is used to expand the range of achievable coverage, no single satellite is able to cover the entire desired coverage area. This would require a considerable expansion of the tradespace of potential maneuvering options, and could result in the development of entirely new approaches to achieving the desired coverage.

Straub also considered the use of drifting ground tracks over time to follow mobile events (i.e. not fixed in one location over time), and this work was continued by Morgan, who considered the possibility of aligning the rate of a drifting orbit with the speed of a mobile event of interest, applying this specifically to hurricane tracks.[47] Paek developed variants upon sun-synchronous RGTs with ground tracks that drift

at a predetermined rate, 'multi-sun-synchronous' orbits that provide identical solar angles at multi-day intervals (i.e. the sun-synchronous equivalent of a $n/2$ or $n/3$ RGT), and a combined multi-sun-synchronous orbit with drifting ground tracks.[26] Implementing such orbits as part of the ReCon model would broaden the range of applications for which the benefits of reconfigurability may be demonstrated and usefully applied.

An alternate expansion for generating target areas of interest could be to consider a requirement to maneuver to specific orbital locations, rather than maneuvering to achieve coverage over a ground location. This could be used to optimize reconfiguration options for reconstitution of a damaged satellite constellation. This might be an especially complex decision space if a small number of reconfigurable satellites were used as support for a much larger mega-constellation, temporarily filling gaps in each orbital plane where units have been lost during launch or due to later failures. Once additional static satellites can be launched to permanently replace these losses, the support satellites could then return to a neutral mode awaiting the next need for reconstitution in a different location. This would provide an application in which reconfiguration may be used to support a larger static constellation, rather than competing to fulfil the same objectives.

Another useful area for future work would be to propagate uncertainty in the cost models used in the ReCon codebase, and to update the unit of the cost estimates from FY2010 US dollars to the present day, taking into account both inflation and changing prices for launch options. Straub explored cost and performance trades between chemical and electric propulsion for reconfigurable constellations, and investigated updates to available launch providers as of 2020 for the ReCon model.[48] Launch vehicle options will continue to evolve over time due to commercial pressures and the availability of different vehicles over time, and so this area of the model requires consistent updates to be made to reflect real-world options.

Adding the simulation of operating costs would provide a useful update to the ReCon model, enabling the determination of whether these remain relatively consistent or show large variation between different scenarios. If operations costs are significantly higher for reconfigurable constellations due to the increased need for operator decisions during the mission lifetime compared to static constellations, this may reduce the calculated benefits of reconfigurability across the design space. Modelling operational costs would reduce uncertainty in the total system cost and allow for statistical comparisons to be made for the distributions of cost and performance scores against different unpredictable scenarios. This would allow for more thorough evaluation of the relative benefits of scoring constellations on iso-performance and iso-cost metrics. Comparing the relative value supplied by reconfigurable and static designs on an iso-cost basis is still expected to supply greater reliability and lower variance than comparisons on an iso-performance basis, as long as the constellation cost variation between scenarios is found to be less than 25% of the existing cost estimates.

Applying variance reduction techniques to the ReCon model achieved a reduction in standard error of mean performance, which is the current metric used for design optimization. If an alternative probabilistic metric is applied to the optimization in future work, the standard error of this metric should also be evaluated in order to assess whether further improvements in efficiency can be made. This reduced error was used to enable reductions in runtime for the ReCon code when optimizing constellation designs of comparable size to past work. Runtime savings for the ReCon model were achieved by a combination of variance reduction techniques and resource allocation parameter tuning, following upgrades to the MIT Supercloud. These efficiency improvements and runtime savings may be used to enable the future investigation of larger constellation sizes, or increased complexity in potential designs.

An area of interest for future work that could be enabled by these efficiency improvements is that of overlaps in the temporal distribution of events of interest. Investigations could be carried out into how satellite operators might react to simul-

taneous events of interest, or to overlaps between the end of the period of interest in one location and the start of interest in another location. This would involve simulating additional decision options for operators, adding to the body of work on how to actually operate a reconfigurable constellation. Modelling such temporal overlaps would better reflect the complexity of real-world decision-making, improving the authenticity of the ReCon model.

The final section of Chapter 6 considers possible effects of delaying reconfiguration, and how satellite operators must assign relative weighting to the competing priorities of conserving propellant for future reconfigurations and achieving the fastest coverage of a location of interest. This section is primarily left as an area for future work, due to the need for additional design optimization runs to establish whether the preliminary results demonstrate a genuine effect or a model discretization error.

Appendix A

MIT Supercloud code tuning

In 2021, upgrades were carried out to the MIT Supercloud to introduce a new 'Triples Mode' to the HPC cluster. Triples Mode is named for the three resource parameters that must be specified as part of job submission when using this mode: the number of nodes on the cluster that are used to run the code, the number of processes that should run per node, and the number of threads that should run per process. This supplies a mode of submitting pMATLAB jobs to the Supercloud cluster with active resource allocation management by the user, resulting in flexible control over the memory and threads assigned to the task. Use of this mode improves performance when compared to the use of default resource settings that may not be optimized for a specific simulation of interest.

Triples Mode schedules whole nodes exclusively for individual submitted tasks. This has the benefit of ensuring that no other users are sharing the node, as this sharing of resources can impact code performance. When multiple users submit memory-intensive jobs that are assigned to the same compute node, this has a particularly detrimental effect on the runtime of these competing tasks.

The benefits of Triples Mode arise from its improved efficiency due to calibrating resource requests to the specific application. This requires an initial time investment in carrying out a tuning process to determine the optimal settings for the three job

submission parameters. However, the Supercloud website cites expected improvements of "a $\sim 2\times$ increase [in] efficiency for many users", and highlights that once the best settings have been found for a specific codebase, these parameter values may be reused as long as significant changes have not been introduced within the code.[124]

A.1 Tuning process

The first step of the tuning process is to develop a metric for the rate at which the codebase is running. Establishing a rate for work done over time allows for comparisons to be drawn for the speed of different settings. For the ReCon model, the first set of output is saved to a .mat MATLAB data file when 100 functional evaluations have been carried out. The amount of time taken to complete and save this initial set of 100 functional evaluations was used to supply a runtime metric for comparing the Triples Mode parameters being tuned.

The ReCon model is memory-intensive, falling close to the limits of the amount of RAM allocated per core on the Supercloud. This results in the submitted HPC jobs being 'memory bound', meaning that the limiting factor for the amount of time taken to complete the task is the amount of memory needed to hold the working data. The overall runtime is dominated by the amount of available memory and the speed of access to it. The opposite of this is a job that is 'compute bound', where the number of steps in the computation or the speed of the central processor is the determining factor that affects the overall runtime.

For memory bound tasks, the recommendation is to tune the number of processes per node first. The three parameters are included as an input in the job submission function in a triplet form written as [*NN NPPN NTPP*], where *NN* is the number of nodes, *NPPN* is the number of processes per node, and *NTPP* is the number of threads per process. The recommended values used for *NPPN* and *NTPP* with the Xeon-G6 nodes on the Supercloud are 1, 2, 4, 8, 16, 20, 32 and 40, due to the fact

that each one of these nodes is made up of 40 cores. The maximum user allocation of Xeon-G6 nodes per user is 8, so the recommended values for *NN* are 1, 2, 4 and 8.

Figure A-1 shows the tuning process carried out for the number of processes per node. The submitted job was set to use one node (*NN*=1) and one thread per process (*NTPP*=1), while *NPPN* was increased from 1 to 40. This means that for this step of the tuning, the triplet settings were entered in the form of [1 *NPPN* 1], with a gradually increasing value of *NPPN*. *NPPN* values of 1 and 2 did not generate any output, and the ReCon codebase stalled without producing any results at this low level of resource allocation. This was theorized to be due to the leader-worker setup used for the optimization, with all of the small amount of resources available in this job submission devoted to allocating tasks with no workers available to actually complete these tasks.

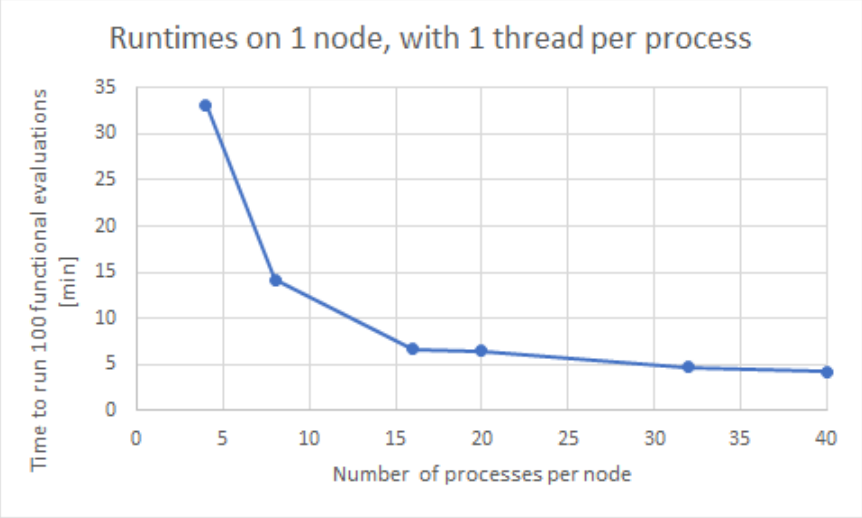


Figure A-1: Effect of the assigned number of processes per node on length of runtime to achieve 100 functional evaluations when using 1 node and 1 thread per process

Significant improvements in runtime to carry out 100 functional evaluations may be observed for the increase in *NPPN* from 4 to 16, dropping from 33.1 min for 4 processes per node to 6.7 min for 16 processes per node. However, there is a diminishing rate of returns for increases in *NPPN* above this point, with runtime only dropping

to 4.3 min when 40 processes per node are assigned.

The second step of tuning is to freeze *NPPN* at the optimal value found from the first step, and compare runtime results while varying the value of *NTPP*. As 40 processes per node gave the lowest runtime, this value was assigned (*NPPN*=40), and the triplet settings were entered as [1 40 *NTPP*] while the number of threads per process was gradually increased.

The third step of tuning is to compare performance for different numbers of nodes (*NN*). However, the ability to vary the number of nodes was somewhat limited by the availability of resources on the MIT Supercloud at any given time, due to high user demand. Due to the need to carry out tuning runs whenever the appropriate number of nodes was available, this stage was conducted somewhat in parallel with the second stage.

Figure A-2 shows combined results for the tuning of *NTPP* and *NN*, highlighting that the optimal setting for *NTPP* actually varies depending on the number of nodes assigned to the job. When using 1 or 2 nodes and 40 processes per node, the optimal number of threads per process is found to be 1. This results in a runtime of 4.27 min using 1 node, and 3.04 min using 2 nodes. When using 4 nodes, the optimal number of threads per process is 2, resulting in a runtime of 2.91 min. When using 8 nodes, the optimal number of threads per process is 20, resulting in a runtime of 2.81 min.

Due to the amount of variation in runtime results and optimal values found for *NTPP* depending on the number of nodes used, tuning was repeated for *NPPN*, using fixed values of 8 nodes and 20 threads per process to assess whether 40 processes per node was still found to result in the lowest runtime. Figure A-3 shows the runtime results for this round of tuning. Output was obtained for *NPPN* values between 2 and 40, once again showing that *NPPN*=40 resulted in the lowest runtime, taking 2.81 min to complete 100 functional evaluations.

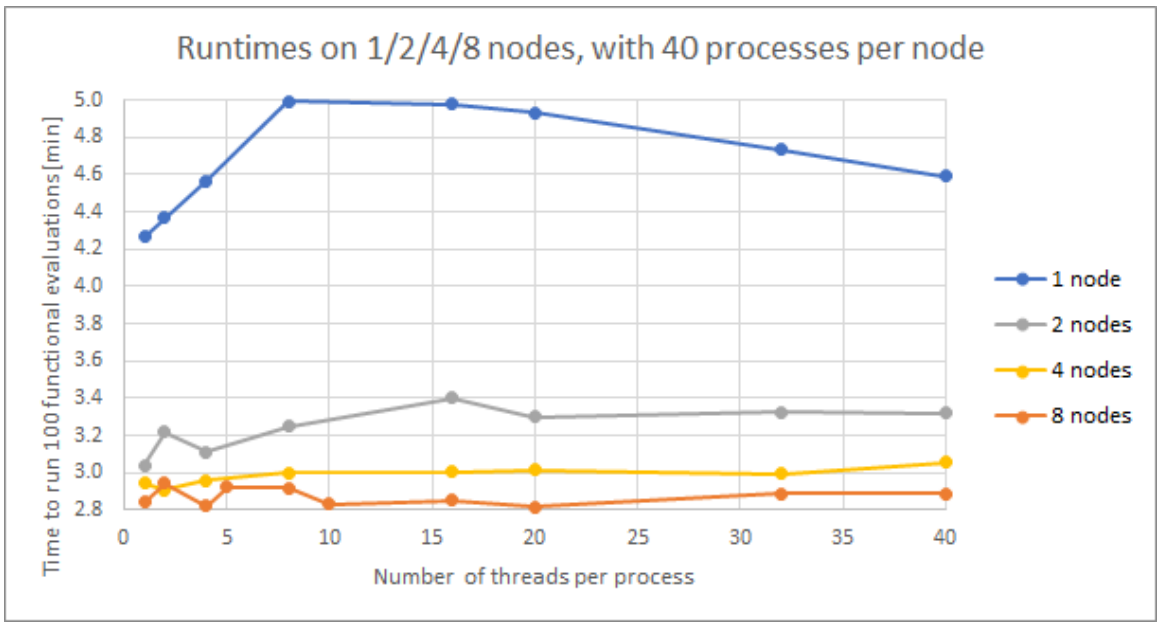


Figure A-2: Effect of number of threads per process on runtime, shown for four different number of nodes and 40 processes per node

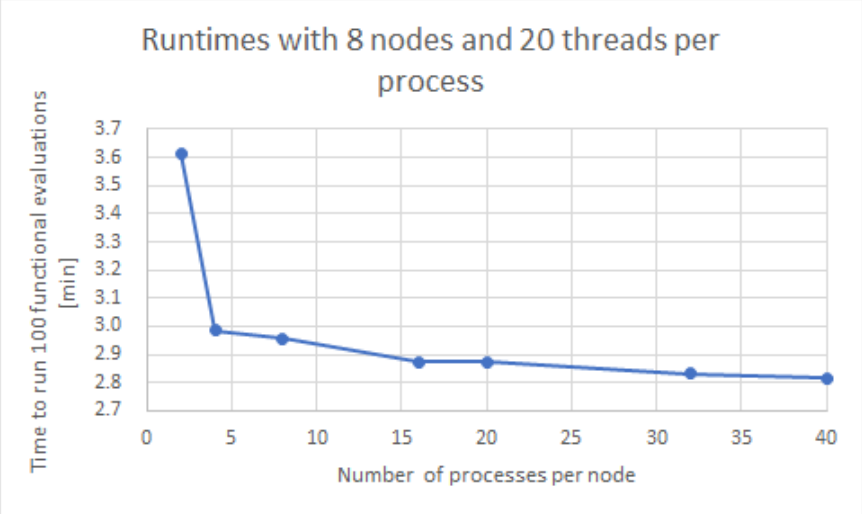


Figure A-3: Effect of number of processes per node on runtime, when using 8 nodes and 20 threads per process

Due to the results in Figure A-2 showing that 1 thread per process gave the lowest runtime for code runs using 1 or 2 nodes, a separate tuning check was carried out to compare runtimes for 40 processes per node and 1 thread per process against different numbers of nodes. The runtimes obtained when 1, 2, 4 and 8 nodes are used

to carry out the submitted job are shown in Figure A-4. The greatest improvement in runtime is observed when increasing *NN* from 1 to 2, with a drop from 4.27 min to 3.04 min. Increasing the number of nodes to 4 or 8 shows considerably diminished improvements, with runtime decreasing only to 2.95 min and 2.87 min respectively.

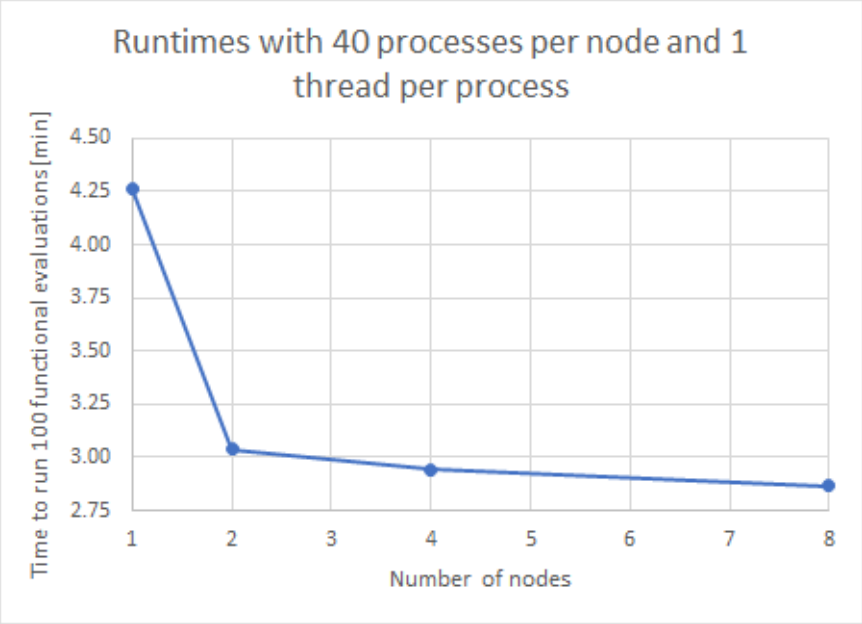


Figure A-4: Effect of number of nodes on runtime, when using 40 processes per node and 1 thread per process

A.2 Tuning conclusions

A series of 51 code tuning runs of the ReCon codebase were carried out on the MIT Supercloud in 2021. 48 of these runs generated output, consisting of the data from 100 functional evaluations and the length of time taken to carry out these evaluations and save the output to a MATLAB data file. The remaining 3 runs used extremely constrained resource allocation parameters and failed to complete 100 functional evaluations or generate and save any output.

The Triples Mode tuning process showed that the lowest runtimes were consis-

tently obtained using 40 processes per node. For code runs using 1 or 2 nodes and 40 processes per node, the lowest runtimes were obtained using 1 thread per process. For code runs using 4 nodes and 40 processes per node, the lowest runtime was obtained using 2 threads per process. For code runs using 8 nodes and 40 processes per node, the lowest runtime was obtained using 20 threads per process. However, for runs using 4 or 8 nodes with 40 processes per node, all runtimes fell within a narrow band (2.91–3.06 min for 4 nodes and 2.81–2.94 min for 8 nodes).

There was found to be a lack of consistent availability of nodes on the MIT Supercloud due to high user demand, and so most runs of the ReCon code were carried out using 1–2 nodes depending on resource availability. For this number of nodes, the settings of 40 processes per node and 1 thread per process were used to run design optimizations.

Using the tuned Triples Mode job submission parameters, there was a mean runtime reduction of 31.1% for the ReCon codebase when compared to previous design optimization runs carried out on the MIT Supercloud. The highest runtime improvement recorded was 46.1% for optimization runs carried out using 24 target decks, with slightly lower efficiency improvements recorded for optimization runs using fewer target decks. This was theorized to be due to the tuning process being carried out on the HPC cluster specifically using 24 target decks. After the sampling improvements and reduction to 22 target decks described in Chapter 5 were implemented, future work could reverify the Triples Mode tuning process described here to assess whether these settings are still the optimal values for use with the updated ReCon model.

Bibliography

[1] Kessler, Donald J. and Cour-Palais, Burton G. "Collision frequency of artificial satellites: The creation of a debris belt", Journal of Geophysical Research: Space Physics 83.A6 (1978): 2637-2646.

[2] Johnson, Nicholas L. "Orbital debris: the growing threat to space operations", NASA Johnson Space Center, 2010.

[3] Sweeting, Martin N. "Modern small satellites-changing the economics of space", Proceedings of the IEEE 106.3 (2018): 343-361.

[4] Foreman, Veronica L., Afreen Siddiqi, and Olivier De Weck. "Large satellite constellation orbital debris impacts: Case studies of OneWeb and SpaceX proposals." AIAA SPACE and Astronautics Forum and Exposition. 2017.

[5] Radtke, Jonas, Christopher Kebschull, and Enrico Stoll. "Interactions of the space debris environment with mega constellations—Using the example of the OneWeb constellation." Acta Astronautica 131 (2017): 55-68.

[6] De Neufville, Richard, and Stefan Scholtes. "Flexibility in engineering design", MIT Press, 2011.

[7] Savage, Sam L., and Harry M. Markowitz. "The flaw of averages: Why we underestimate risk in the face of uncertainty", John Wiley & Sons, 2009.

[8] Legge Jr, Robert S. "Optimization and valuation of reconfigurable satellite constellations under uncertainty", PhD thesis, Massachusetts Institute of Technology, 2014.

[9] Schürer, Rudolf. "A comparison between (quasi-) Monte Carlo and cubature rule based methods for solving high-dimensional integration problems", Mathematics and Computers in Simulation 62.3-6 (2003): 509-517.

[10] McKay, Michael D., Richard J. Beckman, and William J. Conover. "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code", Technometrics 21.2 (1979): 239-245.

[11] McGrath, Ciara. "Analytical methods for satellite constellation reconfiguration and reconnaissance using low-thrust manoeuvres", PhD thesis, University of Strathclyde, 2018.

[12] Guier, William H., and George C. Weiffenbach. "Genesis of satellite navigation", Johns Hopkins APL Technical Digest 18.2 (1997): 179.

[13] Fortescue, Peter, Graham Swinerd, and John Stark, eds. "Spacecraft Systems Engineering", John Wiley & Sons, 2011.

[14] Walker, John Gerard. "Circular orbit patterns providing continuous whole earth coverage", No. RAE-TR-70211. Royal Aircraft Establishment Farnborough (United Kingdom), 1970.

[15] Walker, John G. "Continuous whole-earth coverage by circular-orbit satellite patterns", No. RAE-TR-77044. Royal Aircraft Establishment Farnborough (United Kingdom), 1977.

[16] Walker, John G. "Satellite constellations", Journal of the British Interplanetary Society 37 (1984): 559-572.

[17] Lansard, Erick, Eric Frayssinhes, and Jean-Luc Palmade. "Global design of satellite constellations: a multi-criteria performance comparison of classical walker patterns and new design patterns", Acta Astronautica 42.9 (1998): 555-564.

[18] Ruggieri, Marina, et al. "The Flower Constellation Set and its Possible Applications", ACT Final report, Aridana 5 (2006): 4108.

[19] Mortari, Daniele, Matthew P. Wilkins, and Christian Bruccoleri. "The flower constellations", Journal of Astronautical Sciences 52.1 (2004): 107-127.

[20] Wilkins, Matthew Paul, Christian Bruccoleri, and Daniele Mortari. "Constellation design using flower constellations", Paper AAS (2004): 04-208.

[21] Mortari, Daniele, Mauro De Sanctis, and Marco Lucente. "Design of flower constellations for telecommunication services", Proceedings of the IEEE 99.11 (2011): 2008-2019.

[22] Avendaño, Martín E., Jeremy J. Davis, and Daniele Mortari. "The 2-D lattice theory of flower constellations", Celestial Mechanics and Dynamical Astronomy 116.4 (2013): 325-337.

[23] Ortore, Emiliano, Marco Cinelli, and Christian Circi. "A ground track-based approach to design satellite constellations", Aerospace Science and Technology 69 (2017): 458-464.

[24] Shtark, Tomer, and Pini Gurfil. "Low Earth orbit satellite constellation for regional positioning with prolonged coverage durations", Advances in Space Research 63.8 (2019): 2469-2494.

[25] Luo, Xin, et al. "Constellation design for earth observation based on the characteristics of the satellite ground track", Advances in Space Research 59.7 (2017): 1740-1750.

[26] Paek, Sung Wook, et al. "Sun-synchronous repeat ground tracks and other useful orbits for future space missions." The Aeronautical Journal 124.1276 (2020): 917-939.

[27] Chaize, Mathieu. "Enhancing the economics of satellite constellations via staged deployment and orbital reconfiguration", SM thesis. Massachusetts Institute of Technology, 2003.

[28] Scialom, Uriel. "Optimization of satellite constellation reconfiguration", SM thesis. Massachusetts Institute of Technology, 2003.

[29] Lee, Hang Woon, et al. "Optimization of satellite constellation deployment strategy considering uncertain areas of interest", Acta Astronautica 153 (2018): 213-228.

[30] Appel, Leonid, Moshe Guelman, and David Mishne. "Optimization of satellite constellation reconfiguration maneuvers", Acta Astronautica 99 (2014): 166-174.

[31] Siddiqi, Afreen, Jason Mellein, and Olivier de Weck. "Optimal reconfigurations for increasing capacity of communication satellite constellations", 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. 2005.

[32] De Weck, Olivier L., Uriel Scialom, and Afreen Siddiqi. "Optimal reconfiguration of satellite constellations with the auction algorithm", Acta Astronautica 62.2-3 (2008): 112-130.

[33] Davis, Jeremy John. "Constellation reconfiguration: Tools and analysis", PhD thesis, Texas A&M University, 2010.

[34] Ferringer, Matthew, et al. "Pareto-hypervolumes for the Reconfiguration of Satellite Constellations", AIAA/AAS Astrodynamics Specialist Conference and Exhibit. 2008.

[35] Ferringer, Matthew P., David B. Spencer, and Patrick Reed. "Many-objective reconfiguration of operational satellite constellations with the large-cluster epsilon non-dominated sorting genetic algorithm-II", 2009 IEEE Congress on Evolutionary Computation. IEEE, 2009.

[36] Ferringer, Matthew Phillip. "General framework for the reconfiguration of satellite constellations", PhD thesis, Pennsylvania State University (2008).

[37] Paek, Sung Wook. "Reconfigurable satellite constellations for geo-spatially adaptive Earth observation missions", PhD thesis, Massachusetts Institute of Technology, 2012.

[38] Co, Thomas C., Constantinos Zagaris, and Jonathan T. Black. "Responsive satellites through ground track manipulation using existing technology", Journal of Spacecraft and Rockets 50.1 (2013): 206-216.

[39] Ingraham, Steven P. "Dynamic Constellation Tasking and Management", MS thesis, Air Force Institute of Technology (2013).

[40] Co, Thomas C., and Jonathan T. Black. "Responsiveness in low orbits using electric propulsion", Journal of Spacecraft and Rockets 51.3 (2014): 938-945.

[41] Co, Thomas C. "Operationally Responsive Spacecraft Using Electric Propulsion", PhD thesis, Air Force Institute of Technology (2012).

[42] McGrath, Ciara, and Malcolm Macdonald. "Design of a reconfigurable satellite constellation." 66th International Astronautical Congress, IAC 2015.

[43] McGrath, Ciara N., and Malcolm Macdonald. "Analytical low-thrust satellite maneuvers for rapid ground target revisit." AIAA SPACE 2016, p5294.

[44] McGrath, Ciara, et al. "Applications of responsive small satellites with MIT TILE electrospray propulsion", 15th Reinventing Space Conference. 2017.

[45] McGrath, Ciara N., and Malcolm Macdonald. "General perturbation method for satellite constellation reconfiguration using low-thrust maneuvers", Journal of Guidance, Control, and Dynamics 42.8 (2019): 1676-1692.

[46] Morgan, Sarah J., Ciara McGrath, and Olivier L. De Weck. "Mobile target tracking using a reconfigurable low earth orbit constellation." ASCEND 2020, p4247.

[47] Morgan, Sarah J. "Reconfigurable satellite constellations for mobile target tracking", SM thesis, Massachusetts Institute of Technology, 2021.

[48] Straub, Alexandra N. "Expanded tradespace analysis and operational considerations for reconfigurable satellite constellations", SM thesis, Massachusetts Institute of Technology, 2020.

[49] Gentgen, Chloé. "Hybrid chemical-electric propulsion systems for CubeSats", SM thesis, Massachusetts Institute of Technology, 2022.

[50] Paek, S. W. "Concurrent design optimization of Earth observation satellites and reconfigurable constellations", Journal of the British Interplanetary Society 70 (2017): 19-35.

[51] Paek, Sung Wook, Sangtae Kim, and Olivier de Weck. "Optimization of Reconfigurable Satellite Constellations Using Simulated Annealing and Genetic Algorithm", Sensors 19.4 (2019): 765.

[52] Ross, Adam M., et al. "Multi-attribute tradespace exploration as front end for effective space system design", Journal of Spacecraft and Rockets 41.1 (2004): 20-28.

[53] La Tour, Paul Alexis. "Combining tradespace exploration with system dynamics to explore future space architectures", PhD thesis, Massachusetts Institute of Technology, 2016.

[54] Uebelhart, Scott Alan. "Non-deterministic design and analysis of parameterized optical structures during conceptual design", PhD thesis, Massachusetts Institute of Technology, 2006.

[55] Jilla, Cyrus D. "A Multiobjective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems", PhD thesis, Massachusetts Institute of Technology, 2002.

[56] Jilla, Cyrus D., and David W. Miller. "Multi-objective, multidisciplinary design optimization methodology for distributed satellite systems", Journal of Spacecraft and Rockets 41.1 (2004): 39-50.

[57] Paek, Sung Wook, et al. "Satellite constellation design algorithm for remote sensing of diurnal cycles phenomena", Advances in Space Research 62.9 (2018): 2529-2550.

[58] Ferringer, Matthew P., and David B. Spencer. "Satellite constellation design tradeoffs using multiple-objective evolutionary computation", Journal of Spacecraft and Rockets 43.6 (2006): 1404-1411.

[59] Ferringer, Matthew P., Ronald S. Clifton, and Timothy G. Thompson. "Efficient and accurate evolutionary multi-objective optimization paradigms for satellite constellation design", Journal of Spacecraft and Rockets 44.3 (2007): 682-691.

[60] Reed, Patrick M., et al. "Parallel evolutionary multi-objective optimization on large, heterogeneous clusters: an applications perspective", Journal of Aerospace Computing, Information, and Communication 5.11 (2008): 460-478.

[61] Shah, Nirav B., et al. "Quantifying flexibility for architecting changeable systems", 6th Conference on Systems Engineering Research, Los Angeles, CA. 2008.

[62] Cardin, Michel-Alexandre, and Richard De Neufville. "A survey of state-of-the-art methodologies and a framework for identifying and valuing flexible design opportunities in engineering systems", Massachusetts Institute of Technology, Cambridge (2008).

[63] Cardin, Michel-Alexandre, et al. "Empirical evaluation of procedures to generate flexibility in engineering systems and improve lifecycle performance", Research in Engineering Design 24.3 (2013): 277-295.

[64] Ross, Adam M., Donna H. Rhodes, and Daniel E. Hastings. "Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value", Systems Engineering 11.3 (2008): 246-262.

[65] Ross, Adam M., Donna H. Rhodes, and Daniel E. Hastings. "Defining System Changeability: Reconciling Flexibility, Adaptability, Scalability, and Robustness for Maintaining System Lifecycle Value", INCOSE International Symposium. Vol. 17. No. 1. 2007.

[66] Viscito, Lauren, and Adam Ross. "Quantifying flexibility in tradespace exploration: Value-weighted filtered outdegree", AIAA SPACE 2009 Conference & Exposition. 2009.

[67] Cardin, Michel-Alexandre, et al. "Extracting Value from Uncertainty: A Methodology for Engineering Systems Design", INCOSE International Symposium. Vol. 17. No. 1. 2007.

[68] De Neufville, Richard. "Architecting/designing engineering systems using real options", Engineering Systems Division Working Paper Series, Massachusetts Institute of Technology (2002).

[69] Neely III, James E., and Richard De Neufville. "Hybrid real options valuation of risky product development projects", International Journal of Technology, Policy and Management 1.1 (2001): 29-46.

[70] Haubelt, Christian, et al. "System design for flexibility", 2002 Proceedings: Design, Automation and Test in Europe Conference and Exhibition. IEEE, 2002.

[71] Wang, Tao, and Richard De Neufville. "Real options "in" projects", Real Options Conference, Paris, France. 2005.

[72] Wang, Tao, and Richard De Neufville. "Building real options into physical systems with stochastic mixed-integer programming", Engineering Systems Division Working Paper Series, Massachusetts Institute of Technology (2005).

[73] De Weck, Olivier L., Richard De Neufville, and Mathieu Chaize. "Staged deployment of communications satellite constellations in low earth orbit", Journal of Aerospace Computing, Information, and Communication 1.3 (2004): 119-136.

[74] Hassan, Rania, Richard De Neufville, and Douglas McKinnon. "Value-at-risk analysis for real options in complex engineered systems", 2005 IEEE International Conference on Systems, Man and Cybernetics. Vol. 4. IEEE, 2005.

[75] Walton, Myles A., and Daniel Hastings. "Applications of Uncertainty Analysis Applied to Architecture Selection of Satellite Systems", Journal of Spacecraft and Rockets 41.1 (2004): 75-84.

[76] Yao, Wen, et al. "Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles." Progress in Aerospace Sciences 47.6 (2011): 450-479.

[77] Elms, D. G. "Structural safety–issues and progress." Progress in Structural Engineering and Materials 6.2 (2004): 116-126.

[78] Sondecker IV, George Ralph. "Identification and evolution of quantities of interest for a stochastic process view of complex space system development." SM thesis. Massachusetts Institute of Technology, 2011.

[79] Stout, Kevin Dale. "Bayesian-based simulation model validation for spacecraft thermal systems." PhD thesis. Massachusetts Institute of Technology, 2015.

[80] Sankararaman, Shankar. "Uncertainty Reduction using Bayesian Inference and Sensitivity Analysis: A Sequential Approach to the NASA Langley Uncertainty Quantification Challenge." 18th AIAA Non-Deterministic Approaches Conference. 2016.

[81] Shaw, Graeme B., D. Miller, and Daniel E. Hastings. "Development of the quantitative generalized information network analysis (GINA) methodology for satellite systems." 1999 IEEE Aerospace Conference. Proceedings (Cat. No. 99TH8403). Vol. 5. IEEE, 1999.

[82] Shaw, Graeme Barrington. "The generalized information network analysis methodology for distributed satellite systems." PhD thesis. Massachusetts Institute of Technology, 1999.

[83] Biswas, Abhijit, and Sabino Piazzolla. "Deep-space optical communications downlink budget from Mars: System parameters." IPN Progress Report 42.154 (2003): 0-1.

[84] Clements, Emily Baker. "Probabilistic methods for systems engineering with application to nanosatellite laser communications." PhD thesis. Massachusetts Institute of Technology, 2018.

[85] Stahl, H.P. "Survey of cost models for space telescopes." Optical Engineering, 49(5), 2010. doi:10.1117/1.3430603.

[86] Wertz, James R., David F. Everett, and Jeffery J. Puschell. "Space mission engineering: the new SMAD." Microcosm Press, 2011.

[87] Lang, T.J. "A parametric examination of satellite constellations to minimize revisit time for low earth orbits using a genetic ", Advances in the Astronautical Sciences, 109:625–640, 2001.

[88] Williams, E. A., W. A. Crossley, and T. J. Lang. "Average and maximum revisit time trade studies for satellite constellations using a multiobjective genetic algorithm", Spaceflight Mechanics 2000, volume 105 of Advances in the Astronautical Sciences, Pt. 1, pages 653–666, 2000.

[89] Williams, E. A., W. A. Crossley, and T. J. Lang. "Average and maximum revisit time trade studies for satellite constellations using a multiobjective genetic algorithm", Journal of the Astronautical Sciences, 49(3):385–400, 2001.

[90] Leachtenauer, J. C., W. Malila, J. Irvine, L. Colburn, and N. Salvaggio. "General image quality equation: GIQE", Applied Optics, 36(32):8322–8328, 1997.

[91] Thurman, S. T., and J. R. Fienup. "Analysis of the general image quality equation", Proceedings of SPIE 6978, Visual Information Processing XVII, 69780F, 2008.

[92] Dilley, Maxx. "Natural disaster hotspots: a global risk analysis, volume 5". World Bank Publications, 2005.

[93] Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II". IEEE Transactions on Evolutionary Computation, 6(2):182–197, 2002.

[94] Reuther, Albert, et al. "Interactive supercomputing on 40,000 cores for machine learning and data analysis", 2018 IEEE High Performance Extreme Computing Conference (HPEC). IEEE, 2018.

[95] "MIT Supercloud Systems and Software", Available online at: http://supercloud.mit.edu/systems-and-software Accessed on October 25, 2019.

[96] Lincoln Laboratory Supercomputing Center | MIT Lincoln Laboratory. Retrieved on 12/02/2021 from https://www.ll.mit.edu/about/facilities/lincoln-laboratory-supercomputing-center

[97] Shapiro, Samuel Sanford, and Martin B. Wilk. "An analysis of variance test for normality (complete samples)." Biometrika 52.3/4 (1965): 591-611.

[98] Efron, Bradley, and Robert Tibshirani. "Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy." Statistical Science Vol. 1, No. 1, pp54-75, Feb 1986.

[99] Bland, J. Martin, and Douglas G. Altman. "Statistics notes: bootstrap resampling methods." British Medical Journal 2015, Vol. 350, Locator No. 2622.

[100] Chao, Min-Te, and Shaw-Hwa Lo. "A bootstrap method for finite population." Sankhyā: The Indian Journal of Statistics, Series A (1985): p399-405.

[101] Pezzullo, John. "The Bootstrap Method for Standard Errors and Confidence Intervals", from 'Biostatistics for dummies,' John Wiley & Sons, 2013.

[102] Kwak, Sang Gyu, and Jong Hae Kim. "Central limit theorem: the cornerstone of modern statistics." Korean Journal of Anesthesiology 70.2 (2017): 144.

[103] Pek, Jolynn, Augustine CM Wong, and Octavia CY Wong. "Confidence intervals for the mean of non-normal distribution: transform or not to transform." Open Journal of Statistics 7.3 (2017): 405-421.

[104] Shen, Eugene. "Note on the sampling error of the median." Journal of Educational Psychology 26.2 (1935): 154.

[105] Kallner, Anders. "Distributions of Data", from Laboratory Statistics: Methods in Chemistry and Health Sciences. Elsevier, 2017.

[106] Evans, W. Duane. "The standard error of percentiles." Journal of the American Statistical Association 37.219 (1942): 367-376.

[107] Bland, Martin. "Confidence interval for a median and other quantiles", from 'An Introduction to Medical Statistics.' Oxford University Press, 2015.

[108] National Research Council. "Assessing the reliability of complex models: mathematical and statistical foundations of verification, validation, and uncertainty quantification." National Academies Press, 2012.

[109] Bishop, A., and P. Messina. "Scientific grand challenges for national security: The role of computing at the extreme scale", Technical report: US Department of Energy, Washington D.C., 2009. Available online at: https://science.osti.gov/-/media/ascr/pdf/program-documents/docs/Nnsa_grand_challenges_report.pdf

[110] Box, George E.P., and Norman R. Draper. "Empirical model-building and response surfaces." John Wiley & Sons, 1987.

[111] Jeffreys, H. 1967. "Statistical Methods in Seismology", International Dictionary of Geophysics, Pergamon, London, 1967, pp. 1398-1401.

[112] Sullivan, Timothy John. "Introduction to uncertainty quantification", Texts in Applied Mathematics Vol. 63. Springer, 2015.

[113] Saltelli, Andrea. "Sensitivity analysis for importance assessment." Risk Analysis Vol. 22.3 (2002), pp. 579-590.

[114] Gibbons, Jean Dickinson, and Subhabrata Chakraborti. "Nonparametric statistical inference (4th ed.)" Taylor & Francis, 2014.

[115] Sheskin, David J. "Handbook of parametric and nonparametric statistical procedures (5th ed.)", Chapman and Hall/CRC Press, 2003.

[116] Bradley, James V. "Distribution-free statistical tests." Englewood Cliffs, N.J: Prentice-Hall, 1968.

[117] Neave, Henry R., and Peter L. Worthington. "Distribution-free tests." Unwin Hyman, 1988.

[118] Shalizi, Cosma. "Estimating distributions and densities", Lecture 28, Statistics 36-350: Data Mining, Fall 2009. Retrieved on 04/12/2021 from https://www.stat.cmu.edu/ cshalizi/350/lectures/28/lecture-28.pdf

[119] Kamalov, Firuz. "Kernel density estimation based sampling for imbalanced class distribution." Information Sciences Vol. 512 (2020): pp. 1192-1201.

[120] Gurland, John, and Ram C. Tripathi. "A simple approximation for unbiased estimation of the standard deviation." The American Statistician Vol. 25.4 (1971), pp. 30-32.

[121] Erichsen, Peter. "Performance evaluation of spacecraft propulsion systems in relation to mission impulse requirements." European Spacecraft Propulsion Conference. Vol. 398. 1997.

[122] Find minimum of single-variable function on fixed interval - MATLAB fminbnd. Retrieved on 02/23/2022 from https://www.mathworks.com/help/matlab/ref/fminbnd.html

[123] "Gridded Population of the World, Version 4 (GPWv4): Population Density Adjusted to Match 2015 Revision UN WPP Country Totals, Revision 11." NASA Socioeconomic Data and Applications Center (SEDAC) - hosted by CIESIN (Center for International Earth Science Information Network), Columbia University, 31 Dec 2018. https://doi.org/10.7927/H4F47M65

[124] "Submitting Jobs | MIT Supercloud | Triples Mode Tuning", Available online at: https://supercloud.mit.edu/submitting-jobs#triples Accessed on April 25, 2022.