AUTOMATED SWARM DESIGN ARCHITECTURES FOR RECONNAISSANCE OF SMALL BODIES

by

Ravi teja Nallapu

Copyright © Ravi teja Nallapu 2020

A Dissertation Submitted to the Faculty of the
DEPARTMENT OF AEROSPACE & MECHANICAL ENGINEERING
In Partial Fulfillment of the Requirements For the Degree of
DOCTOR OF PHILOSOPHY
In the Graduate College
THE UNIVERSITY OF ARIZONA

2020
As members of the Dissertation Committee, we certify that we have read the dissertation prepared by: Ravi teja Nallapu
titled:

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Jekan Thanga
Date: Dec 4, 2020
Vishnu Reddy
Date: Dec 4, 2020
Eric Butcher
Date: Dec 4, 2020
Roberto Furfaro
Date: Dec 17, 2020
Erik Ian Asphaug
Date: Jan 28, 2021

Final approval and acceptance of this dissertation is contingent upon the candidate’s submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Jekan Thanga
Date: Dec 4, 2020
Aerospace and Mechanical Engineering Department
ACKNOWLEDGEMENTS

Achieving a doctorate in aerospace engineering has been my lifelong goal. This, in no way, could be achieved without the support of my mentors, peers, friends, and family. I use this section to express my gratitude to the key people who supported me throughout this journey. I express my sincere gratitude to my advisor Dr. Jekan Thanga for supporting me all the way through the program. Dr. Thanga’s “think outside the box” approach and emphasis on developing independent research were very useful in shaping my research career. My co-advisor, Dr. Erik Asphaug, played an important role in my career in highlighting the interactions between spacecraft engineering and planetary sciences. I thoroughly enjoyed working as the attitude and GNC engineer on 5 CubeSat mission concepts with Dr. Thanga and Dr. Asphaug. I then thank Dr. Vishnu Reddy, Dr. Eric Butcher, and Dr. Roberto Furfaro for advising me as my doctoral committee members. I am especially thankful to Dr. Butcher and Dr. Reddy for serving as my teachers in my graduate course work. Dr. Butcher taught two of my favorite courses at the University of Arizona. His lectures on spacecraft attitude control and interplanetary mission design were so insightful that most of my research depended on them.

I am also very thankful for the support from several of my lab mates and colleagues, including Dr. Stephen Schwartz, Leonard Vance, Steven Morad, Aman Chandra, Aaditya Ravindran, Tristan Schuler, and Steven Anderson. I am very grateful to Steven Morad for teaching me how to use ROS, Python, and Linux efficiently. These skills allowed me to accelerate my hardware demonstration research in Chapter 8. Additional thanks to my colleague, Bharath Pidaparthi, who was always around to discuss optimization theory. Also, special thanks to my writing group team members: Bokjin Roh, and Tammee Albrecht for supporting me continuously during my dissertation writing stage. My favorite memories during my doctoral degree were those of mentoring students. I am very thankful for all the contributions to my research from my team members at the University of Arizona: Yinan Xu, Abraham Marquez, Akash Gholap, and Emilio Guevara; and at the Arizona State University: Andrew Warren, Shot Aichikawa, and Amit Tallapragada.

Surviving the stress of pursuing a doctorate would not have been possible without a solid support structure. For this ineffable support, I want to thank my parents: Vinodini and Prahlad Nallapu, and my wife, Vrushali Manka. Finally, while the space restriction limits the number of people listed here, I want to thank everyone who has supported me through my journey. The journey of achieving my doctorate is filled with many strong memories of education, character building, and friendships that could not be achieved without you.
DEDICATION

I dedicate this work to

Prahlad, Vinodini, Vrushali, and Snehith
Many parts of this dissertation have appeared in peer-reviewed publications. This thesis has produced 2 peer reviewed journal articles, and 8 conference publications. Elements of Chapter 2 appeared in Nallapu and Thangavelautham (2019d), where the IDEAS architecture was presented along with a case study of designing global surface mapping missions to uniformly rotating asteroids. The architecture was then expanded in Nallapu and Thangavelautham (2019b) to design region of interest observations around uniformly rotating asteroids. Elements of Chapter 3 appeared in Nallapu and Thangavelautham (2019a), and Nallapu et al. (2020a) where the IDEAS architecture was used to design Earth observation, and lunar communication mission concepts respectively. The sensitivity analysis procedure described in Chapter 4 was published in (Nallapu and Thangavelautham, 2021) along with parts of Chapter 6. Many elements of Chapter 5 describing mission concept design to explore planetary moons through hyperbolic flybys were published in Nallapu and Thangavelautham (2020b). Parts of the Chapter 6 were initially published in (Nallapu and Thangavelautham, 2019c) where the IDEAS architecture was used to design global surface mapping missions to planetary moons using co-orbits. This work was expanded with remaining elements in Chapter 6, and Appendix A and was published in Nallapu and Thangavelautham (2020a). Many elements of Chapter 3, and 7, where the IDEAS architecture was used to design reconnaissance mission concepts to tumbling interstellar visitors. Kindly note that many of these papers contain some overlapping content and were intended for different audiences.
# TABLE OF CONTENTS

LIST OF FIGURES .......................................................... 8

LIST OF TABLES ........................................................... 16

ABSTRACT ................................................................. 19

CHAPTER 1 Introduction .................................................. 21

CHAPTER 2 Mission Design Framework of IDEAS ....................... 28
  2.1 Related Work ....................................................... 30
  2.2 The IDEAS Architecture ......................................... 43

CHAPTER 3 Swarm Mission Concept Modeling .......................... 48
  3.1 Reconnaissance Mission Concepts ................................. 48
  3.2 Performance Evaluation of Mission Concepts .................. 50
  3.3 Dynamical Parameters .......................................... 59
  3.4 Statistical Coverage Evaluation ................................. 67

CHAPTER 4 Sensitivity Analysis of Swarm Designs ..................... 73
  4.1 Swarm Gene Structures .......................................... 73
  4.2 Analysis Algorithms ........................................... 75
  4.3 Sensitivity to Dynamical Perturbations .......................... 80

CHAPTER 5 Case Study-1: Moon Reconnaissance Using Hyperbolic Flybys 86
  5.1 Related Work ....................................................... 87
  5.2 Mission Concept Modeling ....................................... 90
  5.3 Numerical Simulations ........................................... 100
  5.4 Sensitivity Analysis ............................................. 107
  5.5 Discussion ....................................................... 114

CHAPTER 6 Case Study-2: Moon Reconnaissance Using Resonant Co-orbits 117
  6.1 Related Work ....................................................... 117
  6.2 Mission Concept Modeling ....................................... 119
  6.3 Numerical Simulations ........................................... 135
  6.4 Sensitivity Analysis ............................................. 145
  6.5 Discussion ....................................................... 153
<table>
<thead>
<tr>
<th>CHAPTER 7</th>
<th>Case Study-3: Reconnaissance of Tumbling Asteroids</th>
<th>162</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Related Work</td>
<td>162</td>
</tr>
<tr>
<td>7.2</td>
<td>Mission Concept Modeling</td>
<td>164</td>
</tr>
<tr>
<td>7.3</td>
<td>Numerical Simulations</td>
<td>171</td>
</tr>
<tr>
<td>7.4</td>
<td>Discussion</td>
<td>184</td>
</tr>
<tr>
<td>CHAPTER 8</td>
<td>Hardware Demonstration</td>
<td>189</td>
</tr>
<tr>
<td>8.1</td>
<td>Related Work</td>
<td>190</td>
</tr>
<tr>
<td>8.2</td>
<td>Experimental Setup</td>
<td>193</td>
</tr>
<tr>
<td>8.3</td>
<td>Mission Concept Design</td>
<td>214</td>
</tr>
<tr>
<td>8.4</td>
<td>Results</td>
<td>220</td>
</tr>
<tr>
<td>8.5</td>
<td>Discussion</td>
<td>228</td>
</tr>
<tr>
<td>CHAPTER 9</td>
<td>Conclusion</td>
<td>232</td>
</tr>
<tr>
<td>9.1</td>
<td>Summary</td>
<td>232</td>
</tr>
<tr>
<td>9.2</td>
<td>Contributions</td>
<td>234</td>
</tr>
<tr>
<td>9.3</td>
<td>Future Work</td>
<td>237</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>Subsystem Inventory and Cost Modeling</td>
<td>239</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>Leader Selection Algorithm</td>
<td>246</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
<td>248</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1.1 Examples of different types of small bodies found in the solar system. 22
1.2 Artistic rendering of future swarm-based mission concepts to small bodies: ANTS mission concept (left), and OpGrav (right). . . . . . 23

2.1 Artistic rendering of the Mars Climate Orbiter spacecraft which terminated in a crash landing due to a measurement unit mismatch between the control and navigation systems. . . . . . . . . . . . . . . 29
2.2 The vision of MBSE showing a shift from traditional standalone design methods to a unified, collaborative design environment. . . . . 31
2.3 Example configurations of spacecraft in different swarm architectures. 33
2.4 Illustration of Evolutionary Algorithms, showing design principles in nature (top), and their application in artificial design (bottom). . . . . 37
2.5 The different types of child genes in the population of GA generation. 39
2.6 Software architecture of the proposed IDEAS software to provide an end-to-end design framework for spacecraft swarm missions. . . . . . 43
2.7 The sequential exchange of information across the different modules of the IDEAS architecture. . . . . . . . . . . . . . . . . . . . . . . . 45

3.1 Geometrical and camera parameters used in designing the flyby encounters of the target body . . . . . . . . . . . . . . . . . . . . . . . 50
3.2 Process diagram showing the different operations that occur inside the Automated Swarm Designer module of IDEAS. . . . . . . . . 51
3.3 Illustration of the illumination operation which filters the faces of the model facing the Sun. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
3.4 Process diagrams for evaluating illuminated shape models (left), and detecting planetary moon eclipses (right). . . . . . . . . . . . . 53
3.5 Illustration of the culling operation which filters the faces of the model facing the spacecraft. . . . . . . . . . . . . . . . . . . . . . . . 54
3.6 Process diagram illustrating the software implementation of the culling operation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
3.7 Illustration of the clipping operation which filters the vertices of the model that fall inside the spacecraft’s FoV. . . . . . . . . . . . . 56
3.8 Process diagram illustrating the software implementation of the clipping operation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
3.9 Process diagram describing the role of the different operations used to estimate the instantaneous surface coverage of the swarm. . . . 59
3.10 Process diagram indicating the cumulative coverage of the swarm for spatial and temporal reconnaissance missions. .......................... 60
3.11 Illustration of a swarm design which contains a potential collision with the target small body. .................................................. 62
3.12 Illustration of a swarm design which contains a potential spacecraft with the target small body. .................................................. 63
3.13 The mapping behavior of the spacecraft in the designed Class 2 swarms. ................................................................. 64
3.14 The communications behavior of the spacecraft in the designed Class 2 swarms. ....................................................... 65
3.15 Geometrical parameters used to model the attitude of the synchronously rotating planetary moons. ........................................ 66
3.16 Illustration of the nominal and Dual Sphere modes used for global mapping coverage evaluation. ............................... 68
3.17 Illustration of various geometrical parameters used to model the attitude of the tumbling asteroids. ................................. 70
4.1 Gene maps of the planetary moon (top) and asteroid mapping (bottom) swarms, showing the different categories of the swarm design variables. .................................................. 74
4.2 Illustration of a swarm encounter with the target body, showing the nominal encounter (left) an encounter with spacecraft outages (right). .......................................................... 76
4.3 Illustration of the effect of perturbing the initial location of the target body. ................................................................. 78
4.4 Illustration of the effect of perturbing the encounter location of the spacecraft during their encounter. .................................... 79
5.1 A comparison of different mission concepts to explore small planetary moons. ................................................................. 87
5.2 Trajectories of the Voyager spacecraft used to explore the moons of Saturn through hyperbolic flybys of the planet (left), and the images of the moons taken during these flybys (right). [Image credit: NASA] 88
5.3 Highest resolution images of Phobos recorded till date by the Mars Express mission. The resolution of the outer image is about 3.9 m/px, while the 7 internal images were imaged at 3 m/px. .................. 89
5.4 Trajectory gene for designing the Earth-Mars trajectories of the Phobos mapping swarm. ....................................................... 92
5.5 Design space of the hyperbolic flyby swarm, showing the swarm size and target location parameters(left), and encounter shaping parameters (right). .................................................. 93
LIST OF FIGURES – Continued

5.6 Gene map of the swarm design problems showing different design variables and their attributes. ......................................................... 93
5.7 The Dual Sphere shape models of Phobos used in the current work. ... 99
5.8 Variation of the trajectory fitness for the global surface mapping mission concept across different GA populations generations showing the individual optimization trials (left), and their statistical distribution (right). .......................................................... 101
5.9 Figures of merit of the feasible trajectory genes identified by the 5 GA optimization trials. ........................................................... 102
5.10 Optimal solution shown on the Earth-Mars Pork-Chop plot, along with the trajectory design parameters. ................................. 103
5.11 Optimal Earth to Mars trajectory of the Phobos mapping swarm. ... 103
5.12 Variation of the swarm fitness across different GA populations generations showing the individual optimization trials (left), and their statistical distribution (right). .............................................. 105
5.13 Figures of merit of the feasible swarm genes identified by the 5 GA optimization trials. ........................................................... 106
5.14 The arrangement of the hyperbolic trajectories in the designed optimal swarm (top) along with a visualization of a sample encounter with Phobos (bottom). .................................................. 108
5.15 Dynamical evolution of the surface coverage of the nominal and Dual Sphere shape models of Phobos noted during its orbital period. ... 109
5.16 The cumulative coverage pattern on the nominal Phobos model (left) and its corresponding Dual Sphere models (right) resulting after a simulated flyby of the swarm. ............................... 110
5.17 Variation in the coverage of Phobos with spacecraft outages in the swarm. ................................................................. 112
5.18 Variation of the cumulative coverage with uncertainty in the orbital location of Phobos. ......................................................... 113
5.19 Variation of the coverage of the swarm with different amplitudes of spacecraft rendezvous uncertainty. .................................. 114
5.20 Magnitude of perturbative accelerations acting on the spacecraft motion. ......................................................................... 115
5.21 Variation of coverage with different levels of motion propagation fidelity.116

6.1 Images of the martian moon Deimos captured by the HiRISE camera of the MRO spacecraft, at a resolution of about 20 m/px. ........ 118
6.2 Capture of an incoming hyperbolic trajectory into a highly eccentric orbit around the central planet. ........................................ 122
LIST OF FIGURES – Continued

6.3 Walk in maneuver performed at the apoapsis of the HEO to initiate the main phase of aerobraking. ........................................... 123
6.4 Main phase of the aerobraking maneuver where the apoapsis altitude of the orbit is reduced upon passing through the atmosphere of the central planet. ........................................... 124
6.5 Walk out phase of aerobraking where the spacecraft raises its periapsis altitude to that of the targeted resonant orbit. ............... 124
6.6 Gene map of the trajectory gene corresponding to the co-orbiting swarm problem. ........................................... 127
6.7 Swarm behaviours for the two visual mapping mission concepts: global surface mapping (left) and RoI observation (right) missions. 129
6.8 Swarm design gene for the global surface mapping swarm missions. 130
6.9 Swarm design gene for the RoI observation swarm missions. ........ 131
6.10 The nominal and Dual Sphere models of Deimos used for designing global surface mapping mission. ............................. 133
6.11 The nominal shape model showing the selected RoI faces used for RoI observation mission design. ............................. 134
6.12 Variation of the trajectory fitness for the global surface mapping mission across different GA generations showing the individual optimization trials (left), and their statistical distribution (right). ........................................... 137
6.13 The figures of merit of the final generation feasible trajectory genes for the global surface mapping mission across all optimization trials. 138
6.14 Variation of the global surface mapping swarm fitness across different GA generations showing the individual optimization trials (left), and their statistical distribution (right). ........................................... 140
6.15 The figures of merit of the final generation feasible swarm genes for the global surface mapping swarm mission across all optimization trials. ........................................... 141
6.16 Arrangement of resonant co-orbits for the global surface mapping mission to Deimos. ........................................... 143
6.17 Simulated encounter of a spacecraft in the global surface mapping swarm with Deimos, showing its instantaneous surface coverage. 144
6.18 Time evolution of the spatial coverage of Deimos noted during a simulated swarm flyby. ........................................... 145
6.19 Cumulative surface coverage of the nominal and Dual Sphere models of Deimos, noted after all the swarm encounters. ............... 146
6.20 Variation of the RoI observation mission trajectory fitness across different GA generations showing the individual optimization trials (left), and their statistical distribution (right). ........................................... 147
6.21 The figures of merit of the feasible generation trajectory genes for the RoI observation mission across all optimization trials. .......................... 148
6.22 Variation of the RoI observation swarm fitness across different GA generations showing the individual optimization trials (left), and their statistical distribution (right). ................................................................. 150
6.23 The figures of merit of the final generation feasible swarm genes for the RoI observation mission across all optimization trials. .................. 151
6.24 Arrangement of resonant co-orbits for the RoI observation mission to Deimos. ........................................................................ 153
6.25 Simulated encounter of a spacecraft in the swarm with Deimos (left), showing its instantaneous coverage (right). .......................... 154
6.26 Time evolution of the temporal (left) and spatial (right) RoI coverage of Deimos noted during a simulated swarm flyby. .................. 155
6.27 Variation in the surface coverage of Deimos with spacecraft outages in the global surface mapping swarm. ................................. 157
6.28 Variation in the temporal (left) and spatial (right) RoI coverage of Deimos with spacecraft outages in the swarm. ......................... 157
6.29 Variation in the coverage of Deimos shape models with with uncertainty in the location of Deimos. .................................................. 158
6.30 Variation in the temporal (left) and spatial (right) RoI coverage of Deimos with with uncertainty in the location of Deimos. ............. 158
6.31 Variation in the coverage of the Deimos shape models with with uncertainty in the location of encounters. ................................. 159
6.32 Variation in the temporal (left) and spatial (right) RoI coverage of Deimos with with uncertainty in the location of encounters. ........ 160
6.33 Perturbative acceleration profiles of an example spacecraft in the global surface mapping swarm. .................................................. 160
6.34 Variation in the coverage of the Deimos shape models with different levels of motion propagation fidelity. ................................. 161
6.35 Variation in the temporal (left) and spatial (right) RoI coverage of Deimos with with different levels of motion propagation fidelity. .... 161

7.1 Radar generated shape model of the asteroid of 4179 Toutatis showing its views about its principal axes. [Source: Hudson and Ostro (1995)]. 164
7.2 Spacecraft images of Toutatis noted from the flyby of Chang’ e 2 showing the imaging distance $D$, flyby epoch $T$, and the imaging resolution $R$. [Source: Huang et al. (2013)]. ................................. 165
7.3 Illustration of the nominal trajectory (left), and swarm deployment (right) segments of the interplanetary trajectory design. ................. 166
LIST OF FIGURES – Continued

7.4 Illustration of the 2 encounter designs (left), and the 6 encounter locations (right) considered at each encounter (right). ............ 168
7.5 Gene map of the trajectory design problem to tumbling asteroids. . . 168
7.6 Illustration of the geometrical parameters involved in designing the swarm encounters with the tumbling asteroid. ................. 169
7.7 Gene map of the swarm design problem to tumbling asteroids. ...... 170
7.8 The nominal and Dual Sphere models of Toutatis used for computing the surface coverage. .............................................. 172
7.9 Variation of the trajectory fitness for the global surface mapping mission across different GA generations showing the individual optimization trials (left), and their statistical distribution (right). .......... 174
7.10 Trajectory figures of merit of feasible solutions identified by all optimization trials. .................................................... 175
7.11 Porkchop plot contours of Earth-Toutatis trajectories showing the $C_{3,E}$, and $v_{\infty,2}$ of the selected solution. ............. 176
7.12 Selected optimal trajectory design of the mothership and swarm deployment. .............................................................. 177
7.13 Variation of the global surface mapping swarm fitness across different GA generations showing the individual optimization trials (left), and their statistical distribution (right). .......... 179
7.14 Performance figures of merit of feasible swarm genes identified by all optimization trials. .............................................. 180
7.15 Heliocentric trajectories of the spacecraft in the swarm showing the encounter locations on the orbit of Toutatis. ............... 182
7.16 Spacecraft trajectories during an encounter with Toutatis (left) showing its instantaneous coverage over the three shape models of Toutatis (right). ...................................................... 183
7.17 Dynamical evolution of the coverage of the nominal and Dual Sphere shape models of Toutatis. ....................................... 184
7.18 Surface coverage patterns over all 3 shape models of Toutatis, generated by the swarm after all spacecraft encounters. ........ 185
7.19 Variation in the surface coverage of Toutatis with spacecraft outages in the designed global surface mapping swarm. ....... 186
7.20 Variation in the surface coverage of Toutatis with uncertainty in its orbital location. ....................................................... 187
7.21 Variation in the surface coverage of Toutatis with uncertainty in spacecraft encounter locations. ................................. 187
7.22 Perturbative accelerations experienced by a sample spacecraft in the swarm during the simulated rotation period of Toutatis. .... 188
LIST OF FIGURES – Continued

7.23 Variation in the coverage of the Toutatis shape models with different levels of motion propagation fidelity. ................................. 188

8.1 Illustration of UAVs serving as hardware platforms to advance simulated spacecraft as functional prototypes. .............................. 190
8.2 Illustration of the Structure from Motion (SfM) operation, where a target feature of interest is imaged from multiple perspectives. .... 192
8.3 Hardware architecture of MAPS testbed showing the interactions between various constituent subsystems. ............................. 193
8.4 The Parrot ANAFI drone which is used as the seed autonomous agent in the MAPS testbed. .............................................. 195
8.5 Hardware architecture of the ANAFI navigation and control system. 197
8.6 Hardware architecture of the ANAFI camera system. ................. 198
8.7 Hardware architecture of the motion tracking system. .................. 199
8.8 Drones configured with object tracking markers (left), which are then used to define skeleton structures in the Vicon Tracker (right) for motion tracking. .................................................. 200
8.9 Different components of the small body simulation system. ............ 201
8.10 The physical small body mock-up (top), and its baseline shape model generated from the SfM pipeline (bottom). ....................... 202
8.11 The baseline nominal and Dual Sphere shape models of the small body mock-up used in the current work. ............................... 203
8.12 The experimental setup of a mapping trial inside the MAPS simulator 204
8.13 Software architecture of the MAPS testbed showing the interactions between various constituent nodes. .............................. 205
8.14 Concept of operations of a seed drone during the mapping mode. .... 206
8.15 Geometrical parameters used for defining the reference attitude of the drone. ................................................................. 208
8.16 Control loop architecture of the seed drone to track the reference trajectory. ................................................................. 209
8.17 Different tasks involved in the result processing pipeline. ............... 211
8.18 The Structure from Motion pipeline implemented inside the Meshroom software. ............................................................. 212
8.19 Pixel-level geometrical parameters used for determining the maximum flyby altitude of the UAVs. ......................................... 215
8.20 Arrangement of the nominal target body shape model showing the different design gene parameters. ................................. 216
8.21 Gene map of the automated swarm design problem of the global surface mapping mission implemented inside the MAPS testbed. .... 217
LIST OF FIGURES – Continued

8.22 Encounter geometry parameters used to determine the reference velocity of the drones. .......................................................... 218
8.23 Variation of the global surface mapping fitness across different GA generations showing the individual optimization trials (left), and their statistical distribution (right). .................................................. 222
8.24 Performance figures of merit of the feasible design genes identified from the final generations of all optimization trials. ................. 223
8.25 Drone trajectories specified by the selected design gene. .............. 224
8.26 Coverage pattern over the nominal and Dual Sphere shape models of the target after the UAV flybys. ............................................. 225
8.27 Vicon frame trajectories of the participating UAVs. ..................... 227
8.28 Cartesian components of the UAV pose topics showing the performance of the designed reference tracking controller. .............. 228
8.29 Sample images generated from the video recordings by the UAVs. ... 229
8.30 The 3D surface reconstruction of the UAV flight path generated from a sample experiment. .............................................. 229
8.31 Scaled target body vertices extracted from the reconstructed flight paths of different mapping experiments. ................................. 230
8.32 Individual and cumulative coverage estimated by the IDEAS architecture. ........................................................... 231
8.33 Effects of adding more drones to the selected design along the angular (top) and radial directions (bottom) .................................. 231
B.1 Process diagram describing the algorithm to select the leader spacecraft in the swarm. ................................................. 247
# LIST OF TABLES

3.1 LoS tracking targets, and corresponding rotation matrices for the visual mapping spacecraft swarm ........................................... 65

4.1 Perturbation accelerations used in the current work to study the variation of spacecraft coverage under different levels of dynamical fidelity. 85

5.1 Input mission requirements for the Phobos mapping swarm. ....... 91
5.2 Dynamical parameters to model the encounter trajectories of the swarm with and Phobos. .................................................. 99
5.3 Shape model parameters of the Phobos model used in the current work.100
5.4 Flyby and camera parameters required to design the Phobos Flyby. . 100
5.5 Input parameters to design the Earth-Mars trajectory of the Phobos mapping swarm. .................................................. 101
5.6 Input parameters to solve the swarm design problem. .................. 104
5.7 Design variables of the selected optimal swarm gene. .................... 107
5.8 Cumulative coverage statistics noted from 20 random initial orientations of Phobos. .................................................. 107
5.9 Design parameters and cost breakdown of the seed spacecraft in the designed swarm. .................................................. 111
5.10 Cost breakdown of the $N_{sw} = 5$ spacecraft swarm mission inflated to the launch year 2020. ........................................... 111
5.11 Input parameters to study the sensitivity of the swarm to spacecraft outages. .................................................. 112
5.12 Input parameters to study the sensitivity of coverage to the location of the moon. .................................................. 113
5.13 Input parameters to study the sensitivity of coverage to the rendezvous location of the swarm. ........................................... 114

6.1 User input mission requirements for the Deimos mapping swarms. ... 120
6.2 Dynamical parameters to model the encounter co-orbits of the swarm with and Deimos. .................................................. 132
6.3 Shape parameters of the Deimos model used in the current work. ... 133
6.4 Shape parameters of the selected region of interest on Deimos. ...... 134
6.5 Flyby and camera parameters to design the reconnaissance co-orbits. 134
6.6 Input parameters to design the Earth-Mars trajectory of the Deimos mapping swarm. .................................................. 136
LIST OF TABLES – Continued

6.7 Design variables of the selected near optimal trajectory design gene for the global surface mapping mission. ............................... 139
6.8 Input parameters to design the global surface mapping swarm at Deimos 139
6.9 Design variables of the selected near optimal swarm design gene for global surface mapping mission. ............................... 142
6.10 Coverage and maneuver figures of merit of the global surface mapping swarm. .......................................................... 142
6.11 Design variables of the selected near optimal trajectory design gene for RoI observation mission. ........................................ 143
6.12 Input parameters to design the RoI observation swarm at Deimos 149
6.13 Design variables of the selected near optimal swarm design gene for RoI observation mission. ........................................ 152
6.14 Coverage and maneuver figures of merit of the RoI observation swarm. 152
6.15 Seed spacecraft design parameters and associated cost breakdown for the two mission concepts. ................................. 155
6.16 Cost breakdown of the two designed swarm mission concepts. 156
6.17 Input parameters to study the sensitivity of spacecraft outage to the coverage performance of the swarms. ......................... 156
6.18 Input parameters to study the coverage sensitivity to the orbital location of Deimos. ..................................................... 156
6.19 Input parameters to study the coverage sensitivity to the encounter location of the spacecraft. ................................. 159

7.1 Parameters used to describe the dynamical environment around Toutatis. ............................................................... 171
7.2 Geometrical properties measured from the nominal and Dual Sphere shape models of Toutatis. ........................................ 173
7.3 Input parameters to design the interplanetary trajectory of the spacecraft swarm. ........................................................ 173
7.4 Design variables of the selected optimal trajectory gene. .......... 174
7.5 Input parameters to design the Toutatis global surface mapping swarm. 178
7.6 Design variables of the selected optimal swarm gene. .......... 181
7.7 Performance figures of merit of the designed global surface mapping swarm. ........................................................ 181
7.8 Seed spacecraft design parameters and its associated estimated costs. 182
7.9 Estimated swarm mission costs associated with the Toutatis mapping swarm. ............................................................... 183
7.10 Input parameters to study the sensitivity of spacecraft outage to the coverage performance of the swarms. ......................... 184
LIST OF TABLES – Continued

7.11 Input parameters to study the coverage sensitivity to the orbital location of Toutatis. ........................................ 186
7.12 Input parameters to study the coverage sensitivity to the encounter location of the spacecraft. ......................... 186

8.1 Technical specifications of the Parrot ANAFI drone. ............ 196
8.2 Camera parameters used to configure the drone recordings. .... 199
8.3 Geometrical parameters of the baseline target body shape models. . 202
8.4 Vicon frame measurements of the workspace, and target location inside the supporting structure. .................. 204
8.5 Mission definition parameters used for designing the global surface mapping mission implemented inside the MAPS tesbed. 221
8.6 Design variables of the selected near optimal design gene in the global surface mapping mission. ........................... 222
8.7 Reference trajectory and control parameters of the two drone used in the current work. ..................................... 226
8.8 Shape model parameters measured from the extracted target body surface. .......................................................... 230

A.1 Dry mass distribution of the seed spacecraft populated from an existing inventory of off-the-shelf subsystems. .............. 242
A.2 Multiplicative inflation factors used in the current work. ........ 243
A.3 Cost estimating relations predicted by the Small Satellite Cost Model. 244
A.4 Cost estimating relations used in the NICM Model. ................ 245
A.5 Launch vehicles used in the current work, and their corresponding launch cost per unit mass. ............................... 245
ABSTRACT

Small body surface exploration has potential benefits to planetary science, space security, and economy. Exploring small-bodies is a challenge due to the low-gravity fields and uncertainty in the gravitational environment. Attempting surface missions with inadequate gravity field information is prone to high-risk of failure. Additionally, gravity environments around these bodies constrain the radius of orbiter missions for reconnaissance. This leaves spacecraft flybys as an alternate strategy for reconnaissance. Flybys may be the only viable approach when it becomes difficult to orbit or land on a small body. The challenge with flybys is that they are time-limited, thus providing only a limited glimpse of the target. These disadvantages can be overcome using a swarm approach where a complex task is delegated to multiple-low cost agents to achieve synergistic performance. While swarms are an important tool for small body exploration, designing their mission concepts is a complex task, and importantly there is no end-to-end tool to design swarm mission concepts. This thesis develops an automated software solution to design small body reconnaissance missions using spacecraft swarm flybys. The developed software is called the Integrated Design Engineering and Automation of Swarms (IDEAS). IDEAS automates the spacecraft, swarm, and trajectory design processes in a swarm mission. The key focus of this thesis is on the development of the Automated Swarm Designer module of IDEAS, which designs minimum sized swarm missions for space-limited and time-limited reconnaissance missions using Evolutionary Algorithms. In this thesis, the mission concept design process inside the IDEAS framework is demonstrated through multiple case studies of small body reconnaissance missions. First, the design of global surface mapping mission concepts to planetary moons using hyperbolic flybys of the central planet is presented using the IDEAS framework. The design process is then demonstrated using a global surface mapping mission
concept to the Martian moon Phobos. The second case study explores the design of co-orbiting missions where the spacecraft enter into resonant co-orbits around the central planet for moon exploration. The design of these co-orbits to space and time-limited visual reconnaissance missions are then explored using the IDEAS framework. Specifically, the design of global surface mapping and region of interest observation mission concepts are presented using the Martian moon Deimos as the case study target. These principles are then extended to design reconnaissance missions to tumbling asteroids using mother-daughter swarm architectures. The design of such a swarms mission, using IDEAS, is demonstrated by designing a global surface mapping mission to the asteroid 4179 Toutatis. These case studies indicate that IDEAS is capable of generating small body reconnaissance missions concepts that overcome the spatial and temporal coverage limitations of flybys. Additionally, the sensitivity of the spatial and temporal coverages of the designed missions to perturbations such as spacecraft outages, uncertainty in encounter locations, and errors due to dynamical modeling is examined. These analyses describe the feasibility of the mission concepts identified by IDEAS. Finally, a testbed called as the Multi-Agent Photogrammetry of Small bodies (MAPS) is developed, which serves as a platform to demonstrate multi-spacecraft reconnaissance algorithms using autonomous UAVs. The UAV mapping mission design is formulated as a global surface mapping problem to generate a near-complete map of the target small body using a minimum number of UAVs. A near-optimal design with two UAVs is identified by IDEAS and is implemented inside the MAPS testbed. The UAV recordings are then processed inside a structure from motion (SfM) pipeline to generate a 3D reconstruction of the target. The reconstruction results demonstrate the successful performance of the mission concepts identified by IDEAS. In this way, the current work develops a software tool that automates the swarm mission concept generation and provides a hardware platform that can be used to demonstrate swarm mission concepts. Such tools will augment human mission designers in developing optimal, safe, and reliable reconnaissance mission concepts, thereby providing new pathways to explore the solar system.
CHAPTER 1

Introduction

Small body exploration sheds fundamental insight into topics such as the origin of life, the origin of Earth, and the origin of the solar system (Committee on the Planetary Science Decadal Survey, 2011). Small bodies broadly refer to asteroids, planetary moons, and comets—some well known solar system small bodies are presented in Figure 1.1. As seen here, these bodies are characterized by irregular shapes, small size, and their corresponding irregular gravity environments. The small bodies are characterized by their irregular shapes, small size, and their corresponding microgravity environments. While ground-based observations of these target bodies provide useful information, these results are limited by the low resolution and albedo effects. These factors require missions that can obtain a closer look at these bodies through flybys, orbital insertions, and touch and go missions. Consequently, the importance of surface exploration of these bodies is also highlighted by the Planetary Science Decadal survey 2013-2022 (Committee on the Planetary Science Decadal Survey, 2011; Castillo-Rogez et al., 2012). Additionally, Near-Earth Asteroids (NEAs) are high-value targets for in-situ missions as they contain resources to facilitate interplanetary travel (Nallapu et al., 2016; Calla et al., 2018). The Space Mission Directory (SMD) at NASA has referred to these destinations as high-priority targets and has focused on developing robotic and human-based strategies (Space Mission Directorate, 2020) to explore them.

However, the uncertainty in physical and dynamical characteristics of small bodies can be challenging for in-situ missions (Dudal and Loisel, 2016). Second, the orbital radii of the spacecraft around the small bodies are constrained by the dynamical environment (Melman et al., 2013). As a result, orbit insertion maneuvers incur significant fuel costs. Therefore, performing reconnaissance without orbiting these bodies is preferred, as they can also allow for touring one or more small bod-
ies. Typically, single spacecraft which carry the reconnaissance payloads are used to conduct these flyby observations. However, returns from a single observer are limited by both the spatial and temporal coverage of the payload. Specifically, for visual mapping missions, nearly half of the surface experiences its orbital night due to the self-shadowing, which limits the maximum coverage for a single observation to roughly 50%. For moons, this can fall to 0% at certain portions of the orbit when the planet eclipses the moon. Furthermore, most planetary moons are tidally locked to their central planet (Escribano et al., 2008; Quillen et al., 2020), which consequently shows the same portion of the moon’s surface for a fixed rendezvous location on its orbit. Additionally, a single large spacecraft may be susceptible to single-point failures. The swarm approaches (Bonabeau et al., 1999) can efficiently handle these challenges. The swarm approach decomposes a complicated task to simple tasks that can be efficiently handled by multiple, low-cost agents (Space Mission Directorate, 2020). The use of spacecraft swarms is also recognized by the National Academies of Sciences (National Academies of Sciences, 2016) as a high-priority goal to accomplish several science-based investigations. Due to their low-cost and improved efficiency, several swarm missions to small bodies are being developed. For instance, one of the advanced mission concepts is the Autonomous-Nano Technology Swarm (ANTS) developed by NASA, whose objective is to use a spacecraft swarm which cooperatively survey multiple NEAs (Curtis et al., 2003; Rouff, 2007). Also, a NASA Innovative Advanced Concepts (NIAC) proposal was
developed by the Applied Physics Laboratory at the John Hopkins University (JHU-APL), which aims to deploy a swarm of distributed optical sensors for gravimetric studies of an asteroid (Atchison, 2017). These mission concepts are illustrated in Figure 1.2.

Figure 1.2: Artistic rendering of future swarm-based mission concepts to small bodies: ANTS mission concept (left), and OpGrav (right).

However, the design of swarm missions faces several critical challenges. The design of a swarm mission is a multi-disciplinary problem as it involves the selection of multiple parameters such as the number of spacecraft, choice of the science payload, and seed spacecraft subsystems (Space Mission Directorate, 2020). Also, designing swarms that meet reconnaissance performance criteria can be an unintuitive process. This process is challenging due to the dynamical environments around small bodies such as lighting constraints, irregular shapes, small body rotation, and low-gravity conditions. Additionally, such missions require demonstration platforms to verify their feasibility. Without proper demonstration platforms, the actual swarm operations cannot be tested until deployment, which may result in space mission failures. This work addresses these challenges by automating the design of visual reconnaissance missions to small bodies. The Integrated Design Engineering and Automation of Swarms (IDEAS) software (Nallapu and Thangavelautham, 2020a)
was developed to automate the end-to-end design of a spacecraft swarm missions. The IDEAS framework uses Evolutionary Algorithms that generate mission concepts which can be unintuitive to human designers. The IDEAS is developed in the MATLAB programming environment because of its inbuilt state-of-the-art numerical propagators and optimizers (Hanselman and Littlefield, 2005). In the IDEAS framework, a mission concept design is handled by three automated design modules (shown in Figure 2.6): Automated Trajectory Design module, Automated Swarm Design module, and Automated Spacecraft Design module. Each of these modules optimizes their respective objective functions, and the Mission Analyzer module checks the collective validity of the design.

This thesis focuses on the development of the Automated Swarm Designer module of IDEAS, which generates candidate visual reconnaissance swarm architectures that meet the performance requirements using a minimum number of spacecraft. Specifically, in this work, the Automated Swarm Designer module is used to design case studies of near-optimal visual mapping missions, where the spacecraft meet spatial and temporal coverage performance missions from small body flybys. To account for the surface irregularities of the small body, a novel probabilistic method called the Dual Sphere method is developed, where the shape model is augmented with two spheres whose radii correspond to the maximum and minimum radius of the nominal shape model. The Dual Sphere method is used to develop a Monte-Carlo framework to design reconnaissance missions with spatial constraints. While the design algorithms assume perfect knowledge of the parameters involved, a real mission is subjected to modeling uncertainties and perturbations. For this reason, the sensitivity of the swarm coverage is examined by introducing random perturbations on the selected optimal design. The small bodies considered in the current work are small planetary moons that are in synchronous rotation with the central planets, and tumbling asteroids whose spin-axis changes with respect to time. While the focus of the current work is mainly on the swarm designer module, the interaction of the swarm design with their interplanetary trajectory and seed spacecraft is also demonstrated. Additionally, in this work, a hardware testbed is developed
to demonstrate multi-spacecraft reconnaissance algorithms. The Multi-Agent Photogrammetry of Small bodies (MAPS) testbed is used as a platform to implement a swarm solution obtained from the automated swarm designer module on a swarm of Unmanned Air Vehicles (UAVs), and demonstrate their operation by performing photogrammetric 3D surface reconstruction of a target small body mock-up encountered during their flyby trajectories. These contributions allow the spacecraft swarm technology to be accessible to a wider mission design community, and thereby result in efficient ways of exploring small bodies Space Mission Directorate (2020).

The organization of the remaining portion of this dissertation is as follows: Chapter 2 describes the mission concept design methodology using the IDEAS architecture. A detailed review of related work done in the field of spacecraft swarm technologies and the contributions of IDEAS to the existing literature is provided. Additionally, a novel classification of different swarm architectures based on the interaction between different spacecraft in the swarm is presented. Following this, the roles of different design modules of IDEAS and their implementations are described.

Chapter 3 develops the models for the visual reconnaissance mission concepts used in the current work. Specifically, linear transformations to evaluate the surface coverage of irregular small bodies from a spacecraft carrying an optical sensor are developed. These transformations are then integrated into the Dual Sphere method, where the coverage is stochastically evaluated over the nominal and Dual Sphere shape models of the target body. These algorithms are then integrated with the trajectories of a spacecraft swarm, to dynamically evaluate the spatial and temporal coverage over the selected small body. Additionally, algorithms to detect spacecraft collisions in a designed swarm architecture are presented.

Chapter 4 describes the algorithms to study the sensitivity of the coverage performance of the swarms to different perturbations. The design and dynamical perturbations on a given swarm architecture and describe their significance are described here. The models used in the current work to introduce these perturbations into the swarm architectures are then presented.

Chapter 5 describes the development of the Automated Swarm Designer mod-
ule of IDEAS to design a global surface mapping mission concept to a planetary moon through hyperbolic flybys of the central planet. A primary challenge here is that its interplanetary trajectory fixes the arrival direction of the incoming swarm hyperbole. Changes to this direction result in fuel-expensive maneuvers. Here, a numerical method to design the hyperbolic trajectories that encounter the planetary moon at a specified location is developed. The trajectory, swarm, and spacecraft design problems are then formulated. The algorithms are then demonstrated using a numerical case study of designing a global surface mapping mission concept to the Martian moon Phobos.

Chapter 6 describes the application of spacecraft swarms to both space-based and time-based missions to planetary moons. In comparison with Chapter 5, the spacecraft enter into co-orbits around the central planet while encountering the moon in a resonant manner. Here the design of resonant co-orbits and their coupling relations with the interplanetary trajectory of the swarm are developed. The swarm, trajectory, and spacecraft design problems are then formulated using the IDEAS framework. The algorithms are then demonstrated through numerical case studies of global surface mapping and region of interest (RoI) observation missions to the Martian moon Deimos.

Chapter 7 describes the development of the Automated Swarm Designer module of IDEAS to design reconnaissance mission concepts to tumbling asteroids. The target small body described here does not exhibit synchronous rotation, but a dynamical tumbling rotation. Additionally, the small body is directly under the gravitational influence of the Sun. To address this, a mother-daughter swarm architecture is developed, where the mothership deploys the spacecraft swarm on its heliocentric trajectory. The design of the swarm mission is then formulated using the IDEAS architecture, and the algorithms are demonstrated through a case study of a global surface mapping mission design to the asteroid 4179 Toutatis.

While Chapters 5-7 describe the design of spacecraft swarm missions, Chapter 8 presents the development of the MAPS testbed to demonstrate the multi-spacecraft mission concepts generated by the IDEAS framework. Here the MAPS testbed is
described in detail. This is followed by identifying the performance constraints on photogrammetric reconstruction using autonomous UAVs. These constraints are used to formulate the swarm design problem, which is solved using the Automated Swarm Designer module of IDEAS. A near-optimal solution identified by the Automated Swarm Designer module is implemented inside the MAPS testbed, and the surface reconstruction from its mapping operation is examined. Finally, Chapter 9 concludes the current work by summarizing the current work. The important contributions of the current work to the state-of-the-art are listed, and the pathways forward to advance IDEAS in becoming an end-to-end mission concept design tool are identified.
MISSION DESIGN FRAMEWORK OF IDEAS

Mission design to explore small bodies is highly desirable due to its promising returns to the science, economy, and space security community (Committee on the Planetary Science Decadal Survey, 2011; Space Mission Directorate, 2020). During the initial mission design phase, several decisions, such as the spacecraft subsystems, fuel requirements, and costs, are often unknown and are treated as free variables (Hirshorn et al., 2017). Traditionally, such design problems are handled by subject matter experts specializing in individual problem disciplines. However, these resulted in subsystem level optimal designs, which practically cannot be integrated. This approach can result in practical bottlenecks such as redesign and design duplication, thus resulting in extended mission life and costs. In extreme cases, such practices can also result in catastrophic mission failures. The Mars Climate Orbiter (see Figure 2.1) is an iconic example of such a mission failure, where a lander spacecraft crashed onto the Martian surface. The crash was caused because of the discrepancy of units using by its distributed systems engineering teams (Sauser et al., 2009). Specifically, its lander was designed to operate with imperial units of force, while the control commands were programmed in SI units (Board, 1999).

Traditionally, swarm missions are classified into two types of architectures: formation flying and constellations (Bandyopadhyay et al., 2016). Interplanetary swarm missions add another level of complexity as the interactions between the spacecraft complicate the design process. Specifically, mission tasks such as command and control, communication, telemetry, and tracking pose practical bottlenecks to an Earth-based ground station. These challenges often result in designs that are complex and large design spaces, making the mission concept design unintuitive to traditional systems engineering practices. Consequently, the software tools to design mission concepts are manifested as large volumes of coding (Wertz et al.,
These challenges can be overcome by developing multidisciplinary mission concept design tools that minimize the need for human intervention. Additionally, automating system design processes have been shown to reduce mission costs (Fukunaga et al., 1997). At the current state-of-the-art, well-established organizations have multidisciplinary teams that follow concurrent systems engineering practices to bridge the gaps in the design processes. The JPL Team X (Oberto, 2002) at the NASA Jet Propulsion Laboratory (JPL), Mission Design Center (Mauro, 2019) at the NASA Ames Research Center and Integrated Mission Design Center (Mason and Beaman, 2002) at the NASA Goddard Space Flight Center are examples of such multidisciplinary teams that practice concurrent design engineering. An automated mission concept design tool can also augment such human teams, thus leading to accelerated mission concept design (Wertz et al., 2011). The Integrated Design Engineering and Automation of Swarms (IDEAS) software is developed (Nallapu and Thangavelautham, 2019d) to provide such an automated design platform. The IDEAS framework provides a unified platform for multidisciplinary design and optimization of spacecraft swarms missions. It should be mentioned here that IDEAS follows a layered design architecture, where the different systems are sequentially
designed. The system models in IDEAS are stored in an inventory Knowledge Base module of IDEAS, which are then optimized using Evolutionary Algorithms. Specifically, the IDEAS allows for the simultaneous design of trajectory, swarm, and spacecraft systems. As mentioned previously, the current work focuses on the development of the swarm designer module of IDEAS, enabling its role as an end-to-end design system. This chapter describes the different modules of IDEAS and the flow of information between these modules, from modeling to design optimization.

2.1 Related Work

The field of space mission engineering develops mission parameters and refining requirements to meet the mission objectives on time at minimum cost and risk (Wertz, 2001; Space Mission Directorate, 2020). Traditionally, the initial design phase of a complex system, such as a space mission, progressed with specialized individual design teams, developing subsystem level optimal designs (Biffl et al., 2014). These designs often need to be redesigned from scratch at the time of integration due to their incompatibility (Biffl et al., 2019). In addition to delays in the integration, the subsystem level designs can also be prone to design duplication, where a design group duplicates the work of another design group (Baker, 1995; Lüder et al., 2018). This lack of a collaborative environment has also resulted in delayed design times, and therefore the total cost (Crowder et al., 2016). The advancement of space systems engineering has underscored the need for such a collaborative design tools (Kapurch, 2010; Space Mission Directorate, 2020).

Additionally, the development of model-based systems engineering (MBSE) promoted a single end-to-end design and analysis environment for all subsystem designs of a product (Friedenthal et al., 2007), as illustrated in Figure 2.2. However, the applications of MBSE principles in space systems is a new field of space systems research and has mainly been used to design Earth-orbiting spacecraft (Spangelo et al., 2013) and their constellations (Nag et al., 2016). The current work develops such an end-to-end, collaborative design framework to develop visual reconnaissance
mission concepts to small bodies. The developed framework encompasses novel tools for both the modeling and optimization aspects of the swarm mission concept design.

Swarm Architectures

Recognizing that multi-spacecraft missions can be designed with a wide spectrum of interactions, the spacecraft swarm architectures can be broadly classified into 5 classes (Nallapu and Thangavelautham, 2019b) which are illustrated in Figure 2.3, and are described as follows:

1. Class 0 Swarms: This is simply a collection of multiple spacecraft that exhibit no coordination either in movement, sensing, or communication.

2. Class 1 Swarms: Each spacecraft coordinates its movement resulting in formation flying, but there is no explicit communication coordination or sensing coordination.

3. Class 2 Swarms: Each spacecraft coordinates movement and communication, including using Multiple-Input-Multiple-Output (MIMO) or parallel channels.
The swarm has collective sensing capabilities but is not optimized with respect to the swarm.

4. Class 3 Swarms: Each spacecraft coordinates sensing/perception with communication and positioning/movement but is not collectively optimized. Individual losses can have uneven outcomes, including the total loss of the system.

5. Class 4 Swarms. Each spacecraft exploits concurrent coordination of positioning/movement, communication, and sensing to perform system-level optimization. This system acts if it is a single entity. Communication, computation, and sensing are evenly distributed within the swarm. Individual losses result in a gradual loss in system performance.

Class 0 swarms have been successfully realized for several Earth applications (Grewal et al., 2007). Constellations such as flower constellations (Mortari et al., 2004) are classified as Class 1 swarms due to the formations of their trajectories. Class 2 swarms are being designed for applications such as the deflection assessment of binary asteroids (Galvez et al., 2013; Atchison, 2017). Class 3 swarms have been used in interplanetary gravimetry applications (Zuber et al., 2013). A Class 4 swarm mission design is yet to be studied. Parametric mission concept design and mission cost modeling is well studied in the literature (Wertz et al., 2011). Existing research has also focused on the development of mission concept design tools for spacecraft swarms around Earth (Conn et al., 2017).

Modeling

One of the critical contributions of the current work is to develop models for coverage of spacecraft swarms from their flyby dynamics. The relevant work done in modeling spacecraft swarm behaviors and their coverage is presented below.

Spacecraft Dynamics The field of spacecraft dynamics is well studied in the literature. The two-body motion is the simplest model where the acceleration is
caused by a single, spherical, and gravitational source (Vallado, 2013). The motion fidelity is increased as additional sources of acceleration are included. Some frequently used perturbing accelerations in three-body (Valtonen and Karttunen, 2006), non-spherical nature of the source (Schmidt et al., 2007; D’Urso, 2014), and solar radiation pressure (Musen, 1960). Due to the computational simplicity, the two-body motion is often used in the trajectory design phase of the mission (Jones, 2016). When two-body motion is assumed, the trajectory design problem can be formulated as Lambert’s problem, where two boundary points and a desired time of flight are specified, and the terminal velocities need to be computed (Curtis, 2019). Several algorithms have been developed to solve Lambert’s problem to address issues such as speed (Russell, 2019), and multiple revolutions (Ochoa and Prussing, 1992). One of the most versatile solutions to Lambert’s problem was developed by Gooding, where the problem used the formulation of universal variables (Gooding, 1990). The method has been proved to be computationally efficient and accurate compared to several existing solvers (Izzo, 2015).

The effect of perturbative accelerations on the trajectory is then studied during the sensitivity analysis studies (Hallman, 1990). When studying the motion
of spacecraft swarms, Class 0 swarms were modeled using decentralized spacecraft
dynamics and control methods (Wertz, 2001). State-of-the-art class 0 swarm design
research has focused on the maximization of payload spatial and temporal coverage
(Cornara et al., 2001). Modern research on Class 4 formation flying swarms focuses
on guidance, navigation, and control (GNC) challenges such as the development of
control laws for formation maintenance, robustness, cooperation, and swarm nav-
igation. The first challenge is to maintain a required formation at times during a
depth space mission, simplifying communication, and Earth tracking under the pres-
ence of environmental perturbations (Morgan et al., 2012). Robust control laws to
maintain formation, prevent collisions (Lee et al., 2014) and delays (Nazari et al.,
2016) have been developed. Another challenge is the inherent nonlinearities devel-
oped in the dynamical modeling of the constituent spacecraft (Alfriend et al., 2009).
The attitude dynamics of tumbling and spinning bodies is well studied. Analyti-
cal solutions to attitude time evolution have been developed for rigid bodies under
torque-free motion (Schaub and Junkins, 2013). The motion of small bodies under
various perturbing torques has also been studied (Scheeres, 2016).

**Spacecraft Swarm Platforms** The requirement to become a cost-effective plat-
form for exploration may, at times, limit the capabilities of the constituent space-
craft (Agasid et al., 2015). To address these capabilities, one line of research has
focused on the development of hardware platforms to test swarm architectures (Na-
tional Academies of Sciences, 2016). With the advent of subsystem technology
for miniature spacecraft, such as CubeSats (Puig-Suari et al., 2001; Space Mission
Directorate, 2020), the feasibility of swarm architecture-based missions is rapidly in-
creasing. Currently, platforms such as Chipsats from Cornell (Weis, 2016), SunCube
FemtoSats from the University of Arizona (Mercedes et al., 2016) and silicon wafer
integrated Femtosats from JPL (Chung and Hadaegh, 2011) are being researched as
hardware platforms for swarm-based space exploration. The advancement of low-
cost small spacecraft technology has also lead to swarm missions populated with
small spacecraft (D’Arrigo and Santandrea, 2006; Wertz et al., 2011).
Payload Coverage Evaluation  Estimation of spacecraft camera coverage is a central aspect of the current work. Traditionally, the problem of spacecraft coverage was largely used in Earth-based applications. Earlier studies assumed that the projection of spacecraft’s field of view (FoV) traced a cone on the surface of a spherical Earth. This assumption enables the estimation of spacecraft coverage through analytical relations (Lüders and Ginsberg, 1974). The conical FoV assumption was later relaxed, and the surface coverage was modeled analytically using spherical trigonometry (Wertz, 2001). However, the target body was still assumed to be a sphere. Recent studies have developed the grid point coverage method, where the coverage of a grid point on the target surface is determined based on its spherical coordinates relative to the spacecraft camera (Dai et al., 2017). While the grid point method allows coverage computation over non-uniform bodies, it becomes computationally intensive when applied over a large number of grid points (Nallapu et al., 2018). However, this problem is well handled in the field of computer graphics using the homogeneous coordinate representation (Foley et al., 1994). In this method, the coverage is evaluated in a 4-dimensional space using the linear camera transformation matrix (Sobel, 1972). As a related application, the homogeneous coordinates are also used in the fields of SLAM (Blanco, 2008), and photogrammetry (Rudin et al., 2005).

Mission Concept Design Environments  At the current state-of-the-art several mission concept design tools are available. The Systems Tool Kit (STK) is one of the widely used mission concept design software (McNeil and Kelso, 2013). STK also allows the design of constellations using the grid point coverage method (Nallapu and Thangavelautham, 2019a). Other state-of-the-art mission concept design software includes Mission analysis, Operations, and Navigation Toolkit Environment (MONTE) developed by NASA-JPL (Evans et al., 2018), and General Mission Analysis Tool (GMAT) developed by NASA-GSFC (Jah et al., 2009). Existing research has also focused on the development of mission concept design tools for spacecraft swarms around Earth (Conn et al., 2017). While most design tools
provide robust platforms for modeling and visualization, mission optimization often needs integration with simulation environments such as MATLAB (Hanselman and Littlefield, 2005) and Python (Fraanje et al., 2016).

**Optimization**

Optimization forms an essential aspect of engineering system design. Optimization is the process of finding the variables that provide the maximum or minimum value of an objective cost function under a set of constraints. Depending on the nature of the variables, the objective function, and the constraints, optimization problems are categorized into several categories (Rao, 2019). Of specific interest to the current work is mixed-integer nonlinear programming (MINLP). The MINLP problems involve optimization of problems with at least one nonlinear objective function or constraints, and where both real and integer variables span the design space (Bussieck and Vigerske, 2010). Traditionally, algorithms to solve optimization problems are categorized into two types: direct search and indirect search (Rao, 2019). Indirect search methods identify the solution of an optimization problem by using derivatives of the objective function and the constraints (Luenberger et al., 1984). Because of this, they are also called gradient-based search methods. The inherent assumption here is that the optimization functions are continuous and differentiable (Powell, 1998). The Direct methods, on the other hand, find an optimal solution by only evaluating only the objective and constrain functions. The state-of-the-art research on optimization has focused on the use of Evolutionary Algorithms to solve complex optimization problems such as MINLP (Costa and Oliveira, 2001).

**Evolutionary Algorithms** Evolutionary Algorithms (EAs) use principles in nature and biology to solve their target problems, as illustrated in Figure 2.4. Specifically, operational principles from evolution (Gen and Lin, 2007), swarm intelligence (Blum and Merkle, 2008), and immune system response (De Castro and Timmis, 2002) have been mimicked to solve optimization problems. They are characterized by three important key features: i. They are population-based. ii. They are
Figure 2.4: Illustration of Evolutionary Algorithms, showing design principles in nature (top), and their application in artificial design (bottom).

fitness-oriented, and iii. They are driven by validation (Yu and Gen, 2010). Some notable evolutionary optimization algorithms include Genetic Algorithms (Conn et al., 1991), Particle Swarm Optimization (Poli et al., 2007), Immune System Algorithm (Coello and Cortés, 2005). Due to their simplicity and ease of implementation, the GA solvers has been one of the versatile tools to solve optimization problem (Rao, 2019). Because of its relative ease of handling MINLP problems, the GA solver is used as the core optimizer in IDEAS.

The Genetic Algorithms were first introduced by John Holland in 1975 in his book ”Adaptation in Natural and Artificial Systems” (Holland et al., 1992). However, the development of similar ideas can be attributed to an earlier time. In the 1960s, German scientists Ingo Rechenberg (Rechenberg, 1971) and Hans-Paul Schwefel (Schwefel, 1977) developed the concept of the Evolutionary Strategies. Around the same time, Hans Bremermann Bremermann (1958), David Fogel (Fogel, 1998) and others from the United States of America implemented Evolutionary
Programming. These ideas developed the use of selection, and mutation, which formed the core of the Darwinian theory of evolution (Darwin et al., 1958). Despite their applications, the Genetic Algorithms was dormant until the 1980s. Their need for computational power is attributed as one of the reasons for such a phase of dormancy (Reeves and Rowe, 2002). The doctoral thesis of Kenneth De Jong, a student of John Holland, presented the first use of GAs to solve optimization problems (De Jong, 1975). This was followed by David Golberg, another doctoral student of John Holland, applying GAs for gas pipeline optimization (Goldberg, 1984). In 1989, Goldberg published the book "Genetic Algorithms in Search, Optimization, and Machine Learning" which allowed GAs to be applied to a diverse set of real-world problems (Goldenberg, 1989). A closely related Evolutionary Algorithm is that of Genetic Programming, which was developed by John Koza in 1990 (Koza, 1990). While traditionally, the Genetic Algorithms assume a fixed design architecture, the Genetic Programming framework also tries to learn the design architecture of the problem (Koza, 1994). While these Evolutionary Algorithms are frequently applied in solving optimization problems, their scope extends much beyond functional optimization (Reeves and Rowe, 2002). Specifically, the GAs have been used in problems such as control design and systems identification (Kristinsson and Dumont, 1992). In his doctoral thesis, Jekanthan Thangavelautham demonstrated that GAs could be applied to control design problems by evolving neural network-based controllers that deliver human-competitive performance in robotic tasks (Thangavelautham, 2008).

When using the Genetic Algorithms, the objective function is referred to as the fitness function, and the set of design variables constitute a decision vector known as the design gene (Rao, 2019). At the start of a GA optimization trial, a pool of design genes is created based on a user defined population size. Some practical implementations create $2N$ genes in the first generation, if a population size of $N$ is specified (Conn et al., 1991). The subsequent generations are then iteratively created based on the populations of the previous generation. In a given population, there are two classes of genes: elites and children. The elite genes are
those that had the highest fitness in the previous generation (Musnjak and Golub, 2004). An elite count ratio is used to specify the proportion of elite genes in a given generation. The children form the remaining portion of the population, in a generation, by transforming the non-elite population in the previous generation with two operations: crossover and mutation (Costa and Oliveira, 2001). The crossover combines a set of two parent genes from a previous generation to form a child. The mutation, on the other hand, subjects a single parent to random changes to its design structure (Hong et al., 2002). The crossover fraction defines the proportion of the non-elite population in the generation that is obtained from the crossover. The remaining children are obtained through the mutation operation. The evolution of the three different types of children in the population of a GA generation is illustrated in Figure 2.5.

![Figure 2.5: The different types of child genes in the population of GA generation.](image)

The parent genes of crossover and mutation are selected into a mating pool based on a selection algorithm. The parents in the mating pool are then subjected to crossover and mutation algorithms to create the non-elite population in the next generation. Several crossover and mutation algorithms have been developed in the literature, which allows a user to customize the optimization solver (Vasconcelos et al., 2001). Since the current work employs the GA solver available in the MAT-
LAB environment, its selection, crossover, and mutation operations are of key interest here (Deep et al., 2009), for mixed-integer problems. The Tournament selection algorithm has been shown to have better convergence and computational efficiency in the case of mixed-integer variables (Goldberg and Deb, 1991). In the Tournament selection, a set of $k$ design genes is randomly selected, of which the gene with the best fitness is sent to the mating pool (Coello and Montes, 2002). Special operations such as Laplace crossover (Deep and Thakur, 2007a), and Power mutation (Deep and Thakur, 2007b) have been developed with integer modifications. The stall stop criterion is used as a standard search termination criterion, i.e., if the value of the best fitness stalls for a predefined number of generations, then terminate the search is terminated, and the corresponding solutions are retrieved (Samanta, 2014). If the specified number of stall generations is sufficiently large, the final generation can contain more than one gene that produces the best fitness. Here, a candidate solution can be identified either by random selection or through a figure of merit (FoM) analysis. It should be noted that while the GA is categorized as a global search method, its convergence to a global optimal solution, during an optimization trial, cannot be guaranteed (Liepins, 1992). However, running multiple optimizer trials can provide statistical reasoning on global optimality. Due to its highly customizable nature, GA has been used in the design optimization of several applications, while providing counter-intuitive solutions. Notable examples in the field of spacecraft systems include: communications system design (Lohn et al., 2004), trajectory optimization (Cage et al., 1994), CubeSat power system design (Kalita and Thangavelautham, 2016), and many more. Genetic Algorithms have also been used to design coverage optimal constellations (Ely et al., 1999; Confessore et al., 2001).

Multi-disciplinary optimization A subfield of optimization that is closely related to the current work is multidisciplinary design optimization (MDO). MDO approaches use numerical optimization strategies to design systems that involve multiple disciplines. MDO is motivated by the observation that a multidisciplinary design problem is not only driven by the performance of an individual system but
also on the interactions between other constituent subsystems. Despite their heavy computational requirements, solving MDO problems earlier in the design process is expected to achieve reduced time and costs associated with the design cycle (Martins and Lambe, 2013). One of the earliest applications of MDO was published in 1960 by Lucien Schmit. Here Schmit developed a multidisciplinary design tool for structural optimization (Schmit, 1960). In 1975, Raphael Haftka developed an automated multidisciplinary tool for the design of aerodynamics, control, and structural subsystems were required (Haftka, 1977). Since then, MDO tools have been developed for a diverse set of design problems such as aircraft (Kroo et al., 1994), rotorcraft (Ganguli, 2004), spacecraft (Braun et al., 1997), automobiles (Kokkolaras et al., 2004), ships (Peri and Campana, 2003), and buildings (Choudhary et al., 2005). The architecture of the MDO problem describes the coupling and solving methodology. These architectures can be either monolithic or distributed (Martins and Lambe, 2013). The monolithic architecture was formalized by Evin Cramer, where the MDO solver handles a single optimization problem (Cramer et al., 1994). This was followed by Balling and Sobieszczanski-Sobieski (1996) identifying several approaches to handle monolithic architectures. The distributed architectures, on the other hand, partition the problem into smaller problems that handle a subset of the design variables and constraints. The convergence of solutions of a distributed architecture has been studied by Alexandrov et al. (1997) in their collection of articles titled "Multidisciplinary Design Optimization: State of the Art." This collection also featured a review of MDO approaches for both monolithic and distributed architectures up to that time (Kroo, 1997). Since then, several surveys of MDO approaches have been periodically published regularly where new architectures and algorithms have been presented (Sobieszczanski-Sobieski and Haftka, 1997; Tosserams et al., 2009; Martins and Lambe, 2013). The choice of architecture and optimization algorithm is typically decided by the developer and on the design problem. However, these choices have a direct impact on the quality and computation runtimes of the solution. For instance, using gradient-based methods can lead to faster computations, while at the risk of presenting local optimal solutions. Since its inception,
several MDO methods for solving single-objective functions with continuous design variables have been developed (Martins and Lambe, 2013). These methods use the Karush-Kuhn-Tucker conditions (Rao, 2019) to verify the optimality of the developed solution. However, modern research has also developed specialized tools to solve multi-objective functions (McAllister and Simpson, 2003) and discontinuous variables (Haftka and Watson, 2006), through the use of global optimization algorithms like the Genetic Algorithms (Rao, 2019). At the state-of-the-art, several MDO tools are available. OpenMDAO is an opensource solution developed by NASA-GR (Gray et al., 2019) that uses gradient-based search methods to solve its MDO problems. MDO problems have also been applied to space system design problems using Evolutionary Algorithms (Kalita, 2020).

Development of IDEAS

This thesis uses MDO principles to design the trajectory, spacecraft, and swarm behaviors of IDEAS. Our previous work on IDEAS focused on developing the Automated Swarm Designer module of IDEAS, where case studies of swarm missions to high priority targets such as moons of Mars, asteroids, cis-lunar space, and Earth observation (Space Mission Directorate, 2020), using Genetic Algorithm optimization were presented. The developed case studies entail Class 0 swarms for Earth observation (Nallapu and Thangavelautham, 2019a) and cis-lunar communications (Nallapu et al., 2020a); Class 1 swarms for global surface mapping of spinning asteroids were also designed (Nallapu and Thangavelautham, 2019d); Class 2 swarms for RoI observation on spinning asteroids (Nallapu and Thangavelautham, 2019b), global surface mapping of tumbling asteroids (Nallapu et al., 2020b). Case studies to planetary moons include global surface through hyperbolic flybys (Nallapu and Thangavelautham, 2020b), and polar resonant co-orbits (Nallapu and Thangavelautham, 2019c, 2020a).
2.2 The IDEAS Architecture

The IDEAS framework architecture divides a swarm mission concept design into three sub-design problems: trajectory, swarm, and spacecraft design. The architecture of the IDEAS software is presented in Figure 2.6. As shown here, the three individual modules: Automated Trajectory Designer, Automated Swarm Designer, and Automated Spacecraft Designer modules will form the Mission Solver module where the three design processes are automated. The design modules receive high-level inputs such as objectives, constraints, and mission parameters through a user interface. The individual sub-components of IDEAS are defined as follows:

Figure 2.6: Software architecture of the proposed IDEAS software to provide an end-to-end design framework for spacecraft swarm missions.

Knowledge Base

The IDEAS architecture uses the Knowledge Base to store the model-related information. The specific models corresponding to system design (trajectory, spacecraft, and swarm) will be referred to as behaviors. An inventory of different trajectory,
spacecraft, and swarm behaviors is maintained in the corresponding sub-modules of the Knowledge Base.

**Trajectory Design Knowledge**  The trajectory design module of IDEAS designs the interplanetary trajectory of the swarm that is launched from Earth and arrives at the target. The design space of trajectory varies with the type of trajectory selected. As an example, the design variables involved in the design of hyperbolic flyby trajectories at the destination planet are different from the trajectories that must be captured into an orbit (Vallado, 2013). Therefore the different trajectory design gene structures and models used to evaluate these genes are stored in the Trajectory Basis Behavior module of the Knowledgebase.

For this reason, a zero-revolution Lambert’s solver, which uses the Gooding’s algorithm, is used to design the interplanetary spacecraft trajectory (Gooding, 1990). It should be noted here that while the focus of the current work is the design of swarm missions, only a single pair of launch and arrival epochs is noted by solving Lambert’s problem. This supports two swarm deployment configurations: a swarm directly deployed at launch from a single launch provider (Schoolcraft et al., 2016), and a swarm that is deployed from a mother spacecraft during its interplanetary cruise (Vance et al., 2019).

**Spacecraft Design Knowledge**  Spacecraft knowledge corresponds to the design of seed spacecraft in the swarm. The design includes problems such as selecting the spacecraft bus, payload and estimating its total mass and cost. An inventory of different behaviors corresponding to each of these is maintained in the spacecraft design knowledge module. A fixed set of dry mass subsystems for the seed spacecraft is used here. The solution of the trajectory and swarm design problems are used to estimate the fuel requirements of the seed spacecraft and thus used to estimate its cost. The seed spacecraft subsystems and behaviors used in the current work are described in Appendix A.
Swarm Design Knowledge  The swarm design problem requires the knowledge of several parameters. These include the design variables in the swarm gene, the swarm architecture, their reconnaissance configurations, and attitudes. An inventory of these behaviors is maintained in the swarm basis behavior inventory. Since the swarm behavior varies with the type of mission selected, the attributes corresponding to different behaviors will be detailed in the corresponding case studies.

Knowledge Generator  The Knowledge Generator module is used to populate the behavior inventories of the Knowledge Base. In cases where existing models are not available for a particular design, artificial models can also be generated by the use of machine learning algorithms. For instance, if a specific spacecraft subsystem information is not available, methods such as neural networks can be used to determine this from existing information on other relevant subsystems. The primary purpose of the Knowledge Generator module is to keep the knowledge inventory current and increase the potential applications of IDEAS.

Figure 2.7: The sequential exchange of information across the different modules of the IDEAS architecture.
Mission Solver

The Mission Solver module is the most computationally intensive module of IDEAS, where the individual designs are optimized. As seen in Figure 2, the Mission Solver module encompasses the three automated designer modules, corresponding to the trajectory, swarm, and spacecraft optimization based on their behavioral attributes. Each automated designer module uses Genetic Algorithms to solve their appropriate constrained optimization problems, as shown in Figure 2.7. The feasibility of the solutions obtained from the different designer modules is analyzed in the Mission Analyzer module. The primary purpose of the Mission Analyzer module is to select a feasible design solution. When multiple solutions exist, the Mission Analyzer implements an FoM based selection scheme specified by the user. When no feasible solutions are found, the Mission Analyzer iterates through the individual design optimization modules.

User Interfaces

The user can access the Mission Solver module of IDEAS through graphical user interfaces (GUI). As such, two GUIs are crucial for user interactions: User Input GUI, and the Results GUI.

Input Interface The User input interface is used to specify the high-level requirements and simulation parameters. The simulation parameters include information such as trajectory, swarm, and spacecraft genes; Information on the target small body; and the FoM selection scheme to select a candidate solution in case of multiple solutions. In addition to these, behavior parameters such as model specifiers and constrain parameters, and optimization parameters such as the number of genes, stall criterion parameters, elite count, and cross over ratios are also supplied through the input interface.

Result Interface When the design requirements are met, the results from IDEAS can be collected at the Results interface. In the current work, the performance of
the final generation genes supplied by the Mission Solver module is examined using an FoM analysis. A near-optimal design is selected from the final generation design solutions in the Mission Analyzer module. The swarm and trajectory performance of the selected design genes is analyzed, and the net seed spacecraft and swarm mission costs are estimated. Finally, the selected near-optimal design is subjected to a sensitivity analysis, where its coverage performance is studied under different categorical perturbations.

**Current Implementation**

The exchange of information across the different modules of IDEAS in the current implementation is summarized in Figure 2.7. In its current implementation, the IDEAS is developed in the MATLAB programming environment because of its in-built state-of-the-art numerical propagation and optimization routines (Hanselman and Littlefield, 2005). All simulations were run on a high-performance computer cluster with a 2.3 GHz Intel Xeon Processor and were executed in a parallelized architecture with 28 processor cores. The main contribution of the current work is on the development of swarm design components of IDEAS: specifically, the swarm knowledgebase and the Automated Swarm Design modules while demonstrating the interaction between the trajectory and spacecraft design modules. Additionally, in the current work, trajectory design problems, whose heliocentric trajectories are modeled as Lambert arcs, are solved. The arrival information of the Lambert arc is used to design the swarm operations near the target small body. For this reason, the optimization of trajectory and swarm in their appropriate design modules is examined. The Spacecraft designer module provides the payload parameters to design the reconnaissance missions. Finally, any maneuver costs noted from the trajectory and swarm design problems are used to estimate the fuel requirements and space mission cost.
This chapter develops models for the visual reconnaissance mission concepts described in this thesis. The developed models are used to specify two significant constraints on the swarm operations: stochastic coverage requirements and collision avoidance. These models are programmed into the Automated Swarm Designer module of IDEAS to generate and identify near-optimal mission concepts. We begin this chapter by identifying coverage requirements of spatial and temporal reconnaissance missions to small bodies. This is followed by developing algorithms to model surface coverage of a small body shape model, along with algorithms to detect spacecraft collisions. Then, a stochastic method of evaluating small body surface coverage, known as the Dual Sphere method, is developed. Finally, this chapter is concluded by extending the Dual Sphere method to design spatial and temporal reconnaissance missions to different small bodies considered in this thesis.

3.1 Reconnaissance Mission Concepts

This thesis focuses on designing two categories of visual reconnaissance mission concepts, which are listed below:

1. Global surface mapping

2. Region of interest (RoI) observations

The global surface mapping swarms observe a large portion of the target body, focusing on the spatial coverage capability of swarms. The region of interest (RoI) mapping swarms, on the other hand, observe a target region on the surface of the target body for a required amount of time, focusing on the temporal coverage aspect
of swarms. A key challenge here is to design swarms whose coverage performance is robust to the orientation of the target body. The coverage requirements of both missions are described as follows:

**Global Surface Mapping**

Global surface mapping swarms are required to meet a minimum required percentage coverage of $P_{\text{map},R}$ of the target body.

**RoI Observation**

The RoI observation swarms are required to observe a target RoI, such that the percentage coverage of the RoI should exceed $P_{\text{RoI},R}$, while the coverage duration of the RoI should exceed $T_{\text{RoI},R}$. In both missions, only observations with a maximum ground resolution of $x_D$ are considered for coverage evaluation.

**Visual Mapping Requirements** For a given spacecraft camera, the maximum flyby altitude $h_{\text{max}}$ to meet the $x_R$ requirement, and half-field of view of the camera $\eta_C$ are computed as (Wertz et al., 2011)

$$h_{\text{max}} = \frac{x_R D_C}{\lambda_C}$$  \hspace{1cm} (3.1)

and

$$\sin \eta_C = \left( \frac{R_{T,\text{max}}}{r_{\text{max}}} \right) \cos \epsilon_f$$  \hspace{1cm} (3.2)

Where $D_C$ and $\lambda_C$ are the aperture diameter of the camera and imaging wavelength of the camera sensor, respectively. The parameter $\epsilon_f$ indicates a slant angle tolerance at the imaging radius $r_{\text{max}}$ given by

$$r_{\text{max}} = R_{T,\text{min}} + h_{\text{max}}$$  \hspace{1cm} (3.3)
Where \( R_{T,max} \) and \( R_{T,min} \) are the maximum and minimum radii of the irregular target shape model. Using an altitude tolerance \( \Delta h_f \), the spacecraft will pass the target at a radial distance \( r_f \) given by

\[
 r_f = r_{max} - \Delta h_f \tag{3.4}
\]

It is noted here that \( r_f \) is not the minimum distance of the spacecraft to the target, but is a point on the spacecraft trajectory where the visual coverage of the spacecraft meets the \( x_R \) requirement. The camera of the spacecraft is modeled as a pinhole camera with a square sensor (Sobel, 1972), whose pyramidal field of view (FoV) is used to model the instantaneous footprint of the camera on the surface of the target. The geometrical and camera parameters described in Equations 3.1 to 3.4 are illustrated in Figure 3.1

![Figure 3.1: Geometrical and camera parameters used in designing the flyby encounters of the target body](image)

### 3.2 Performance Evaluation of Mission Concepts

As described above, the design of the swarm mission concepts occurs inside the Automated Swarm Designer module of IDEAS. Figure 3.2 presents the different
processes that occur inside the Automated Swarm Designer module. The Automated Swarm Designer module evaluates the performance of the generated swarm designs, hereby known as genes, using the basis behaviors programmed into the Knowledge Base module. The dynamics of the generated genes are simulated to estimate the performance indicators, such as the coverage figures of merit (FoM) and collision flags. This section develops the mathematical models for evaluating the coverage FoMs, and for detecting spacecraft collisions in a given swarm gene.

Figure 3.2: Process diagram showing the different operations that occur inside the Automated Swarm Designer module of IDEAS.

Coverage Evaluation

The raw shape model of the target body is described by a unique set of vertices $V_T$, and triangular face connectivity set $F_T$ (Scheeres, 2016). To provide a generalized description of the coverage evaluation for both missions, consider the coverage of sub-region described by faces $F_{TR} \subseteq F_T$. The vertices in $V_T$ are assumed to be expressed in a target body-fixed frame $T$ frame, where the instantaneous rotational transformation from the reference frame of the raw data $R$ to the $T$ frame is denoted by $[TR]$. A series of 3 operations on $V_T$ and $F_{TR}$ are developed here to estimate the
instantaneous and cumulative coverage by the spacecraft’s camera: illumination, LoS culling, and clipping. The three filters are described as follows.

**Illumination** At a time $t$, the set of faces illuminated by the Sun are noted using the inner product between the target-to-Sun direction $T \hat{R}_{TH}$ and the normal vector to each face. If $T \hat{n}_{k,TR}$ denotes the instantaneous normal vector of face $k$ in $F_{TR}$, the face is considered illuminated if

$$N \hat{R}_{TH} \cdot T \hat{n}_{k,TR} > 0.$$

(3.5)

The illumination filter is illustrated in Figure 3.3. As seen here, this operation filters out the set of all illuminated faces $F_i(t) \subset F_{TR}$ at $t$. The process diagram describing the evaluation of the instantaneous illuminated faces in a given shape model is presented in Figure 3.4.

![Figure 3.3: Illustration of the illumination operation which filters the faces of the model facing the Sun.](image)

The illumination filter identifies faces of an irregularly shaped lit by the Sun at any given instant.

**Eclipses** An exception to the above rule occurs when the planet eclipses the Sun. When this happens, no faces will be observed. An instantaneous flag variable
is used to check if the moon is undergoing an eclipse, based on the phase angle between the planet and the moon. When the planet-to-moon angle subtended at the Sun $\phi_{PM}$ is greater than the angular diameter of the planet $\theta_{P,D}$ from the Sun, and when the distances from the Sun to the planet and if the planet distance from the Sun is smaller than the moon’s distance, the moon will be eclipsed entirely, and no portion of its surface will be visible, as illustrated in Figure 3.4.

![Figure 3.4: Process diagrams for evaluating illuminated shape models (left), and detecting planetary moon eclipses (right).](image)

**LoS Culling** The faces in $F_t(t)$ are culled with respect to the line of sight (LoS) from spacecraft to the specified point on the target. The culling operation filters out the illuminated faces that lie towards the side of the spacecraft’s camera, as illustrated in Figure 3.5. Let $T_i \vec{R}$ denote the position vector of spacecraft $i$ with
respect to target, and $\nabla n_{k,l}$ denote the normal vector of face $k$ in $F_i(t)$ at $t$. The culling operation eliminates the faces of the shape model that satisfy

$$T_n \overline{R}_{T_i} \leq 0$$

(3.6)

Figure 3.5: Illustration of the culling operation which filters the faces of the model facing the spacecraft.

The software implementation of the culling operation is illustrated in Figure 3.6. As seen here, faces that satisfy Equation 3.6 will not be observed by the spacecraft. This operation further filters the set of illuminated faces that face the spacecraft $i$ $F_{C,i}(t) \subseteq F_i(t)$.

The culling filter identifies illuminated faces of the which lie in the direction of the spacecraft.

**Clipping** While the above two operations filter the faces, the clipping operation filters the vertices of $F_{C,i}$, which fall inside the FoV of spacecraft $i$, as illustrated in Figure 3.7. In the clipping operation, we construct the camera transformation matrix, which transforms the FoV into a unit cube (Sobel, 1972). The instantaneous camera transformation matrix $G_i$ into the image space of spacecraft $i$ is given by
Figure 3.6: Process diagram illustrating the software implementation of the culling operation.

\[ G_i = VT_1T_2 \]  \hspace{1cm} (3.7)

where

\[ T_1 = \begin{bmatrix} [RB_i] & 0 \\ \hline 0 & 1 \end{bmatrix}_{4 \times 4} \]  \hspace{1cm} (3.8)

\[ T_2 = \begin{bmatrix} [I_{3 \times 3}] & -\bar{R}_{T_i} \\ \hline 0 & 1 \end{bmatrix}_{4 \times 4} \]  \hspace{1cm} (3.9)

and
Figure 3.7: Illustration of the clipping operation which filters the vertices of the model that fall inside the spacecraft’s FoV.

\[ V = \begin{bmatrix} \cot \eta_C & 0 & 0 & 0 \\ 0 & \cot \eta_C & 0 & 0 \\ 0 & 0 & \frac{fa_C + ne_C}{fa_C - ne_C} & \frac{-2fa_C ne_C}{fa_C - ne_C} \\ 0 & 0 & 1 & 0 \end{bmatrix}_{4\times4} \] (3.10)

The matrix \([RB_i]\) is the rotation matrix that transforms the body frame of spacecraft \(i\) to the reference LoS tracking attitude, and \([I_{3\times3}]\) is the \(3 \times 3\) identity matrix. The LoS tracking reference attitude described in Tsiontras et al. (2001) is used to model \([RB_i]\). The parameters \(ne_C\) and \(fa_C\) indicate the near field and far field distance of the spacecraft’s camera. Here, we set \(fa_C = h_{\text{max}}\), and select an arbitrary small value of \(ne_C\) for the spacecraft camera. Let \([N_a N_b N_c]^T\) denote the position vector of a vertex that forms at least one triangular face in \(F_{C,i}(t)\). The vertex is resolved in an intermediate camera space \(Ci\) of spacecraft \(i\) by using the camera transformation.

The clipping filter identifies faces of an irregularly shaped target body which fall inside the pyramidal FoV of the spacecraft camera.

\[
\begin{bmatrix} C_i^a & C_i^b & C_i^c & C_i^d \end{bmatrix}^T = G \begin{bmatrix} N_a & N_b & N_c & 1 \end{bmatrix}^T
\] (3.11)
Where $C_i d$ is the apparent depth of the projection. The camera space is normalized by $I_i d$ to obtain the scaled coordinate in the image frame $I_i$ of spacecraft $i$, i.e.,

$$
\begin{bmatrix}
I_i a \\
I_i b \\
I_i c
\end{bmatrix}^T = \frac{1}{C_i d} \begin{bmatrix}
C_i a \\
C_i b \\
C_i c
\end{bmatrix}^T
$$

(3.12)

The vertex $[N_a \ N_b \ N_c]^T$ will fall inside the FoV of spacecraft $i$ if the absolute value of its coordinates lies in a unit cube (Sobel, 1972).

$$
\text{abs} \left( \begin{bmatrix}
I_i a \\
I_i b \\
I_i c
\end{bmatrix}^T \right) < \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T
$$

(3.13)

If a vertex fails the clipping criterion in Equation 3.13, all faces in $F_{C,i}(t)$ that are formed by it are omitted from the coverage computation. The final result of these three filters is a set of faces $F_{FoV,i}(t)$ whose faces are illuminated by the Sun, and whose vertices fall inside the FoV of spacecraft $i$ at a time $t$. The clipping operation is summarized by the process diagram described in Figure 3.8. As seen here, after performing the clipping operation, the instantaneous coverage of a spacecraft $i$ given by $F_{FoV,i}(t)$ is obtained.

**Instantaneous Coverage** At time $t$, the set of faces observed $F_{Ins}(t)$ is computed as

$$
F_{Ins}(t) = \bigcup_{i=1}^{N_{Sw}} F_{FoV,i}(t)
$$

(3.14)

Where $N_{Sw}$ is the number of total spacecraft in the swarm. To reduce the computational load, only those spacecraft whose distance to the target is below $r_{max}$ are used in the computation of Equation 3.14. The instantaneous coverage
Figure 3.8: Process diagram illustrating the software implementation of the clipping operation.

evaluation algorithm illustrating the role of the three operations described above is presented in Figure 3.9.

**Cumulative Coverage** Similarly, we can note the cumulative faces $F_{Cum}(t_{sim})$ observed up to the simulation time $t_{sim}$ as

$$F_{Cum}(t_{sim}) = \bigcup_{t=0}^{t_{sim}} F_{Ins}(t)$$

(3.15)

The area of surface described by faces in $F_{Cum}(t_{sim})$ is the sum of areas of their triangular areas (Goldman, 1991). Let $A_{Cum}(t_{sim})$ and $A_{TR}$ denote the cumulative surface area of the faces $F_{Cum}(t_{sim})$ and $F_{TR}$ respectively. Then the figure of merit considered in the current work is the cumulative percentage of surface coverage given by,
Figure 3.9: Process diagram describing the role of the different operations used to estimate the instantaneous surface coverage of the swarm.

\[ P_{Cum}(t_{sim}) = \frac{A_{Cum}(t_{sim})}{A_{TR}} \times 100 \quad (3.16) \]

The final values of \( P_{Cum} \) and observation duration are from each simulation to evaluate the surface coverage of the shape model. The coverage evaluation algorithm to determine the spatial \( P_{Cov} \) and temporal \( T_{Cov} \) coverage of the swarm during a given simulation is presented in Figure 3.10.

### 3.3 Dynamical Parameters

It can be seen here from Figure 3.10, that the surface coverage evaluation depends on the dynamical states of the target body and the spacecraft in the swarm. Specifically,
the states of interest are the position of the spacecraft with respect to the target $T\vec{R}_{Ti}$, the attitude of the target $[T\vec{R}]$, and attitude of observer spacecraft $[TB_i]$. Additionally, when designing missions to moons, the heliocentric location of the planet $T\vec{R}_P$ is also required to compute the moon-to-Sun vector. The models for these parameters are described as follows:

**Translatory Motion**

For computational simplicity, only two body gravitational motion (Vallado, 2013) is considered when evaluating coverage in the design optimization modules of IDEAS. Let $N$ be the inertial frame centered at the central source of gravity, and $^N\vec{R}_T$ and $^N\vec{R}_i$ be the position vectors of the target and spacecraft $i$ with-respect to the central body, the evolution of these position vectors is then given by
\[
N \ddot{R}_T = - \frac{\mu}{|N \vec{R}_T|^3} N \vec{R}_T \\
N \ddot{R}_i = - \frac{\mu}{|N \vec{R}_i|^3} N \vec{R}_i
\] (3.17)

Where \(\mu\) is the gravitational parameter of the central source. When designing mission concepts to asteroids, the central source is assumed to be the Sun, while it is assumed to be the central planet when designing missions to moons. The initial condition to propagate \(N \vec{R}_i\) will be selected from the swarm basis behavior inventory described in the individual case studies, along with the initial conditions for \(N \vec{R}_T\).

The motion of the planet is modeled using a Julian epoch based ephemeris of planet (Vallado, 2013). Assuming that the \(T\) frame is related to the \(N\) frame by a pure translation, the position of spacecraft \(i\) with respect to the target can be expressed as

\[
T \vec{R}_{Ti} = N \vec{R}_i - N \vec{R}_T
\] (3.18)

Collision Avoidance

The spacecraft trajectories shall be designed to be ballistically free of collisions using a binary collision flag parameter. Specifically, we check if the trajectories: i. collide with the target, and ii. collide with other spacecraft in the swarm. The target and spacecraft collisions are checked as follows:

**Target Collisions** By propagating the orbital motion of the spacecraft, we note the magnitude of the distance of spacecraft \(i\) to the target \(|N \vec{R}_{Ti}|\) throughout the simulation time. We require that

\[
|N \vec{R}_{Ti}| > r_{T,\text{max}}
\] (3.19)
Figure 3.11: Illustration of a swarm design which contains a potential collision with the target small body.

Where $i = 1, 2, \ldots, N_{Sw}$, in the time span $[0, T_{sim}]$. Here $r_{T,max}$ is the maximum radius of the target shape model. Figure 3.11 presents an illustration of a swarm design which is considered as a spacecraft-target collision check presented in Equation 3.19. As seen here, the target collision check described in Equation 3.19, conservatively identifies those swarm designs where the spacecraft potentially collide with the target body.

**Spacecraft Collisions** Using a similar approach, if $r_{i,l}$ denotes the magnitude of position vector from spacecraft $i$ to spacecraft $l$, we require that

$$r_{il} > 2 \ r_{col}$$

for all $i = 1, 2, \ldots, N_{Sw}$ and $i \neq l$, in the complete simulated time span $[0, T_{sim}]$. Here, $r_{col}$ indicates a collision radius around each spacecraft. Figure 3.12 presents an illustration of a swarm that contains trajectories which result in a potential collision among the participating spacecraft as described by Equation 3.20. As seen here, the spacecraft collision check described in Equation 3.20, conservatively identify those swarm designs where the spacecraft potentially collide amongst themselves.

The collision flag is set to 0 if all spacecraft co-orbits are free of the moon and spacecraft collisions and is 1 otherwise.
Figure 3.12: Illustration of a swarm design which contains a potential spacecraft with the target small body.

**Attitude**

Here the models used in the current work to describe the spacecraft attitude $[TB_i]$ and target body attitude $[TR]$ are presented.

**Spacecraft** The spacecraft are assumed to exhibit Class 2 behaviors where the spacecraft are classified as Leaders and Observers. In the current work, the spacecraft in the Class 2 swarm coordinate for communication. As a result, each spacecraft in the swarm will have two modes of operation: mapping mode and the communications mode. In the mapping mode, all spacecraft cameras orient along the LoS to $F_{TR}$ when $|TR_i| \leq r_{max}$ as illustrated in Figure 3.13.

Once the spacecraft move outside $r_{max}$, they enter their communication mode, where the Observer spacecraft orient their antenna towards the leader, while the Leader spacecraft orients its antenna towards Earth, as illustrated in Figure 3.14.

The rotation matrix that transforms its body frame to a reference line of sight (LoS) tracking attitude is derived here to provide a generalized description of the spacecraft attitude. The spacecraft is assumed to be a rigid body with both its camera and communications antenna placed along its body $z$ axis. The solar panels along the $y$ axis, as illustrated in Figure 3.13. Let a spacecraft be located at $N\hat{R}_i$ in the $N$ frame. The reference attitude which orients $\hat{z}$ along the LoS to $N\hat{R}_Q$, while
Figure 3.13: The mapping behavior of the spacecraft in the designed Class 2 swarms. Trying to orient \( \hat{y} \) along \( \vec{R}_H \) (Tsiontras et al., 2001) is given by

\[
\begin{align*}
N\hat{z}_i &= \frac{\vec{R}_Q - \vec{R}_i}{|\vec{R}_Q - \vec{R}_i|} \\
N\hat{y}_i &= \frac{R\hat{z}_i \times (\vec{R}_H - \vec{R}_i)}{|R\hat{z}_i \times (\vec{R}_H - \vec{R}_i)|} \\
N\hat{x}_i &= R\hat{y}_i \times R\hat{z}_i
\end{align*}
\] (3.21)

The basis vectors are used to define the rotation matrix that transforms the spacecraft’s body frame to track the LoS to the point \( \vec{R}_Q \) is given by as (Schaub and Junkins, 2013)

\[
[QB_i] = \begin{bmatrix}
N\hat{x}_i & N\hat{y}_i & N\hat{z}_i
\end{bmatrix}
\] (3.22)

The variation of targets, and the corresponding rotation matrices of the spacecraft in the swarm during their mapping and communications mode are summarized in Table 3.1

**Target Bodies** The target bodies considered in the current work can be classified into two categories: i. Uniform synchronous rotators, and ii. Complex tumblers.
Figure 3.14: The communications behavior of the spacecraft in the designed Class 2 swarms.

Table 3.1: LoS tracking targets, and corresponding rotation matrices for the visual mapping spacecraft swarm

<table>
<thead>
<tr>
<th>Mode</th>
<th>Observer</th>
<th>Leader</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping $</td>
<td>n_R_T</td>
<td>\leq r_{max}$</td>
</tr>
<tr>
<td>Communications $</td>
<td>n_R_T</td>
<td>&gt; r_{max}$</td>
</tr>
</tbody>
</table>

Specifically, the planetary moons considered in the current work are uniform synchronous rotators, whose spin axis is fixed along with one of their principal axis, and their rotation is tidally locked to the central planet. The complex tumblers, on the other hand, have a non-principal axis spin leading to a tumbling motion (Scheeres, 2016). The rotation matrix to transform the body-fixed rotating frame to the target frame $[TR]$ for both types of targets is presented here.

**Uniform Synchronous Rotators**  Assuming that the spin axis is fixed along the $z$ axis of the shape model as shown in Figure 3.15, the rotation matrix $[TR]$ at a time, $t$ for the case of uniform synchronous rotators is given by
Figure 3.15: Geometric parameters used to model the attitude of the synchronously rotating planetary moons.

\[ [TR] = R_z (\theta_{T,0} + \omega_T t) \] (3.23)

Where \( R_z \) is the principal rotation matrix along \( z \) axis (Schaub and Junkins, 2013), \( \theta_{t,0} \) is the initial rotation angle at the start of the simulation, and \( \omega_T \) is the magnitude of the angular velocity of the target. The value of \( \theta_{t,0} \) is selected based on the specific mission concept and will be presented in a later section.

**Complex Tumblers** The tumbling motion is a more complicated motion than a principal axis rotation (shown in Figure 3.17). In the current work, only the torque-free tumbling motion is considered, as the torques have insignificant effects for the time scales of a spacecraft flyby missions (Scheeres, 2016). The modified Rodriguez parameters (MRPs) are used to describe the tumbling attitude of the target body (Terzakis et al., 2018). Let \( \sigma \) denote the MRP that describes the rotation of the target body, with a magnitude \( \sigma \). We obtain the variation of \( \sigma \) by solving its kinematic differential equation (Schaub and Junkins, 2013)

\[ \dot{\sigma} = \frac{1}{4} \left[ (1 - \sigma^2)[1_{3 \times 3}] + 2[\tilde{\sigma}] + 2\sigma\sigma^T \right] T \tilde{\omega}_T \] (3.24)
Where $T\bar{\omega}$ denotes the angular velocity vector of the target body, and $[\cdot]$ denotes the skew-symmetric version of the input vector. The MRP is switched to its shadow set $\sigma^S$ to avoid singularities, which is given by

$$\sigma = \sigma^S = -\frac{\sigma}{\sigma^2} \quad \text{if } \sigma \geq 1 \quad (3.25)$$

The torque-free dynamics is governed by the Euler’s equation (Schaub and Junkins, 2013) and is expressed as

$$\dot{\bar{\omega}}_T = -[I_{3\times3}]^{-1}[\tilde{\varphi}] [I_{3\times3}] \bar{\omega}_T \quad (3.26)$$

The initial conditions to propagate Equations 3.24 and 3.26 for the global surface mapping missions will be presented in a later section. The MRP can be converted to yield the rotation matrix $[TR]$ as (Schaub and Junkins, 2013)

$$[TR] = [I_{3\times3}] + \frac{8[\tilde{\sigma}]^2 - 4(1-\sigma^2)[\tilde{\sigma}]}{(1+\sigma^2)^2} \quad (3.27)$$

Since attitude propagation is involved, where the shadow set checks in Equation 3.25 have to be performed at every propagation time step, all spacecraft and target motion is propagated using a custom, fixed time step, 4th order Runge-Kutta integrator (Munthe-Kaas, 1998).

### 3.4 Statistical Coverage Evaluation

The statistical figures of merit used for evaluating the coverage performance of swarms for both swarm missions are described as follows.

**Global Surface Mapping Coverage**

Their irregular shapes characterize the small bodies studied in the current work. To reduce the sensitivity of the coverage to these irregularities, the Dual Sphere coverage method is developed, which is described as follows.
Dual Sphere Coverage  In the Dual Sphere coverage method, the coverage evaluation algorithm in Figure 3.10 is employed over two spherical shape models in addition to the nominal shape model. The radius of these two spherical models are equal to $R_{T,max}$ and $R_{T,min}$. The sphere with the radius $r_{T,max}$ is referred to as the view sphere as it specifies the camera FoV, using Equation 3.2, that always covers the complete shape model. The sphere with radius $R_{T,min}$ is referred to as the range since it is used in Equation 3.3 to compute the spacecraft range such that the lowest radius features on the shape model can be observed. Both spheres are located at the centroid of the target and exhibit the same attitude motion as the target body. The geometrical parameters in the Dual Sphere method are presented in Figure 3.16.

The Dual Sphere coverage method provides a stochastic framework to evaluate coverage of a swarm in design global surface mapping mission concepts.
Random Attitude Initialization  The nominal and Dual Sphere models are initialized with a random attitude when designing global surface mapping missions, whose for the two types of small bodies is presented as follows:

**Planetary Moons** At the start of a simulation, the planetary moon is initialized with a rotational angle $\theta_{T,0}$, which is a random number uniformly distributed in [0, 360] deg.

**Tumbling Asteroids** The target asteroid is assumed to be in a long axis mode (LAM) tumbling state with the right ascension $\alpha_T$, and declination $\delta_T$ of its rotational pole determined. The rotation matrix $[T R]$ can then be computed as

$$[T R] = R_z(\alpha_T)R_y(\delta_T) \tag{3.28}$$

The initial MRP $\sigma(t_0)$ can be computed from the rotation matrix using an intermediate conversion to the quaternion representation (Schaub and Junkins, 2013). For typical tumblers, $\delta_T$ is tightly bounded between $[\delta_l, \delta_u]$, while $\alpha_T$ can be located anywhere between $[0, 360]$ deg Scheeres et al. (1998). Therefore, for a given coverage simulation, the right ascension and declination of the pole are initialized as

$$\alpha_T(0) = r(0, 360) \tag{3.29}$$
$$\delta_T(0) = r(\delta_l, \delta_u) \tag{3.30}$$

Where $r(a, b)$ denotes a random number which is uniformly distributed in $[a, b]$. The tumblers considered in the current work are LAM tumblers, whose angular momentum is largely focused about its long axis. In the current work, the shape model is formatted such that the long axis coincides with the $T \hat{x}$ axis. The different parameters used to describe the attitude of the tumbling asteroid are presented in Figure 3.17. Using the convention described in Nazari et al. (2014), the initial angular velocity vector of the tumbling asteroid at the beginning of a coverage evaluation simulation is written as
\[ \bar{\omega}_T(0) = \frac{2\pi}{T_{Rot}} [\tan \eta_T \quad 0 \quad -1]^T \] (3.31)

Figure 3.17: Illustration of various geometrical parameters used to model the attitude of the tumbling asteroids.

Where \( \eta_T \) is the angle between the angular velocity vector and the longest principal axis \( T\hat{x} \), and \( T_{Rot} \) is the effective rotation period of the target. The variation of \( \eta_T \) for a given tumbler can be bounded between \([\eta_l, \eta_u]\) (Scheeres et al., 1998; Nazari et al., 2014). Therefore, each simulation is initialized with a randomized angular initial angular velocity vector, which is characterized by

\[ \eta_T(0) = r(\eta_l, \eta_u) \] (3.32)

Figure of Merit Since each simulation is initialized with a randomized target attitude, we run \( N_{mon,1} \) Monte-Carlo simulations, where, in each simulation, the mean and standard deviation of the coverage of the nominal models and its Dual Sphere models are evaluated. Let \( \langle P_{Cov} \rangle \mid_{N_{mon,1}} = \langle P_{Cov,V} \rangle, \langle P_{Cov,N} \rangle, \langle P_{Cov,R} \rangle \rangle \mid_{N_{mon,1}} \) denote an array containing mean coverages of the the view sphere,
nominal model, and range sphere respectively, which is evaluated over $N_{mon,1}$ Monte-Carlo simulations. Similarly, let $S_{Cov|N_{mon,1}} = [S_{Cov,V}, S_{Cov,N}, S_{Cov,R}]|_{N_{mon,1}}$ denote an array containing $1-\sigma$ standard deviation of coverages of the corresponding shape model evaluated over $N_{mon,1}$ Monte-Carlo simulations. Then the key figure of merit used to estimate the global mapping performance $P_{FoM,map}$ of the swarm is the lowest value of the difference between the mean and standard deviation of the coverages, given by

$$P_{FoM,map} = \min \left( \langle P_{Cov} \rangle |_{N_{mon,1}} - S_{Cov}|_{N_{mon,1}} \right)$$  \hspace{1cm} (3.33)$$

**RoI Coverage**

The RoI missions considered in the current work are only targeted at uniformly rotating synchronous planetary moons, due to the uniformity of their spin. Since the RoI is a fixed feature on the surface of the shape model. However, to ensure that the designed performance is robust to a spin angle uncertainty $\theta_{max}$, at the beginning of each simulation, the moon is initialized at a spin angle $\theta_{T,0}$

$$\theta_{T,0} = f_{0} + r(\theta_{max})$$  \hspace{1cm} (3.34)$$

Where $f_{0}$ is the orbital true anomaly of the moon at the start of the simulation, and $r(\theta_{max})$ is a random number uniformly distributed in $[-\theta_{max},\theta_{max}]$. The implication of Equation 3.34 is that when $r(\theta_{max}) = 0$, the rotation angle of the moon, computed from Equation 3.23, is always equal to its true anomaly if the moon is in circular orbits (Aleshkina, 2009). Therefore Equation 3.34 nominally enforces the tidal locking phenomenon of the moon when its orbit is nearly circular. The amplitude of uncertainty, $\theta_{max}$ is defined by the user in the mission definition phase. The parameter $f_{0}$ is noted from the first encounter location with the target and will be presented in the case studies.
Figures of Merit  Similar to the global surface mapping missions, we run $N_{\text{mon,1}}$ Monte-Carlo simulations, where, in each simulation the mean and standard deviations of the spatial and temporal RoI coverages are evaluated. Let $< P_{\text{RoI}} > |_{N_{\text{mon,1}}}$ and $S_{P,\text{RoI}}|_{N_{\text{mon,1}}}$ denote the mean and $1 - \sigma$ standard deviation of the spatial coverage, and $< T_{\text{RoI}} > |_{N_{\text{mon,1}}}$, and $S_{T,\text{RoI}}|_{N_{\text{mon,1}}}$ denote the mean and $1 - \sigma$ standard deviation of the temporal coverage of the RoI evaluated over $N_{\text{mon,1}}$ simulations, the spatial $P_{\text{FoM,RoI}}$ and temporal $T_{\text{FoM,RoI}}$ figures of merit used in the current work are given by

$$\begin{align*}
P_{\text{FoM,RoI}} &= < P_{\text{RoI}} > |_{N_{\text{mon,1}}} - S_{P,\text{RoI}}|_{N_{\text{mon,1}}} \quad (3.35)
\end{align*}$$

and

$$\begin{align*}
T_{\text{FoM,RoI}} &= < T_{\text{RoI}} > |_{N_{\text{mon,1}}} - S_{T,\text{RoI}}|_{N_{\text{mon,1}}} \quad (3.36)
\end{align*}$$

The collision and coverage figures of merit described in this chapter can be used to filter out genes that do not meet the design requirements. These are then used by the swarm optimizer module to identify the optima gene that meets performance requirements, as indicated in Figure 3.2.
CHAPTER 4

Sensitivity Analysis of Swarm Designs

The concept design phases typically rely on analytical models to make critical mission design decisions. These models assume perfect knowledge of the parameters involved. However, real-world implementations are always susceptible to errors resulting from perturbations and model uncertainties. Thus, it is important to study the performance variation of the nominal designs when subjected to different types of perturbations. This chapter describes the swarm design parameters and the algorithms used in the current work to analyze the sensitivity of the swarm performance when subjected to different classes of perturbations. We begin by an in-depth examination of the swarm design gene, which describes the construction of trajectories of the spacecraft of the swarm during their encounter with the target body. The swarm gene allows us to define three categories of perturbation: spacecraft outages, target location at the encounter, and encounter shaping as described in (Nallapu and Thangavelautham, 2021). Additionally, a fourth perturbation category of dynamical sensitivity is presented where the motions of the spacecraft and the target body are propagated by adding various dynamical perturbative forces. The objective of this chapter is to provide an extensive description of the Monte-Carlo algorithms that study the variation of the surface coverage of the designed swarms when subjected to different categories of perturbations presented in this chapter.

4.1 Swarm Gene Structures

In this thesis, we design swarm missions to map the surface of planetary moons and asteroids. The primary difference between these two types is that planetary moons exhibit short term periodic motion that is on the order of few days, around the central planet, which is the primary source of gravitational motion (Vallado,
The swarm size category of variables specifies the number of visits to the small body, along with the number of spacecraft in each visit. The ideal mission would have all visiting spacecraft contributing to generating the visual coverage of the target body. We study the sub-optimal case, where some spacecraft in the designed swarm cannot contribute to the cumulative coverage due to a subsystem failure.
Target Location at Encounter

The small body locations during each swarm visit are specified by the planetary true anomaly in the case of the planetary moons, and the time past the heliocentric arrival epoch in the case of the asteroids. In order to evaluate the coverage, the target body is initialized at a location shortly before the first swarm encounter location. Therefore, to study the effect of the target’s location, we introduce a perturbation on this initial location and study the variation of the coverage of the swarm.

Encounter Shaping

During an encounter, spacecraft location with respect to the target is described by the spherical coordinates: flyby radius $r_f$, azimuth $\theta_{x,i}$, and elevation angles $\theta_{y,i}$ as

$$T \bar{R}_{Ti} = r_f [\cos \theta_{y,i} \cos \theta_{x,i} \cos \theta_{y,i} \sin \theta_{x,i} \sin \theta_{y,i}]^T \quad (4.1)$$

While the flyby radius is constrained by the image resolution requirements using Equation 3.4, $\theta_{x,i}$, and $\theta_{y,i}$ are noted through the swarm design optimization, where the coverage requirements are met through a minimum number of spacecraft. As a result, to study the sensitivity of these encounter shaping parameters, a perturbative position vector is added to the encounter location of each spacecraft. The cumulative coverage resulting from these flybys is then noted.

4.2 Analysis Algorithms

We now describe the algorithms used in the current work to study the performance of the swarm when subjected to different perturbations placed on the categories of design variables described here. Additionally, we present the algorithms to evaluate the performance of coverage when the fidelity of the spacecraft motion propagation is improved. While typically, the designed coverage requirements on the designs may be aggressive ($\geq 90\%$), small perturbations can easily result in performance
that falls below the original requirement. Therefore, we use a relaxed coverage requirement of $P_{\text{thr}}$ to study the sensitivity of these perturbations. In all cases, the minimum expected coverage over all three models: nominal, range, and view sphere are noted.

**Sensitivity to Spacecraft Outages**

The sensitivity to outage describes how many spacecraft in the swarm can be lost before the coverage deteriorates below $P_{\text{thr}}$. To study this, let us assume that the designed mission concept consists of a swarm of $N_{sw}$ spacecraft. In this swarm, $P_{\text{op}}\%$ of spacecraft are functional, while the remaining are assumed to be defunct due to a random outage. As a result, the encounters with the target will now contain defunct spacecraft that do not contribute to the surface coverage by the swarm, as seen in Figure 4.2. Therefore, when outages are modeled, only $N_{op}$ of $N_{sw}$ spacecraft are assumed to be operational, which is expressed as

$$N_{op} = \text{floor} \left( N_{sw} \frac{P_{\text{op}}}{100} \right)$$

(4.2)

Figure 4.2: Illustration of a swarm encounter with the target body, showing the nominal encounter (left) an encounter with spacecraft outages (right).
For a given value $P_{op}$, we conduct $N_{mon,2}$ Monte-Carlo simulations where $N_{op}$ spacecraft from the swarm are randomly selected, and the remaining are declared as defunct. The surface coverage of the swarm is then computed as described in the previous section, by only using the observations of the operational spacecraft. It can be noted that some values of $P_{op}$ will do not require $N_{mon,2}$ trials due to a small number of possibilities, for example if only one spacecraft is defunct, this can happen in $N_{sw}$ possibilities (assuming $N_{mon,2} > N_{sw}$). Therefore, the maximum number of simulations is capped to their binomial coefficient $\binom{N_{sw}}{N_{op}}$ when $N_{mon,2}$ exceeds this binomial coefficient. The stochastic variation of the coverage by noting the mean and $1 - \sigma$ standard deviation for a given value of $P_{op}$ is noted. The value of $P_{op}$ is then varied within bounds to study if and when the mean coverage deteriorates to below $P_{thr}$.

Simulating spacecraft outages allows us to study the variation of coverage when multiple spacecraft in the swarm are unable to contribute to the cumulative coverage of the target body.

**Sensitivity to Target Location**

The sensitivity of coverage to the location of the target body indicates the maximum error in the knowledge of the target’s location before the coverage deteriorates below $P_{Low}$. The target’s location during an encounter $j$ is described by the true anomaly $f_{v,j}$ in the case of the moons and time past arrival $T_j$ in the case of the asteroids. In order to employ a generalized framework, let $f t_j$ denote the generalized parameter that represents the location of the target body during an encounter $j$. To evaluate the cumulative coverage of the target body by the swarm, the target body is initialized at a small offset $\Delta f t_0$ before the first encounter given by

$$f t_0 = f_{v,1} - \Delta f t_0 \quad (4.3)$$

To quantify the sensitivity of coverage to the uncertainty of the target’s location,
its initial location is perturbed by $\Delta ft_{0,P}$ given by

$$ft_{0,P} = ft_0 + \Delta ft_{0,P} \quad (4.4)$$

Figure 4.3: Illustration of the effect of perturbing the initial location of the target body.

An illustration of the effect of perturbing the target’s location is presented in Figure 4.3. The perturbation $\Delta ft_{0,P}$ is varied within bounds to study the variation of the cumulative swarm coverage. Each value of $\Delta ft_{0,P}$ is subjected to $N_{\text{mon,2}}$ Monte-Carlo simulations, where the shape models are initialized with random initial orientations. The expected coverage value over all three shape models is noted, and the variation of the expected coverage is then studied.

Perturbing the moon’s position allows us to study the variation of coverage under the presence of uncertainties in the moon’s location.

**Sensitivity to Encounter Locations**

In order to study the effect of perturbation this encounter location on the coverage of the swarm, we introduce a perturbation position vector $\Delta \tilde{R}_i$ on each spacecraft, whose amplitude is given by $\Delta R_i$ and the direction is given by random azimuth and
elevation angles: $\theta_{Rx,i}$, and $\theta_{Ry,i}$ respectively. Therefore, the new encounter location $\bar{R}_{Ti}'$ of spacecraft $i$ in the swarm with the target will be given by

$$T \bar{R}_{Ti}' = T \bar{R}_{Ti} + T \Delta \bar{R}_i \quad (4.5)$$

where

$$T \Delta \bar{R}_i = \Delta R \begin{bmatrix} \cos \theta_{Ry,i} \cos \theta_{Rx,i} & \cos \theta_{Ry,i} \sin \theta_{Rx,i} & \sin \theta_{Ry,i} \end{bmatrix}^T \quad (4.6)$$

Figure 4.4: Illustration of the effect of perturbing the encounter location of the spacecraft during their encounter.
The effect of perturbing the encounter locations is presented in Figure 4.4. The perturbative position vector $\Delta \vec{R}_i$ is applied in a frame located at the nominal encounter location of the corresponding spacecraft $i$, while the basis vectors are parallel to those of the target body, as shown in Figure 4.4. As a result of the new encounter location, the trajectories of the spacecraft in the swarm will change, based on the models described in Chapters 5 and 6. The coverage, on these perturbed trajectories, will only be evaluated when the flyby altitude with respect to the target body falls below the imaging altitude described in Equation 3.1, in order to meet the resolution requirements. The perturbation amplitude $\Delta \vec{r}$ is then varied within specified bounds, while the angular components: $\theta_{x,r}$ and $\theta_{y,r}$ are introduced as uniformly distributed random numbers between $[0, 2\pi]$ and $[-\pi/2, \pi/2]$ respectively. The coverage variation is then noted by simulating $N_{\text{mont},2}$ Monte-Carlo simulations, where each simulation produces a different set of spherical coordinate perturbation $(\theta_{Rx,i}, \theta_{Ry,i})$ on each spacecraft, along with initializing the target shape models at random initial attitudes. The mean and $1 - \sigma$ standard deviation of the cumulative swarm coverage over all three shape models are noted at each amplitude.

Perturbing the encounter position allows us to study the variation of coverage under the presence of uncertainties in the spacecraft’s location.

### 4.3 Sensitivity to Dynamical Perturbations

While the above-described perturbations describe the variation of coverage with respect to the category of the swarm design variables, the dynamical perturbations described in this section present the variation of coverage when the motion of the spacecraft is perturbed from the nominally designed trajectories.

Dynamical perturbations allow us to study the variation of coverage under the presence of modeling errors.
Perturbations

In the current work, the uniform two-body gravitational model (Vallado, 2013) is used to design the nominal spacecraft and target body trajectories. Let \( \vec{R} \) and \( \vec{V} \) denote the generalized position and velocity vectors with respect to the central gravitational source of mass \( M_s \). The generalization implies that \( \vec{R} \) and \( \vec{V} \) can describe the dynamical state of the spacecraft or the target body. The two-body dynamics presented in Equation 3.17 can be expressed as

\[
\dot{\vec{R}} = \vec{V} \\
\dot{\vec{V}} = \vec{A}_{2B} = -\frac{\mu_s}{r^3} \vec{r}
\]

(4.7)

Where \( \mu_s = GM_s \) is the gravitational parameter of the source, \( G \) is the universal gravitational constant, and \( r \) is the magnitude of \( \vec{R} \). In order to define the effect of perturbations, we define the perturbed position \( \vec{R}' \) and velocity vectors \( \vec{V}' \), whose dynamics is given by

\[
\dot{\vec{R}}' = \vec{V}' \\
\dot{\vec{V}}' = \vec{A}_{2B} + \vec{A}_P
\]

(4.8)

Where \( \vec{A}_P \) is the perturbative acceleration vector. In the current work, we increase the fidelity of propagation incrementally in levels, where at each subsequent level, a different perturbative acceleration is added to model the dynamics. For planetary moons, the effect of third body gravity, asphericity, and solar radiation pressure (SRP) on the propagation are considered. The perturbative acceleration components are described as follows:

**Third Body Gravitation** The third body perturbation adds the effect of the gravity from the target central body. Let the target body have a mass \( m_T \), and be located at \( \vec{R}_T \) from the primary source of gravitation, and at \( \vec{R}_{iT} \) respect to spacecraft \( i \). The third body acceleration \( \vec{A}_{3B,i} \) is given by on the spacecraft \( i \) is given by (Vallado, 2013):
\[ \bar{A}_{3B,i} = -\mu_T \left( \frac{\bar{R}_T}{r_T^3} + \frac{\bar{R}_{iT}}{r_{iT}^3} \right) \]  

(4.9)

Where \( r_T \) and \( r_{iT} \) are magnitudes of \( \bar{R}_T \) and \( \bar{R}_{iT} \) respectively, and \( \mu_T = Gm_T \).

Since the target body causes the third body motion, the third body acceleration will not influence the dynamics of the target body.

**Asphericity** In moon mapping missions, the non-spherical shape of the planet influences the dynamics of both the spacecraft and the target. The spherical harmonics (Vallado, 2013) describe a series of perturbative accelerations arising from incrementally deviating the shape of the central body from a perfect sphere. The oblateness, also called the \( J_2 \) effect, is the first and the dominant deviation from the spherical assumption. As defined above, let \( \bar{R} \) be the generalized position vector, with respect to the central planet, with Cartesian components \([r_x \ r_y \ r_z]^T\). The \( J_2 \) perturbative acceleration (Sengupta, 2004) is given by

\[ \bar{A}_{Asp} = -\frac{3\mu_s J_2 R_p^2}{r^5} \begin{bmatrix} r_x \left( 1 - 3 \left( \frac{r_z}{r} \right)^2 \right) \\ r_y \left( 1 - 3 \left( \frac{r_z}{r} \right)^2 \right) \\ r_z \left( 3 - 3 \left( \frac{r_z}{r} \right)^2 \right) \end{bmatrix} \]  

(4.10)

Where \( J_2 \) and \( R_p \) denote the oblateness coefficient and the equatorial radius of the central planet, respectively. As denoted by the generalized coordinates, the \( J_2 \) acceleration will impact the dynamics of the target and the swarm when used.

**Dynamical Perturbations around Asteroids** While the accelerations described in Equations 4.9 and 4.10 can be used to describe perturbations around planetary moons, their models vary when describing dynamics around asteroids. One of the primary sources of differences is that a moon orbits closely around its central planet, which is the primary source of their dynamical motion. On the other hand, asteroids are sufficiently far away from the Sun, which is their primary source of dynamical motion. The sphere of influence is used as a figure of merit to describe
when the dynamical models change when planning missions to asteroids. The radius of the sphere of influence $r_{\text{SoI}}$ is used to switch dynamical models, which is expressed as (Vallado, 2013)

$$
r_{\text{SoI}} = \left( \frac{\mu_T}{\mu_s} \right)^{\frac{2}{3}} a_T \tag{4.11}
$$

Where $a_T$ corresponds to the semi-major axis of the target. The acceleration models for third body and asphericity models can now be described as follows.

**Third Body** Using the radius of sphere influence as a threshold, the perturbative acceleration is defined as follows:

$$
\bar{A}_{3B,i} = \begin{cases} 
-\mu_T \left( \frac{\bar{R}_T}{r_i^2} \right) + \mu_s \left( \frac{\bar{R}_s}{r_i^2} \right) & r_{iT} \leq r_{\text{SoI}} \\
-\mu_T \left( \frac{\bar{R}_T}{r_i^2} + \frac{\bar{R}_s}{r_i^2} \right) & r_{iT} > r_{\text{SoI}} 
\end{cases} \tag{4.12}
$$

Where $\bar{R}_i$ describes the heliocentric position of spacecraft $i$ whose magnitude is given by $r_i$. It should be noted here that the third body acceleration when $r_{Ti} \leq r_{SoI}$ contains the gravity due to asteroid, which is the primary source of gravity. Therefore, while $\bar{A}_{3B,i}$ cannot strictly be classified as a perturbation, the summation $\bar{A}_{2B,i} + \bar{A}_{3B,i}$ is used to describe the perturbed third body dynamics (Vallado, 2013) around asteroids.

**Asphericity** The non-uniform shape of the asteroids results in perturbed spacecraft motion (Scheeres et al., 1998), especially when the spacecraft is inside the sphere of influence of the asteroid. The second-order spherical harmonic perturbations are used to model the effect of perturbations due to asphericity. These perturbations are evaluated in Misra et al. (2015) as

$$
\bar{A}_{\text{Asp}} = -\frac{3\mu_T}{r_{Ti}^5} \begin{bmatrix} 
 r_x \left( C_{20} - 5C_{20} \left( \frac{r_{y}^2 + r_{z}^2 - 2r_{x}^2}{2r_{Ti}^2} \right) \right) - 6C_{22} + 15C_{22} \left( \frac{r_{y}^2 - r_{z}^2}{r_{Ti}^2} \right) \\
 r_y \left( C_{20} - 5C_{20} \left( \frac{r_{x}^2 + r_{z}^2 - 2r_{y}^2}{2r_{Ti}^2} \right) \right) + 6C_{22} + 15C_{22} \left( \frac{r_{x}^2 - r_{z}^2}{r_{Ti}^2} \right) \\
 r_z \left( -2C_{20} - 5C_{20} \left( \frac{r_{x}^2 + r_{y}^2 - 2r_{z}^2}{2r_{Ti}^2} \right) \right) + 15C_{22} \left( \frac{r_{x}^2 - r_{y}^2}{r_{Ti}^2} \right)
\end{bmatrix} \tag{4.13}
$$
Where $C_{20}$ and $C_{22}$ are the second degree and order gravity coefficients of the target asteroid. When the spacecraft is outside the sphere of influence of the asteroid, the effects of asphericity are not strongly felt by the spacecraft, so we set $\bar{A}_{Asp} = 0$.

**Solar Radiation Pressure** The Solar radiation pressure describes the effect of the Solar photons incident on the spacecraft or the target body. The Cannonball model is used, which developed from the LAGEOS missions (Lucchesi, 2002), to model the SRP effect on the spacecraft and the target. Let $m$ denote the generalized mass describing both the spacecraft and the target body, whose net cross-sectional area exposed to the Sun is $A_S$. Let the Sun be located at a distance $r_S$ with respect to the body of interest at a direction given by $\hat{u}_S$. The SRP acceleration acting on the target body is then given by

$$\bar{A}_{SRP} = -A_S C_r \Phi_E \left( \frac{1 \text{ AU}}{r_S} \right)^2 \hat{u}_S$$

(4.14)

Where $\Phi_E$ is the solar flux at 1 AU, $c$ is the speed of light, and $C_r$ is the effective coefficient of reflection. In the case of artificial objects like spacecraft, the effective coefficient of reflection (Kenneally, 2016) can be written as

$$\bar{C}_{r,i} = 1 + s_k + \frac{2}{3} \rho_k$$

(4.15)

Where $s_k$ is the coefficient of specular reflection, and $\rho_k$ is the coefficient of diffuse reflection of the exposed face. While in case of the target small bodies, the effective coefficient of (Scheeres, 2016) as

$$\bar{C}_{r,T} = 1 + \rho_T$$

(4.16)

Where $\rho_T$ is the albedo of the target body.

**Propagation Scheme**

Since the dynamical environment varies with the type of small bodies being studied, the perturbations applied to the spacecraft, and target body motion will vary
Table 4.1: Perturbation accelerations used in the current work to study the variation of spacecraft coverage under different levels of dynamical fidelity.

<table>
<thead>
<tr>
<th>Level</th>
<th>Spacecraft</th>
<th>Perturbative Accelerations</th>
<th>Asteroid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Spherical gravity</td>
<td>Spherical gravity (Planet)</td>
<td>Spherical gravity (Sun)</td>
</tr>
<tr>
<td>2</td>
<td>Spherical gravity</td>
<td>Spherical gravity (Planet)</td>
<td>Spherical gravity (Sun)</td>
</tr>
<tr>
<td></td>
<td>3rd body (Target)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Spherical gravity</td>
<td>Spherical gravity (Planet)</td>
<td>Spherical gravity (Sun)</td>
</tr>
<tr>
<td></td>
<td>3rd Body (Target)</td>
<td>Planetary $J_2$ (Planet)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asphericity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Spherical gravity</td>
<td>Spherical gravity (Planet)</td>
<td>Spherical gravity (Sun)</td>
</tr>
<tr>
<td></td>
<td>3rd Body (Target)</td>
<td>Planetary $J_2$ (Planet)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asphericity</td>
<td>SRP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SRP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

depending on the class of small body being studied. Additionally, since there are multiple sources of perturbations, the dynamical simulations are divided into multiple levels of fidelity, where at each level, a perturbative acceleration is added to that of the previous levels. This hierarchy also provides an insight into the dominant contributor of perturbation that would deteriorate the coverage below $P_{thr}$. The perturbative acceleration models used at each propagator fidelity level is presented in Table 4.1. At each level, $N_{mon,2}$ Monte-Carlo simulations are evaluated, where each simulation initializes the shape models of the target at a random attitude. The variation of the cumulative coverage on all three shape models is noted at each fidelity level.
Missions to planetary moons offer a unique set of challenges. The gravity of the host planet primarily influences the spacecraft and the Moon’s dynamics. This constrains the radius of spacecraft orbits around the moons. The challenge is aggravated with small Planetary moons such as the Martian moons. In these cases, the sphere of influence is closer to or inside the Moon’s surface, making it virtually impossible to orbit the Moon using Keplerian trajectories (Zamaro and Biggs, 2016). This reduces the potential mission concepts to explore small planetary moons. Performing the flybys through hyperbolic or co-orbital trajectories of the central planet or surface landers are viable mission concepts. Each of these mission concepts has its own set of distinguishing features, as seen in Figure 5.1. Since the focus of this thesis is on flyby reconnaissance mission concepts, we only examine moon reconnaissance from hyperbolic and co-orbital flybys. This chapter describes the design of hyperbolic flyby mission concepts, while Chapter 6 explores the design of co-orbital reconnaissance of planetary moons.

This chapter also presents the first case study of the current work, where the design of visual reconnaissance mission concepts to planetary moons is explored. In this case study, we design swarm missions that compensate for spatial coverage limitations flyby reconnaissance. The swarms will be deployed on hyperbolic trajectories with respect to the central planet. In this chapter, we present a numerical algorithm to design the swarm’s hyperbolic trajectories such that the encounters occur at a fixed orbital location of the Moon. We begin this chapter by reviewing related work done in the field of planetary moon exploration. The trajectory, swarm, and spacecraft design problems for global surface mapping missions to planetary moons are then formulated and are then handled by the IDEAS framework. The Dual Sphere coverage method is used to evaluate the coverage performance of
5.1 Related Work

Planetary moons are a particular class of small bodies that orbit larger planetary bodies instead of the Sun. Because of this property, they are referred to as natural satellites or planetary satellites. Currently, more than 200 planetary moons have been identified in the Solar system, with a vast majority of moons orbiting around outer giant planets like Jupiter (Stevenson, 2001) and Saturn (Canup, 2010). The planetary moons exhibit different shapes, sizes, and compositions (Graykowski and Jewitt, 2018). The exploration of planetary moons has been pursued to understand the solar system processes, build planetary habitats, and develop in-situ resource utilization strategies. Virtually, all planetary moons serve as potential sources to

Figure 5.1: A comparison of different mission concepts to explore small planetary moons.
describe the solar system evolution processes (Committee on the Planetary Science Decadal Survey, 2011; Castillo-Rogez et al., 2012). The moons of outer planets such as the Jovian moon Europa, and the Saturnian moon Titan (Battison, 2011) are being evaluated as feasible candidates to serve as extra-terrestrial habitats. Due to this reason, several missions to these moons have been planned. Notable examples to these moons include the Europa Clipper mission to Europa (Phillips and Pappalardo, 2014) and the Dragonfly mission to Titan (Lorenz et al., 2018). Planetary moon encounters have served as the critical source of planetary moon exploration. Historical flybys of the Pioneer (Fimmel et al., 1980) and Voyager missions (Kohlhase and Penzo, 1977) have immensely contributed to the knowledge of the moons of outer planets such as Jupiter and Saturn. The flyby trajectories of two Voyager spacecraft near Saturn and their photo contributions of the Saturnian
moons are presented in Figure 5.2. At the existing state-of-the-art, the design of these moon-encountering hyperbolic trajectories has been treated as a timing problem. Here, once a hyperbola of suitable shape and size is selected, the spacecraft’s launch timing is adjusted such that it passes through the encounter location with respect to the Moon. However, when designing global surface mapping missions, in addition to timing, the design should also account for the three-dimensional location of the encounter to allow imaging from sufficient perspectives angles.

Figure 5.3: Highest resolution images of Phobos recorded till date by the Mars Express mission. The resolution of the outer image is about 3.9 m/px, while the 7 internal images were imaged at 3 m/px.

While the outer planets host a vast majority of the planetary moons, the martian moons are characterized by several key distinct features, making them primary targets for scientific and engineering research (Space Mission Directorate, 2020). One such key distinguishing features of these moons is the uncertainty in their origin (Veverka and Burns, 1980). The inner Moon Phobos is shown in Figure 5.3 specifically has received much attention due to radial distance to Mars. It is suspected
that the Moon might form a debris ring around Mars in the near future (Black and Mittal, 2015). The Marian moons have also been studied as an active research target for ISRU activities en-route to Mars and deep space destinations (Muscatello et al., 2012; Abercromby et al., 2015). Due to these reasons, they have been studied by several space missions, such as the Mariner, Viking, Martian Reconnaissance Orbiter (MRO) (Duxbury et al., 2014). One of the key distinguishing features of these small moons is their virtually absent spheres of influence. The SoI radius of Phobos (maximum radius: 13.4 km) is 7.3 km while that of Deimos (maximum radius: 7.2 km) is 7.5 km (Wallace et al., 2012) – which suggests that it is virtually impossible to perform reconnaissance from Keplerian orbits around these moons (Zamaro and Biggs, 2016). Therefore, to date, these moons’ reconnaissance has been performed from flyby encounters with single spacecraft as their secondary objectives. To date, the highest resolution images of Phobos were recorded by the High Resolution Stereo Camera (HRSC) onboard the Mars Express mission (Witasse et al., 2014), whose resolutions were about 3.9 – 3 m/px resolution as shown in Figure 5.3. Having surveyed state-of-the-art planetary moon exploration research, in the current chapter, we develop numerical algorithms that design hyperbolic trajectories of spacecraft swarms, which account for both timing, and locations of the spacecraft encounters. The trajectories are then configured to design a global surface mapping mission concept to the Martian moon Phobos.

5.2 Mission Concept Modeling

This section presents the crucial parameters that are passed as input arguments to the IDEAS design framework. The coverage requirements and critical parameters to design trajectory and spacecraft are presented here.

Coverage Requirements

The objective of the mission concept in question is to obtain a global surface map of a planetary moon through hyperbolic flybys around the central planet. As described
in Chapter 3, a global surface mapping swarm is required to produce surface maps that have a maximum ground resolution of $x_D$ with a minimum cumulative coverage of $P_{map,R}$. The mission requirement parameters used in the current chapter to are summarized in Table 5.1.

Table 5.1: Input mission requirements for the Phobos mapping swarm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage requirement, $P_R$</td>
<td>90 %</td>
</tr>
<tr>
<td>Ground resolution requirement, $x_R$</td>
<td>1 m</td>
</tr>
<tr>
<td>Minimum elevation angle, $\varepsilon_R$</td>
<td>5 deg</td>
</tr>
</tbody>
</table>

**Camera and Flyby Parameters**

The dry mass of the seed spacecraft is assumed to populated with subsystems presented in Appendix A. As noted there, the spacecraft payload is a visual camera with an aperture diameter of $D_C = 8$ cm, whose flyby altitude is calculated using the red spectrum as a baseline wavelength. The values $r_{max}$ and $\eta_C$ computed using Equations 3.3, and 3.2 are 120 km and 3.04 deg respectively. Using a $\Delta h = 5$ km attitude tolerance, the flyby radius of the spacecraft, computed using Equation 3.4, is located at $r_f = 115$ km from the target moon.

**Trajectory Design**

The trajectory design addresses the launch of spacecraft from the Earth and arrival at the target planet. All spacecraft in the swarm are assumed to be launched from Earth on the same launch epoch $D_L$ and arrive at the sphere of influence (SoI) of a selected arrival epoch $D_A$. The launch and arrival epochs can define a heliocentric Lambert arc between the launch and arrival planets. The Lambert arc provides the excess velocity asymptotes at the launch $\bar{V}_{\infty,1}^+$ and arrival $\bar{V}_{\infty,2}^-$. The mission-critical parameters like launch energy ($C_3$), time of flight ($ToF$), and excess velocity magnitude at arrival $v_{\infty,2}^-$ can be easily extracted from these asymptotes (Vallado, 2013). Since the encounter velocities with the target moon are directly related to
the spacecraft’s excess velocities at the arrival planet, the ease of imaging could be improved by selecting trajectories with low $v_{\infty,2}$. Additionally, the trajectories selected should have practical bounds on the launch energy $C_{3,\text{max}}$, and time of flight $ToF_{\text{max}}$. Additionally, the right ascension $RAA$, and declination $DAA$ of the arrival asymptote can also be extracted from $\bar{V}_{\infty,2}$, which will be used to construct the swarm trajectories. The automated trajectory design problem then is a search for launch and arrival epochs which produces a feasible trajectory with the least excess velocity at arrival, posed as

$$\min J_T = v_{\infty,2}$$

s.t. \quad C_3 \leq C_{3,\text{max}}$$

$$ToF \leq ToF_{\text{max}}$$

(5.1)

Where the mission designer defines the bounds on launch and arrival epochs. The design gene of the automated trajectory design problem is shown in Figure 5.4.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Launch date</th>
<th>Arrival date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>$D_L$</td>
<td>$D_A$</td>
</tr>
<tr>
<td>Range</td>
<td>Integer</td>
<td>Integer</td>
</tr>
<tr>
<td></td>
<td>$[D_{L,\text{min}}, D_{L,\text{max}}]$</td>
<td>$[D_{A,\text{min}}, D_{A,\text{max}}]$</td>
</tr>
</tbody>
</table>

Figure 5.4: Trajectory gene for designing the Earth-Mars trajectories of the Phobos mapping swarm.

Swarm Design

Let us assume that during a single pass, a swarm of $N_j$ spacecraft will pass by the planetary Moon at an altitude $h'_j$. In order to design swarm encounters that meet the coverage requirement, the spacecraft $i$ in the swarm performs a flyby of the Moon at azimuth $\theta_{x,i}$, and elevation angles $\theta_{y,i}$ as shown in Figure 5.5.

If the Moon’s surface is completely illuminated, a single pass of the Moon is sufficient to generate a global surface map. However, since at any given time, roughly 50% of the Moon is in the hard shadow, the swarm to generates the global map in
Swarm size and target location parameters

Encounter shaping parameters

Figure 5.5: Design space of the hyperbolic flyby swarm, showing the swarm size and target location parameters (left), and encounter shaping parameters (right).

$N_v$ passes. Furthermore, to generalize these designs, each visit has $N_{v,j}$ spacecraft, where $j$ is the index corresponding to the pass number. The swarm size can now be computed as

$$N_{Sw} = \sum_{j = 1}^{N_v} N_j$$  \hspace{1cm} (5.2)

The swarm design gene of the moon mapping swarm is presented in Figure 5.6.

<table>
<thead>
<tr>
<th>Parameter</th>
<th># spacecraft visits</th>
<th># Spacecraft each visit</th>
<th>True anomaly of the moon</th>
<th>Spacecraft RA at visit</th>
<th>Spacecraft Dec at visit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N_v$</td>
<td>$N_{v,1}$ \ldots $N_{v,N_v}$</td>
<td>$f_{v,1}$ \ldots $f_{v,N_v}$</td>
<td>$\theta_{x,1}$ \ldots $\theta_{x,N_{Sw}}$</td>
<td>$\theta_{y,1}$ \ldots $\theta_{y,N_{Sw}}$</td>
</tr>
<tr>
<td>Variable</td>
<td>Integer</td>
<td>Integer</td>
<td>Real</td>
<td>Real</td>
<td>Real</td>
</tr>
<tr>
<td>Range</td>
<td>$[1, N_{1,\text{max}}]$</td>
<td>$[1, N_{2,\text{max}}]$</td>
<td>$[0, 2\pi]$</td>
<td>$[0, 2\pi]$</td>
<td>$[-\frac{\pi}{2}, \frac{\pi}{2}]$</td>
</tr>
</tbody>
</table>

Figure 5.6: Gene map of the swarm design problems showing different design variables and their attributes.
Swarm Behavior  The basis behavior of spacecraft in the swarm when mapping the target moon through hyperbolic flybys are presented as follows.

Encounter Hyperbola Construction  The trajectory design problem described above provides the incoming asymptote at planetary arrival. The challenge now is to ensure that the encounters share the same incoming hyperbolic asymptote, as shown in Figure 5.5. Let $N\vec{V}_{\infty,2}^{-}$ denote the incoming excess velocity asymptote of the spacecraft at the SoI of the planet, resolved in the inertial $N$ reference frame centered at the planet. The flyby radius $r_f$, the right ascension $\theta_{x,i}$, and the declination angles $\theta_{y,i}$, specify the position vector of spacecraft $i$ with respect to the moon using Equation 4.1. During a moon encounter $j$, when the Moon is located at $\vec{R}_{T,j}$ corresponding to the true anomaly $f_{v,j}$, the position vector of spacecraft $i$ with respect to the planet can be found as

$$N\vec{R}_i = T\vec{R}_{Ti} + N\vec{R}_{T,j}$$  \hspace{1cm} (5.3)

While it appears that the vectors $T\vec{R}_{Ti}$ and $N\vec{R}_{T,j}$ are resolved in different reference frames, it should be reminded here that the $T$ frame is obtained by purely translating the $N$ frame to the center of the target body as noted in Chapter 3. It is now apparent that Equation 5.3 specifies the inertial location of the spacecraft at the encounter epoch, while the incoming $N\vec{V}_{\infty,2}^{-}$ is noted at an epoch when the spacecraft are at the planet’s SoI. Therefore to determine the complete spacecraft state at an encounter, a boundary value problem is solved, which allows us to compute the velocity vector $\vec{V}_i$ with respect to the planet at the encounter epoch. Let $\vec{V}_i = [v_x \ v_y \ v_z]^T$ be this unknown velocity vector. The Vis-Viva equation Vallado (2013) allows us to constrain this velocity vector as

$$\left|N\vec{V}_i\right|^2 = v_i^2 = v_x^2 + v_y^2 + v_z^2 = \left(v_{\infty,2}^{-}\right)^2 + \left(\frac{2\mu_P}{r_i}\right)$$  \hspace{1cm} (5.4)

Where, $\mu_P$ is the gravitational parameter of the planet, and $r_i$ is the magnitude
of $^NR_i$. The semi-major axis $a_i$ and eccentricity vector $^N\varepsilon_i$ of the hyperbola can be written as (Vallado, 2013)

$$a_i = -\frac{\mu_P}{v_{\infty,2}^2}$$

(5.5)

and

$$^N\varepsilon_i = \frac{1}{\mu_P} \left( \left( v_i^2 - \frac{\mu_P}{r_i} \right)^N \bar{R}_i - \left( ^N\bar{R}_i, ^N\bar{V}_i \right)^N \bar{V}_i \right)$$

(5.6)

Where $v_{\infty,2}$ is the magnitude of $^N\bar{V}_{\infty,2}$. The specific angular momentum $^N\bar{h}_i$, and normal vector $^N\hat{n}_i$ can be expressed as

$$^N\bar{h}_i = ^N\bar{R}_i \times ^N\bar{V}_i$$

(5.7)

and

$$^N\hat{n}_i = ^N\hat{k} \times \frac{^N\bar{h}_i}{|^N\bar{h}_i|} = ^N\hat{k} \times ^N\bar{h}_i$$

(5.8)

Where, $^N\hat{k} = [0 \ 0 \ 1]^T$ is the z axis of the central planet’s inertial frame. It should be noted here that the vectors expressed in Equations 5.6-5.8 are functions of the unknown velocity components. These can be used to define the orientation elements of the orbit. The right ascension $\Omega_i$, inclination $i_{in_i}$, and argument of periapsis $\omega_{p,i}$ of the spacecraft can be determined as (Vallado, 2013)

$$\Omega_i = \begin{cases} \arccos \left( \frac{^N\hat{n}_i[1]}{|^N\hat{n}_i|} \right) & ^N\hat{n}_i[1] \geq 0 \\ 2\pi - \arccos \left( \frac{^N\hat{n}_i[1]}{|^N\hat{n}_i|} \right) & ^N\hat{n}_i[1] < 0 \end{cases}$$

(5.9)

$$i_{in_i} = \arccos \left( \frac{^N\bar{h}_i[3]}{|^N\bar{h}_i|} \right)$$

(5.10)
\[ \omega_{p,i} = \begin{cases} \arccos \left( \hat{N}_i \cdot \hat{e}_i \right) & N\hat{e}_i[3] \geq 0 \\ 2\pi - \arccos \left( \hat{N}_i \cdot \hat{e}_i \right) & N\hat{e}_i[3] < 0 \end{cases} \quad (5.11) \]

Finally, the hyperbolic true anomaly \( f_{\infty,i} \) is computed as

\[ f_{\infty,i} = \arccos \left( \frac{-1}{e_i} \right) \quad (5.12) \]

Where \( e_i \) is the magnitude of \( \hat{N}_i \), and the numbers inside \([\quad]\) indicate the components of the corresponding vector. On the entering asymptote, we set the true anomaly of the as \( f_i = -f_{\infty,i} \). The asymptotic excess velocity vector can be expressed as

\[ ^N\vec{V}_{\infty,2} = R_3(-\Omega_i)R_1(-i\eta_i)R_3(-\omega_{p,i})^P\vec{V}_{\infty,2} \quad (5.13) \]

Where \( R_1 \) and \( R_3 \) are principal rotation matrices about axes 1 and 3 respectively Schaub and Junkins (2013), and \(^P\vec{V}_{\infty,2}\) is the incoming excess velocity vector resolved in the perifocal frame of the planet given by (Vallado, 2013).

\[ ^P\vec{V}_{\infty,2} = \sqrt{\mu_P \over a_i (1 - e_i^2)} \begin{bmatrix} \sin (f_{\infty,i}) \\ -e_i + \cos (f_{\infty,i}) \\ 0 \end{bmatrix} \quad (5.14) \]

The right ascension \( RAA_i \) and declination \( DAA_i \) of spacecraft \( i \) can now be extracted from the asymptote as

\[ RAA_i = \arctan \left( \frac{\vec{V}_{\infty,i}[2]}{\vec{V}_{\infty,i}[1]} \right) \quad (5.15) \]

and
\[ DAA_i = \arcsin \left( \frac{\bar{V}_{\infty,i} \hat{3}}{|\bar{V}_{\infty,i}|} \right) \] (5.16)

This allows us to place final constraints on the trajectories as

\[ RAA_i = RAA \] (5.17)

and

\[ DAA_i = DAA \] (5.18)

Equations 5.17 and 5.18 are applied to all spacecraft in the swarm. These constraints ensure that all spacecraft share the same arrival asymptote. It should be noted here that Equations 5.4, 5.17, and 5.18 are nonlinear functions of the spacecraft velocity components, and are solved using a nonlinear root solvers. Through experimentation, the Moon’s velocity during the encounter was found to produce faster convergences to the solutions if used as an initial guess. The solution allows us to compute the spacecraft’s position and velocity on the hyperbola during the encounter epoch.

**Automated Swarm Design** The automated swarm design problem can now be posed as an optimization problem, where we are interested in determining the swarm trajectories, which can meet a coverage requirement \( P_{Req} \) with a minimum number of spacecraft. In addition to the coverage constraint, we require that the swarm be free of any undesired collisions. The algorithms to evaluate coverage and checks for spacecraft-target and spacecraft-spacecraft collisions are presented in Chapter 3. Since the spacecraft trajectories are described with respect to the central planet, the collision with the central planet can be avoided by placing a lower bound on the periapsis altitude \( h_{p,\text{min}} \) of each spacecraft \( h_{p,i} \) with respect to the central planet. Therefore, the swarm design problem can be posed as
min \quad J_{Sw} = N_{Sw} \\
\text{s.t.} \quad P_{\text{FoM, map}} \geq P_{\text{Req}} \\
\quad \min (h_{p,i}) \geq h_{p,\text{min}} \quad \forall \quad i \leq N_{Sw} \\
\text{collision flag} = \text{False} 

(5.19)

It should be noted that the swarm gene described in this chapter allows us to compute the spacecraft locations at the encounter instant. However, the location of the Moon is initialized by offsetting the lowest encounter true anomaly by $\Delta f_0$ as

$$f_0 = \min (f_{v,j}) - \Delta f_0 \quad (5.20)$$

The true anomalies of all spacecraft are then phased such that when the Moon reaches a true anomaly of $f_{v,j}$, the spacecraft corresponding to this encounter reach a true anomaly of 180 deg (Vallado, 2013). The cumulative coverage is noted by propagating spacecraft and moon motion for one orbital period of the Moon.

**Phobos Model**

The orbital elements and dynamical parameters of Phobos are presented in Table 5.2. A 32k triangular model of Phobos (Frieger, 2018) is used as the nominal shape model of the target moon. The view and range spheres are generated from the nominal shape model with 5000 triangular faces each. The radius of the range and view spheres are noted to be 14 km and 8.12 km respectively. The nominal shape model of Phobos, along with its Dual Sphere shape models, is presented in Figure 5.7, and the critical parameters noted from these models are summarized in Table 5.3. To study the swarm coverage performance, all three shape models were initialized with a random $z$ axis rotation uniformly distributed in $[0, 2\pi]$.

**Swarm Parameters**

Using the trajectory and the shape model parameters, the required swarm parameters are computed as follows. From Equation 3.1, the maximum altitude for the
Table 5.2: Dynamical parameters to model the encounter trajectories of the swarm with and Phobos.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phobos semi-major axis</td>
<td>9376 km</td>
</tr>
<tr>
<td>Phobos eccentricity</td>
<td>0.015</td>
</tr>
<tr>
<td>Phobos inclination</td>
<td>1.08 deg</td>
</tr>
<tr>
<td>Phobos RAAN</td>
<td>208 deg</td>
</tr>
<tr>
<td>Phobos argument of periapsis</td>
<td>150 deg</td>
</tr>
<tr>
<td>Phobos orbital period</td>
<td>0.319 days</td>
</tr>
<tr>
<td>Phobos Mass</td>
<td>$1.07 \times 10^{16}$ kg</td>
</tr>
<tr>
<td>Phobos Albedo</td>
<td>0.071</td>
</tr>
<tr>
<td>Mars gravitational parameter</td>
<td>43824 km$^3$/s$^2$</td>
</tr>
<tr>
<td>Mars $J_2$</td>
<td>0.0019</td>
</tr>
<tr>
<td>Mars equatorial radius</td>
<td>3394 km</td>
</tr>
</tbody>
</table>

selected spacecraft camera is 114 km. Using an additional 5 km tolerance, the minimum flyby altitude for the spacecraft swarm is located at $h'_f = 109$ km. The spacecraft FoV required to image the view sphere at the required elevation angle is about $\eta_D = 13.6$ deg. The camera and flyby parameters are summarized in Table 5.4.
Table 5.3: Shape model parameters of the Phobos model used in the current work.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Shape Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Range Sphere</td>
</tr>
<tr>
<td># Faces</td>
<td>5120</td>
</tr>
<tr>
<td>Radius (km)</td>
<td>8.12</td>
</tr>
<tr>
<td>Area (km²)</td>
<td>828</td>
</tr>
</tbody>
</table>

Table 5.4: Flyby and camera parameters required to design the Phobos Flyby.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required flyby altitude, $h_f$</td>
<td>114 km</td>
</tr>
<tr>
<td>Altitude tolerance, $\Delta h$</td>
<td>5 km</td>
</tr>
<tr>
<td>Designed flyby altitude, $h'_f$</td>
<td>109 km</td>
</tr>
<tr>
<td>Required spacecraft FoV, $2\theta_D$</td>
<td>13.6 deg</td>
</tr>
</tbody>
</table>

Sensitivity Study Parameters

To study the sensitivity of the coverage of swarm do different uncertainties, the optimal design is perturbed using different categorical perturbations described in Chapter 4. Here, four categories of perturbations are considered: Spacecraft outage, error in the knowledge of Phobos’ location, error in encounter shaping, and sensitivity of coverage performance to the high fidelity dynamics. In these studies, the objective is to determine the worst-case perturbation that makes the coverage drop below a threshold $P_{thr}$.

5.3 Numerical Simulations

We now demonstrate the results of the design of the optimal swarm, to map the surface of Phobos through Martian flybys. We begin with the description of the Earth to Mars interplanetary trajectory design of the swarm. Following this, the mapping performances of the designed swarm is examined. Additionally, the coverage sensitivity of the designed swarm to different gene-based and dynamical perturbations are examined.
Trajectory Design

The optimization problem in Equation 5.1 is solved using a Genetic Algorithm (GA) optimizer (Conn et al., 1991). The parameters described in Table 5.5 are passed as input arguments to the optimizer.

Table 5.5: Input parameters to design the Earth-Mars trajectory of the Phobos mapping swarm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{L,\text{min}}$,$D_{L,\text{max}}$</td>
<td>Jan 1st 2020 – Dec 31st 2020</td>
</tr>
<tr>
<td>$D_{A,\text{min}}$,$D_{A,\text{max}}$</td>
<td>May 1st 2020 – Dec 31st 2021</td>
</tr>
<tr>
<td>$C_{3,\text{max}}$</td>
<td>20 km$^2$/s$^2$</td>
</tr>
<tr>
<td>$ToF_{\text{max}}$</td>
<td>200 days</td>
</tr>
</tbody>
</table>

Figure 5.8: Variation of the trajectory fitness for the global surface mapping mission concept across different GA populations generations showing the individual optimization trials (left), and their statistical distribution (right).

Optimization Performance  The results of 5 GA optimization trials of the trajectory optimization are presented in 5.8. Each generation of a trial evaluated 1000
trajectory design genes. The subsequent search generations were populated with 20 % elite genes, while 80 % of the remaining genes were achieved through the crossover operation. A uniform stochastic selection criterion was used to select the parents for crossover and mutation operations. All searches terminated when the constraints were satisfied, and the minimum value of the objective function stalled for 100 generations. The final generations of all 5 trials identified 2751 trajectory design solutions whose figures of merit are presented in Figure 5.9. The selected trajectory had a $C_{3,E} = 16.7 \text{ km}^2/\text{s}^2$ at launch and a heliocentric cruise duration of $ToF = 200$ days.

Figure 5.9: Figures of merit of the feasible trajectory genes identified by the 5 GA optimization trials.

Optimal Trajectory As seen in Figure 5.8, the optimal Earth to Mars trajectory starts from Earth on 13th August 2020, and will reach Mars 3rd March 2021. The 200 day cruise from Earth to Mars will require a $C_3$ energy of 19.5 km$^2$/s$^2$, and would arrive at Earth, with an excess velocity magnitude of 2.47 km/s. The optimality of the trajectory is presented in Figure 5.10, along with key trajectory
design parameters. The asymptote angles presented here are expressed in Heliocentric ecliptic reference frame. The optimal Earth-Mars trajectory of the swarm is presented in Figure 5.11.

Figure 5.10: Optimal solution shown on the Earth-Mars Pork-Chop plot, along with the trajectory design parameters.

Figure 5.11: Optimal Earth to Mars trajectory of the Phobos mapping swarm.

Swarm Design

The trajectory design provided the optimal arrival asymptote, presented in Figure 5.10. Once inside the sphere of influence of Mars, the swarm will use these asymptotes to flow on trajectories, which result from solving the MINLP optimization problem in Equation 5.19. The true anomaly of Phobos, offset by $\Delta f_0$ from the
first encounter true anomaly in the swarm. The spacecraft true anomalies are then obtained by solving the Kepler’s equation iteratively. The parameters input to the swarm design problem are presented in Table 5.6.

Table 5.6: Input parameters to solve the swarm design problem.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum periapsis altitude, $h_{p,min}$</td>
<td>300 km</td>
</tr>
<tr>
<td>Initial true anomaly offset, $\Delta f_0$</td>
<td>1 deg</td>
</tr>
<tr>
<td>Number of random target initializations, $N_{mon,1}$</td>
<td>20</td>
</tr>
<tr>
<td>Maximum spacecraft visits, $N_{1,max}$</td>
<td>5</td>
</tr>
<tr>
<td>Maximum spacecraft per visit, $N_{2,max}$</td>
<td>5</td>
</tr>
</tbody>
</table>

**Optimization Performance**  The automated swarm design problem presented in Equation 5.19 is solved using the modified, real-valued mixed-integer GA optimizer (Deep et al., 2009; Deb, 2000). The results of 5 Genetic Algorithm trials of the swarm optimization are presented in Table 5.7. The GA generation of each trial spanned 50 swarm design genes, out of which elite genes span 20 %. The remainder of the population was spanned by 80 % crossover children, and 20 % mutation children. Each GA trial was configured to run for a maximum of 300 generations, during which the optimizer run was terminated when the best fitness stalled for 100 generations. As seen here, the trials identified a minimum swarm size of 5 spacecraft.

The final generations of all 5 trials identified 199 feasible swarm solutions whose performance figures of merit are illustrated in Figure 5.13. The selected gene had a minimum swarm size of $N_{sw} = 5$ spacecraft along with a maximum coverage figure of merit $P_{FoM,map} = 91.3 \%$.

**Swarm Performance**  The selected optimal swarm gene is presented in Table 5.7. As seen here, the selected optimal design consists of a swarm of $N_{sw} = 5$ spacecraft that visit Phobos individually in 5 different locations. The hyperbolic trajectories of the 5 spacecraft swarm are presented in Figure 5.14 along with the first designed Phobos encounter. The Phobos true anomalies, along with spacecraft
azimuth and elevation angle, can be noted from Table 5.7. The minimum Martian periapsis altitude of the swarm was noted as 1269 km. The encounter velocities of the spacecraft with respect to Phobos were found to range between $2.08 - 5.90 \text{ km/s}$.

The cumulative coverage performance of the swarm over the nominal and Dual Sphere models of Phobos noted as a result of 20 randomly oriented initializations is presented in Table 5.8. As seen here, the minimum expected coverage is noted as 91.3 %, which does exceed the cumulative coverage requirement of 90 % defined in Table 5.1.

The dynamical evolution of the surface coverage over different shape models of Phobos, noted from a random initialization, is presented in Figure 5.15. The coverage shown here is evaluated for one orbital day of Phobos. As noted from Figure 5.15, the first three encounters each cover $30 - 40$ % of the surface of Phobos, while the remaining two encounters cover about $15 - 16$ % of the surface. The coverage trial described here resulted in a cumulative coverage of 91.3 % over the nominal and view sphere model of Phobos along with a 93.8 % coverage of its range sphere.
Figure 5.13: Figures of merit of the feasible swarm genes identified by the 5 GA optimization trials.

The resulting coverage patterns are presented in Figure 5.16.

**Seed Spacecraft Design**

The cost of the swarm mission concept is estimated assuming the node spacecraft are populated using the subsystems and cost models described in Appendix A. Since the mission concept is designed as a sequence of flybys, the spacecraft will not expend any deterministic maneuver fuel. A conservative $\Delta v$ of 100 m/s per spacecraft as a baseline to estimate any probabilistic corrective maneuvers to account for uncertainties and perturbations. The corresponding fuel mass for the seed spacecraft is noted as 0.91 kg using Equation A.1, which results in a net spacecraft mass of 19.8 kg. The design parameters and cost breakdown of the seed spacecraft are presented in Table 5.9. The bus and operations costs presented here are calculated using the SSCM10 model in Table A.3, while the net payload cost is calculated using the NICM model in Table A.4. The best-case launch vehicle for the spacecraft was noted as Falcon 9 rocket, whose interplanetary launch cost was noted as 0.833 ± 0.25 LY$M$. The
Table 5.7: Design variables of the selected optimal swarm gene.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_w$</td>
<td>5</td>
</tr>
<tr>
<td>$N_{ij}$</td>
<td>1 1 1 1 1</td>
</tr>
<tr>
<td>$f_{ij}$ (deg)</td>
<td>66.2, 152, 258, 288, 343</td>
</tr>
<tr>
<td>$[\theta_x, \theta_z]$ (deg)</td>
<td>191, 21.4, 267, 40.2, 59.9, -12.7, 269, 55.3, 190, 53.6</td>
</tr>
</tbody>
</table>

Table 5.8: Cumulative coverage statistics noted from 20 random initial orientations of Phobos.

<table>
<thead>
<tr>
<th>Model</th>
<th>Cumulative Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal model</td>
<td>$91.5 \pm 0.23 %$</td>
</tr>
<tr>
<td>View sphere</td>
<td>$91.4 \pm 0.03 %$</td>
</tr>
<tr>
<td>Range sphere</td>
<td>$93.8 \pm 0.03 %$</td>
</tr>
</tbody>
</table>

launch year costs associated with a seed spacecraft mission is $0.39.0 \pm 1.2 \text{ LY$M.}$
Since the launch year of the mission is 2020, the inflation factor from 2010 to 2020 is noted as 1.217 from Table A.2.

**Swarm Cost** The cost of the swarm mission in the launch year 2020 is noted using Equation A.3. The cost breakdown of the swarm mission is presented in Table 5.10. As seen here the total space segment cost of the $N_{sw} = 5$ spacecraft swarm is noted as $80.5 \pm 2.5 \text{ LY$M.}$

5.4 **Sensitivity Analysis**

This section presents the sensitivity of coverage performance to the swarm to perturbations from the optimal design. The sensitivity of cumulative surface coverage
Figure 5.14: The arrangement of the hyperbolic trajectories in the designed optimal swarm (top) along with a visualization of a sample encounter with Phobos (bottom).

to spacecraft outages, Moon, and spacecraft locations are examined. Finally, the sensitivity of cumulative coverage to varying levels of dynamic fidelity is noted.
Figure 5.15: Dynamical evolution of the surface coverage of the nominal and Dual Sphere shape models of Phobos noted during its orbital period.

**Sensitivity to Outages**

The parameters input to studying the coverage sensitivity to spacecraft outages is presented in Table 5.11. The results of the sensitivity analysis are presented in Figure 5.17.

As seen in Figure 5.17, the coverage performance decays as the operational percentage reduces. When the swarm is operating at 80% (4 spacecraft), the expected coverage still over all three spheres is still above $P_{thr}$, after which the loss of any additional spacecraft will fail to meet the minimum coverage requirement.

**Sensitivity to Moon’s location**

The input parameters to study the coverage sensitivity to the Moon’s initial position is presented in Table 5.12. The results of the simulations are presented in Figure 5.18. It can be noted from Figure 5.18 that the performance of the swarm is more sensitive to true anomaly lags (negative offsets) than leads. The coverage of all three models (nominal model, view sphere, and the range sphere) is shown to meet the
design requirement of 90 % for a maximum lag of $-0.15 \, \text{deg}$, and a maximum lead of $0.23 \, \text{deg}$. Similarly, the minimum requirement of 80 % is satisfied for a maximum true anomaly lag of $-0.44 \, \text{deg}$, while in the case of true anomaly lead, the minimum coverage requirement was met for a maximum value of 1.00 deg. Thus a conservative bound of $\pm 0.44 \, \text{deg}$ can be used as a critical threshold on the location uncertainty of Phobos.

**Sensitivity to Rendezvous location**

The parameters for studying the sensitivity of the swarm’s coverage to their rendezvous location are presented in Table 5.13. The results of the study are presented in Figure 5.19. It can be noted from Figure 5.19, that the coverage of the swarm is reasonably robust to uncertainties in the rendezvous location of the spacecraft. For amplitudes below 6.93 km, the coverage of all three models is shown to meet the designed requirement of 90 %. The coverage of all three models is above the minimum coverage requirement for a maximum perturbative amplitude of 49.5 km.
Table 5.9: Design parameters and cost breakdown of the seed spacecraft in the designed swarm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta v_{net}$ (m/s)</td>
<td>100</td>
</tr>
<tr>
<td>$m_{fuel}$ (kg)</td>
<td>0.91</td>
</tr>
<tr>
<td>$m_{wet}$ (kg)</td>
<td>19.8</td>
</tr>
<tr>
<td>Launch year</td>
<td>2020</td>
</tr>
<tr>
<td>Interplanetary payload cost (LY$$M)</td>
<td>$2.72 \pm 0.71$</td>
</tr>
<tr>
<td>Interplanetary bus and operations cost (LY$$M)</td>
<td>$14.9 \pm 0.85$</td>
</tr>
<tr>
<td>Launch cost (LY$$M)</td>
<td>$0.39 \pm 0.12$</td>
</tr>
</tbody>
</table>

Table 5.10: Cost breakdown of the $N_{sw} = 5$ spacecraft swarm mission inflated to the launch year 2020.

<table>
<thead>
<tr>
<th>Component</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission cost (LY$$M)</td>
<td>$78.5 \pm 2.5$</td>
</tr>
<tr>
<td>Launch cost (LY$$M)</td>
<td>$2.01 \pm 0.27$</td>
</tr>
<tr>
<td>Total space segment cost (LY$$M)</td>
<td>$80.5 \pm 2.5$</td>
</tr>
</tbody>
</table>

Perturbations greater than this amplitude are shown to cause rapid fluctuations in the coverage, as seen here. Therefore, the swarm encounters the target with a positional uncertainty as large as 49.5 km and still meet the minimum performance requirement.

**Dynamical Sensitivity**

To study the coverage performance of the swarms when subjected to perturbative dynamical accelerations, the motion of the spacecraft and Phobos is propagated using four different levels of propagators. Each level adds an external source of perturbative acceleration to the equation of motion. At each fidelity level, the swarm was subjected to $N_{mon,2} = 200$ random initializations of Phobos. The objective here is to note how the external perturbations impact the surface coverage of the swarm. The magnitudes of different perturbative accelerations that act on a space-
Table 5.11: Input parameters to study the sensitivity of the swarm to spacecraft outages.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum coverage requirement, $P_{thr}$ (%)</td>
<td>80</td>
</tr>
<tr>
<td>Minimum operational percentage, $P_{opr}$ (%)</td>
<td>20</td>
</tr>
<tr>
<td>Study resolution (# spacecraft)</td>
<td>1</td>
</tr>
</tbody>
</table>

![Graph showing surface coverage vs. % of operational spacecraft](image)

Figure 5.17: Variation in the coverage of Phobos with spacecraft outages in the swarm.

craft in the swarm during the simulation time are presented in Figure 5.20. It can be noted from Figure 5.20 that the dominant source of acceleration is the spherical two-body gravity of Mars, which is then followed by its $J_2$ oblateness effect. The third body acceleration is shown to reach a peak value of 0.003 m/s$^2$ when the spacecraft encounters Phobos and is shown to quickly decay to about $10^{-9}$ m/s$^2$ as the spacecraft flyby. The contribution of the solar radiation pressure (SRP) was noted to be reasonably constant on the order of $10^{-8}$ m/s$^2$. The coverage performance when subjected to different levels of perturbative accelerations, is presented in Figure 5.21.
Table 5.12: Input parameters to study the sensitivity of coverage to the location of the moon.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum coverage requirement, ( P_{thr} ) (%)</td>
<td>80</td>
</tr>
<tr>
<td>Number of random target initializations, ( N_{mon,2} )</td>
<td>200</td>
</tr>
<tr>
<td>Range of true anomaly error, ( \Delta f_{op} ) (deg)</td>
<td>[-1.5, 1.5]</td>
</tr>
<tr>
<td>Study size (# true anomaly offsets)</td>
<td>201</td>
</tr>
</tbody>
</table>

Figure 5.18: Variation of the cumulative coverage with uncertainty in the orbital location of Phobos.

As seen in Figure 5.21, the performance is shown to meet the minimum coverage requirement across all the simulated levels of fidelity considered in Table 4.1. Specifically, the performance is unaffected by the gravity of Phobos, and therefore the performance of the swarm exceeded the actual designed 90 % in the case of all three shape models. The addition of the Martian \( J_2 \) gravity model brought down the minimum expected coverage to 89.6 %, which was noted for the view sphere. Finally, the inclusion of SRP acceleration is also not shown to cause any effect in the swarm’s coverage performance. While these models show that the swarm can still meet the minimum performance requirement despite the perturbations, additional
Table 5.13: Input parameters to study the sensitivity of coverage to the rendezvous location of the swarm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum coverage requirement, ( P_{thr} ) (%)</td>
<td>80</td>
</tr>
<tr>
<td>Number of random target initializations, ( N_{mon,2} )</td>
<td>200</td>
</tr>
<tr>
<td>Maximum perturbation amplitude, ( \Delta R ) (km)</td>
<td>100</td>
</tr>
<tr>
<td>Study size (# perturbation amplitudes)</td>
<td>100</td>
</tr>
</tbody>
</table>

Figure 5.19: Variation of the coverage of the swarm with different amplitudes of spacecraft rendezvous uncertainty.

corrective maneuvers can be placed to compensate for these effects.

5.5 Discussion

This chapter presented a case study where the Automated Swarm Designer module of IDEAS was applied to design and analyze a mission concept to map a planetary moon’s surface. We began with a generalized objective, where the surface of the Moon has to be mapped with a required ground resolution. The design problems corresponding to the spacecraft, swarm, and trajectory of the mission concept were
then formulated. The algorithms were demonstrated by designing a global surface mapping mission concept to the Martian moon Phobos. The appropriate design modules of IDEAS then solved the formulated problems, and a 4 spacecraft swarm was selected as the near-optimal mission concept. The spacecraft in the swarm approach Phobos on a common arrival asymptote at Mars, given by the trajectory design module. Following this, the cost of the swarm mission was estimated using publicly available cost estimating relations. Finally, the surface coverage sensitivities to different perturbations such as spacecraft outages, uncertainty in the Moon’s location, uncertainty in spacecraft rendezvous location, and dynamic perturbations were examined. In these cases, the critical perturbation/uncertainty threshold that would ensure that the swarm can meet a minimum coverage requirement is noted.
Figure 5.21: Variation of coverage with different levels of motion propagation fidelity.
CHAPTER 6

Case Study-2: Moon Reconnaissance Using Resonant Co-orbits

While the previous chapter demonstrated that spatial coverage requirements to planetary moons could be met using hyperbolic flybys of the central planet, this chapter will demonstrate the application of swarms to both spatial and temporal reconnaissance mission concepts. The spacecraft in the swarm will co-orbit the central planet along with the moon, which will be designed to provide encounters based on a resonant pattern. The co-orbits are designed such that the spacecraft use two deterministic maneuvers to enter into them. First, a planar orbit insertion maneuver uses the aerobraking maneuver at the central planet to be captured into the target orbit; second, an orientation change maneuver that rotates the incoming $\vec{V}_{\infty,2}$ vector in order to facilitate a planar capture. The algorithms described are then demonstrated using numerical case studies to the Martian moon, Deimos. We begin this chapter by reviewing state-of-the-art co-orbital missions to small bodies. The mission architecture describing the trajectory, swarm, and spacecraft behaviors are then presented, followed by case studies of designing global surface mapping, and RoI observation mission concepts to Deimos, followed by their coverage sensitivity analysis.

6.1 Related Work

The motivation to explore planetary moons was described in Chapter 5. Therefore, in this section, the emphasis is placed on the co-orbital exploration of planetary moons. At the state-of-the-art, several missions have used planetary co-orbits to explore the planetary moons. Notable examples include exploration of the Martian moons by the Mars Reconnaissance Orbiter (MRO) (Zurek and Smrekar, 2007), and Saturn’s moons by the Cassini mission (Elachi et al., 2004). In the near future,
118
dedicated missions to planetary moons such as the MMX (Campagnola et al., 2018),
JUICE (Grasset et al., 2013), and Europa Clipper mission (Phillips and Pappalardo,
2014) have been planned. Resonant Co-orbits were studied as viable trajectories to
explore Europa (Buffington, 2014).

Figure 6.1: Images of the martian moon Deimos captured by the HiRISE camera of
the MRO spacecraft, at a resolution of about 20 m/px.

In this work, Deimos, the second moon of Mars (Shor, 1975), is used as the target
body to design simulated case studies. Deimos has been actively researched as an
in-situ resource utilization (ISRU) base around Mars (Abreu, 2018). Additionally,
Deimos has also been studied as a suitable location for a Mars base camp, as it
has areas on its surface that have better illumination, and access to Earth, than
Phobos (Pratt and Hopkins, 2011). The highest resolution images of Deimos include:
portions of Deimos imaged at about 3 m/px resolution by the Viking 2 spacecraft
(Duxbury and Veverka, 1978), and about 20 m/px images taken by the HiRISE
camera of the Martian Reconnaissance Orbiter (MRO) (McEwen et al., 2010).

Existing work on co-orbit design focused on the derivation of orbital elements
assuming equatorial and circular moon orbits (Conte, 2014). An essential aspect of
the trajectory design described in this chapter is the use of aerobraking at the cen-
tral planet (Spencer and Tolson, 2007). When using the aerobraking maneuver, the spacecraft orbits whose periapsis passes through the atmosphere, use the drag force to reduce its apoapsis radius (Vallado, 2013). While this maneuver is used to save fuel, the participating spacecraft trade this with mission time before deployment (Smith Jr and Bell, 2005). This chapter builds on the existing work by developing automated design architectures where the spacecraft in the swarm enter into resonant co-orbits around the central planet to explore the moon with space-based and time-based coverage restrictions.

6.2 Mission Concept Modeling

In this section, the models and parameters corresponding to the design of the two mission concepts are presented. The coverage requirements are presented first, followed by the formulation of the automated trajectory and swarm design problems. The models used in this chapter were also published in Nallapu and Thangavelautham (2020a, 2021).

Coverage Requirements

The coverage requirements associated with the two mission concepts are listed as follows:

**Global Surface Mapping** In these missions, the spacecraft are required to meet a minimum cumulative surface coverage requirement of 90%, with a maximum ground resolution of 1 m in one orbit of the moon.

**RoI Observation** The swarm should observe a target region of interest (RoI) on the surface of Deimos which located at longitude 45 deg and latitude 45 deg and has a square angular spread of 15 deg. The RoI should be imaged at a maximum resolution of 1 m and be observed for a minimum duration of 20 mins. The minimum cumulative coverage requirement of the RoI is 90% in one orbit of the Deimos. The
definition parameters corresponding to the two missions are summarized in Table 6.1.

Table 6.1: User input mission requirements for the Deimos mapping swarms.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground resolution requirement, $x_R$</td>
<td>1 m</td>
</tr>
<tr>
<td>Minimum elevation angle, $\varepsilon_R$</td>
<td>5 deg</td>
</tr>
<tr>
<td>Global Coverage requirement, $P_{G,R}$</td>
<td>90 %</td>
</tr>
<tr>
<td>Rol location [$\phi_x, \phi_y$]</td>
<td>[45, 45] deg</td>
</tr>
<tr>
<td>Rol angular spread, $\Phi_E$</td>
<td>15 deg</td>
</tr>
<tr>
<td>Rol Coverage requirement, $P_{G,R}$</td>
<td>95 %</td>
</tr>
<tr>
<td>Rol Coverage duration requirement, $T_R$</td>
<td>20 mins</td>
</tr>
</tbody>
</table>

Camera and Flyby Parameters

The seed spacecraft bus is maintained the same as Chapter 5, which was obtained using the inventory described in Appendix A. As a result, the imaging radius and half FoV computed using Equations 3.3, and 3.2 are also 120 km and 3.04 deg respectively. However, the fuel mass in the case of the orbiting swarms has to account for the transition from the incoming hyperbolic trajectory into the designed co-orbits. This transition is split into two maneuvers: plane change of the hyperbola to facilitate an apoapsis encounter, and planar orbit insertion maneuver where the spacecraft on the hyperbolic trajectory is captured into the resonant co-orbit. The planar maneuver $\Delta v$ cost is estimated using the trajectory designer module, which is used to estimate the worst-case orbit insertion $\Delta v$, where the spacecraft are assumed to use an aeroassist from the central planet to be captured into the co-orbits. The maximum plane change $\Delta v$ in the swarm is used to estimate the fuel required for this maneuver. The two maneuvers are then used to estimate the fuel requirements and cost of the seed spacecraft.
Trajectory Design

As described in the previous section, the current case study's trajectory design problem will minimize the worst-case magnitude of the planar orbit insertion $\Delta v$ required to capture the spacecraft from the incoming hyperbolic trajectory into the designed co-orbit using an aeroassist from the central planet. Therefore, the design gene, in this case, has to account for the optimal aerobraking maneuver, in addition to the launch window problem described in Chapter 5. This section describes the construction of the aerobraking maneuvers, followed by the formulation of the automated optimal trajectory design algorithms.

Launch Window Design  The launch window design in the case of the co-orbiting swarms is similar to the design problem in Chapter 5. The trajectory must meet the bounds on $v_{\infty,2}$ and $C_{3,E}$. However, since the spacecraft in the interplanetary trajectory will be captured into orbits for aerobraking, and then subsequently into the resonant orbits, the trajectory design will produce trajectories that minimize the worst-case $\Delta v$ required to be captured into the final resonant orbits. The behaviors of the aerobraking maneuver and resonant co-orbits corresponding to the trajectory design are described as follows:

Aerobraking Maneuver Behaviour  The aerobraking maneuver will occur in four stages (Vallado, 2013), during which the spacecraft perform impulsive tangential maneuvers on the apse lines:

- HEO Capture
- Walk In
- Main Phase
- Walk Out

These stages can be described as follows.
**HEO Capture**  Upon arrival at the target planet, the spacecraft will perform an initial tangential maneuver at the planet’s periapsis to get captured into a highly eccentric orbit (HEO), as shown in Figure 6.2. The maneuver cost \( \Delta V \) required for the HEO capture maneuver (Vallado, 2013) is given by

\[
\Delta v_C = \sqrt{v_{\infty}^2 + \frac{2\mu_P}{h_{p,0} + R_P}} - \sqrt{\frac{\mu_P (1 + e_0)}{h_{p,0} + R_P}}
\]  

(6.1)

Where \( R_P \) is the central planet radius, \( h_{p,0} \) is the periapsis altitude of the hyperbola, and \( e_1 \) is the eccentricity of the achieved HEO.

**Walk In**  Once captured in an HEO, the spacecraft will perform a maneuver at apoapsis to reduce periapsis to pass through the atmosphere, as shown in Figure 6.3. Let \( h_{AB} \) be the altitude where the planet has sufficient atmosphere to assist the spacecraft. The walk in burn magnitude is written as

\[
\Delta v_{WI} = \sqrt{\frac{\mu_P (1 - e_1)}{r_{a,1}}} - \sqrt{\frac{\mu_P (1 - e_2)}{r_{a,1}}}
\]  

(6.2)
Figure 6.3: Walk in maneuver performed at the apoapsis of the HEO to initiate the main phase of aerobraking.

Where $r_{a,1}$ is the apoapsis radius of the HEO. The eccentricity $e_2$ denotes the eccentricity of the orbit passing through the upper atmosphere. The eccentricities $e_1$ and $e_2$ can be computed from the elliptical geometry since the orbital apses are known (Vallado, 2013).

**Main Phase** Following the Walk in phase, the spacecraft enters the main phase, where apoapsis altitude is reduced, as shown in Figure 6.4. The apoapsis decay is caused due to the atmospheric drag experienced during the periapsis passage. The aerobraking phase ends when the spacecraft arrives at the target apoapsis altitude of the science orbit. Therefore, the spacecraft in the main phase is assumed not to expend any additional fuel. However, it is pointed out here that practical implementations of the main phase involve performing small maneuvers to hold the periapsis at its planned value (Spencer and Tolson, 2007).

**Walk Out** At the end of the main aerobraking phase, the spacecraft is located on its science orbit’s apoapsis. The spacecraft will perform a walk out burn to raise its periapsis altitude from the aerobraking altitude $h_{AB}$ to the periapsis altitude of its resonant orbit, as shown in Figure 6.5. Let $r_{a,B}$ denote a vector that describes
Figure 6.4: Main phase of the aerobraking maneuver where the apoapsis altitude of the orbit is reduced upon passing through the atmosphere of the central planet.

the maximum and minimum boundary values of the apoapsis magnitude, described by eccentricity boundary values $e_B$ of the targeted resonant co-orbit. The boundary values of the walk-out burn can be expressed as:

$$\Delta v_{WO,B} = \sqrt{\frac{\mu_P (1 - e_{2,B})}{r_{a,B}}} - \sqrt{\frac{\mu_P (1 - e_B)}{r_{a,B}}} \quad (6.3)$$

Where $e_{2,B}$ corresponds to the eccentricity of the orbit whose apoapsis is at $r_{a,B}$ while the periapsis altitude is at $h_{AB}$. The bounds on the apoapsis value can be used to estimate the maximum $\Delta V$ for the final Walk-Out burn. Additionally, a
constraint placing the minimum periapsis altitude above the desired value avoids atmospheric effects from the central planet on the spacecraft’s sensing orbit.

**Planar Orbit Insertion**  The worst case maneuver cost associated with planar orbit insertion using aerobraking can now be expressed as:

$$\Delta v_{OI,max} = \Delta v_C + \Delta v_{WI} + \max(\Delta v_{WO,B})$$  \hspace{1cm} (6.4)

**Resonance Co-orbit Behaviour**  The resonant behaviour of the spacecraft ensures that the spacecraft encounters with the moon will repeat after every \( p \) orbits of the moon or after every \( q \) orbits of the spacecraft. This allows us to determine the semi-major axis of the resonant co-orbit using Kepler’s third law as (Vallado, 2013):

$$ \left( \frac{a_T}{a_{Sw}} \right)^{\frac{3}{2}} = \frac{p}{q} $$  \hspace{1cm} (6.5)

Where \( a_T \) and \( a_{Sw} \) are the semi-major axis of the moon and the spacecraft, respectively. It is noted from Equation 6.5 that \( p, q, \) and \( a_{Sw} \) do not depend on the location of the moon. Since the designed spacecraft encounters with the moon occur on the apoapsis, the apoapsis vector \( ^N \bar{R}_{a,i} \) for co-orbiting swarms is expressed as

$$ ^N \bar{R}_{a,i} = ^N \bar{R}_{Tj} + ^N \bar{R}_{Tji} $$  \hspace{1cm} (6.6)

where \( ^N \bar{R}_{Tj} \) is the position vector of the moon to the central planet at a true anomaly \( f_{v,j} \), and \( ^N \bar{R}_{Tji} \) is the designed encounter point is given by

$$ ^N \bar{R}_{Tji} = r_f \begin{bmatrix} \cos \theta_{x,i} \cos \theta_{y,i} & \sin \theta_{x,i} \cos \theta_{y,i} & \sin \theta_{y,i} \end{bmatrix}^T $$  \hspace{1cm} (6.7)

It should be noted that while the left-hand side of Equation 6.7 is the spacecraft location with respect to the moon in the \( T \) frame, we set \( ^T \bar{R}_{Tji} = ^N \bar{R}_{Tji} \) following the assumption that the \( T \) is constructed by a pure translation of the \( N \) frame. The eccentricity of the spacecraft co-orbit \( e_{sw,i} \) and its periapsis altitude \( h_{p,i} \) can then be determined as (Vallado, 2013)
\[ e_{sw,i} = \left( \frac{r_{a,i}}{a_{sw}} \right) - 1 \] (6.8)

and

\[ h_{p,i} = a_{sw} (1 - e_{sw,i}) - R_{Pl} \] (6.9)

Where \( r_{a,i} \) is the magnitude of \( N \bar{R}_{a,i} \), and \( R_{Pl} \) is the radius of the central planet.

**Boundary Values** Using Equation 6.6, \( r_{a,i} \) is bounded through the triangle inequality (Marghitu and Dupac, 2012) as

\[ \min (r_{T,j}) - r_f \leq r_{a,i} \leq \max (r_{T,j}) + r_f \] (6.10)

Where \( r_{T,j} \) denotes the magnitude of \( N \bar{R}_{T,j} \). The values of \( \min (r_{T,j}) \), and \( \max (r_{T,j}) \) occur at the moon’s periapsis (\( f_{v,j} = 0 \) deg), and apoapsis (\( f_{v,j} = 180 \) deg) respectively (Vallado, 2013). Let \( r_{a,B} \) denote the set of minimum and maximum boundary values of \( r_{a,i} \), which allows us to compute the set of bounding eccentricities \( e_B \) using Equation 6.8. Additionally, the minimum periapsis altitude of the co-orbit \( h_{p,min} \) is also computed using Equation 6.9. The boundary values are used to evaluate the feasibility of a specified resonance.

**Automated Trajectory Design** The trajectory design problem for the co-orbiting swarm can now be expressed as follows:

\[
\begin{align*}
\min J_{Tr} &= \Delta v_{OI,max} \\
\text{s.t.} \quad C_3 &\leq C_{3,max} \\
ToF &\leq ToF_{max} \\
v_{\infty,2} &\leq v_{\infty,max} \\
\min(h_{p,i}) &\geq h_{p,min}
\end{align*}
\] (6.11)

The gene map of the trajectory optimization problem in Equation 6.11 is shown in presented in Figure 6.6.
Figure 6.6: Gene map of the trajectory gene corresponding to the co-orbiting swarm problem.

**Swarm Design**

The semi-major axis of the becomes a fixed design parameter, as noted through the Trajectory design problem. Furthermore, since all encounters are designed to occur on the orbital apoapsis, the spacecraft true anomaly during the co-orbits also becomes a fixed parameter, with a value of 180 def. However, the remaining COEs still need to be defined such that these resonant apoapsis encounters of the spacecraft occur at a specified location on the moon’s orbit, i.e., when it is located at a true anomaly \( f_{v,j} \). The orbital eccentricity corresponding to such an encounter can be computed using Equation 6.8. The task at hand then is to compute the orientation elements such: the right ascension of the ascending node (RAAN), inclination, and argument of periapsis, such that \( N \bar{R}_{a,i} \) of spacecraft \( i \) passes through the encounter location specified by \( f_{v,j} \). Since, for planar orbit insertions, the orientation orbital elements are constrained by the hyperbolic tube defined by the incoming excess velocity vector \( N V_{\infty,2} \), and orbits whose orientation elements are not supported by the tube require an orientation change maneuver (Vallado, 2013). This constraint emphasizes the need to couple the co-orbit design of the swarm with their interplanetary trajectory. Here the Triad algorithm (Schaub and Junkins, 2013) is used to determine the perifocal to \( N \) frame rotation matrix \([NP_i]\) of spacecraft \( i \), and to estimate the maneuver cost associated with the orientation change \( \Delta v_{OC,i} \).

**Co-orbit Orientation** Since \( N V_{\infty,2} \), and the periapsis altitude of the arrival hyperbola \( h_{p,0} \) are known through trajectory design. The \( N \) frame vectors \( N \bar{R}_{a,i} \) and

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Launch date</th>
<th>Arrival date</th>
<th>Hyperbolic Periapsis altitude</th>
<th>HEO Eccentricity</th>
<th>Resonance Parameters # Target Orbits</th>
<th># spacecraft Orbits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>( D_L )</td>
<td>( D_A )</td>
<td>( h_{p,0} )</td>
<td>( e_1 )</td>
<td>Integer ( p )</td>
<td>Integer ( q )</td>
</tr>
<tr>
<td>Range</td>
<td>Real ([D_{L,\text{min}}, D_{L,\text{max}}])</td>
<td>Real ([D_{A,\text{min}}, D_{A,\text{max}}])</td>
<td>Real ([h_{p,\text{min}}, \infty])</td>
<td>Real ([0, e_{1,\text{max}}])</td>
<td>Integer ( [p_{\text{min}}, p_{\text{max}}] )</td>
<td>Integer ( [q_{\text{min}}, q_{\text{max}}] )</td>
</tr>
</tbody>
</table>
$N V^{-}_{\infty,2}$ can now be resolved in the perifocal frame $P$ of the planet. Equation 5.14 describes the perifocal resolution of the incoming asymptote, while the apoapsis vectors can be resolved in the perifocal frame as (Vallado, 2013).

\[ P \bar{R}_{a,i} = r_{a, [-1 \ 0 \ 0]^T} \]  

(6.12)

A possible rotation matrix $[NP]$ can now be defined from the Triad algorithm as (Schaub and Junkins, 2013)

\[ [NP] = [\hat{n}_1 \ \hat{n}_2 \ \hat{n}_3]\hat{p}_1 \ \hat{p}_2 \ \hat{p}_3]^T \]  

(6.13)

where the unit basis vectors are defined as

\[
\begin{align*}
\hat{n}_1 &= \frac{N \bar{R}_{a,i}}{r_{a,i}} \\
\hat{n}_2 &= \frac{N \bar{R}_{a,i} \times N V^{-}_{\infty,2}}{|N \bar{R}_{a,i} \times N V^{-}_{\infty,2}|} \\
\hat{n}_3 &= \hat{n}_1 \times \hat{n}_2 \\
\hat{p}_1 &= \frac{P \bar{R}_{a,i}}{r_{a,i}} \\
\hat{p}_2 &= \frac{P \bar{R}_{a,i} \times P V^{-}_{\infty,2}}{|P \bar{R}_{a,i} \times P V^{-}_{\infty,2}|} \\
\hat{p}_3 &= \hat{p}_1 \times \hat{p}_2
\end{align*}
\]  

(6.14)

The RAAN $\Omega_i$, inclination $in_i$, and argument of periapsis $\omega_{p,i}$ can be extracted from $[NP]$ (Schaub and Junkins, 2013) as

\[
\begin{align*}
\tan \Omega_i &= \frac{[NP](3,1)}{-[NP](3,2)} \\
\cos in_i &= [NP](3,3) \\
\tan \omega_{p,i} &= \frac{[NP](1,3)}{[NP](2,3)}
\end{align*}
\]

**Correction Maneuver** While it appears that Equations 6.13, and 6.14 always describe a valid $[NP]$, the triad algorithm often looses information about the second reference vector, due to the cross product operation in defining $\hat{n}_2$ and $\hat{p}_{i,2}$. This allows us to write the error vector $\Delta v_{OC,i}$ as

\[ \Delta v_{OC,i} = [NP]P V^{-}_{\infty,2} - N V^{-}_{\infty,2} \]  

(6.15)

The $\Delta v_{OC,i}$ can be interpreted as the maneuver cost of spacecraft $i$ to correct $N V^{-}_{\infty,2}$ and have the same orientation elements of the selected co-orbit.
Swarm Behaviours  Similar to the hyperbolic swarms in Chapter 5, the co-orbital swarms will also be configured to visit the moon over $N_v$ encounters during which the moon is located at $f_{v,j}$, $j = 1, 2, ..., N_v$. However, since the visual coverage requirements of the two mission concepts considered in the current work are different, their co-orbital configuration will vary accordingly, as illustrated in Figure 6.7.

![Figure 6.7: Swarm behaviours for the two visual mapping mission concepts: global surface mapping (left) and RoI observation (right) missions.](image)

Specifically, each encounter in the global surface mapping mission will contain multiple spacecraft, while in the case of the RoI observation missions, each encounter will only contain a single spacecraft. Therefore, the number of spacecraft in the case of the global surface mapping mission can be expressed as

$$N_{Sw,\text{map}} = \sum_{j=1}^{N_v} N_{v,j}$$

(6.16)

where $N_{v,j}$ is the number of spacecraft in visit $j$. In the case of the RoI observation swarm, the number of spacecraft in the swarm can be expressed as

$$N_{Sw,\text{RoI}} = \sum_{j=1}^{N_v} 1 = N_v$$

(6.17)

Automated Swarm Design  The Automated Swarm Designer module to design swarms that can meet the coverage, collision, and trajectory performance require-
ments with a minimum number of spacecraft. The design problems specific to each problem is expressed as follows:

**Global Surface Mapping** In the global surface mapping application, the objective function $J_{Sw,\text{map}} = N_{Sw,\text{map}}$ is minimized such that the coverage figure of merit $P_{FoM,\text{map}}$ described in chapter 3 of the swarm exceeds the coverage requirement $P_{map,R}$ without any collisions, with the maximum value of $\Delta v_{OC,i}$ upper bounded by $\Delta v_{OC,max}$. The problem can be expressed as

$$\min \quad J_{Sw,\text{map}} = N_{Sw,\text{map}}$$
$$\text{s.t.} \quad P_{FoM,\text{map}} \geq P_{map,R}$$
$$\text{Collision flag} \leq 0$$
$$\max(\Delta v_{OC,i}) \leq \Delta v_{OC,max}$$

(6.18)

The swarm design gene corresponding to the global surface mapping swarm missions is presented in Figure 6.8.

<table>
<thead>
<tr>
<th>Parameter</th>
<th># Moon encounters</th>
<th># Spacecraft each encounter</th>
<th>True anomaly of the moon</th>
<th>Spacecraft azimuth at encounter</th>
<th>Spacecraft elevation at encounter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>$N_v$</td>
<td>$N_{v,1}$</td>
<td>...</td>
<td>$N_{v,N_v}$</td>
<td>$f_{v,1}$</td>
</tr>
<tr>
<td>Range</td>
<td>Integer $[1, N_{1,\text{max}}]$</td>
<td>Integer $[1, N_{2,\text{max}}]$</td>
<td>Real $[0, 360]$ deg</td>
<td>Real $[0, 360]$ deg</td>
<td>Real $[-90, 90]$ deg</td>
</tr>
</tbody>
</table>

Figure 6.8: Swarm design gene for the global surface mapping swarm missions.

**RoI Observation** In the RoI observation mission, $J_{Sw,\text{RoI}} = N_{Sw,\text{RoI}}$ is minimized such that the RoI coverage requirements $P_{FoM,\text{RoI}}$ and $T_{FoM,\text{RoI}}$ exceed the spatial requirement $P_{RoI,R}$, and temporal requirement $T_{RoI,R}$, without collisions. The maximum value of $\Delta v_{OC,i}$ is also assumed to be upper bounded by $\Delta v_{OC,max}$. The gene map of the RoI observation swarm design problem is presented in Figure 6.9. The RoI swarm design problem can now be expressed as
\[
\begin{align*}
\min & \quad J_{Sw, RoI} = N_{Sw, RoI} \\
\text{s.t.} & \quad P_{FoM, RoI} \geq P_{RoI, R} \\
& \quad T_{FoM, RoI} \geq T_{RoI, R} \\
& \quad \text{Collision flag } \leq 0 \\
& \quad \max(\Delta v_{OC,i}) \leq \Delta v_{OC,max}
\end{align*}
\]

(6.19)

Figure 6.9: Swarm design gene for the RoI observation swarm missions.

The bounds on all design variables shown in Figures 6.8 and 6.9 are provided through the user interface. It should be noted that the swarm genes described in this chapter allow us to compute the spacecraft locations at the encounter instant. However, at the start of simulations, we initialize the moon’s location by offsetting the lowest encounter true anomaly by \( \Delta f_0 \), similar to the initialization in Chapter 5 as

\[
f_0 = \min (f_{v,j}) - \Delta f_0
\]

(6.20)

The true anomalies of all spacecraft are then phased such that when the moon reaches a true anomaly of \( f_{v,j} \), the spacecraft corresponding to this encounter reach a true anomaly of 180 deg (Vallado, 2013). The cumulative coverage is noted by propagating spacecraft and moon motion for one orbital period of the moon.

**Spacecraft Design**

Since the co-orbital swarms described in this chapter perform two deterministic maneuvers to capture into the target resonant orbits, these maneuver costs are used
to estimate their fuel requirements and mission costs.

**Wet Mass** To ensure that the spacecraft in the swarm is capable of the two maneuvers: orbit insertion, and orientation change, the seed spacecraft is allotted fuel enough to perform a total velocity change \( \Delta v_{\text{net}} \) given by

\[
\Delta v_{\text{net}} = 1.3 (\Delta v_{\text{OI, max}} + \max (\Delta v_{\text{OC,i}}))
\]  

(6.21)

The factor 1.3 is used as a 30% margin to account for correction maneuvers due to any unmodeled dynamics (Vallado, 2013; Scheeres, 2016). The fuel mass of the seed spacecraft and its wet mass at launch is noted using Equation A.1.

**Deimos Models**

The orbital elements and dynamical parameters of Deimos are presented in Table 6.2. The shape models of Deimos used for the two mission concepts are described as follows:

Table 6.2: Dynamical parameters to model the encounter co-orbits of the swarm with and Deimos.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deimos semi-major axis</td>
<td>23485 km</td>
</tr>
<tr>
<td>Deimos eccentricity</td>
<td>0.0002</td>
</tr>
<tr>
<td>Deimos inclination</td>
<td>1.79 deg</td>
</tr>
<tr>
<td>Deimos RAAN</td>
<td>261 deg</td>
</tr>
<tr>
<td>Deimos argument of periapsis</td>
<td>123 deg</td>
</tr>
<tr>
<td>Deimos orbital period</td>
<td>1.26 days</td>
</tr>
<tr>
<td>Deimos mass</td>
<td>(1.8 \times 10^{15}) kg</td>
</tr>
<tr>
<td>Deimos albedo</td>
<td>0.07</td>
</tr>
<tr>
<td>Mars gravitational parameter</td>
<td>43824 km(^3)/s(^2)</td>
</tr>
<tr>
<td>Mars aerobraking altitude</td>
<td>150 km</td>
</tr>
<tr>
<td>Mars (j_2)</td>
<td>0.0019</td>
</tr>
<tr>
<td>Mars equatorial radius</td>
<td>3394 km</td>
</tr>
</tbody>
</table>
Global Surface Mapping  A 5040 triangular face model (Frieger, 2018; Thomas et al., 2000) is used for the nominal shape model of Deimos. The view and range spheres are generated from the nominal shape model, respectively, using 1280 triangular faces each. The nominal shape model of Deimos, along with its Dual Sphere shape models, is presented in Figure 6.10 and the critical parameters noted from these models are summarized in Table 6.3.

![Figure 6.10: The nominal and Dual Sphere models of Deimos used for designing global surface mapping mission.](image)

Table 6.3: Shape parameters of the Deimos model used in the current work.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range Sphere</th>
<th>Shape Model</th>
</tr>
</thead>
<tbody>
<tr>
<td># Faces</td>
<td>1280</td>
<td>5040</td>
</tr>
<tr>
<td>Radius (km)</td>
<td>3.58</td>
<td>5.82 (mean)</td>
</tr>
<tr>
<td>Area (km²)</td>
<td>160</td>
<td>519</td>
</tr>
</tbody>
</table>

RoI Observation  The target RoI is modeled using a set of 58 triangular faces, which are shown in Figure 6.11. The shape model parameters corresponding to the RoI faces are presented in Table 6.11.

Swarm Parameters

Using the trajectory design and the shape model parameters, the required swarm parameters are computed as follows. As explained above, the maximum altitude
Figure 6.11: The nominal shape model showing the selected RoI faces used for RoI observation mission design.

Table 6.4: Shape parameters of the selected region of interest on Deimos.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>RoI Values</th>
</tr>
</thead>
<tbody>
<tr>
<td># Faces</td>
<td>58</td>
</tr>
<tr>
<td>Area (km²)</td>
<td>5.94</td>
</tr>
</tbody>
</table>

for the selected spacecraft camera is 114 km. The camera and flyby parameters are summarized in Table 6.5.

Table 6.5: Flyby and camera parameters to design the reconnaissance co-orbits.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required flyby altitude, $h_f$</td>
<td>114 km</td>
</tr>
<tr>
<td>Altitude tolerance, $\Delta h$</td>
<td>5 km</td>
</tr>
<tr>
<td>Designed flyby altitude, $h'_f$</td>
<td>109 km</td>
</tr>
<tr>
<td>Required spacecraft FoV, $2\eta_p$</td>
<td>6.04 deg</td>
</tr>
</tbody>
</table>

Sensitivity Study Parameters

To study the sensitivity of the performance of swarm do different uncertainties, the optimal design is perturbed using different categorical perturbations described in
Chapter 4. In this chapter, four categories of perturbations are considered: Spacecraft outage, error in the knowledge of Deimos’ location, error in encounter shaping, and sensitivity of coverage performance to the high fidelity dynamics. In these studies, the worst-case perturbations that make the coverage drop below $P_{thr}$ are noted.

6.3 Numerical Simulations

This section presents the results of the two swarm mission concepts to explore the surface of Deimos through resonant Martian co-orbits. The optimizer performance for different design optimization problems is presented, followed by examining the coverage performance of the designed swarms.

Global Surface Mapping

The global surface mapping swarm is designed to provide a coverage figure of merit of at least 90 % of the surface of Deimos. The observations are required to be of a maximum resolution of 1 m/px. The design of the trajectory, swarm, and seed spacecraft of the global surface mapping mission is described below.

Trajectory Design  As seen from Equation 6.11, the trajectory design problem is a mixed-integer optimization problem is solved using which are solved by using the modified real-valued Genetic Algorithm (GA) described in (Deb, 2000). Each trajectory search optimization was set to run for a maximum of 1000 generations, where each generation searched for a total of 1000 design genes. A stall stop criterion of 100 generations was used to identify the optimal solution of the corresponding trial. The parameters described in Table 6.6 are passed as input arguments to the optimizer.

Optimization  The results of 5 GA trials for the trajectory design is presented in Figure 6.12. The trials were able to converge to their optimal solutions in about
Table 6.6: Input parameters to design the Earth-Mars trajectory of the Deimos mapping swarm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[D_{L,\text{min}}, D_{L,\text{max}}]$</td>
<td>[Jan 1st 2020 – Dec 31st 2020]</td>
</tr>
<tr>
<td>$[D_{A,\text{min}}, D_{A,\text{max}}]$</td>
<td>[May 1st 2020 – Dec 31st 2021]</td>
</tr>
<tr>
<td>$C_{3,\text{max}}$</td>
<td>20 km$^2$/s$^2$</td>
</tr>
<tr>
<td>$v_{\infty,\text{max}}$</td>
<td>3 km/s</td>
</tr>
<tr>
<td>$T_{\text{Of max}}$</td>
<td>200 days</td>
</tr>
<tr>
<td>$h_{p,\text{min}}$</td>
<td>300 km</td>
</tr>
<tr>
<td>$e_{l,\text{max}}$</td>
<td>0.98</td>
</tr>
<tr>
<td>$[\varphi_{\text{min}}, \varphi_{\text{max}}]$</td>
<td>[1.10]</td>
</tr>
<tr>
<td>$[\rho_{\text{min}}, \rho_{\text{max}}]$</td>
<td>[1.20]</td>
</tr>
</tbody>
</table>

200 generations, after which they were stopped when the minimum fitness stalled for 100 generations.

The final generations of all 5 trials identified 4114 near-optimal solutions that satisfied the launch and arrival constraints. The performance of these feasible genes is presented in Figure 6.13. The selected near-optimal trajectory had a minimum $\Delta v_{OI,\text{max}}$ of 0.666 km/s as shown in Figure 6.13.

The selected optimal trajectory gene is presented in Table 6.7. It can be seen here that the selected belongs to the same launch window with the Phobos mapping interplanetary trajectory in Chapter 5, and therefore shares similar properties with those in Figure 5.10. The selected optimal trajectory has a time of flight of 200 days. The trajectory requires a $C_{3,E}$ of 18.6 km$^2$/s$^2$, and arrives at Mars with an excess velocity of 2.47 km/s. The eccentricity and periapsis altitude of the captured HEO can be noted from Table 6.7. Upon aerobraking from these orbits, the swarm enters into a 4 : 9 co-orbit resonance with Deimos, which requires a maximum planar $\Delta v_{OI,\text{max}}$ of 0.672 km/s. The minimum periapsis altitude of these resonant co-orbits was noted as 349 km.

**Swarm Design** The optimization problem in Equation 6.18 is solved using the constrained GA solver (Deb, 2000). The parameters described in Table 6.8 are
Figure 6.12: Variation of the trajectory fitness for the global surface mapping mission across different GA generations showing the individual optimization trials (left), and their statistical distribution (right).

passed as input arguments to the optimizer.

**Optimization Results**  The optimization solver was set up to evaluate a population of 100 global surface mapping swarm genes per generation. A maximum of 300 generations was placed, during which the search was terminated when the best fitness stalled for 100 generations. The results of 5 swarm design trials using the modified real-valued GA optimizer (Deep et al., 2009) is presented in Figure 6.14.

The final generations of all 5 trials identified 352 feasible design genes. The performance of these feasible swarm genes is presented in Figure 6.15. As seen here, the final generations identified several near-optimal solutions that satisfied the coverage, maneuver, and collision constraints. The selected near-optimal swarm gene has a minimum swarm size of 6 spacecraft, and its constituent design variables are presented in Table 6.9.
Swarm Performance  The configuration of the resonant co-orbits for the global surface mapping mission is presented in Figure 6.16. The maximum value of $\Delta v_{OI,i}$ is noted as $2.81 \text{ km/s}$. As seen here, the swarm encounters Deimos in 4 locations, whose true anomalies can be noted from Table 6.9. The encounter velocities of spacecraft with respect to Deimos during the encounters were noted to range between $0.692 - 2.05 \text{ km/s}$. An example encounter of Deimos with a single participating spacecraft is presented in Figure 6.17.

The key performance figures of merit of the global surface mapping swarm are summarized in Table 6.10. As seen here, the minimum mean-1σ surface coverage, noted from $N_{Mon,1} = 20$ random initializations of Deimos, was found to be 90.0 %. The time evolution of surface coverage noted during a simulated swarm flyby of Deimos is presented in 6.18. The cumulative coverage of the swarm over the nominal model was found to be 90.7 %, while it was found to be 90.0 % and 93.4 % over the View and range spheres, respectively. The cumulative coverage of the nominal and Dual Sphere models of Deimos are presented in Figure 6.19.
Table 6.7: Design variables of the selected near optimal trajectory design gene for the global surface mapping mission.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_L$</td>
<td>Aug 11$^{th}$ 2020, 15:47:00</td>
</tr>
<tr>
<td>$D_A$</td>
<td>Feb 27$^{th}$ 2021, 10:42:00</td>
</tr>
<tr>
<td>$h_{p,0}$</td>
<td>347 km</td>
</tr>
<tr>
<td>$e_1$</td>
<td>0.979</td>
</tr>
<tr>
<td>$p$</td>
<td>4</td>
</tr>
<tr>
<td>$q$</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 6.8: Input parameters to design the global surface mapping swarm at Deimos

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{\text{map,}k}$</td>
<td>90%</td>
</tr>
<tr>
<td>$\Delta v_{\text{thr},\text{max}}$</td>
<td>3 km/s</td>
</tr>
<tr>
<td>$N_{1,\text{max}}$</td>
<td>4</td>
</tr>
<tr>
<td>$N_{2,\text{max}}$</td>
<td>4</td>
</tr>
<tr>
<td>$N_{\text{non,}1}$</td>
<td>20</td>
</tr>
<tr>
<td>$T_{\text{sim}}$</td>
<td>1.4 days</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>5 sec</td>
</tr>
<tr>
<td>$\Delta f_0$</td>
<td>1 deg</td>
</tr>
<tr>
<td>$r_{\text{col}}$</td>
<td>1 m</td>
</tr>
<tr>
<td>$[\alpha_1, \alpha_2]$</td>
<td>[0.4, 0.6]</td>
</tr>
</tbody>
</table>

RoI Observation Swarm Design

The RoI observation swarm is designed to provide a coverage figure of merit of at least 95% of the surface of selected RoI. The RoI needs to be observed for at least 20 mins of duration in an orbital day of Deimos. The observations are required to be of a maximum resolution of 1 m/px. The design of the trajectory, swarm, and seed spacecraft of the RoI observation mission is described below.

Trajectory Design  To distinguish the two mission concepts, The trajectory design problem in Equation 6.11 is solved separately for the RoI observation swarms using the input parameters in Table 6.6.
Optimization Results The results of 5 GA trials for the trajectory design is presented in Figure 6.20. The trials were able to converge to their optimal solutions in about 200 generations, after which they were stopped when the minimum fitness stalled for 100 generations.

The final generations of all 5 optimization trials identified 2476 feasible trajectory genes that satisfied the launch and arrival constraints. The performance of these feasible genes is presented in Figure 6.21. The selected near-optimal trajectory had a minimum $\Delta v_{OI,max}$ of 0.672 km/s as shown in Figure 6.21.

The selected optimal trajectory gene is presented in Table 6.11. Similar to the global surface mapping swarm, the selected optimal trajectory has a time of flight of 200 days. The trajectory requires a $C_{3,E}$ of 17.9 km$^2$/s$^2$, and arrives at Mars with an excess velocity of 2.5 km/s. The eccentricity and periapsis altitude of the captured HEO can be noted from Table 6.11. Upon aerobraking from these orbits, the swarm enters into a 4 : 9 co-orbit resonance with Deimos, which requires a maximum planar $\Delta v_{OI,max}$ of 0.672 km. The minimum periapsis altitude of these
The resonant co-orbits is at 349 km, as shown in Figure 6.21.

**Swarm Design** The optimization problem in Equation 6.19 is solved using the constrained GA solver (Deb, 2000). The parameters described in Table 6.12 are passed as input arguments to the optimizer.

**Optimization Results** The optimization solver was set up to evaluate a population of 250 RoI swarm genes per generation. A maximum of 300 generations was placed, during which the search was terminated when the best fitness stalled for 100 generations. The results of 5 swarm design trials using the modified real-valued GA optimizer (Deep et al., 2009) is presented in Figure 6.22.

The final generations of all 5 optimization trials identified 957 feasible RoI observation swarm genes that satisfied the coverage, maneuver, and collision constraints. The performance figures of merit of these feasible swarm genes are presented in Figure 6.23. The selected near-optimal swarm has a minimum swarm size of 5 spacecraft, as shown in Figure 6.23. The selected optimal RoI swarm gene is presented...
Table 6.9: Design variables of the selected near optimal swarm design gene for global surface mapping mission.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_p$</td>
<td>4</td>
</tr>
<tr>
<td>$N_{v,i}$</td>
<td>[1, 1, 2, 2]</td>
</tr>
<tr>
<td>$f_{x,i}$ (deg)</td>
<td>[62.8, 133, 230, 300]</td>
</tr>
<tr>
<td>$[\theta_{x,i}, \theta_{y,i}]$ (deg)</td>
<td>[296, 15.9, 252, 38.7]</td>
</tr>
<tr>
<td></td>
<td>[191, –36.9, 228, 25.5]</td>
</tr>
<tr>
<td></td>
<td>[203, –37.1, 272, –35.6]</td>
</tr>
</tbody>
</table>

Table 6.10: Coverage and maneuver figures of merit of the global surface mapping swarm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal model</td>
<td>$90.9 \pm 0.48 %$</td>
</tr>
<tr>
<td>View sphere</td>
<td>$90.3 \pm 0.30 %$</td>
</tr>
<tr>
<td>Range sphere</td>
<td>$93.5 \pm 0.15 %$</td>
</tr>
<tr>
<td>$\max(\Delta v_{O,I,i})$</td>
<td>$2.81 \text{ km/s}$</td>
</tr>
</tbody>
</table>

in Table 6.13.

**Swarm Performance** The encounter configuration in the RoI observation missions is presented in Figure 6.24 along with the direction of the incoming $N V_{\infty,-2}$ asymptote. The maximum value of $\Delta v_{O,I,i}$ of the swarm is noted as 1.24 km/s. As seen here, the swarm approaches Deimos is through 5 serial flybys on the resonant co-orbits. The encounter velocities with respect to Deimos were found to range between 0.754 – 0.995 km/s. An example encounter of a spacecraft in the swarm with Deimos is presented in Figure 6.25.

The coverage evaluation of the swarm noted from $N_{Mon,1} = 20$ random initializations of Deimos suggested that the swarm can observe about $99.2 \pm 1.1 \%$ of the surface of the RoI, with an estimated observation time of $20.7 \pm 0.15 \text{ mins}$, which
allow the swarm to meet the design requirements. The time evolution of the temporal and spatial RoI coverage noted during a simulated swarm flyby of Deimos is presented in 6.26. During this simulation, the swarm was able to observe the about 98.8 % of the surface of the RoI for about 20.7 mins. The performance figures of merit of the RoI observation swarm are summarized in Table 6.14.

**Seed Spacecraft Design**

The design parameters of the two spacecraft swarm mission concepts are described here. The seed spacecraft’s launch mass, required for the orbit insertion and orientation change maneuvers, is estimated. The launch mass is then used to estimate the space segment cost of the seed spacecraft and the swarm mission using the
Global Surface Mapping  As described above, the maneuver costs associated with the orbit insertion and orientation change of the global surface mapping swarm were found to be 0.666 km/s and 2.81 km/s respectively. The net maneuver cost associated with the two maneuvers noted from Equation 6.21 is found to be 4.53 km/s, which requires that the seed spacecraft should be budgeted with a hydrazine mass of 135 kg. The resultant design parameters of the seed spacecraft in the global surface mapping swarm are presented in Table 6.15.

RoI Observation  The net maneuver $\Delta v_{net}$ computed from Equation 6.21 is noted as 2.5 km/s. The resultant seed spacecraft requires a fuel mass of 41.2 kg of
Figure 6.18: Time evolution of the spatial coverage of Deimos noted during a simulated swarm flyby.

Hydrazine. The seed spacecraft design parameters and the associated cost breakdown are summarized for both the mission concepts in Table 6.15. The breakdown of the swarm mission costs of the selected optimal designs is presented in Table 6.16 for both mission concepts. As seen here, the global surface mapping mission has a total estimated space segment cost of $129 \pm 3.8$ LY$\$$M, while that of the RoI observation swarm mission is $87.8 \pm 2.7$ LY$\$$M.

### 6.4 Sensitivity Analysis

The coverage sensitivity of the designed swarms to different perturbations is examined here. A coverage lower bound of $P_{thr} = 80\%$ is used to evaluate the worst-case perturbation in the case of the global surface mapping swarms. In case of the RoI observation swarms, the lower bounds used on the temporal and spatial coverage were $P_{thr} = 85\%$ and $T_{thr} = 15$ mins respectively.
Outage

The input parameters to study the coverage sensitivities to spacecraft outages are presented in Table 6.17. The results of the outage sensitivity studies for the two mission concepts are presented as follows:

Global Surface Mapping  The variation of coverage over the three shape models: nominal and Dual Sphere models of Deimos, with spacecraft outages in the swarm, are presented in Figure 6.27. As seen here, the swarm is capable of meeting the designed coverage requirement of 90 % when there are no outages. However, the loss of a single spacecraft can lead to a decay in the surface coverage below the minimum threshold. When the swarm is operating at about 82 % efficiency (one defunct spacecraft), the coverage over the nominal shape model of Deimos was found to be $87.1 \pm 4.1 \%$, while the coverages over the View and Range spheres is found to be $82.1 \pm 5.7 \%$ and $90.4 \pm 3.5 \%$ respectively. While the expected coverage over
all three models was noted to be larger than $P_{thr}$, the coverage figure of merit over the View sphere from Equation 3.33 is noted to fall below this threshold.

**RoI Observation**  The variation of the spatial and temporal coverage of Deimos, with spacecraft outages in the swarm, are presented in Figure 6.28. As expected, the spacecraft outages are shown to lead to decay in temporal and spatial coverage of Deimos. When operating at about 80% efficiency (one defunct spacecraft), the temporal coverage of the RoI is noted as 16.7 ± 0.23 mins, while the spatial coverage of the RoI was found to be 93.5 ± 6%. The loss of more than one spacecraft is shown to cause coverage decay below $P_{thr}$ and $T_{thr}$ as seen in Figure 6.28.

**Moon location**

The input parameters to study the coverage sensitivities to perturbations in the location of Deimos are presented in Table 6.17. The coverage variation in the two mission concepts are described as follows:
Global Surface Mapping  The variation of the coverage over the nominal and Dual Sphere shape models of Deimos, with true anomaly perturbations, is presented in Figure 6.29. It was noted that all the coverage of all three shape models exceeded the design requirement of 90 % when the true anomaly perturbation was below 0.09 deg. As expected, the coverage decayed with an increase in the perturbation amplitude. The analysis indicated that the swarm could tolerate a worst-case lead perturbation of 0.45 deg, where the View sphere coverage was noted as 80.7 ± 0.45 %. Additionally, the worst-case lag perturbation was found to be 0.27 deg, where the view sphere coverage was found to be 79.5 ± 0.27 %. Therefore, the swarm was found to be robust to a true anomaly uncertainty of ±0.27 deg.

RoI Observation  The variation in the spatial and temporal RoI coverage of Deimos with true anomaly perturbations is presented in Figure 6.30. The analysis indicated that the temporal coverage was robust to a worst-case lead perturbation of 0.18 deg, where the RoI coverage was duration found to be 15.5 ± 0.85 mins. The worst-case lag perturbation was also found as 0.18 deg where the coverage
Table 6.12: Input parameters to design the RoI observation swarm at Deimos

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{RoI,R}$</td>
<td>95 %</td>
</tr>
<tr>
<td>$T_{RoI,R}$</td>
<td>20 mins</td>
</tr>
<tr>
<td>$\Delta v_{OE,max}$</td>
<td>2 km/s</td>
</tr>
<tr>
<td>$N_{1,max}$</td>
<td>15</td>
</tr>
<tr>
<td>$N_{mon,1}$</td>
<td>20</td>
</tr>
<tr>
<td>$T_{sim}$</td>
<td>1.4 days</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>5 sec</td>
</tr>
<tr>
<td>$\theta_{max}$</td>
<td>5 deg</td>
</tr>
<tr>
<td>$\Delta f_0$</td>
<td>1 deg</td>
</tr>
<tr>
<td>$r_{col}$</td>
<td>1 m</td>
</tr>
<tr>
<td>$[\alpha_1, \alpha_2]$</td>
<td>[0.4, 0.6]</td>
</tr>
</tbody>
</table>

duration was $17.1 \pm 0.04$ mins. The results indicate that the RoI coverage is tolerant of lagging perturbations as opposed to lead perturbations in the true anomalies of Deimos. As seen in Figure 6.30, a similar trend was noted in the spatial coverage of RoI. The worst-case lead perturbation for the spatial coverage was found to be 0.27 deg, where the spatial coverage was noted as $92.2 \pm 0.27 \%$. In the case of the lag perturbations, the swarm could tolerate a worst-case perturbation of 0.54 deg, where the RoI spatial coverage was $99.5 \pm 0.5 \%$. Therefore the temporal and spatial coverage of the swarm is found to be robust to a worst-case true anomaly perturbation of $\pm 0.18$ deg.

**Encounter Shaping**

The input parameters to study the coverage sensitivities to uncertainties in the encounter locations of the spacecraft are presented in Table 6.19. The coverage variation in the two mission concepts are described as follows:

**Global Surface Mapping** The variation of the coverage over the nominal and Dual Sphere models of Deimos, with encounter location uncertainties, is presented in Figure 6.31. The analysis suggested that the swarm can meet the designed 90 %
coverage requirement up to a worst-case uncertainty of 18.2 km, where the lowest expected coverage of 91 ± 0.55% was noted over the View sphere. The swarm was able to meet the minimum threshold requirement up to a maximum uncertainty of 27.3 km, where the View sphere coverage was noted as 88.8 ± 0.28%. Larger perturbations in the encounter locations caused the coverage to fluctuate rapidly, where some uncertainties produced a global surface coverage that met the $P_{thr}$ threshold, as seen in Figure 6.31. Therefore, the swarm is robust to a worst-case encounter location uncertainty of 27.3 km.

**RoI Observation** The RoI coverage variation with uncertainties in spacecraft encounter locations is presented in Figure 6.32. It can be seen here that the spatial coverage of the RoI is virtually unaffected by the uncertainties in the spacecraft locations. The temporal RoI coverage, on the other hand, was able to meet the designed duration requirement of 20 mins up to a maximum uncertainty of 20.6 km. While larger uncertainties caused the coverage to fluctuate rapidly, the analysis
indicated that the swarm was able to tolerate a worst-case uncertainty of 81.8 km, where the RoI observation duration was noted as $14.7 \pm 0.3$ mins.

Dynamic Sensitivity

The coverage sensitivities of the two swarm mission concepts, when the motion is propagated with different levels of fidelity, are examined here. The perturbative accelerations used at different fidelity levels are the same as described in Table 4.1. The coverage variation in the two swarm mission concepts is described as follows:

**Global Surface Mapping**  The magnitudes of different perturbative accelerations used to model the motion of the spacecraft are presented in Figure 6.33. As seen here, the two-body perturbation is still the dominant acceleration with a peak value of about 2.72 m/s$^2$. The $J_2$ oblateness is seen to be the largest perturbative acceleration consistently and is about 2 – 3 orders of magnitude smaller than the two-body effect. The third-body acceleration is seen to peak during the spacecraft encounters with Deimos, where its magnitude is nearly equal to the $J_2$ effect. Following the
Table 6.13: Design variables of the selected near optimal swarm design gene for RoI observation mission.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_p )</td>
<td>5</td>
</tr>
<tr>
<td>( f_{\psi}(\text{deg}) )</td>
<td>147, 148, 153, 160, 169</td>
</tr>
<tr>
<td>([\theta_{x,1}, \theta_{y,1}] (\text{deg}))</td>
<td>[122, 13.4], [271, 8.2], [134, 41.3], [113, 58.0], [272, -30.5]</td>
</tr>
</tbody>
</table>

Table 6.14: Coverage and maneuver figures of merit of the RoI observation swarm.

<table>
<thead>
<tr>
<th>Figure of merit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{\text{Fom,RoI}} )</td>
<td>98.1 %</td>
</tr>
<tr>
<td>( T_{\text{Fom,RoI}} )</td>
<td>20.5 mins</td>
</tr>
<tr>
<td>( \text{max}(\Delta v_{\text{DC,1}}) )</td>
<td>1.24 km/s</td>
</tr>
</tbody>
</table>

encounter, the third body effect is shown to quickly decay and become the least dominant source of perturbation with magnitudes of the order \( 10^{-10} \) m/s\(^2\). The SRP acceleration is seen to produce a near-constant perturbation acceleration of magnitude of about \( 10^{-8} \) m/s\(^2\).

The coverage variation of the global surface mapping swarm with different levels of propagator fidelities is presented in Figure 6.34. As seen here, coverage of the swarm is unaffected when modeling propagating the motion with the third-body perturbations from Deimos. However, as noted from Figure 6.34, including the \( J_2 \) perturbation causes the coverage performance to decay below the \( P_{\text{thr}} = 80 \) % threshold. The combined third body and \( J_2 \) effects cause a minimum expected coverage of 49.5 ± 0.34 % over the View sphere. Therefore, trajectory correction maneuvers are required to counteract these perturbations and to ensure that the coverage performance of the swarm meets the design requirements. The coverage
Figure 6.24: Arrangement of resonant co-orbits for the RoI observation mission to Deimos.

performance of the swarm is seen to be unaffected by the SRP perturbation, as seen in Figure 6.34.

**RoI Observation**  Since the spacecraft in the RoI observation swarm are deployed on the same 4 : 9 resonant co-orbits as those on the global surface mapping swarm, their accelerations will be similar to the magnitudes noted in Figure 6.33. The coverage variation of the RoI observation swarm with different levels of propagator fidelities is presented in Figure 6.35. As seen here, the temporal and spatial coverage of the selected RoI is unaffected by the different perturbative accelerations considered in the current work. This robustness is primarily caused by the LoS tracking behavior of the spacecraft camera to the center of RoI, which suggests that the resonant co-orbits can be viable trajectories for RoI observation missions to planetary moons.

### 6.5 Discussion

This chapter presented a novel way of utilizing spacecraft swarm flybys to perform visual reconnaissance of planetary Moons using a case study of Deimos exploration. In comparison to the hyperbolic flyby behavior described in Chapter 5, the spacecraft in the swarm now enters into resonant co-orbits around the central planet, which
enables repeated flybys of the moon. The orbit capture is constrained by an insertion maneuver that is split into two components: a planar orbit capture maneuver that uses aerobraking to deploy the spacecraft on the desired resonant orbit. An out of plane maneuver that corrects the incoming hyperbola to facilitate a planar orbit capture into the targeted orbit. The configuration of the swarm was described for two types of mission concepts: global surface mapping and region of interest observation. The two mission concepts were then designed to explore the surface of Deimos using the IDEAS architecture, and the performance sensitivities of the selected near-optimal solutions, to different perturbations, were then examined.
Figure 6.26: Time evolution of the temporal (left) and spatial (right) RoI coverage of Deimos noted during a simulated swarm flyby.

Table 6.15: Seed spacecraft design parameters and associated cost breakdown for the two mission concepts.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mission Concept</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Global Surface Mapping</td>
<td>Rol Observation</td>
<td></td>
</tr>
<tr>
<td>$\Delta v_{net}$ (km/s)</td>
<td>3.48</td>
<td>1.24</td>
<td></td>
</tr>
<tr>
<td>$m_{fuel}$ (kg)</td>
<td>135</td>
<td>41.2</td>
<td></td>
</tr>
<tr>
<td>$m_{wet}$ (kg)</td>
<td>154</td>
<td>60.0</td>
<td></td>
</tr>
<tr>
<td>Launch year</td>
<td>2020</td>
<td>2020</td>
<td></td>
</tr>
<tr>
<td>Interplanetary payload cost (LY$\text{M}$)</td>
<td>$2.72 \pm 0.71$</td>
<td>$2.72 \pm 0.71$</td>
<td></td>
</tr>
<tr>
<td>Interplanetary bus and operations cost (LY$\text{M}$)</td>
<td>$18.4 \pm 1.04$</td>
<td>$15.8 \pm 0.89$</td>
<td></td>
</tr>
<tr>
<td>Launch cost (LY$\text{M}$)</td>
<td>$3.03 \pm 0.91$</td>
<td>$1.18 \pm 0.35$</td>
<td></td>
</tr>
</tbody>
</table>
Table 6.16: Cost breakdown of the two designed swarm mission concepts.

<table>
<thead>
<tr>
<th>Component</th>
<th>Mission Concept</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Global Surface Mapping</td>
<td>Roll Observation</td>
</tr>
<tr>
<td>Mission cost (LY$M)</td>
<td>111 ± 3.1</td>
<td>82.0 ± 2.6</td>
</tr>
<tr>
<td>Launch cost (LY$M)</td>
<td>18.2 ± 2.2</td>
<td>5.91 ± 0.79</td>
</tr>
<tr>
<td>Total space segment cost (LY$M)</td>
<td>129.4 ± 3.8</td>
<td>87.9 ± 2.7</td>
</tr>
</tbody>
</table>

Table 6.17: Input parameters to study the sensitivity of spacecraft outage to the coverage performance of the swarms.

| Parameter                                      | Mission Concept |         |
|                                               |                 |         |
|                                               | Global Surface Mapping | Roll Observation |
| Minimum coverage requirement, \( P_{thr} \) (%) | 80              | 85      |
| Minimum coverage duration requirement, \( T_{thr} \) (mins) | N.A.          | 15      |
| Minimum operational percentage, \( P_{opm} \) (%) | 16             | 20      |
| Study resolution (# spacecraft)                | 1               | 1       |

Table 6.18: Input parameters to study the coverage sensitivity to the orbital location of Deimos.

| Parameter                                      | Mission Concept |         |
|                                               |                 |         |
|                                               | Global Surface Mapping | Roll Observation |
| Minimum coverage requirement, \( P_{thr} \) (%) | 80              | 85      |
| Minimum coverage duration requirement, \( T_{thr} \) (mins) | N.A.          | 15      |
| Minimum operational percentage, \( P_{opm} \) (%) | 16             | 20      |
| Study resolution (# spacecraft)                | 1               | 1       |
Figure 6.27: Variation in the surface coverage of Deimos with spacecraft outages in the global surface mapping swarm.

Figure 6.28: Variation in the temporal (left) and spatial (right) RoI coverage of Deimos with spacecraft outages in the swarm.
Figure 6.29: Variation in the coverage of Deimos shape models with uncertainty in the location of Deimos.

Figure 6.30: Variation in the temporal (left) and spatial (right) RoI coverage of Deimos with uncertainty in the location of Deimos.
Table 6.19: Input parameters to study the coverage sensitivity to the encounter location of the spacecraft.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mission Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Global Surface Mapping</td>
</tr>
<tr>
<td>Minimum coverage requirement, $P_{thr}$ (%)</td>
<td>80</td>
</tr>
<tr>
<td>Minimum coverage duration requirement, $T_{thr}$ (mins)</td>
<td>N.A.</td>
</tr>
<tr>
<td>Number of random target initializations, $N_{mon,2}$</td>
<td>200</td>
</tr>
<tr>
<td>Range of true anomaly error, $\Delta f_{ap}$ (deg)</td>
<td>$[-1,1]$</td>
</tr>
<tr>
<td>Study size (# true anomaly offsets)</td>
<td>21</td>
</tr>
</tbody>
</table>

Figure 6.31: Variation in the coverage of the Deimos shape models with uncertainty in the location of encounters.
Figure 6.32: Variation in the temporal (left) and spatial (right) RoI coverage of Deimos with uncertainty in the location of encounters.

Figure 6.33: Perturbative acceleration profiles of an example spacecraft in the global surface mapping swarm.
Figure 6.34: Variation in the coverage of the Deimos shape models with different levels of motion propagation fidelity.

Figure 6.35: Variation in the temporal (left) and spatial (right) RoI coverage of Deimos with with different levels of motion propagation fidelity.
This chapter develops the swarm architectures for reconnaissance mission concepts to tumbling asteroids. These architectures should account for two key challenges. First, the spin axis of the asteroids is not fixed, unlike the uniform rotators. This dynamic variation of the spin axis, coupled with their irregular shapes, results in large fluctuations in the surface coverage of the swarms. An additional challenge is that here is the asteroids tend to have slower rotation periods, which results in longer periods between spacecraft passages. In this chapter, the Dual Sphere method is used to design global surface mapping missions to tumbling asteroids while addressing these challenges. We begin this chapter by surveying the relevant work done on missions to asteroids. The behaviors of the spacecraft, swarm, and the trajectory are then described. The applications of these principles are then demonstrated using a case study of designing a mission concept to explore the asteroid 4179 Toutatis. Finally, the sensitivity of the coverage performance of the selected near-optimal swarms is then examined under different categorical perturbations.

7.1 Related Work

Asteroids form a key destination of in-situ exploration Committee on the Planetary Science Decadal Survey (2011). Asteroids are traditionally viewed as the rocky remnants of the solar system formation (Carry, 2012). They are primarily classified based on their composition (Bowell et al., 1978) and location (Tancredi, 2014). Based on their Location, the asteroids are divided into three categories: main-belt, trojans, and near-Earth asteroids. Main-belt objects are found in between the orbits of Mars and Jupiter (Masiero et al., 2011). Trojan asteroids are typically found near the triangular Lagrange points ($L4$ and $L5$) of the Sun-planet three-body systems.
(Gradie and Veverka, 1980). Near-Earth asteroids (NEAs) have orbits in the vicinity of Earth (Bottke et al., 1994). A sub-category of the NEAs are the potentially hazardous asteroids (PHAs). They are named so because of their collision risk and are categorized by a minimum orbit intersection is less than 0.05AU (Perna et al., 2015). Most asteroids are categorized as uniform rotators that spin about a body-fixed principal axis. However, a large number of tumbling asteroids have also been discovered (Scheeres and Schweickart, 2004; Pravec et al., 2005). This vast diversity among asteroids has made them prime candidates for spacecraft missions. Of specific interest to the current chapter are missions to tumbling asteroids. Much work has been done regarding the complex dynamics of a spacecraft in the vicinity of tumbling asteroids (Scheeres, 2016). Spacecraft trajectories for orbits (Scheeres et al., 1998), rendezvous (Fitz-Coy and Liu, 1995), hovering (Nazari et al., 2014), and landing (Misra et al., 2015) have been derived. Existing work has also developed algorithms to detumble a tumbling asteroid for facilitating its capture has been presented (Bazzocchi and Emami, 2018). Their applications to PHAs have also been studied.

The asteroid 4179 Toutatis is classified as a PHA, which is in a 4 : 1 resonance with the Earth (Whipple and Shelus, 1993). Toutatis is assumed to be a S-class asteroid, with a mass of about $5 \times 10^{13}$ kg. The orbital period of Toutatis is about 4.02 yrs (Siregar and Soegiartini, 2013). the first shape model and rotational state of Toutatis were generated based on the radio observations from the Aricebo and Goldstone observatories (Hudson and Ostro, 1995), as shown in Figure 7.1. During one of its close flybys of the Earth, Toutatis was observed by the lunar probe Chang’e-2 in 2012 (Ji et al., 2015), which were used to refine its physical, and orbital properties (Bu et al., 2014). During this flyby high resolution images in the range $2.25 - 8.3$ m/px were generated (Huang et al., 2013), which are presented in Figure 7.2. Current estimates suggest that the asteroid is in a long axis mode (LAM) tumble (Scheeres, 2016) with an effective rotation period of 3.92 days, with a pole declination between 39.5 – 40.4 deg (Takahashi et al., 2013). The radar generated shape model of Toutatis with 3196 triangular faces is used as the nominal shape
Figure 7.1: Radar generated shape model of the asteroid of 4179 Toutatis showing its views about its principal axes. [Source: Hudson and Ostro (1995)].

model of Toutatis.

7.2 Mission Concept Modeling

This section models the critical mission design parameters for the global surface mapping missions to tumbling asteroids. The coverage requirements and spacecraft field of view model considered in the global surface mapping mission are identical to those described in Chapter 5. The spacecraft, swarm, and trajectory behaviors are developed here.
Figure 7.2: Spacecraft images of Toutatis noted from the flyby of Chang’e 2 showing the imaging distance $D$, flyby epoch $T$, and the imaging resolution $R$. [Source: Huang et al. (2013)].

Swarm Behavior

The design of mission concepts to asteroids poses an interesting combination of the previous two case studies. The trajectory escaping the Earth is a hyperbola. However, the remaining trajectory is most likely an elliptical orbit around the Sun. As described in previous chapters, the launch energy at Earth, $C_{3,E}$, is a restricting constraint to design escaping hyperbolas. However, there are two fundamental differences between missions to moons and missions to asteroids. First, during the arrival at the target body, the spacecraft does not enter into the gravity well of a central planet. If the encounter velocity, $v_{\infty,2}$ of the spacecraft with respect to the target is large, the gravitational perturbations of the target can be inconsequential to the spacecraft. However, large $v_{\infty,2}$ values can impede the observations with motion blurs and reduced sensor integration times. Therefore, the trajectories that have both low $C_{3,E}$ and low $v_{\infty,2}$ should be selected. The second key difference here is that, since the trajectory is mainly heliocentric, its deployment in the heliocentric
frame must also be considered.

For this reason, the swarm is assumed to be launched on the epoch $D_L$, with a time of flight $ToF_N$. However, this single mothership trajectory is interrupted after a time $T_{Dep}$ has passed since its launch. At this time, the mothership deploys all participating spacecraft in the swarm on their interplanetary journey to the target asteroid. The design space of the interplanetary trajectory of the swarm is illustrated in Figure 7.3.

Figure 7.3: Illustration of the nominal trajectory (left), and swarm deployment (right) segments of the interplanetary trajectory design.

Upon deployment, the spacecraft in the swarm use their Class 2 behavior to visit the target body in the remainder of their journey. Specifically, the nominal trajectory identifies a trajectory that departs from Earth’s center on $D_L$ and arrives at the center of the target bodies center after $ToF_N$. On the other hand, the deployment behavior modifies the trajectories such that upon deployment on $T_{Dep}$ after $D_L$, the swarm will arrive at their respective encounter points characterized by the spherical coordinates: azimuth angle $\theta_{x,i}$, elevation angle $\theta_{y,i}$ and flyby radius $r_f$, at encounter time $T_j$, past the arrival epoch $D_L + ToF_N$. The deployment results in a trajectory that is different from its nominal trajectory, as illustrated in Figure 7.3,
and thus requires an additional maneuver cost $\Delta v_{Dep}$. Modeling the two trajectories: pre and post-deployment as Lambert arcs, the deployment cost of spacecraft $i$ can be expressed as

$$\Delta v_{Dep,i} = \left| \vec{V}^{-}(T_{Dep}) - \vec{V}_{i}^{+}(T_{Dep}) \right| \quad (7.1)$$

Where $\vec{V}^{-}(T_{Dep})$ denotes the velocity vector on the nominal trajectory on $D_L + T_{dep}$, and $\vec{V}_{i}^{+}(T_{Dep})$ denotes the velocity vector of spacecraft $i$ on $D_L + T_{dep}$ to resume it’s remaining heliocentric journey and arrive at its corresponding encounter point. The wet mass of the seed spacecraft in the swarm can then be estimated by noting the fuel requirements of the worst-case deployment maneuver. This allows us to define the trajectory and swarm design problems as follows:

**Trajectory Design**

The trajectory design problem will find an optimal swarm configuration whose worst-case launch, encounter, and deployment fuel requirements are minimized. The nominal trajectory design using launch and time of flight is similar to the trajectory design approach described in Chapters 5 and 6. However, after the deployment, two encounters with the target on arrival are considered. The first encounter occurs on $D_L + ToF_N$, while the second occurs after one rotation period $T_{Rot}$ of the target, i.e., on $D_L + ToF_N + T_{Rot}$. At each of these encounters, the deployment costs associated with 6 extreme encounter configurations are evaluated at 6 extreme $\theta_{x,i}$, and $\theta_{y,i}$ values illustrated in Figure 7.4.

Let the maximum deployment maneuver cost noted from these 2 bounding encounters, each with 6 different encounter locations be denoted by $\Delta v_{Dep,max}$. The trajectory design objective can now be formulated as

$$J_{Tr} = w_1 \left( \frac{v_{\infty,2}}{v_{\infty,max}} \right)^2 + w_2 \left( \frac{C_{3,E}}{C_{3,max}} \right) + w_3 \left( \frac{\Delta v_{Dep,max}}{\Delta v_{max}} \right)^2 \quad (7.2)$$

Where $w_1$, $w_2$, $w_3$ are user defined weights emphasizing the relative importance of minimizing $C_{3,E}$, $v_{\infty,2}$, and $\Delta v_{Dep,max}$ respectively. Similarly, the parameters $v_{\infty,max}$,
Figure 7.4: Illustration of the 2 encounter designs (left), and the 6 encounter locations (right) considered at each encounter (right).

\[ C_{3,\text{max}}, \text{ and } \Delta v_{\text{max}} \] are user defined upper bounds on the corresponding parameters.

The trajectory design gene is presented in Figure 7.5. In order to avoid trajectories that deploy immediately after launch, i.e., \( T_{\text{Dep}} = 0 \), we place a lower bound on \( T_{\text{Dep}} \), such that it is at least a fraction \( a_1 \) of \( ToF_N \) as shown in Figure 7.5. The trajectory design problem is formulated in Equation 7.3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Launch epoch</th>
<th>Nominal time of flight</th>
<th>Deployment Time past launch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>( D_L )</td>
<td>( ToF_N )</td>
<td>( T_{\text{Dep}} )</td>
</tr>
<tr>
<td>Range</td>
<td>Real</td>
<td>Integer</td>
<td>Integer ( \text{round}(0.9a_1 ToF_N) )</td>
</tr>
</tbody>
</table>

Figure 7.5: Gene map of the trajectory design problem to tumbling asteroids.
min $J_{Tr}$

s.t. $v_{\infty,2} \leq v_{\infty,max}$

$C_{3,E} \leq C_{3,max}$

$\Delta v_{Dep,max} \leq \Delta v_{max}$

(7.3)

**Swarm Design**

The swarm design problem will search for configurations that, after deployment, can generate the required coverage $P_{map,R}$ with a minimum number of spacecraft, without any collisions. The spacecraft in the swarm will have multiple encounters with the target body, each of which occurs at a time $T_j$ past the arrival epoch $D_L + ToF_N$. Each encounter contains $N_{e,j}$ spacecraft, whose encounter location is parameterized by the spherical coordinates: $\theta_{x,i}$, $\theta_{y,i}$, and $r_f$ as shown in Figure 7.6. Additionally, the maximum deployment maneuver cost of all spacecraft in the swarm $\max(\Delta v_{Dep,i})$ is also assumed to be upper bounded by $\Delta v_{max}$. The swarm design problem can, therefore, be formulated as

Figure 7.6: Illustration of the geometrical parameters involved in designing the swarm encounters with the tumbling asteroid.
\[
\begin{align*}
\text{min } & J_{\text{Sw, map}} = N_{\text{Sw, map}} \\
\text{s.t. } & P_{\text{FoM, map}} \geq P_{\text{map, R}} \\
& \text{Collision flag } \leq 0 \\
& \max(\Delta v_{\text{Dep, i}}) \leq \Delta v_{\text{max}} \\
& \max(v_{\infty, i}) \leq \Delta v_{\infty, \text{max}}
\end{align*}
\] (7.4)

Where the swarm size \( N_{\text{Sw, map}} \) is determined using Equation 6.16. The swarm design gene listing the above-described swarm design parameters is presented in Figure 7.7.

<table>
<thead>
<tr>
<th>Parameter</th>
<th># Asteroid encounters</th>
<th># Spacecraft each encounter</th>
<th>Encounter time past arrival epoch</th>
<th>Spacecraft azimuth at encounter</th>
<th>Spacecraft elevation at encounter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>( N_p )</td>
<td>( N_{p, 1} ) ... ( N_{p, N_p} )</td>
<td>( T_1 ) ... ( T_{N_p} )</td>
<td>( \theta_{x, 1} ) ... ( \theta_{x, N_{\text{Sw}}} )</td>
<td>( \theta_{y, 1} ) ... ( \theta_{y, N_{\text{Sw}}} )</td>
</tr>
<tr>
<td>Range</td>
<td>Integer ( [1, N_{i, \text{max}}] )</td>
<td>Integer ( [1, N_{f, \text{max}}] )</td>
<td>Real ( [0, T_{\text{Rot}}] )</td>
<td>Real ( [0, 360] ) deg</td>
<td>Real ( [-90, 90] ) deg</td>
</tr>
</tbody>
</table>

Figure 7.7: Gene map of the swarm design problem to tumbling asteroids.

**Spacecraft Design**

Since the mothership is injected with the required \( C_{3, E} \) energy by the launch providers Wertz et al. (2011), the spacecraft in the swarm must deterministically compensate only for the deployment maneuver. Therefore, the seed spacecraft in the swarm is budgeted with \( \Delta v_{\text{net}} \) that corresponds to a worst-case maneuver cost \( \max(\Delta v_{\text{Dep, i}}) \) with a 30 % surplus fuel for unaccounted errors. The net budgeted \( \Delta v \) is therefore expressed by Equation 7.5. The fuel mass is then computed using Equation A.1.

\[
\Delta v_{\text{net}} = 1.3 \max(\Delta v_{\text{Dep, i}})
\] (7.5)
Sensitivity Study Parameters

To study the sensitivity of the coverage performance of the swarms to different uncertainties, the optimal design is perturbed using different categorical perturbations described in Chapter 4. Similar to the previous case studies, four categories of perturbations are considered: Spacecraft outage, error in the knowledge of the asteroid’s location, error in encounter shaping, and sensitivity of coverage performance to the high fidelity dynamics. In these studies, the worst-case perturbation that makes the coverage drop below $P_{thr}$ is noted.

Table 7.1: Parameters used to describe the dynamical environment around Toutatis.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ephemeris epoch</td>
<td>10 Apr 2010</td>
</tr>
<tr>
<td>Semi-major axis</td>
<td>2.54 AU</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.629</td>
</tr>
<tr>
<td>Inclination</td>
<td>0.446 deg</td>
</tr>
<tr>
<td>RAAN</td>
<td>124 deg</td>
</tr>
<tr>
<td>Argument of periapsis</td>
<td>279 deg</td>
</tr>
<tr>
<td>Mean anomaly at epoch</td>
<td>127 deg</td>
</tr>
<tr>
<td>$T_{Rot}$</td>
<td>5.42 days</td>
</tr>
<tr>
<td>$[\delta_l, \delta_u]$</td>
<td>$[39.5, 40.5]$ deg</td>
</tr>
<tr>
<td>$[\eta_l, \eta_u]$</td>
<td>$[20.5, 21.9]$ deg</td>
</tr>
<tr>
<td>Mass</td>
<td>$5.5 \times 10^{13}$ kg</td>
</tr>
<tr>
<td>$[I_{xx}, I_{yy}, I_{zz}]$</td>
<td>$[0.437, 1.53, 1.57] \times 10^{13}$ kgm$^2$</td>
</tr>
<tr>
<td>$i_{Sol}$</td>
<td>85.4 km</td>
</tr>
<tr>
<td>$C_{20}$</td>
<td>0.778 km$^2$</td>
</tr>
<tr>
<td>$C_{22}$</td>
<td>$-0.016$ km$^2$</td>
</tr>
<tr>
<td>Albedo</td>
<td>0.185</td>
</tr>
</tbody>
</table>

### 7.3 Numerical Simulations

This section demonstrates the architectures described above to design a global surface mapping mission to the tumbling asteroid 4179 Toutatis. The objective of the
mission concept is to provide a surface coverage figure of merit of at least 90% of the surface of Toutatis. The coverage requirement parameters are the same as the ones used in the global surface mapping swarm described in Table 5.1. As a result the designed flyby altitude $h_f$ was noted as 110 km. The camera half FoV $\eta_C$ computed using Equation 3.2 was noted as 1.26 deg. The parameters used to model the dynamical environment (Giorgini, 2015; Scheeres, 2016) of Toutatis are presented in Table 7.1.

Figure 7.8: The nominal and Dual Sphere models of Toutatis used for computing the surface coverage.

A 3300 triangular face model of Toutatis generated from Doppler imaging (Scheeres et al., 1998) was used as the nominal shape model of the target body. The maximum and minimum radii of the model were noted as 2.42 and 0.713 km, respectively, which were then used to construct the Dual Sphere shape models. The nominal and Dual Sphere shape models of Toutatis used in the current work are presented in Figure 7.8. The geometrical parameters measured from all the shape models are presented in Table 7.2. The design of the interplanetary trajectory, swarm, and seed spacecraft are now described as follows.

Trajectory Design

The automated trajectory design problem presented in Equation 7.3 is solved by using the modified real-valued Genetic Algorithm (GA) described in (Deb, 2000).
Table 7.2: Geometrical properties measured from the nominal and Dual Sphere shape models of Toutatis.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range Sphere</th>
<th>Nominal Model</th>
<th>View Sphere</th>
</tr>
</thead>
<tbody>
<tr>
<td># Faces</td>
<td>1280</td>
<td>3196</td>
<td>1280</td>
</tr>
<tr>
<td>Radius (km)</td>
<td>0.713</td>
<td>1.37 (mean)</td>
<td>2.42</td>
</tr>
<tr>
<td>Area (km²)</td>
<td>73.5</td>
<td>22.5</td>
<td>6.36</td>
</tr>
</tbody>
</table>

Each trajectory optimization trial was configured to run for a maximum of 1000 generations, where each generation spanned a total of 1000 design genes. A stall stop criterion of 100 generations was used to identify the optimal solution in an optimization trial. The parameters described in Table 7.3 are passed as input arguments to the optimizer. The weights were selected, such that the minimization of encounter velocity magnitude was weighted relatively larger than the $C_{3,E}$, and $\Delta v_{Dep,max}$. However, bound constraints were placed on the remaining parameters to make sure that the generated feasible solutions were practical.

Table 7.3: Input parameters to design the interplanetary trajectory of the spacecraft swarm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[D_{L,min},D_{L,max}]$</td>
<td>[Aug 1st 2020 – Dec 31st 2022]</td>
</tr>
<tr>
<td>$ToF_{max}$</td>
<td>2000 days</td>
</tr>
<tr>
<td>$a_1$</td>
<td>0.25</td>
</tr>
<tr>
<td>$v_{so,max}$</td>
<td>1 km/s</td>
</tr>
<tr>
<td>$C_{3,max}$</td>
<td>70 km²/s²</td>
</tr>
<tr>
<td>$\Delta v_{max}$</td>
<td>1 km/s</td>
</tr>
<tr>
<td>$[w_1, w_2, w_3]$</td>
<td>[0.4, 0.3, 0.3]</td>
</tr>
</tbody>
</table>

**Optimization** The results of 5 GA optimization trials for the trajectory design is presented in Figure 7.9. The trials were able to converge to their optimal solutions in about 110 generations, after which they were stopped when the minimum fitness stalled for 100 generations. The solutions converged to a minimum value of $J_{Tr} = 0.327$. The final generation identified 4526 feasible solutions whose trajectory
constraint figures of merit are presented in Figure 7.10. The selected solution had a lowest $C_{3,E}$ of 64.9 km$^2$/s$^2$.

Figure 7.9: Variation of the trajectory fitness for the global surface mapping mission across different GA generations showing the individual optimization trials (left), and their statistical distribution (right).

Table 7.4: Design variables of the selected optimal trajectory gene.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_L$</td>
<td>Oct 31st 2020, 08:09:00</td>
</tr>
<tr>
<td>$ToF_N$</td>
<td>1274 days</td>
</tr>
<tr>
<td>$T_{Dep}$</td>
<td>382 days</td>
</tr>
</tbody>
</table>

**Trajectory Performance** The selected optimal trajectory gene is presented in Table 7.4. As seen here, the nominal time of flight of the mothership trajectory is 1274 days, which results in an estimated arrival date of 27 Apr, 2024. The estimated encounter velocity of the mothership with Toutatis was noted as $v_{\infty,2} = 0.349$ km/s as shown in the porkchop plot in Figure 7.11.

The mothership then deploys the swarm 382 days after its launch on 11 Apr 2023. The interplanetary trajectory of the mothership and the swarm are presented in
Figure 7.10: Trajectory figures of merit of feasible solutions identified by all optimization trials.

Figure 7.12. The maximum deployment $\Delta v_{Dep,max}$ required for Toutatis encounters at the 6 boundary points presented in Figure 7.4 was noted as noted as 2.77 m/s.

**Swarm Design**

The automated swarm design problem presented in Equation 7.4 is solved using the real-valued GA (Deb, 2000). Each trajectory optimization trial was configured to run for a maximum of 300 generations, where each generation spanned a total of 25 design genes. A stall stop criterion of 100 generations was used to identify the optimal solution in an optimization trial. The parameters described in Table 7.5 are passed as input arguments to the optimizer.

**Optimization** The results of 5 GA trials for the trajectory design is presented in Figure 7.13. The number of generations required for convergence varied between 129 – 300, where optimization was terminated when the minimum fitness stalled for 100 generations. The solutions converged to a minimum value of $J_{Sw} = 5$
spacecraft. The final generation identified 72 feasible genes whose performance constraint figures of merit are presented in Figure 7.14. The selected solution had the maximum coverage figure of merit $P_{FoM,map}$ of 92.1 %.

Swarm Performance The design parameters of the selected near-optimal swarm gene are presented in Table 7.6. As seen here, the swarm consists of 5 spacecraft which serially map the surface of Toutatis through their encounters. The time past encounters, azimuth and elevation angles of these encounters can be noted from Table 7.6. The heliocentric trajectories of the spacecraft showing the locations of different encounter is presented in Figure 7.15. The swarm configuration has a maximum deployment maneuver requirement of $\max(\Delta v_{Dep,i}) = 2.72 \text{ m/s}$, and encounters Toutatis with a maximum velocity of $\max(v_\infty,i) = 0.349 \text{ km/s}$.

A visualization of a sample encounter of the swarm with the Toutatis is presented in Figure 7.16. The sample encounter corresponds to the first encounter of the swarm with Toutatis. The visualization presents the instantaneous coverage over the three
shape models of Toutatis. Additionally, it can be seen from Figure 7.16 that only illuminated faces of the models are used for computing the surface coverage, as the shadowed faces are shaded dark. The performance figures of merit of the swarm are presented in Table 7.7.

As noted from Table 7.7, the coverage performance of the swarm estimated over $N_{mon,1} = 50$ Monte-Carlo coverage trials is noted as 92.1 %. The time evolution of coverage over the nominal and Dual Sphere models of Toutatis noted over a randomly initialized attitude of Toutatis is presented in Figure 7.17. As noted from here, the first two encounters in the swarm each contribute to nearly 40 % of the total surface coverage, while the remaining 3 encounters cover up to 15 % of the remaining surface of Toutatis.

At the end of all encounters, the above simulation resulted in surface coverage of 97.4 % over the nominal model of Toutatis, along with 96.24 % over its Dual Sphere shape models. The resulting coverage patterns over all three shape models are presented in Figure 7.18.
Table 7.5: Input parameters to design the Toutatis global surface mapping swarm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{map,R}$</td>
<td>90 %</td>
</tr>
<tr>
<td>$\Delta v_{max}$</td>
<td>1 km/s</td>
</tr>
<tr>
<td>$N_{1,max}$</td>
<td>5</td>
</tr>
<tr>
<td>$N_{2,max}$</td>
<td>5</td>
</tr>
<tr>
<td>$N_{mon,1}$</td>
<td>20</td>
</tr>
<tr>
<td>$T_{sim}$</td>
<td>5.42 days</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>10 sec</td>
</tr>
<tr>
<td>$T_0$</td>
<td>1 deg</td>
</tr>
<tr>
<td>$r_{col}$</td>
<td>1 m</td>
</tr>
<tr>
<td>$[\alpha_1, \alpha_2]$</td>
<td>[0.4, 0.6]</td>
</tr>
</tbody>
</table>

Seed Spacecraft Design

As noted above, the maneuver costs associated with the swarm deployment were found to be 2.72 m/s. The net maneuver cost associated with the two maneuvers noted from Equation 7.5 is found to be $\Delta v_{net} = 3.56$ m/s, which requires that the seed spacecraft should be budgeted with a hydrazine mass of 0.034 kg. The resultant design and mission cost parameters associated with the seed spacecraft in the global surface mapping swarm are presented in Table 7.8. The cost breakdown of the swarm which uses the seed spacecraft described here is presented in Table 7.9. As seen here, the estimated cost of the total space segment of the swarm mission is $81.2 \pm 2.5 \text{LY}\$M in the launch year 2020.

Sensitivity Analysis

The coverage sensitivity of the designed swarms to different perturbations is now examined. The effect of the three gene-based perturbations is examined, followed by studying the coverage sensitivity to the fidelity of trajectory propagation. A coverage lower bound of $P_{thr} = 80 \%$ is used to evaluate the worst-case perturbations.
Outage  The input parameters to study the coverage sensitivities to spacecraft outages are presented in Table 7.10. The results of the outage sensitivity studies are presented as follows:

The variation of coverage over the three shape models: nominal and Dual Sphere models of Toutatis, with spacecraft outages in the swarm, are presented in Figure 7.19. As seen here, the swarm is capable of meeting the designed coverage requirement of 90% when there are no outages. The loss of a single spacecraft is shown to bring down the minimum coverage over the three shape models to $92 \pm 2.2\%$, which meets the design requirement of $P_{map,R} = 90\%$. The loss of additional spacecraft results in the decay of the minimum surface coverage to $81.3 \pm 13.7\%$. While the mean coverage is still sufficient to meet the $P_{thr} = 80\%$ requirement, the fluctuations are still large enough to result in a poor coverage performance. Therefore, the safe operating limit of the swarm is at about 80% efficiency (with one defunct spacecraft). The coverage over both the Dual Sphere surfaces was noted as $94.3 \pm 2.9\%$, respectively.
Figure 7.14: Performance figures of merit of feasible swarm genes identified by all optimization trials.

**Asteroid Location**  The input parameters to study the coverage sensitivities to perturbations in the location of Toutatis are presented in Table 7.11. The coverage variation over the shape models of Toutatis is described as follows.

Figure 7.20 describes the effect of its location uncertainties over the surface coverage of Toutatis. As noted from here, the coverage of all three models is found to be greater than the designed $P_{\text{map},R} = 90\%$ requirement when the orbital trajectory of Toutatis is lagging by 3 secs or leading by 4.2 secs. The coverage over all three shape models is seen to fall below $P_{\text{thr}} = 80\%$ for a maximum lag uncertainty of 3.6 secs or a maximum lead uncertainty of 7.6 secs. Therefore, the coverage is seen to meet the design requirements when the target location uncertainty is within $\pm 3$ secs and falls below the minimum threshold when the uncertainty is within $\pm 3.6$ secs.

**Encounter Shaping**  The input parameters to study the coverage sensitivities to perturbations in the spacecraft encounter locations are presented in Table 7.12. The coverage variation over the shape models of Toutatis is presented in Figure
Table 7.6: Design variables of the selected optimal swarm gene.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_v$</td>
<td>5</td>
</tr>
<tr>
<td>$N_{v,i}$</td>
<td>[1, 1, 1, 1]</td>
</tr>
<tr>
<td>$T_i$ (days)</td>
<td>0.16</td>
</tr>
<tr>
<td>$[\theta_{v,i, \phi_{v,i}}]$ (deg)</td>
<td>[142, 72.5]</td>
</tr>
</tbody>
</table>

Table 7.7: Performance figures of merit of the designed global surface mapping swarm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal model</td>
<td>95.4 ± 3.2 %</td>
</tr>
<tr>
<td>View sphere</td>
<td>96.3 ± 2.3 %</td>
</tr>
<tr>
<td>Range sphere</td>
<td>96.5 ± 2.3 %</td>
</tr>
<tr>
<td>$P_{FoM_{map}}$</td>
<td>92.1</td>
</tr>
<tr>
<td>$\max(\nu_{v,i})$</td>
<td>0.349 km/s</td>
</tr>
<tr>
<td>$\max(\Delta v_{Dep,i})$</td>
<td>2.72 m/s</td>
</tr>
</tbody>
</table>

7.21. As noted from here, the coverage is noted to be above the design requirement for a maximum encounter location uncertainty of 19.3 km. However, intermediate fluctuations caused the coverage figure of merit $P_{FoM_{map}}$ to fall below this design requirement. The coverage fell below the minimum threshold requirement for an encounter location uncertainty of 27.7 km. After this peak uncertainty, the coverages fluctuated rapidly, as seen in Figure 7.21 leading to intermittent satisfactory coverage. Therefore, the designed swarm is robust to a maximum encounter location uncertainty of 19.3 km, while can meet the minimum threshold up to 27.7 km.
Figure 7.15: Heliocentric trajectories of the spacecraft in the swarm showing the encounter locations on the orbit of Toutatis.

Table 7.8: Seed spacecraft design parameters and its associated estimated costs.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Δν_{net} (m/s)</td>
<td>3.56</td>
</tr>
<tr>
<td>m_{fuel} (kg)</td>
<td>0.034</td>
</tr>
<tr>
<td>m_{wet} (kg)</td>
<td>20.4</td>
</tr>
<tr>
<td>Launch year</td>
<td>2020</td>
</tr>
<tr>
<td>Interplanetary payload cost (LYSM)</td>
<td>2.72 ± 0.71</td>
</tr>
<tr>
<td>Interplanetary bus and operations cost (LYSM)</td>
<td>15.1 ± 0.86</td>
</tr>
<tr>
<td>Launch cost (LYSM)</td>
<td>0.403 ± 0.12</td>
</tr>
</tbody>
</table>

**Dynamic Sensitivity**  The sensitivity of the coverage of the swarm was studied using by using switched-dynamical models described in Chapter 4. The sphere of influence of Toutatis is noted from Table 7.1 as 85.4 km. In the designed configuration, 2 spacecraft passed through the sphere influence of Toutatis. The acceleration profiles of a participating spacecraft which passed through the sphere of influence of Toutatis is presented the semilog plot in Figure 7.22.

As seen here, the solar spherical gravity is the dominant source of spacecraft dynamics. However, the third body dynamics cause significant perturbations as the spacecraft approaches its encounter location. The third body perturbative accelera-
Figure 7.16: Spacecraft trajectories during an encounter with Toutatis (left) showing its instantaneous coverage over the three shape models of Toutatis (right).

Table 7.9: Estimated swarm mission costs associated with the Toutatis mapping swarm.

<table>
<thead>
<tr>
<th>Component</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission cost (LYSM)</td>
<td>$79.2 \pm 2.5$</td>
</tr>
<tr>
<td>Launch cost (LYSM)</td>
<td>$2.01 \pm 0.27$</td>
</tr>
<tr>
<td>Total space segment cost (LYSM)</td>
<td>$81.2 \pm 2.5$</td>
</tr>
</tbody>
</table>

...tion is seen to vary between $4.37 \times 10^{-13}$ m/s² at the beginning of the simulation, and is seen to reach a maximum of $7 \times 10^{-4}$ m/s² as the spacecraft approaches Toutatis. The solar radiation pressure is seen to have a near-constant perturbative acceleration with a magnitude of about $10^{-10}$ m/s. Once inside the sphere of influence the aspherical acceleration is seen to reach a peak value of $3.25 \times 10^{-10}$ m/s². However, this interaction is only for a brief duration of about 4.32 mins. As a result of this short term effect of the low-magnitude three-body and aspherical perturbations, the coverage performance is not affected by dynamical perturbations, as seen in Figure 7.23.
7.4 Discussion

This chapter developed the design architectures for reconnaissance missions to tumbling asteroids using the IDEAS framework. Such architectures need to address several critical challenges. First, the target bodies do not have a fixed spin axis and therefore undergo a tumbling motion. Second, the irregular shape of the target body results in a significant variation in the surface coverage by the swarm when the target is initialized with a random attitude. Additionally, the spacecraft are primarily under the influence of solar gravity, where they follow elliptical trajectories. The challenge, therefore, is to configure the solar trajectories of the
spacecraft swarms such that their encounters are capable of meeting the coverage performance requirements. To address these challenges, the current chapter developed a mother-daughter configuration of the swarm, where the swarm was launched on a mothership, which deployed the individual spacecraft at a designed location on its heliocentric trajectory. Additionally, models to describe the interactions of the swarm and its coverage performance were developed. Following this, the trajectory, swarm, and spacecraft designs were formulated. To address the coverage variation due to the non-uniform shape of the asteroid, the coverage evaluations require a larger number of Monte-Carlo trials. The algorithms described in the current work were demonstrated using a numerical case study of a global surface mapping mission design to the tumbling asteroid 4179 Toutatis, and examining the sensitivity of the coverage performance of the designed swarm to various perturbations.
Figure 7.19: Variation in the surface coverage of Toutatis with spacecraft outages in the designed global surface mapping swarm.

Table 7.11: Input parameters to study the coverage sensitivity to the orbital location of Toutatis.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{thr}$ (%)</td>
<td>80</td>
</tr>
<tr>
<td>$N_{mon,2}$</td>
<td>200</td>
</tr>
<tr>
<td>$\Delta T_0$ (deg)</td>
<td>$[-1, 1]$</td>
</tr>
<tr>
<td>Study size (# true anomaly offsets)</td>
<td>201</td>
</tr>
</tbody>
</table>

Table 7.12: Input parameters to study the coverage sensitivity to the encounter location of the spacecraft.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{thr}$ (%)</td>
<td>80</td>
</tr>
<tr>
<td>$N_{mon,2}$</td>
<td>200</td>
</tr>
<tr>
<td>$\Delta R$ (km)</td>
<td>100</td>
</tr>
<tr>
<td>Study size (# spacecraft offsets)</td>
<td>201</td>
</tr>
</tbody>
</table>
Figure 7.20: Variation in the surface coverage of Toutatis with uncertainty in its orbital location.

Figure 7.21: Variation in the surface coverage of Toutatis with uncertainty in spacecraft encounter locations.
Figure 7.22: Perturbative accelerations experienced by a sample spacecraft in the swarm during the simulated rotation period of Toutatis.

Figure 7.23: Variation in the coverage of the Toutatis shape models with different levels of motion propagation fidelity.
The previous chapters used numerical simulations to demonstrate swarm reconnaissance mission concepts to small bodies. While simulations are essential in the early stages of mission design, hardware demonstration of mission concepts is crucial to building the required spacecraft prototypes. Building laboratory demonstration platforms for spacecraft is challenging due to the dynamical environment and the degrees of freedom. Specifically, the motion of a spacecraft is primarily influenced by the gravity of the central source. Other forces, including the control actions, have smaller magnitudes and can act for shorter durations. Additionally, a spacecraft can exhibit completely decoupled motion along all the 6 degrees of freedom. Recreating these conditions is challenging for ground-based platforms. As a result, there are no platforms to demonstrate spacecraft mission concepts around small bodies. This chapter describes the development of a hardware laboratory platform to advance the simulation mission concepts into functional spacecraft prototypes, as shown in Figure 8.1. Specifically, this chapter develops a hardware platform to demonstrate multi-spacecraft mission concepts identified by IDEAS. Here, we use unmanned air vehicles (UAVs) to simulate the spacecraft in the designed concepts. The UAVs perform autonomous visual reconnaissance of a target body that is encountered along their flight path. The visual feed from the UAVs is used to reconstruct the three-dimensional (3D) surface of the target using photogrammetry principles. This chapter demonstrates that the principles of IDEAS can also be generalized to multi-agent autonomous systems such as UAVs, which, in turn, can serve as functional simulators for demonstrating multi-spacecraft reconnaissance mission concepts. The Multi-Agent Photogrammetry of Small bodies (MAPS) testbed serves as a hardware demonstration platform created to test and study the multi-spacecraft reconnaissance algorithms. We begin this chapter by surveying related work done in
multi-agent autonomous systems. The MAPS testbed and its constituents are described in detail. The design of a reconnaissance mission concept with a multi-agent autonomous system is then formulated as a global surface mapping mission problem that is solved by the Automated Swarm Designer module of IDEAS. A near-optimal solution identified by IDEAS is then implemented inside the MAPS testbed, and the feed from the participating UAVs is used to reconstruct the surface of the target body.

Figure 8.1: Illustration of UAVs serving as hardware platforms to advance simulated spacecraft as functional prototypes.

8.1 Related Work

Unmanned Air Vehicles (UAVs) are defined as uninhabited and reusable motorized air vehicles (van Blyenburgh, 2000). Other common descriptions of UAVs include drones, Remotely Piloted Vehicles (RPVs), and micro air vehicles (Finn and Wright, 2012). Based on their built, UAVs are classified into 4 types: fixed-wing, tilt-wing, single rotor, and multi-rotor vehicles (Bura, 2015). Multi-rotor UAVs are widely used in engineering applications due to their low cost, low weight, and operational ease. However, they also suffer from limitations such as low payload capacity and susceptibility to wind disturbances (Erceg et al., 2017). Traditionally, UAV applications were motivated by military goals such as unmanned inspection Rakha and Gorodetsky (2018), surveillance (Cavoukian, 2012), and mapping hostile ter-
ritories (Corcoran, 2014). However, with the advancement of low-cost technology, UAVs have also been integrated into civilian applications. Notable civilian sectors that have adopted UAVs include Fire fighting (Ghamry et al., 2017), energy sector (Quater et al., 2014), agriculture (Honkavaara et al., 2013), remote sensing (Everaerts et al., 2008), postal services (Jung and Kim, 2017), and communications (Cetin and Zagli, 2012). Low-cost UAVs such as quadcopters have been used as research platforms for several engineering applications (Bürkle et al., 2011). Existing research has focused on developing formation flying control algorithms for UAVs (Wang et al., 2007). They have also been used as functional mock-ups to study the formation flying operations of spacecraft (Richards et al., 2002). At the current state-of-the-art several off-the-shelf UAV solutions are available. Notable UAV suppliers include Parrot Drones SAS, Shenzen Da-Jiang Innovations (SZ DJI) Technologies, and Skydio Inc. Most commercial solutions provide their proprietary flight control libraries, which are then used to develop customized in-house flight software tailored to specific applications. These libraries are programmed using real time operating systems such as the robotic operating system (ROS) to facilitate real time data transfer and hardware integration (Quigley et al., 2009). This chapter focuses on using commercial off-the-shelf UAVs for photogrammetric surface reconstruction of a target body.

Photogrammetry deals with image-based measurements and interpretation to derive information about an object’s shape and location from one or more images (Luhmann et al., 2013). Photogrammetric principles have numerous applications, such as generating 2D surface maps and orthophotos of areas of interest (Ruzgienë et al., 2015), generating 3d models of buildings and terrains (Shashi and Jain, 2007), medical image alignment (Patias, 2008), and visual odometry of autonomous vehicles (Yoon and Kim, 2019). Photogrammetry has been performed from terrestrial (Yao et al., 2011), aerial (Nex and Remondino, 2014), and orbital platforms (Pearse et al., 2018) that carry different types of visual sensors. Relevant to the current work, the field of photogrammetry, which deals with constructing a 3D point cloud of a target feature from multiple images taken from either a single moving sensor or
multiple sensors, is known as Structure from Motion (SfM). The difference between SfM and traditional photogrammetric methods is that the sensors’ location and orientation need not be known apriori but are computed from the sourced images (Westoby et al., 2012). The images are required to capture the target from multiple perspective views, as illustrated in Figure 8.2. It should be noted that SfM is not a single operation on a set of input images, but a broad sequence of multiple operations which typically include feature extraction (Lowe, 1999), sparse point cloud generation (Snavely et al., 2008), dense point cloud generation (Furukawa and Ponce, 2007), and texturing (Baumberg, 2002). Several commercial and open source solutions that implement different versions of the SfM pipelines are currently available. Some notable open-source SfM software are COLMAP (Schoenberger and Frahm, 2016), and Meshroom (Moulon et al., 2012a), VisualSFM (Wu, 2013). Existing research on SfM algorithms has largely been restricted to computer vision (Dellaert et al., 2000; Schoenberger and Frahm, 2016) and geology (Mancini et al., 2013; Dietrich, 2014) disciplines. This chapter improves the state-of-the-art research described here by building a hardware testbed for SfM with multiple UAVs. The participating UAVs reconstruct the surface of a target small body mock-up, which
is encountered along their flight path. The Automated Swarm Designer module of IDEAS is used for identifying the final design solution, which is implemented on the developed MAPS testbed.

8.2 Experimental Setup

In-order to demonstrate small body reconnaissance operations using UAVs, we develop the Multi-Agent Photogrammetry of Small bodies (MAPS) testbed. The MAPS testbed provides an experimental setup that allows autonomous flybys of autonomous UAVs, whose image feed is used to reconstruct the surface of a small body mock-up, which is placed along its flight path.

Hardware Architecture

The hardware architecture showing different physical elements of the MAPS testbed are presented in Figure 8.3. The constituent subsystems are described as follows.

![Figure 8.3: Hardware architecture of MAPS testbed showing the interactions between various constituent subsystems.](image)

**Master Computer** The master computer of the MAPS testbed has several important roles to play in running a reconstruction experiment. First, upon receiving the user’s start command, the master computer signals the motion tracking system
to initiate object tracking and the individual flight computers to send flight control commands to individual drones. Upon completing a mapping experiment, the video feed from individual drones is transferred to the master computer. The video feed collection is then processed using a video processing pipeline for 3D reconstruction of the flight path, model extraction, and area measurement. The mission parameters corresponding to autonomous reconnaissance, which are described in the following sections, are programmed into the master computer. These parameters are then passed onto the individual flight computers at the start of the trial. The master computer is a Lenovo S-130 laptop, which consists of a 1.1 GHz Intel Celeron processor, with 4 GB RAM, and 64 GB storage memory. The master computer initiates the robotic operating system (ROS) environment and is configured as the ROS-Master computer for communication between various nodes. Inside the ROS environment, the master computer communicates with the motion tracking system and the individual flight computers using wireless networking connections. The communication with the motion tracking system is achieved using the onboard WiFi adapter, which uses single-receiver, and single-transmitter channels running on the IEEE 802.11 AC protocol. To communicate with the individual drone computers, the master computer broadcasts ROS messages using a USB WiFi dongle by TP-Link, which operates at a bandwidth of 2.4 GHz.

**Flight Control Computers** The individual flight control computers run the flight control software and are used to send control commands to the individual UAVs. The flight control software is developed using the AR-Drone software development kit (ARSDK) (Krajník et al., 2011). The flight control computers used in the current work are physically identical to the master computer. However, their main objective is to ensure that the individual drones adhere to the mission parameters supplied by the master computer. The mission parameters are used to define reference trajectories for the drone, which are then enforced using a reference tracking control law that is programmed into the flight control software. These computers receive the broadcasted messages from the master computer using their onboard
WiFi adapter. The communication with the individual drones is achieved using a WiFi USB dongle operating at 2.4 GHz. The communication with the drones is programmed as a two-way interaction: Each flight control computer transmits flight control commands to the drones during the mapping mode. Once the drones finish the sensing task, they transmit their video recordings to their corresponding flight computer. The individual feeds are then gathered at the master computer.

![Figure 8.4: The Parrot ANAFI drone which is used as the seed autonomous agent in the MAPS testbed.](image)

**Drones** The drones are autonomous reconnaissance agents in the mapping experiments. As illustrated in Figure 8.3. The drones record the video of their flight path based on their distance to the target small body. The flight path is controlled by the reference tracking control commands sent by the individual flight computers. In the current work, the quadcopter Parrot ANAFI, shown in Figure 8.4, is used as the seed autonomous agent inside the MAPS testbed. As shown here, the drone is equipped with reflective object markers that enable its tracking through the motion tracking system. The performance specifications of the selected drone are presented
in Table 8.1. The embedded motherboard controls the operation of all onboard drone subsystems. Relevant to the current work are two crucial subsystems of the drone: First, its embedded navigation and control system, which accepts commands from the flight control computer, and second its camera, which begins recording when the distance to the target falls below the imaging radius. The two subsystems are described as follows:

Table 8.1: Technical specifications of the Parrot ANAFI drone.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total mass</td>
<td>0.32 kg</td>
</tr>
<tr>
<td>Bounding dimensions</td>
<td>$175 \times 240 \times 65 \text{ mm}^{3}$</td>
</tr>
<tr>
<td>Battery capacity</td>
<td>2500 mAh, at 3.6 V</td>
</tr>
<tr>
<td>Maximum flight time</td>
<td>25 mins</td>
</tr>
<tr>
<td>Maximum vertical speed</td>
<td>4 m/s</td>
</tr>
<tr>
<td>Maximum horizontal speed</td>
<td>15.3 m/s</td>
</tr>
<tr>
<td>Maximum control range</td>
<td>4 km</td>
</tr>
<tr>
<td>Altitude sensors</td>
<td>Ultrasound sensor, Vertical camera</td>
</tr>
<tr>
<td>Inertial navigation</td>
<td>3-axis accelerometer, 3-axis gyroscope, Barometer, Magnetometer</td>
</tr>
<tr>
<td>Absolute navigation</td>
<td>GPS sensor</td>
</tr>
</tbody>
</table>

**Navigation & Control System**  The hardware architecture of the navigation and control system is presented in Figure 8.5. As seen here, the input motion control commands from the flight control computer are passed to the drone’s motherboard, where they are converted to individual rotor voltage commands. The input motion commands to the motherboard are formatted using the ARSDK libraries as a 4 component vector with signed integer entries between $[-100, 100]$. The 4 command vector components denote the percentages of requested vertical throttle $U_T$, roll $U_R$, pitch $U_P$, and yaw rate $U_{YR}$ over their corresponding maximum limits.

The resultant drone trajectory is measured using a suite of navigation sensors presented in Table 8.1. However, to realize an autonomous flight, few critical challenges need to be addressed here. First, the measurements of the altitude and
Figure 8.5: Hardware architecture of the ANAFI navigation and control system.

Inertial sensors are handled internally by the motherboard and are unavailable to the end-user. Second, the poor indoor performance of the GPS sensor prevents using the onboard absolute navigation system. Finally, the conversion of the motion control commands to the rotor voltages, along with onboard sensor data processing, is done using proprietary manufacturer algorithms and therefore needs to be used as ‘black box’ solutions. To address these challenges, a motion tracking system is used for the absolute navigation of the drone. This is achieved by placing reflective markers on the body of the drone, as shown in Figure 8.4. The navigation information noted from the motion tracking system is then used by the flight control computer to generate the 4 motion control commands described above.

**Drone Camera System** The hardware architecture of the camera system of the drone is presented in Figure 8.6. The motherboard also controls the camera system of the drone. As shown in Figure 8.4, the drone’s camera is mounted on a 1-axis gimbal, which is controlled through a tilt command $\theta_T$ to the motherboard from the flight control computer. The motherboard converts the tilt command to a voltage command, which is then supplied to the gimbal servo. The gimbal tilt angle
commands are limited restricted between $[-90, +90]$ deg.

Figure 8.6: Hardware architecture of the ANAFI camera system.

The camera of the drone carries a 10.7 mm CMOS sensor with a 35 mm equivalent focal length. The camera supports multiple modes of high-resolution photography and videography. To prevent shutter related issues during autonomous mapping operations, the drones are configured to operate in the video recording mode. The recorded videos are then converted to constituent images during data processing. The critical configuration parameters of the recorded videos are presented in Table 8.2. As seen here, the camera is configured to operate in a 4k resolution mode where the angular camera resolution is calculated to be 0.018 deg/px. The drone video recordings are gathered in the MP4, where the incoming image stream is recorded at 30 fps.

**Motion Tracking System** The motion tracking system used is comprised of 8 identical Vicon Vantage 5 cameras that track objects in the near-infrared spectrum. The cameras are hoisted on a rectangular metallic ring of dimensions 12 m. The supporting ring is located at a distance of 10 m from the ground. The cameras are arranged on the ring in a $2 \times 4$ configuration, where the 2 rows that hoist 4 cameras
Table 8.2: Camera parameters used to configure the drone recordings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel resolution</td>
<td>$3840 \times 2160$ px</td>
</tr>
<tr>
<td>Angular field of view (Horizontal)</td>
<td>69 deg</td>
</tr>
<tr>
<td>Angular resolution</td>
<td>0.018 deg/px</td>
</tr>
<tr>
<td>Recording framerate</td>
<td>30 fps</td>
</tr>
</tbody>
</table>

each are 8m long. The safety structure of the experiment is located on the ground and is contained inside the metallic ring. The different hardware components of the motion tracking system are illustrated in Figure 8.7. As seen here, the Vicon cameras are connected to a motion capture (MoCap) computer through a multi-port Ethernet switch. The MoCap computer runs Vicon Tracker, a proprietary software that allows users to configure and operate the motion tracking system. The motion control computer interfaces with the master computer and the individual flight control computers using a ROS-based MoCap package (Gargioni et al., 2019).

![Figure 8.7: Hardware architecture of the motion tracking system.](image)

The primary goal of the motion tracking system is to establish an inertial frame $N$, which allows us to track the pose and odometry of objects of interest. The origin
Figure 8.8: Drones configured with object tracking markers (left), which are then used to define skeleton structures in the Vicon Tracker (right) for motion tracking. and orientation of the $N$ frame are specified during the calibration of the Vicon cameras. Objects that need to be tracked are configured with reflective markers. In the current work, we use spherical reflective markers of diameter 12.7 mm to configure the participating drones. To track each drone individually, the markers are placed in unique, asymmetrical configurations on the top surface of each drone. These configurations are then used to define the skeleton structure of the drones, as shown in Figure 8.8, and are stored into the Vicon Tracker V3.5 software to avoid object reconfiguration. The defined drone objects are then used to compute the pose and odometry information. The pose data stream refers to the drone configuration space: Cartesian position vector of each drone, $^N\bar{R}_i$, and attitude of each drone expressed as a quaternion (Schaub and Junkins, 2013) $q_i$. The odometry data stream describes each drone’s velocity $^N\bar{V}_i$, and its angular velocity $\bar{\omega}_i$. In its existing configuration, the positional tracking of the Vicon cameras was accurate up to 0.1 mm. The MoCap ROS driver is configured to publish the pose and odometry information in the ROS environment at 100 Hz.
Small Body Simulation System  A small body simulation system is developed, which is used to simulate the target small body for the reconstruction experiments. The small body simulation system consists of a small body mock-up hoisted on a vertical stand and a background simulator, as shown in Figure 8.9.

![Figure 8.9: Different components of the small body simulation system.](image)

Small Body Mock-Up  The small body mock-up used in the current work was crafted from papier-mâché. The crafted model was baselined on the nominal shape model of Deimos Thomas et al. (2000), where its maximum diameter was scaled to be 0.56 m. This results in a model which is a 1 : 23247 replica of Deimos. To measure the area, and generate the digital shape models (nominal and Dual Sphere), video recordings of the target are recorded from a pair of ground-based mobile cameras. The recordings are subjected to the SfM pipeline described later in this chapter. The small body mock-up and its digital SfM shape model are presented in Figure 8.10. The SfM model was then used to generate the nominal shape model with about 31k triangular and the Dual Sphere models with 5k faces, which are presented in Figure 8.11. The geometrical parameters computed from the shape models are presented in Table 8.3. The nominal and Dual Sphere shape models are
used in the current work to design the global surface mapping mission.

Figure 8.10: The physical small body mock-up (top), and its baseline shape model generated from the SfM pipeline (bottom).

Table 8.3: Geometrical parameters of the baseline target body shape models.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Nominal Model</th>
<th>View Sphere</th>
<th>Range Sphere</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius (m)</td>
<td>0.232 (mean)</td>
<td>0.323</td>
<td>0.159</td>
</tr>
<tr>
<td>Area (m²)</td>
<td>0.654</td>
<td>1.13</td>
<td>0.32</td>
</tr>
<tr>
<td># Faces</td>
<td>31412</td>
<td>5120</td>
<td>5120</td>
</tr>
</tbody>
</table>

**Background Simulator** The background simulator is used to simulate the deep space environment around the small bodies. The simulator consists of four 65” televisions (TVs) that are hoisted on individual vertical TV stands, as shown in Figure 8.9. The TVs are arranged in the shape of a circular arc to generate an immersive, panoramic background. The simulator displays a looping orbital day of Deimos as observed from its center, in a 180 deg panoramic configuration. It should be pointed out here that the UAV mapping missions inside the MAPS testbed are
carried out under standard indoor lighting conditions. These conditions allow the SfM pipeline to reconstruct the shape model of the target body from the recordings of the UAV cameras. However, at the state-of-the-art, ultra-high sensitivity cameras exist (Tanioka et al., 2003) that can compensate for the low lighting conditions of a space mission. These payloads allow us to extend the results of the SfM mission carried under standard lighting conditions to that performed under low lighting conditions.

Safety Structure  The MAPS testbed is equipped with a safety supporting structure to prevent damage from drone run-offs and crashes. The safety structure is a box-shaped cage with dimensions of $7.2 \times 4.3 \times 2.4$ m$^3$. The skeleton of the cage was built from PVC tubes and is used to support a nylon safety net with 4” square holes along its walls. In addition to the cage structure, styrofoam cushions and foam landing pads are placed inside the safety structure to mitigate any damages to the drone from impact. The origin of the inertial Vicon $N$ frame is located inside the supporting structure, as shown in Figure 8.12. The $N$ frame measurements of the working space inside the cage and the location of the target body are presented in Table 8.4.
Figure 8.12: The experimental setup of a mapping trial inside the MAPS simulator

Table 8.4: Vicon frame measurements of the workspace, and target location inside the supporting structure.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_x$ bounds</td>
<td>$[-2, 2]$ m</td>
</tr>
<tr>
<td>$N_y$ bounds</td>
<td>$[-3, 3]$ m</td>
</tr>
<tr>
<td>$N_z$ bounds</td>
<td>$[0, 2.4]$ m</td>
</tr>
<tr>
<td>Target center location</td>
<td>$[0, 0, 1.3]^T$ m</td>
</tr>
</tbody>
</table>

Software Architecture

The software architecture of the MAPS terminal is presented in Figure 8.13. As seen here, a majority of the interactions between various nodes occur inside the ROS environment. The Melodic version of the ROS environment is used in the current work and is used to interact between different computers that run Linux Ubuntu 18.00 operating system. In a ROS environment, individual software programs are referred to as nodes, which either subscribe or publish information streams labeled as topics (Quigley et al., 2009). Figure 8.13 also illustrates a key advantage of ROS-based software architecture: Nodes inside the ROS architecture can communicate with each other in realtime using properly formatted topics. The roles and contents of individual nodes are described as follows:
205

Figure 8.13: Software architecture of the MAPS testbed showing the interactions between various constituent nodes.

**Command Node** The command node is used to broadcast operational mode commands for the experiment. Starting the command node initiates data logging of an experiment and provides a terminal for a human operator to enter their command inputs. The operational modes for the autonomous agents include several testing modes such as group take-off, hover, homing, land, and record modes. Of crucial interest to the current work is the mapping mode, which signals the flight control nodes to send control commands to the individual drones.

**Motion Capture Node** The motion capture node runs the MoCap ROS package described above, enabling tracking information of the drones to be accessed by the individual flight control nodes. As shown in Figure 8.13, both the command and motion capture nodes run on the master computer. The objects that need to be tracked are manually defined as skeleton objects in the MoCap computer using the Vicon Tracker software. The tracking information of each drone is available as ROS topics corresponding to its assigned name during the object definition. The individual flight control nodes subscribe to the pose and odometry topics. The high-frequency noise in the pose and odometry topics is filtered by the Vicon tracker.
software using a low-pass 1–Euro filter (Casiez et al., 2012). As described above, the motion capture node is configured to publish the pose and odometry topics 100 Hz, which are then subscribed by the flight control nodes.

**Flight Control Node**  The flight control node undertakes the crucial responsibility of communicating with the UAVs. The flight control node sends flight and camera commands using the ARSDK libraries through the USB WiFi dongle of the flight control computers. The flight control node is programmed such, starting the node puts the drone in an armed state where it can read the ARSDK commands. Standard tasks such as individual drone take-off, land, and record videos are available as macro-commands in the ARSDK libraries. Since the mapping task is a custom operation, the control software was developed in-house. The operation of a seed drone during the mapping mode is illustrated in Figure 8.14. As seen here, in the mapping mode, the drone is commanded to take-off and then commanded to track the reference trajectory. On the reference trajectory, the drone starts to record the video when its distance to the target \( r_{Ti} \) falls below the maximum imaging radius \( r_{max} \). Upon reaching the end of the trajectory, the drone is commanded to land, followed by transmitting the recorded video to its flight control computer. Since the take-off, record, and landing commands are available as pre-programmed commands in the ARSDK library, the reference trajectory and tracking control algorithms are developed here and are described as follows.

![Figure 8.14: Concept of operations of a seed drone during the mapping mode.](image-url)
Reference Trajectory Generation. The objective of the designed reference trajectory is to ensure that the camera of the drone tracks the line of sight (LoS) to the target body. The reference trajectory is modeled using Cartesian coordinates to denote the translatory motion of the drone and the $z - y - x$ Euler angle sequence to describe its attitude. Additionally, since the drone’s attitude and trajectory are coupled, we use the camera tilt motion to track the LoS without disrupting the drone’s translatory trajectory. As shown in Figure 8.14, the translational reference trajectory of the drone is a horizontal line parallel to the ground, in the decreasing $y$ direction of the $N$ frame. This means that for a given drone $i$, its $N$ frame Cartesian coordinates $x_i$, and $z_i$ remain constant throughout the reference trajectory. The $y$ coordinate is expressed as

$$y_i = y_0 - v_{y,i}t_i \quad \text{if} \quad t_i \leq ToF_i$$

$$= y_f \quad \text{if} \quad t_i > ToF_i$$

(8.1)

Where $y_0$ and $y_f$ denote the take-off and landing $y$ coordinates respectively, and $v_{y,i}$ denotes the magnitude of the drone’s $y$ component velocity. The time since the start of trajectory is denoted by $t_i$, and $ToF_i$ denotes its designed time of flight. It should be noted here that all drones share the same $y_0$ and $y_f$. This is to ensure that the reference trajectories of all drones share the same spatial shape, while their velocities are allowed to be different. To define the reference attitude, we define a drone centered reference frame $B$, as shown in Figure 8.15.

The $B$ frame of each drone is defined during the object definition in the Vicon Tracker, using the convention that the $x$ axis points from the center of the Vicon object skeleton towards the camera lens when the camera is at 0 deg tilt. The $z$ axis is constructed normal to the skeleton plane of the Vicon object, and the $y$ axis completes the right-hand rule. The origin of the $B$ frame $BO$ is located at the center of the skeleton. Its location in the $N$ frame $N\vec{R}_i = [x_i, y_i, z_i]^T$ is noted from the pose topic through the motion capture node. Since the target body is placed directly above ($z_T > 0$) on the $N$ frame origin, the yaw Euler angle $\psi_i$ of the drone, which orients the drone’s camera towards the target’s azimuth, is given by
\[ \psi_i = \arctan \left( \frac{0 - y_i}{0 - x_i} \right) = \arctan \left( \frac{-y_i}{-x_i} \right) \]  
(8.2)

The compensation for the elevation difference is provided by the camera’s tilt angle \( \theta_{T,i} \) as

\[ \theta_{T,i} = \arcsin \left( \frac{z_T - z_i}{r_{T,i}} \right) \]  
(8.3)

Therefore, the reference trajectory of each drone is parameterized by a vector of 5 coordinates: \( T_{R,i} = [x_i^*, v_{y,i}^*, z_i^*, \psi_i^*, \theta_{T,i}^*]^T \), where the * superscript indicates that a reference value. The control law used to ensure that the drone tracks the presented reference trajectory is described as follows.

**Trajectory Control Algorithm**  
The control loop of the drones is illustrated in Figure 8.16. As seen here, the main objective of the designed controller is to generate the control inputs \( U_{T,i}, U_{R,i}, U_{P,i}, U_{YR,i}, \) and \( \theta_{T,i} \), which modify the dynamics
Figure 8.16: Control loop architecture of the seed drone to track the reference trajectory.

of the drone to track the reference trajectory. Several models have been proposed to describe the nonlinear dynamics of a quadrotor drone accurately; however, they cannot describe all perturbative actions of the drone (Kim et al., 2009) or are not available to an end-user (Bristeau et al., 2011). The ARSDK (Krajník et al., 2011) firmware has developed its control libraries such that the drone dynamics can be described by a linearized model when its corresponding control inputs are used. The linear model is given by

\[
\begin{align*}
\dot{B}v_{x,i} & = M_{1,i}U_{R,i} - M_{2,i}v_{x,i} \\
\dot{B}v_{y,i} & = M_{3,i}U_{P,i} - M_{4,i}v_{y,i} \\
N\dot{v}_{z,i} & = M_{5,i}U_{T,i} - M_{6,i}v_{z,i} \\
\ddot{\psi} & = M_{7,i}U_{YR,i} - M_{8,i}\omega_{z,i}
\end{align*}
\]

(8.4)

The parameters $M_{1,...,8,i}$ denote empirical model constants of a drone $i$. The left superscripts indicate the reference frame in which the individual components are resolved. This model has been adopted by existing research, primarily because it describes a decoupled input to output mapping between the drone states and AR drone control inputs (Engel et al., 2012; Hernandez et al., 2013; Falcón et al., 2013). Furthermore, this model allows the drone motion to be described by 4 states: $Bv_{x,i}$, $Bv_{y,i}$, $Nv_{z,i}$ in lieu of its complex nonlinear dynamics (Santana et al., 2014). Under such model, a PD control law, as described in Doukhi et al. (2017) and Boudjiet and Larbes (2015) is implemented such that
\[
\begin{bmatrix}
N_{U,R,i} \\
N_{U,P,i} \\
N_{U,T,i}
\end{bmatrix}
= K_P \begin{bmatrix}
N x_i^* - N x_i \\
N y_i^* - N y_i \\
N z_i^* - N z_i
\end{bmatrix}
+ K_D \begin{bmatrix}
N v_{x,i}^* - N v_{x,i} \\
N v_{y,i}^* - N v_{y,i} \\
N v_{z,i}^* - N v_{z,i}
\end{bmatrix}
\] (8.5)

Where \( K_P \) and \( K_D \) are 3 \times 3 positive definite diagonal matrices that carry proportional and derivative control gains, respectively. The \( N \) frame commands are then converted to body frame commands \( B\bar{U} \) as

\[
\begin{bmatrix}
U_{R,i} \\
U_{P,i} \\
U_{T,i}
\end{bmatrix}
= [B N] N \bar{U} = R_z(\psi_i) N \bar{U}
\] (8.6)

The yaw control is specified by

\[
U_{YR,i} = k_{p,y} (\psi_i^* - \psi_i)
\] (8.7)

Where \( k_{p,y} \) is a positive yaw control gain. Since the ARSDK libraries allow for direct tilt angle commands to be passed, the camera tilt \( \theta_{T,i}^* \) computed using Equation 8.3 is sent as a command through the flight control node. The commands computed in Equations 8.6, and 8.7 are normalized and rounded off such that each input is an integer in \([-100, 100]\), which are sent from the flight control node at 20 Hz. After completing the mapping mission, the drones send their video recordings via the USB dongle port to the individual flight computers. The flight computers send the gathered videos to the master computer through the SSH file transfer protocol (SFTP) method. The video feeds of all the drones are now subjected to the result processing pipeline described as follows.

**Result Processing Pipeline**

The main objective of the result processing pipeline is to generate a 3D shape model of the target body and measure its surface area using the videos recorded by all the
drones. The different tasks involved in the result processing pipeline are illustrated in figure 8.17. The individual tasks are described as follows.

**Image Decomposition** In this task, each video is broken down into constituent image frames. The scene filter of the VideoLan Client (VLC) player (Müller and Timmerer, 2011). The videos are decomposed such that 1 image is generated for every 1 second of the video. This allows us to derive an empirical relation between the duration of recorded video $D_v$, and the total number of images generated from all recordings $N_{im}$ as

$$N_{im} \approx N_D N_{Sw} = \text{floor} \left( D_v \right) N_{Sw} \quad (8.8)$$

Where $N_D$ denotes the number of images resulting from each drone, and $D_v$ is measured in secs. The pixel resolution of the image is maintained the same as the source video.

**SfM Pipeline** The array of $N_{im}$ images resulting from the image decomposition task is then sent to the SfM pipeline. The SfM pipeline of the Meshroom (Moulon et al., 2012a) is used in the current work to construct the dense point cloud and 3D surfaces resulting from the flight paths of the drones. The different tasks involved in the SfM pipeline is presented in Figure 8.18. As seen here, the pipeline contains
7 sequential tasks that result in a textured 3D surface describing the flight path of the drones.

**Feature Extraction**  Feature extraction corresponds to identifying groups of pixels that share 'similar' characteristics or features. Typically these pixels are invariant under varying camera angles. The Scale-Invariant Feature Transform (SIFT) algorithm (Ives and Delbracio, 2014) is used to identify such features. The SIFT features resulting from this operation are then sent for image matching.

**Image Matching**  Image matching operation finds groups of images observing the same aspects of the scene. To achieve this, the features identified during extraction are sorted into a tree data structure known as a vocabulary tree (Nister and Stewenius, 2006). The distance between the vocabulary trees of each image can then be used to judge whether the participating images describe the same scene.

Figure 8.18: The Structure from Motion pipeline implemented inside the Meshroom software.

**Feature Matching**  The feature matching stage uses the vocabulary trees and computes pairwise comparisons of features between trees. This is typically done in two stages: First, a photometric comparison between two feature sets is compared using the Approximate Nearest Neighbours algorithm (Muja and Lowe, 2009). The geometric features of the photometrically similar features are then compared using the RANdom SAmple Consensus (RANSAC) framework (Derpanis, 2010). The feature matching stage reveals relations between image pairs, which are then used to understand the structure of the scene.
**Point Cloud Generation** The image pair relations generated from feature matching are then used to infer the scene and camera poses in the point cloud generation stage. In this stage, the image features are first fused to obtain points observed in multiple images. This is done first by picking the best image pair that provides a good start for the reconstruction process, and then incrementally common points with the remaining images (Moulon et al., 2012b). The first two images are also used to define a coordinate system for camera pose estimation through photogrammetric methods such as fundamental matrix estimation (Boufama and Mohr, 1995). The generated point cloud and camera pose parameters are then refined using the Bundle Adjustment algorithm (Triggs et al., 1999). This stage generates points observed by multiple images, along with an estimate of the camera pose.

**Depth Map Generation** The camera poses estimated through the point cloud generation stage are used to compute the depth of the generated point cloud. Meshroom uses the Semi-Global Matching algorithm (Hirschmuller, 2007) to generate the depth maps, which provides a parallelized method to calculate depth maps between multiple groups of cameras. In the current work, this stage generates a set of $3D$ vertices $V_F$ of the flight path of the drones.

**Meshing** The vertices generated from depth maps are used for constructing surfaces. The Delaunay triangulation algorithm (Field, 1988) is used to construct the surfaces from these vertices. The triangulation results in a set $F_F$, which describes the order of connectivity of $V_F$ along the flight path of the drones.

**Texturing** The texturing operation is the final stage of the SfM pipeline of Meshroom and generates color maps for faces resulting from the previous stages. Meshroom uses a least-squares algorithm (Lévy et al., 2002) to infer the color of a face by using the color values of its vertices. This results in a texturized $3D$ surface reconstruction of the complete flight path of the drone, which will now be processed.
to extract the target model surfaces.

**Model Extraction and Scaling** Since the SFM pipeline generates the surface of the complete flight path of the drones, in this stage, only the surface corresponding to the target body shape model is extracted. This is achieved by manually filtering $V_F$ and $F_F$ of the 3D surface generated from the SfM pipeline, using MeshLab (Cignoni et al., 2008) an open-source mesh handling software. Furthermore, the model vertices are scaled here to correct for unknown camera intrinsic camera properties using a reference dimension. In the current work, the maximum diameter presented in Figure 8.11 is used as the reference diameter to scale the model vertices. As a result, a vertex set vertices $V_T$, and a set of triangular connectivity between the faces $F_T$ of the target body.

**Area Measurement** The calibrated model is then used to compute the surface area of the generated shape model using the approach described in Chapter 3. If $A_k$, $B_k$, and $C_k$ are the vertices of the $k^{th}$ triangle in $F_T$, the area of the shape model $A_M$ is computed as (Goldman, 1991):

$$A_M = \frac{1}{2} \sum_{k=1}^{N_f} \left| A_k \bar{B}_k \times A_k \bar{C}_k \right|$$

(8.9)

Where $N_f$ is the number of faces noted from the model. This allows us to generate the 3D shape model of the small body mock-up and measure its surface area through autonomous flybys of the UAVs. The design of the multi-agent mission concept that meets the coverage and performance requirements using the Automated Swarm Designer module is now presented here.

**8.3 Mission Concept Design**

This section formulates the automated swarm design problem to design a global surface mapping mission with UAVs. The sensed video feed of the UAVs is used for photogrammetric 3D surface reconstruction of the target small body mock-up. The
Performance Requirements

As described in previous chapters, the objective of the global surface mapping mission is to generate a cumulative coverage figure of merit $P_{FoM,map}$ exceeds a lower bound requirement $P_{map,R}$ with a maximum pixel resolution of $x_R$. Since the angular resolution, $\theta_R$ of the drone is known, the maximum flyby altitude $h_{\text{max}}$ can be determined by noting the geometrical relations in Figure 8.19. The maximum flyby altitude can now be determined as

$$ h_{\text{max}} = \frac{x_R}{2 \tan \left( \frac{\theta_R}{2} \right)} \quad (8.10) $$

Equation 8.10, allows us to define $r_{\text{max}}$, and $r_f$ using the Dual Sphere relations described in Chapter 3. During these flybys, the UAVs must generate a total of $N_{im}$ images. Using Equation 8.8, this corresponds to a video recording duration of $D_v = N_{im}/N_{Sw}$ secs per UAV. The coverage evaluation to estimate $P_{FoM,map}$ is
described as follows.

Figure 8.20: Arrangement of the nominal target body shape model showing the different design gene parameters.

Monte-Carlo Coverage Evaluation During each mapping experiment, the target body was mounted on its hoist stand, shown in Figure 8.9. The target body was nominally mounted such that its center was directly above the origin and with its longest dimension oriented parallel to the $N$ frame $y$ axis, as shown in Figure 8.20. However, the model is subjected to placement and orientation errors. Therefore, each coverage simulation trial was designed such that the location of the target was perturbed from $^N\bar{R}_T$ as

$$^N\bar{R}'_T = ^N\bar{R}_T + ^N\Delta \bar{R}_T$$  \hspace{1cm} (8.11)$$

Where $^N\Delta \bar{R}_T$ is a $3 \times 1$ vector with uniformly distributed random elements in $[-a, a]$. Additionally, since the mount arrested rotations about $y$ axis, the rotational perturbation of the nominal shape model $[TR]$ was modeled as

$$[TR] = R_x(\theta_1)R_z(\theta_3)$$  \hspace{1cm} (8.12)$$
Where $\theta_1, \text{ and } \theta_3$ are uniformly distributed random angular perturbations in $[-b, b]$, and $[-c, c]$ respectively. The user defines the bounds on the perturbations $[a, b, c]$ during the mission definition stage.

**Table 8.21**: Gene map of the automated swarm design problem of the global surface mapping mission implemented inside the MAPS testbed.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Number of Drones in the Swarm</th>
<th>Elevation of the encounter point</th>
<th>Azimuth of the encounter point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>$N_{Sw}$</td>
<td>$\theta_y 1$</td>
<td>$\theta_x 1$</td>
</tr>
<tr>
<td>Range</td>
<td>Integer $[1, N_{max}]$</td>
<td>Real $[-75, 75]$ deg</td>
<td>Real $[1, 179]$ deg</td>
</tr>
</tbody>
</table>

Design Gene Parameters

Similar to the global surface mapping architectures described in the previous chapters, the current reconnaissance mission concept will also contain multiple UAVs visiting the target body. However, since the target body is static, only a single encounter with multiple UAVs is considered. The $N_{Sw}$ UAVs encounter the target body at different azimuth $\theta_{x,i}$, and elevation angles $\theta_{y,i}$ with respect to the target body as shown in Figure 8.20. The design gene used in the current work is presented in Figure 8.21.

Reference Trajectory Parameters

As described in the previous section, the reference trajectory of the drone is parameterized by 5 coordinates $T_{R,i} = [x_i^*, v_y^*, v_z^*, \psi_i^*, \theta_{T,i}^*]^T$. While the angular parameters $\psi_i^*$ and $\theta_{T,i}^*$ vary dynamically with the location of the drones, the translational parameters $x_i^*, v_y^*, v_z^*$ remain fixed on the reference trajectory as illustrated in Figure 8.14. Here the relations between the design gene parameters and their corresponding transnational reference trajectory parameters are described. The $x$ and $z$ coordinates can be expressed by examining the geometrical relations illustrated in Figure 8.20 as
Figure 8.22: Encounter geometry parameters used to determine the reference velocity of the drones.

\[ x^*_i = r_{\text{max}} \cos(\theta_{x,i}) \cos(\theta_{y,i}) \]  \hspace{1cm} (8.13)

and

\[ z^*_i = z_T + r_{\text{max}} \sin(\theta_{y,i}) \]  \hspace{1cm} (8.14)

In order to relate the \( y \) component velocity consider the geometry presented in Figure 8.22. As shown here, the reference trajectory traces a chord inside the recording zone, which is a sphere of radius \( r_{\text{max}} \). Since the \( x \) and \( z \) coordinates are held constant, the reference motion is contained along the \( y \) direction. Additionally, since the target is located directly above the center, the reference trajectory is symmetrical about the \( x-z \) plane. It should be noted here that using the convention of the defined \( N \) frame, the \( y \) coordinate of the drone at the end of the recording
is a negative mirror image of its value at the start of recording, as illustrated in Figure 8.22. Therefore, the displacement of the drone along the $y$ axis $y_{D,i}$ can be expressed as

$$y_{D,i} = -2r_{\text{max}} \text{abs}(\cos(\theta_{x,i}) \sin(\theta_{y,i}))$$ \hspace{1cm} (8.15)

Since the drones spend a duration of $D_v$ inside the recording zone, the reference velocity is computed as

$$v_{y,i}^* = \frac{y_{D,i}}{D_v}$$ \hspace{1cm} (8.16)

Using the expression for velocity, the time of flight $ToF_i$ can be determined as

$$ToF_i = \frac{\text{abs}(y_0 - y_f)}{v_{y,i}^*}$$ \hspace{1cm} (8.17)

Equations 8.13-8.17 allow us to specify the reference trajectories of the individual drones by specifying a design gene.

**Automated Swarm Design**

The design of the global surface mapping mission inside the MAPS testbed can now be formulated as an optimization problem to minimize the number of UAVs that generate the target’s required surface coverage without collisions. It can be inferred that UAVs with slow velocities can theoretically generate sufficient coverage at the cost of increased computation costs associated with downlinking and processing the videos. For this reason, an upper bound $ToF_{\text{max}}$ is placed on the maximum time of flight of the drones. Furthermore, to ensure that the drones do not collide with the target body, a lower bound is placed on the minimum value of $r_{T_i}$ to be greater than the view sphere radius $r_{T,\text{max}}$. The swarm design problem can be formulated as
\[
\begin{align*}
\min & \quad J_{Sw} = N_{Sw} \\
\text{s.t.} & \quad P_{FoM,\text{map}} \geq P_{\text{map},R} \\
& \quad \max(ToF_i) \leq ToF_{\text{max}} \\
& \quad \min(r_{Ti}) \geq r_{T,\text{max}}
\end{align*}
\] (8.18)

8.4 Results

This section presents the results of designing a multi-agent mapping mission and implementing it inside the MAPS testbed. The swarm design problem presented in Equation 8.18 is solved using the Automated Swarm Designer module to identify a near-optimal mission concept. The selected mission concept is then implemented inside the MAPS testbed. The surface reconstruction of the target generated from the video feed of the autonomous flybys is then examined.

Mission Requirements

The objective of the case study considered in the current work is to generate the shape model of the small body mock-up through a global surface mapping mission performed by autonomous drone flybys. The UAVs are expected to cover a minimum of \( P_{\text{map},R} = 95 \% \) of the surface of the model using a maximum baseline resolution of \( x_R = 0.5 \text{ mm/px} \). The participating UAVs should generate a total of \( N_{im} = 120 \) using the video feed of all the drones. The used-defined parameters required for the mission definition are presented in Table 8.5.

The maximum imaging radius \( r_{max} \) required by drone camera described in Table 8.2 in order to meet the coverage requirements is noted as 1.72 m. The desired flyby radius is noted from Table 8.5 as 1.62 m, which is used as the far field distance in the coverage computation during the Monte-Carlo trials.
Table 8.5: Mission definition parameters used for designing the global surface mapping mission implemented inside the MAPS testbed.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{map,R}$</td>
<td>90 %</td>
</tr>
<tr>
<td>$x_D$</td>
<td>0.5 mm/px</td>
</tr>
<tr>
<td>$\Delta h_f$</td>
<td>0.1 m</td>
</tr>
<tr>
<td>$N_{im}$</td>
<td>120</td>
</tr>
<tr>
<td>$N_{max}$</td>
<td>5</td>
</tr>
<tr>
<td>$N_{mon,1}$</td>
<td>10</td>
</tr>
<tr>
<td>$ToF_{max}$</td>
<td>3 mins</td>
</tr>
<tr>
<td>$[y_0, y_f]$</td>
<td>[2, -2] m</td>
</tr>
<tr>
<td>$a$</td>
<td>1 cm</td>
</tr>
<tr>
<td>$b$</td>
<td>1 deg</td>
</tr>
<tr>
<td>$c$</td>
<td>10 deg</td>
</tr>
</tbody>
</table>

Optimization Results

The optimization solver was set up to evaluate a population of 20 design genes per generation. Each generation carried 5% elite genes, and 80% of the remaining genes was formed from crossover children. A maximum of 200 generations was placed, during which the search was terminated when the best fitness stalled for 100 generations. The results of the 5 Genetic Algorithm (GA) optimization trials are presented in Figure 8.23.

As seen here, all optimizations converged to a design of size $N_{su} = 2$ UAVs. The final generations of all design trials identified 95 feasible candidate solutions whose constraint figures of merit are presented in Figure 8.24. The selected optimal design contained the least maximum time of flight of 1.5 mins. The design gene of the selected near-optimal solution is presented in Table 8.6
Figure 8.23: Variation of the global surface mapping fitness across different GA generations showing the individual optimization trials (left), and their statistical distribution (right).

Table 8.6: Design variables of the selected near optimal design gene in the global surface mapping mission.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{SW}$</td>
<td>2</td>
</tr>
<tr>
<td>$[\theta_{xi}, \theta_{yi}]$ (deg)</td>
<td>$[47.3, 4.65]$ [102, 8.18]</td>
</tr>
</tbody>
</table>

Mission Concept Performance

The simulated flight path of the two drones is illustrated in Figure 8.25. The azimuth and elevation angles of the drone trajectories with respect to the shape models are noted from Table 8.6. The value of $\min(r_{Ti})$ was noted as 0.537 m.

The expected coverage figures of merit sampled over 10 Monte-Carlo simulations are $99.1\pm0.03\%$, $97.1\pm0.06\%$, and $99.03\pm0.05\%$ over the nominal, view and range sphere respectively. This provides an effective figure of merit of $P_{FoM,map} = 97.0\%$ which meets the designed coverage requirements. The coverage pattern over all the 3 shape models of the target body, resulting from a simulated flyby of the UAVs, is
Figure 8.24: Performance figures of merit of the feasible design genes identified from the final generations of all optimization trials.

presented in Figure 8.26. The estimated cumulative surface coverages noted during this simulation, over the nominal, view, and range spheres, are 99.2, 97.0, and 99.4 %, respectively. The implementation of the selected mission concept in the MAPS testbed is described as follows.

**Hardware Implementation**

The reference trajectory and control parameters of the drones used in the current work are presented in Table 8.7. As seen here, the velocities of the drones on their reference velocities were 4 and 5 cm/s. This can be compared to the global surface mapping missions to Deimos in Chapter 6, where the spacecraft relative velocities were in the range 0.692 – 2.05 km/s. Thus the MAPS testbed is used to simulate spacecraft operations scaled up to about \((1.73 - 4.1) \times 10^4\) to the demonstrated capability. The trajectories of the two drones noted from 5 experiments are presented in Figure 8.27. The take-off, reference tracking, and landing sequences of the drone can be noted here, along with the segments when the drones start recording their
The individual pose components of the drone trajectories noted from the 5 experiments are presented in Figure 8.28. The slope of the $y$ coordinates corresponds to the reference velocities of the drones presented in Table 8.7. As seen here, the drones are shown to track the designed reference trajectory closely. Each drone is inside the recording zone for about 60 secs. Therefore, the two drones collectively meet the total image time constraint of 120 secs. The results obtained by processing the video feed of individual drones to reconstruct the surface of the target body are described as follows.

**Surface Reconstruction**

**Image Decomposition** The videos recorded by the $N_{Sw} = 2$ drones were converted into images using a conversion ratio of $1/30$, which generated 120 images. Figure 8.29 presents sample images generated from each of the drone’s recordings noted during a mapping experiment. It should be noted here that all images had a
Figure 8.26: Coverage pattern over the nominal and Dual Sphere shape models of the target after the UAV flybys.

pixel resolution of $3840 \times 2160$ pixels, as described in Table 8.2.

**SfM Pipeline** The images generated from each experiment were processed using the SfM pipeline. Figure 8.30 presents a 3D surface reconstruction of the flight path of the drones resulting from the SfM pipeline. The surfaces of the target body and the estimated camera poses can be seen here.

**Model Extraction** The reconstructed flight paths were then processed in MeshLab, where surfaces that did not belong to the target body were cropped. The models were then scaled using the maximum dimension of the target body, as illustrated in Figure 8.10. The extracted shape models of the target from the reconstruction of all 5 experiments are presented in Figure 8.31. As seen here, the shape models generated provided a dense coverage of the model in the top and the front views, while the coverage deteriorated in the bottom view. This is mainly because the drone’s reference trajectories mostly accessed the top and front views of the target.
Table 8.7: Reference trajectory and control parameters of the two drone used in the current work.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Drone-1</th>
<th>Drone-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_i^*$ (m)</td>
<td>1.2</td>
<td>-0.5</td>
</tr>
<tr>
<td>$z_i^*$ (m)</td>
<td>1.3</td>
<td>1.4</td>
</tr>
<tr>
<td>$v_{xy}^*$ (m/s)</td>
<td>0.04</td>
<td>0.05</td>
</tr>
<tr>
<td>$ToF_i$ (mins)</td>
<td>1.58</td>
<td>1.22</td>
</tr>
<tr>
<td>Maximum vertical speed (m/s)</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Maximum roll (deg)</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Maximum pitch speed (deg)</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Maximum yaw rate (deg/s)</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>$K_p$</td>
<td>diag([9,9,5])</td>
<td>diag([9,9,1])</td>
</tr>
<tr>
<td>$K_d$</td>
<td>diag([18,18,1])</td>
<td>diag([18,18,4])</td>
</tr>
<tr>
<td>$k_{py}$</td>
<td>1.2</td>
<td>1.2</td>
</tr>
</tbody>
</table>

body, as indicated in Figure 8.25.

Model Measurements  The calibrated shape models were then imported into MATLAB for estimating its surface area and other geometrical properties. The estimated model properties of all models are presented in Table 8.8. As seen here, the measured surface area of the model varied between 98.9 – 99.7%. The shape models of all experiments were decimated to have 50k triangular faces. The estimated surface area noted from the 5 experiments presented in Table 8.8 was found to be 0.651 ± 0.002 m² which accounts for 99.4 ± 0.31% of the baseline shape model. Additionally, as described above, the coverage of the nominal shape model of the target was estimated to be about 99.1% by the swarm designer module. Therefore, the mean difference between the estimation and measurement is 0.3%. The computed radii, along with the estimated surface area, are similar to the geometrical parameters of the baseline model noted in Table 8.3.

Scalability  While this chapter presented a hardware platform to demonstrate multi-spacecraft algorithms, its extension to spacecraft swarms ($N_{Sw} > 2$) needs to be discussed. The individual and cumulative drone coverages of the shape model are presented in Figure 8.32. In the identified solution, Drone 1 can map 70.1%, while
Drone 2 can map 86.1 % of the surface of the nominal model. Assuming a nominal coverage estimate of 99.18 % leaves an overlapping area of about 56.0 % covered by both drones. While this coverage is dominant in the northern hemisphere of the shape model, some portions of the southern hemisphere are not covered by either drone, as seen in Figures 8.26 and 8.32. This highlights some critical areas where adding more drones to the identified design can be beneficial.

Firstly, adding additional drones to the design can lead to a complete surface coverage of the target. In this case, the drone would be deployed at southern elevation angles $\theta_{y,i}$ with respect to the target. A second advantage of adding more drones can result in an increased overlapping area observed by multiple drones. This can be done by distributing the drones along the azimuthal $\theta_{x,i}$ or elevation $\theta_{y,i}$ directions. In this case, adding more drones can identify more common features in the SfM pipeline in Figure 8.18. This can result in high definition shape models obtained from the SfM operation. Additionally, adding more drones to the design can also lead to imaging the target at higher resolutions. This can be done by re-
Figure 8.28: Cartesian components of the UAV pose topics showing the performance of the designed reference tracking controller.

Reducing the imaging radius $r_{max}$ of the MAPS testbed. In this case, the drones fly by closer altitudes with respect to the target and consequently have smaller individual coverages. This improves the ground resolution of the videos recorded by the drones, and consequently, in high definition shape models obtained from the SfM algorithms. The algorithms developed in this chapter to design and implement the drone trajectories in the MAPS testbed can easily scale up to accommodate the additional drones into the swarm. However, under such designs, care must also be taken to program collision avoidance maneuvers into flight software architecture. The advantage of adding multiple drones can be summarized in Figure 8.33.

8.5 Discussion

This chapter described the development of the MAPS testbed, which demonstrated multi-spacecraft reconnaissance mission concepts around small bodies. The testbed simulates spacecraft behaviors using UAVs as autonomous agents. The UAVs perform photogrammetric 3D reconstruction of a target small body mock-up placed
Figure 8.29: Sample images generated from the video recordings by the UAVs.

Figure 8.30: The 3D surface reconstruction of the UAV flight path generated from a sample experiment.

along their flight path. The different components of the MAPS testbed were described here. These components were then used to define the multi-agent autonomous architectures for visual reconnaissance missions. Specifically, this chapter demonstrated global surface mapping missions that were implemented inside the MAPS testbed. The mission design problem was formulated as an optimization problem to search for the minimum number of UAVs that generate the required spatial coverage of the target body. We then developed reference trajectories and reference tracking control laws for a specified design gene. The Automated Swarm Designer module of IDEAS was used to identify a near-optimal mission concept that uses 2 drones as autonomous agents to map at least 95% of the small body. This identified mission concept was then implemented inside the MAPS testbed. The trajectories of the drones exhibited successful performance of the developed refer-
Figure 8.31: Scaled target body vertices extracted from the reconstructed flight paths of different mapping experiments.

Table 8.8: Shape model parameters measured from the extracted target body surface.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Experiment-1</th>
<th>Experiment-2</th>
<th>Experiment-3</th>
<th>Experiment-4</th>
<th>Experiment-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max radius (m)</td>
<td>0.333</td>
<td>0.32</td>
<td>0.322</td>
<td>0.367</td>
<td>0.322</td>
</tr>
<tr>
<td>Min radius (m)</td>
<td>0.139</td>
<td>0.141</td>
<td>0.162</td>
<td>0.127</td>
<td>0.158</td>
</tr>
<tr>
<td>Mean radius (m)</td>
<td>0.236</td>
<td>0.24</td>
<td>0.24</td>
<td>0.223</td>
<td>0.242</td>
</tr>
<tr>
<td>Computed surface area (m²)</td>
<td>0.648</td>
<td>0.651</td>
<td>0.650</td>
<td>0.653</td>
<td>0.652</td>
</tr>
<tr>
<td>Area mapped (%)</td>
<td>98.9</td>
<td>99.5</td>
<td>99.2</td>
<td>99.7</td>
<td>99.7</td>
</tr>
</tbody>
</table>

ence trajectories and tracking control laws. The video feed of the UAVs was then used to reconstruct the surface of the target body using an SfM software pipeline. The measurements from the surface reconstruction indicated that the UAVs were able to map about 98.9 % as estimated by the mission concept identified by IDEAS. In this way, the MAPS testbed supports the IDEAS framework in demonstrating multi-spacecraft reconnaissance algorithms. Therefore, this chapter augmented the state-of-the-art mission design research with powerful tools to demonstrate multi-spacecraft mission concepts to small bodies.
Figure 8.32: Individual and cumulative coverage estimated by the IDEAS architecture.

Figure 8.33: Effects of adding more drones to the selected design along the angular (top) and radial directions (bottom)
CHAPTER 9

Conclusion

In this chapter, the essence of this dissertation is summarized through an examination of critical discussion points highlighted by the individual chapters. The critical contributions of the current thesis to the existing state-of-the-art research on space technology are then identified. Specifically, the contributions to the fields of spacecraft swarm, small body reconnaissance, robotics, and mission concept design research are presented. Finally, the current chapter is concluded by identifying pathways forward for the IDEAS framework and the MAPS testbed.

9.1 Summary

The design of reconnaissance mission concepts using spacecraft swarms is a complex and unintuitive task. The challenges to design such mission concepts include large and mixed-integer design spaces. Furthermore, the mission concept design is complicated by dynamical constraints such as micro-gravity environment, surface illumination, and irregular surfaces of the target. Additionally, swarm mission concepts require hardware platforms to demonstrate critical aspects of the mission. This thesis addressed these challenges by proposing and developing an end-to-end mission concept design software known as IDEAS. The focus of the current work was to develop the Automated Swarm Designer module of IDEAS to design visual reconnaissance missions to small bodies that meet spatial and temporal coverage requirements. In addition to this, the current work developed the MAPS testbed to demonstrate spacecraft swarm behaviors identified by the IDEAS framework.

The IDEAS architecture and its approach to designing swarm mission concepts were presented in Chapter 2. IDEAS provides a unified design environment for seed spacecraft, swarms, and interplanetary trajectory design. In this chapter, a
review of the existing state-of-the-art spacecraft swarm technologies and the contribution of IDEAS to the existing literature was provided. This was followed by presenting a new classification scheme swarm architectures to account for a broad spectrum of interactions possible between different spacecraft. The roles of different modules of IDEAS in designing a swarm mission concept were examined, and the implementation of these modules in a series of different case studies was described.

Chapter 3 developed the models for visual reconnaissance mission concepts for Class 2 swarms to small bodies. Specifically, coverage evaluation algorithms that use linear camera transformations to evaluate the surface coverage of an irregular small body from a spacecraft carrying an optical sensor were developed. The Camera transformation formed the core of the Dual Sphere coverage method, where the coverage is evaluated over the nominal shape model of a target body, along with two spheres whose radii correspond to the maximum and minimum radius of the target body model. Additionally, algorithms to detect spacecraft collisions in a designed swarm architecture were presented. Chapter 4 described the algorithms to study the sensitivity of the spatial and temporal coverage performance of the swarms to different classes of perturbations. The design and dynamical perturbations on a given swarm architecture and their significance were described. The models used in the current work to introduce these perturbations into the selected swarm architectures were then presented.

Chapters 5-7 presented case studies of applications of the IDEAS architecture to design visual reconnaissance mission concepts to small bodies, which were demonstrated by numerical simulations. Chapter 5 described swarm architectures to planetary moons through hyperbolic flybys of the central planet, which were demonstrated by designing a global surface mapping mission to Phobos. Chapter 6 developed swarm architectures to explore planetary moons using resonant co-orbits. The co-orbits were configured for global surface mapping and RoI observation missions to planetary moons, which were demonstrated through designing mission concepts to explore Deimos. Chapter 7 described swarm architectures to explore tumbling asteroids. Here a mother-daughter architecture, where a mothership spacecraft deployed
the spacecraft swarm on its heliocentric trajectory, was developed. The design of this architecture was then demonstrated through a simulated global surface mapping mission to 4179 Toutatis.

Chapter 8 presented the development of the MAPS testbed, which is used to demonstrate the swarm behaviors generated by IDEAS. Here a detailed description of the MAPS testbed was provided, which placed performance constraints on photogrammetric surface reconstruction using flybys of autonomous UAVs. The constraints were then used to formulate the swarm design problem, which was solved using the Automated Swarm Designer module of IDEAS. The obtained solution was implemented inside the MAPS testbed using autonomous UAVs, and their visual recordings were used to reconstruct the surface of the target body. The surface reconstruction indicated that the UAVs were able to achieve the coverage performance estimated by the Automated Swarm Designer module of IDEAS.

9.2 Contributions

The current work contributed to the state-of-the-art space technology by providing end-to-end mission concept design architectures for spacecraft swarms. The architectures automated the generation and selection of reconnaissance mission concepts using Evolutionary Algorithms. Additionally, a hardware platform to demonstrate swarm reconnaissance algorithms was developed. The MAPS testbed supported the IDEAS framework by providing a hardware platform to demonstrate multi-agent reconnaissance mission concepts. These technologies are intended for both technical and non-technical users to design and demonstrate otherwise complicated and unintuitive mission concept design processes. To achieve this, the current work makes several technical contributions, which are listed as follows.

**Spacecraft Swarms**

The contributions of this thesis to the field of spacecraft swarms are listed as follows.

1. Developed swarm architectures for visual reconnaissance of small bodies, where
the spacecraft in the swarm are decentralized agents that coordinate for communication.

2. Designed optimal swarm architectures that synergistically improve reconnaissance performance by achieving the desired spatial and temporal coverage requirements with a minimum number of spacecraft.

3. Provided a novel classification for swarm architectures based on the level of interactions between different participating spacecraft. This classification presented a new class of swarms that has not yet been realized to date.

**Small Body Reconnaissance**

To advance the state-of-the-art in small body reconnaissance, the current work contributed as follows.

1. Developed and demonstrated numerical algorithms to design hyperbolic trajectories of a spacecraft swarm to encounter a planetary moon at a required orbital location.

2. Developed and demonstrated analytical algorithms to design resonant co-orbits to encounter a planetary moon at a required orbital location.

3. Developed and demonstrated numerical algorithms to design the interplanetary trajectory and swarms launched on a mother-daughter architecture for reconnaissance mission concepts to tumbling asteroids.

4. Developed the Dual Spheres method, which is a novel algorithm to stochastically evaluate the coverage performance of a small body by a designed swarm.

5. Developed a new coverage evaluation method, which uses the concept of camera transformations to estimate the surface coverage of irregularly shaped objects. This method was used to estimate the spatial and temporal surface coverage of small bodies in the designed mission concepts.
Robotics

The current work developed a novel hardware testbed to demonstrate swarm behaviors identified by IDEAS. The contribution of this work to state-of-the-art robotic demonstration platforms are as follows:

1. Demonstrated the operation of the MAPS testbed by implementing the near-optimal mission concept identified by IDEAS and reconstructing the surface of the target body from the visual feed of the autonomous UAVs.

2. Developed hardware and software architectures for a multi-UAV system that performs photogrammetric surface reconstruction of simulated small bodies.

3. Developed time-varying reference trajectories for the UAVs to track the line of sight to the target, based on the design gene identified by IDEAS.

4. Implemented reference tracking PD controllers using the ARSDK software libraries, that ensures that the UAVs track the reference trajectories based on their pose and odometry state feedback.

Mission Design Architectures

The contributions of this thesis to existing state-of-the-art space mission design research can be listed as follows:

1. Developed an end-to-end architecture that provided a unifying design environment for the swarm, trajectory, and the participating spacecraft.

2. Demonstrated the interaction between swarm, trajectory, and spacecraft designs in visual reconnaissance missions to small bodies.

3. Formulated trajectory and swarm design problems of small body reconnaissance mission concepts as MINLP optimization problems.

4. Automated the generation and selection of feasible solutions of swarm and trajectory design problems by using Evolutionary Algorithms.
5. Provided global figures of merit such as seed spacecraft and swarm mission cost estimates to inform mission designers regarding the feasibility of selected mission concepts.

9.3 Future Work

The primary focus of the current work was on presenting end-to-end automated solutions to design swarm mission concepts. While the current work presented the interaction between the different modules of IDEAS, it would be of great value to improve the flexibility of the seed spacecraft in the swarm. The spacecraft design is complicated by pitchfork bifurcations caused by in-house and off-the-shelf manufactured subsystems. The spacecraft design spaces when its subsystems are populated under these two manufacturing configurations would be strikingly different. Specifically, analytical formulations can be applied to the design of in-house manufactured subsystems. However, designs that employ off-the-shelf subsystems require a frequently updated knowledgebase of spacecraft subsystem inventory. Multi-disciplinary optimization (MDO) provides promising tools to handle such complex design spaces. Therefore, a future line of research can build a frequently updated knowledgebase of spacecraft behaviors to form the seed spacecraft design space, while using MDO strategies to generate the seed spacecraft design. On a similar note, the continually changing spacecraft technology results in a corresponding change in cost estimation strategies. Therefore, periodically updating the cost estimation algorithms results in more accurate mission concept designs using IDEAS.

The current work focused on the design of visual reconnaissance mission concepts to small bodies using Class 2 swarms. To improve the range of application, future work can explore the application of IDEAS to a variety of mission scenarios. Possible applications include the design of swarms that use multi-spectral sensing such as UV, infrared, and radio sensing. Additional work on IDEAS can be used to design multiple swarm classes in addition to a class 2 architecture. Such architectures can be used to design complex scenarios such as distributed sensing and modeling
various spacecraft life cycle tasks.

Another line of future work can focus on improving the dynamical fidelity of IDEAS. The current thesis developed swarm mission concepts using the two-body gravitational modules. While the motion of spacecraft was indeed dominantly influenced by two-body dynamics, the presence of dynamical perturbations was shown to decay in the coverage performance of swarms. Such decays can be eliminated by the placement of correction maneuvers when modeling spacecraft swarm trajectories. Therefore future work can explore the integration of correction maneuvers into IDEAS architecture when the spacecraft in the swarm are subjected to high fidelity dynamics. This augmentation can lead to accurate prediction of spacecraft fuel requirements, spacecraft sizing, and, consequently, global mission parameters such as cost and lifetime.

Finally, an additional line of future research can focus on improving the range of applications that can be demonstrated on the MAPS testbed. While the MAPS testbed was intended to demonstrate spacecraft swarm concepts to small bodies, critical dynamical differences exist between UAVs and interplanetary spacecraft. These differences include dynamical coupling between attitude and translation states of the agents and the dominant source of accelerations. In the current work, the MAPS testbed was used to demonstrate a photogrammetric surface reconstruction mission. However, future work on the MAPS testbed can demonstrate additional swarm mission concepts such as studying the coverage performance under simulated perturbations and collision avoidance maneuvers. These streams of research will enable IDEAS to generate holistically optimal swarm mission concepts, whose performance can be demonstrated using the MAPS testbed. Therefore, these streams of research provide end-to-end tools that automate the mission concept design processes and also provide hardware demonstration platforms, thus advancing the state-of-the-art by providing powerful tools to explore the solar system.
APPENDIX A

Subsystem Inventory and Cost Modeling

Spacecraft Inventory

This section presents the progress of the Spacecraft Designer portion of the IDEAS software. In reality, the design of a spacecraft swarm mission, and constituent spacecraft design are coupled with each other. For example, the flyby altitude of the swarm depends on the camera aperture diameter, as indicated by Equation 3.1. Additionally, the subsystems of the spacecraft populating the designed swarm must be able to meet the operational mission requirements, i.e., handle the required attitude behaviors, provide a feasible data transfer rate. This section provides the parameters required to define a single seed spacecraft in the swarm. Since the swarm is effectively realized on small spacecraft platforms, the subsystems of the spacecraft are populated, assuming off-the-shelf components where information is available. The spacecraft bus cost is estimated through the 2010 version of the Small Satellite Cost Model (SSCM10) cost model by Aerospace Corporation (Wertz et al., 2011), whose minimum spacecraft bus mass requirement is 20 kg. For this reason a 27U spacecraft is used which supports a maximum bus mass of 54 kg (Hevner et al., 2011). The following subsections provide the subsystems used to populate the sample 27U spacecraft.

Payload The payload of the seed spacecraft is assumed to be a camera of aperture diameter $D_c = 8$ cm. The camera of interest was demonstrated on the ASTERIA mission, which is a low Earth orbit CubeSat based telescope (Pong et al., 2010). The red spectrum ($\lambda = 0.7 \mu m$) is used as the baseline wavelength to identify the minimum flyby altitude requirement of the spacecraft (Wertz et al., 2011). The payload is assumed to weigh a total of 0.57 kg. The camera is assumed to consume
a power of 2 W and last for a design life of 4 yrs.

**Attitude Control System**  The attitude control system of the seed spacecraft is assumed to be consisting of the XACT-50 module from Blue Canyon Technologies (BCT). The XACT-50 is an integrated attitude determination and control system (ADCS) which uses a star tracker and an inertial measurement sensor suite for its attitude determination, and 3-axis reaction wheels for attitude control. The selected ADCS unit weighs a total mass of 1.23 kg.

**Communications**  The communications subsystem of the seed spacecraft is assumed to be the Iris radio manufactured by the Jet Propulsion Laboratory (JPL), which was demonstrated through the recent MarCO mission to Mars (Schoolcraft et al., 2016). The Iris radio serves as the telemetry, tracking, and command (TTC) module of spacecraft through the X-Band operational channel. The radio is built to be compatible to be tracked by the Deep Space Network (DSN) has been demonstrated to operate at a downlink data transfer rate of 8.4 GHz and an uplink data transfer rate of 7.2 GHz. The Iris radio weighs a total of 1.2 kg.

**Command and Data Handling**  The Command and Data Handling (CDH) subsystem of the seed spacecraft is assumed to be the CubeSat Processor (CSP) by Space Micro Inc. The CSP is a single radiation-hardened chip of dimensions $8.81 \times 8.95 \text{ cm}^2$. Going by the datasheet of the CSP subsystem, the CDH module weighs a total of 60 g.

**Power**  The seed spacecraft is assumed to have a CubeSat grade Lithium battery manufactured by Pumpkin Space Inc. The battery module weighs a total of 0.71 kg. Power to the spacecraft subsystems is distributed to the spacecraft subsystems through the electrical power supply (EPS) system from the same vendor, which weighs a total of 0.155 kg. The solar panels of the spacecraft are assumed to be the MMA Hawk solar panels, which for a 2 wing configuration providing up to
36 W of power will result in a mass of 0.558 kg. Therefore, the total power system is assumed to weigh a total of 1.42 kg.

**Structure** Since the seed spacecraft described here is assumed to belong to small spacecraft class (Wertz et al., 2011), a net supporting structural mass of 10 kg, and an additional 2 kg mass for thermal protection is assumed as a conservative estimate.

**Propulsion** The propulsion system of the spacecraft is assumed to be made of the MPS-120XL CubeSat propulsion system from Aerojet Rocketdyne Holdings Inc. This propulsion system has a dry mass of 2.4 kg. Thus by using the mass information of all the above systems accounts for a total spacecraft dry mass of \( m_{\text{dry}} = 20.6 \) kg. The fuel mass \( m_{\text{fuel}} \) required for a total maneuver cost \( \Delta v_{\text{net}} \) is estimated from the rocket equation (Vallado, 2013) as

\[
m_{\text{fuel}} = m_{\text{dry}} \left( \exp \left( \frac{\Delta v_{\text{net}}}{I_{\text{sp}g_0}} \right) - 1 \right) \tag{A.1}
\]

The mass of spacecraft at launch \( m_{\text{wet}} \) can then be determined as

\[
m_{\text{wet}} = m_{\text{dry}} + m_{\text{fuel}} \tag{A.2}
\]

The spacecraft fuel is assumed to Hydrazine which can provide a specific impulse of \( I_{\text{sp}} = 220 \) s. The resulting dry mass distribution of the spacecraft is summarized in Table A.1. The sum of fuel mass \( m_{\text{fuel}} \) and the dry mass of the propulsion system \( m_{\text{dry}} \) will be the wet mass of the propulsion system \( m_{\text{Prop,Wet}} \) which is used in predicting the spacecraft and launch costs.
Table A.1: Dry mass distribution of the seed spacecraft populated from an existing inventory of off-the-shelf subsystems.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mass (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload</td>
<td>0.57</td>
</tr>
<tr>
<td>ADCS</td>
<td>1.23</td>
</tr>
<tr>
<td>Communication</td>
<td>1.20</td>
</tr>
<tr>
<td>CDH</td>
<td>0.06</td>
</tr>
<tr>
<td>Power</td>
<td>1.42</td>
</tr>
<tr>
<td>Structure</td>
<td>10.0</td>
</tr>
<tr>
<td>Thermal</td>
<td>2.0</td>
</tr>
<tr>
<td>Propulsion dry mass</td>
<td>2.4</td>
</tr>
<tr>
<td><strong>Total dry mass</strong></td>
<td><strong>18.9</strong></td>
</tr>
</tbody>
</table>

**Cost Estimation**

This section will present the results of the spacecraft mission cost estimation, which is achieved by implementing the empirical cost models publicly available in the literature. These models provide estimated costs in the fiscal year 2010 US dollars FY10$, which is converted to the launch year dollars LY$ based on its inflation factor. A list of inflation factors (Wertz, 2001) for converting FY10$ to LY$ up to a launch year of 2025 is presented in Table A.2.

**Spacecraft Bus and Mission Cost** As mentioned in the previous section, the cost of the spacecraft bus, other mission costs are estimated using the 2010 version of the Small Satellite Cost Model (SSCM10) cost model by Aerospace Corporation (Wertz et al., 2011), whose minimum spacecraft bus mass requirement is 20 kg. The itemized cost estimating relations used in the SSCM10 model is presented in Table A.3. As seen here, the bus cost includes the cost of all spacecraft subsystems, excluding the payload. The standard errors in case of the cumulative costs (bus cost, and the total costs) are obtained by taking the root square sum (RSS) of their corresponding variables shown in Table A.3. Additionally, In case of a subsystem cost whose standard error model was not specified, this was estimated through a user defined multiplicative uncertainty factor over the corresponding subsystem cost.
Table A.2: Multiplicative inflation factors used in the current work.

<table>
<thead>
<tr>
<th>Year</th>
<th>2010 Inflation Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>1.0000</td>
</tr>
<tr>
<td>2011</td>
<td>1.0154</td>
</tr>
<tr>
<td>2012</td>
<td>1.0308</td>
</tr>
<tr>
<td>2013</td>
<td>1.0515</td>
</tr>
<tr>
<td>2014</td>
<td>1.0734</td>
</tr>
<tr>
<td>2015</td>
<td>1.0958</td>
</tr>
<tr>
<td>2016</td>
<td>1.1192</td>
</tr>
<tr>
<td>2017</td>
<td>1.1431</td>
</tr>
<tr>
<td>2018</td>
<td>1.1675</td>
</tr>
<tr>
<td>2019</td>
<td>1.1925</td>
</tr>
<tr>
<td>2020</td>
<td>1.2179</td>
</tr>
<tr>
<td>2021</td>
<td>1.2439</td>
</tr>
<tr>
<td>2022</td>
<td>1.2704</td>
</tr>
<tr>
<td>2023</td>
<td>1.2976</td>
</tr>
<tr>
<td>2024</td>
<td>1.3252</td>
</tr>
<tr>
<td>2025</td>
<td>1.3536</td>
</tr>
</tbody>
</table>

**Payload Cost** The cost of the spacecraft payload is estimated through the NASA Instrument Cost Model (NICM), assuming that the payload is a planetary imaging camera. The manufacturing cost and the associated wrap cost estimating relations involved in the NICM model are presented in Table A.4.

**Launch Cost** The launch cost is estimated using a database of launch providers in the United States of America (USA) and their GTO capabilities (Wertz et al., 2011) as a baseline. The database of launch providers and their cost per unit mass for the year 2010 are presented in Table A.5.

**Interplanetary Cost Factor** Since the SSCM and the Launch costs were estimated primarily for Earth-based spacecraft, a multiplicative interplanetary factor of 1.29, as indicated by the QuickCost model (Wertz et al., 2011), is used to convert an Earth-based mission costs to interplanetary mission cost. The payload cost estimated through the NICM, on the other hand, is estimated for an interplanetary
Table A.3: Cost estimating relations predicted by the Small Satellite Cost Model.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subsystem</th>
<th>Cost Estimate</th>
<th>Standard Error</th>
<th>Independent Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Structure</td>
<td>0.407 + 0.02x log x</td>
<td>1.1 x 10^{-3}</td>
<td>x- Structural mass (kg)</td>
</tr>
<tr>
<td>2</td>
<td>Thermal</td>
<td>0.335 + 0.006x^2</td>
<td>1.2 x 10^{-4}</td>
<td>x- Thermal mass (kg)</td>
</tr>
<tr>
<td>3</td>
<td>Attitude Determination and Control (ADCS)</td>
<td>1.85 + 0.012x^2</td>
<td>1.1 x 10^{-3}</td>
<td>x- ADCS mass (kg)</td>
</tr>
<tr>
<td>4</td>
<td>Power</td>
<td>1.26 + 0.54x^{0.72}</td>
<td>9.1 x 10^{-4}</td>
<td>x- Power mass (kg)</td>
</tr>
<tr>
<td>5</td>
<td>Propulsion</td>
<td>0.089 + 0.003x^{1.26}</td>
<td>3.1 x 10^{-4}</td>
<td>x- Power wet mass (kg)</td>
</tr>
<tr>
<td>6</td>
<td>Telemetry, Tracking, and Control (TTC)</td>
<td>0.486 + 0.055x^{1.35}</td>
<td>6.3 x 10^{-4}</td>
<td>x- TTC mass (kg)</td>
</tr>
<tr>
<td>7</td>
<td>Command and Data Handling (CDH)</td>
<td>0.658 + 0.075x^{1.35}</td>
<td>8.5 x 10^{-4}</td>
<td>x- CDH mass (kg)</td>
</tr>
<tr>
<td>8</td>
<td>Bus Cost</td>
<td>Sum of rows 1-6</td>
<td>RSS of rows 1-6</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Integration and Testing</td>
<td>x x 10^{-4}</td>
<td>xy x 10^{-4}</td>
<td>x- Bus Cost y- Estimation uncertainty (%)</td>
</tr>
<tr>
<td>10</td>
<td>Program Level</td>
<td>2.3x x 10^{-4}</td>
<td>2.3xy x 10^{-4}</td>
<td>x- Bus Cost y- Estimation uncertainty (%)</td>
</tr>
<tr>
<td>11</td>
<td>Launch and Orbital Operations Support (LOOS)</td>
<td>6.1x x 10^{-5}</td>
<td>6.1xy x 10^{-5}</td>
<td>x- Bus Cost y- Estimation uncertainty (%)</td>
</tr>
<tr>
<td>12</td>
<td>Ground Support</td>
<td>6.6x x 10^{-5}</td>
<td>6.6xy x 10^{-5}</td>
<td>x- Bus Cost y- Estimation uncertainty (%)</td>
</tr>
<tr>
<td>13</td>
<td>Total Cost</td>
<td>Sum of rows 8-13</td>
<td>RMS of rows 8-13</td>
<td></td>
</tr>
</tbody>
</table>

payload and so the interplanetary cost factor is not multiplied.

**Swarm Cost Factor** The mission cost per spacecraft is expected to decrease when realized using identical spacecraft (Wertz et al., 2011). However, no such trend is discussed for launch costs. Therefore, for an \( N_{Sw} \) spacecraft swarm, the total mission cost \( TCS \) is estimated as

\[
TCS = TM_1 \times \frac{1 + \frac{\ln S}{\ln 2}}{N_{Sw}} + (N_{Sw} \times LC_1) \tag{A.3}
\]

where \( TM_1 \) and \( LC_1 \) are the mission cost and launch costs corresponding to a single spacecraft, and \( S \) is a learning curve factor. A conservative value of \( S = 0.95 \)
Table A.4: Cost estimating relations used in the NICM Model.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Component</th>
<th>Cost Breakdown</th>
<th>Standard Error (%)</th>
<th>Independent Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sensor</td>
<td>$0.328 , m^{0.426} , p^{0.414} , d^{0.375}$</td>
<td>39</td>
<td>$m$- Camera mass (kg) $p$ - Camera power (w) $d$- design life (months)</td>
</tr>
<tr>
<td>2</td>
<td>Management</td>
<td>$7.1 \times 10^{-5}$</td>
<td>$y$</td>
<td>$x$- Sensor cost $y$- Estimation uncertainty (%)</td>
</tr>
<tr>
<td>3</td>
<td>Systems Engineering</td>
<td>$4.93 \times 10^{-4}$</td>
<td>$y$</td>
<td>$x$- Sensor cost $y$- Estimation uncertainty (%)</td>
</tr>
<tr>
<td>4</td>
<td>Product Assurance</td>
<td>$1.43 \times 10^{-4}$</td>
<td>$y$</td>
<td>$x$- Sensor cost $y$- Estimation uncertainty (%)</td>
</tr>
<tr>
<td>5</td>
<td>Integration and Testing</td>
<td>$1.45 \times 10^{-4}$</td>
<td>$y$</td>
<td>$x$- Sensor cost $y$- Estimation uncertainty (%)</td>
</tr>
<tr>
<td>6</td>
<td>Total Cost</td>
<td>Sum of rows (1-5)</td>
<td>RSS of rows (1-5)</td>
<td></td>
</tr>
</tbody>
</table>

Table A.5: Launch vehicles used in the current work, and their corresponding launch cost per unit mass.

<table>
<thead>
<tr>
<th>Launch Vehicle</th>
<th>Maximum GTO Capacity (kg)</th>
<th>GTO Launch cost per mass (FY10Sk/kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minotaur Taurus</td>
<td>448</td>
<td>55.7</td>
</tr>
<tr>
<td>Falcon 9</td>
<td>4540</td>
<td>12.5</td>
</tr>
<tr>
<td>Atlas V</td>
<td>8200</td>
<td>25.3</td>
</tr>
<tr>
<td>Delta 4 Heavy</td>
<td>13130</td>
<td>16.4</td>
</tr>
<tr>
<td>Space Shuttle</td>
<td>5900</td>
<td>69.2</td>
</tr>
</tbody>
</table>

is used, which can be changed by the user in the spacecraft definition block. The standard error for the swarm mission concept is estimated as:

\[
SE_S = \sqrt{N_{Sw} \left( SE_{M1}^2 + SE_{L1}^2 \right)}. \tag{A.4}
\]

Where, $SE_{M1}$ and $SE_{L1}$ are the standard errors corresponding to the mission segment and the launch segment, respectively, assuming a single spacecraft.
APPENDIX B

Leader Selection Algorithm

This section presents the real time leader selection algorithm used in the current work to decide the communications spacecraft. The leader selection scheme is used to avoid practical bottlenecks of tracking multiple interplanetary spacecraft from a ground station. Furthermore, implanting an algorithm that operates in realtime can make the swarm robust to single-point failures. In the current work, the leader selection is performed by computing a cost function $J_{L,i}$, which is the weighted sum of distances to individual spacecraft, and distance to Earth. Let $^{N}{\vec{R}}_{Ei}$ denote the position vector of spacecraft $i$ with respect to Earth, and $^{N}{\vec{R}}_{ji}$ denote its position vector with respect to spacecraft $j$, at a sampled instant. The leader selection cost is given by

$$J_{L,i} = \alpha_1 \left( \sum_{j=1}^{N_{sw}} |^{N}{\vec{R}}_{ji}| \right) + \alpha_2 \left|^{N}{\vec{R}}_{Ei}\right|$$  \hspace{1cm} (B.1)

Where $N_{sw}$ is the number of spacecraft in the swarm, and $\alpha_1$ and $\alpha_2$ are user-specified weights. When multiple spacecraft have the same cost, a leader is selected randomly amongst all the spacecraft, which have the least value of $j_{L,i}$. In the current work, the leader cost is only evaluated at the start of the simulation, i.e., at $t = 0$. The leader selection algorithm is summarized in the process diagram presented in Figure B.1.
Figure B.1: Process diagram describing the algorithm to select the leader spacecraft in the swarm.
REFERENCES


Oberto, R. (2002). Mars sample return, a concept point design by Team-X (JPL’s advanced project design team). In Proceedings, IEEE Aerospace Conference, volume 2, pp. 2–2. IEEE.


