UC San Diego
UC San Diego Electronic Theses and Dissertations

Title
Large-scale Multidisciplinary Optimization of CubeSat Swarms

Permalink
https://escholarship.org/uc/item/4t96c4c8

Author
Yang, Aobo

Publication Date
2020

Supplemental Material
https://escholarship.org/uc/item/4t96c4c8#supplemental

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA SAN DIEGO

Large-scale Multidisciplinary Optimization of CubeSat Swarms

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Engineering Science (Mechanical Engineering)

by

Aobo Yang

Committee in charge:

Professor John T. Hwang, Chair
Professor Boris Kramer
Professor Aaron J. Rosengren

2020

Copyright
Aobo Yang, 2020
All rights reserved.

The thesis of Aobo Yang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2020

EPIGRAPH

We will freely glide through the darkness to the unknown. Joy is by my side.

—Joyside

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my deep thanks to my advisor Prof. John Hwang for his guidance and teaching during these past two years. I am very grateful for his patience and encouragement when I encountered a bottleneck in this project.

I would like to express my gratitude to the other members of my thesis committee. Prof. Boris Kramer and Prof. Aaron Rosengren, for attending my thesis defense and giving me feedback.

Next, I would like to thank my project partner Victor Gandarillas, our cooperation helped us work efficiently on the project. Thanks also to the other team members in LSDO (Large-scale Design Optimization) Lab, especially Ning Li and Jiayao Yan, for their sive advice and my friends Yuzhe Qin, Cheng Qian, and Juefan Xie for discussing supporting technical details with me.

Last but not least, I would like to dedicate the thesis to my parents. Thanks for their unending love, which supports me in joyfully exploring the unknown.

Some of the methods presented in the thesis have been submitted for publication in Victor E. Gandarillas, Aobo Yang, Hyeongjun Park and John T. Hwang. "Large-Scale Multidisciplinary Design Optimization of CubeSat Swarms with Application to a Virtual Telescope Mission" SciTech 2021 - AIAA Science and Technology Forum and Exposition. This thesis is currently being prepared for submission for publication of the material. The thesis author is one of the primary investigators and authors of this material.

VITA

2013-2017	B. S. in Flying Vehicle and Power Engineering, Beihang University
2018-2020	M. S. in Mechanical and Aerospace Engineering, University of California San Diego

ABSTRACT OF THE THESIS

Large-scale Multidisciplinary Optimization of CubeSat Swarms

by

Aobo Yang

Master of Science in Engineering Science (Mechanical Engineering)

University of California San Diego, 2020

Professor John T. Hwang, Chair

Multidisciplinary design optimization is playing an increasingly important role in the design of engineering systems. One example is the design of a CubeSat. Multidisciplinary optimization provides a way to evaluate complex tradeoffs involving tight power and mass budgets. However, existing methods are not able to consider swarms of CubeSats, which are becoming increasingly common. This thesis presents a new multidisciplinary optimization and modeling method for CubeSat swarms operation, including the multiple disciplines of orbital mechanics, attitude control, propulsion, and communication. The approach efficiently handles thousands of variables and successfully achieves high-fidelity optimization with respect to strict CubeSats alignment and separation constraints, and limited propellant and power. In terms of

optimization results, the CubeSat swarm increases the total data download of chief communication spacecraft by 52.9 % compared to the original design, yielding improvements of around 15 % improvements in the delta-v and succeeding in controlling the alignment of three CubeSats within a 300 mm threshold during scientific observation phase.

Chapter 1

Introduction

Small satellites are useful for various purposes, including communication, navigation, space exploration, and scientific research. Compared to large satellites, small satellites are more effective at reducing the high economic cost of launch vehicles and the costs associated with construction. In recent years, researchers have shown increasing interest in small satellites and put forward many new concepts [2]. As one such miniaturized satellite, CubeSats are commonly launched as secondary payloads and successfully deployed in orbit. Through technological innovation, CubeSats enable a wide range of activities in space and serve multiple purposes [3].

Multiple studies in the literature have applied modeling and optimization to satellite design and operation. Sun et al. [4] created an integrated system for design, analysis, system simulation, and evaluation of the small satellite. Wu et al. [5] solved the multidisciplinary design optimization problem with a gradient-based approach [6]. However, the number of variables that can be handled by Wu's algorithm is minimal. Hwang et al. [1] applied the gradient-based multidisciplinary optimization to small-satellite design and operation. They created a new modeling framework and optimization of the multidisciplinary problem based on the adjoint method. The algorithm is applied in CubeSat investigating atmospheric density response to extreme driving (CADRE) [7], which studies the response of the outermost layer of Earth's atmosphere to auroras caused by solar wind [8]. The CADRE optimization algorithm modeled multiple disciplines, including orbit dynamics, attitude dynamics, cell illumination, temperature, solar power, energy storage, and communication. Compared with previous attempts at multidisciplinary optimization, the number of design parameters involved in the multidisciplinary optimization of CADRE's model increased significantly. However, the model still has some shortcomings. First, CADRE models single-satellite operation and is deficient expressing the relationship between multiple moving satellites, so the design platform cannot be applied to solve the CubeSats swarm operation optimization problem. Second, the relative motion relationship between the satellites needs to be represented. We must produce a more comprehensive model to characterize satellite attitude. Third, as the task becomes more complex, the number of objective functions to be optimized increases, as do the optimization constraints. A completely physics-based simulation of small satellite swarms is required.

The Virtual Super-resolution Optics with Reconfigurable Swarms (VISORS) mission conducted by NASA [9] is intended to reveal individual energy-release sites in the solar corona to test fundamental theories of coronal heating like why solar corona is much hotter than the visible surface. The goal of VISORS project is to design a baseline mission involving a virtual telescope comprising three 3U CubeSats, including sunshade spacecraft (SSC), optics spacecraft (OSC), and detector spacecraft (DSC) [10]. According to the design requirements of the VISORS project, CubeSats need to send observation data back to the ground stations run by team members. Maximizing data download is the top priority. Then, taking into account multiple observation phases and ensuring effective operation, we need to satisfy the power budget. Finally, we need to perform attitude control of the satellite swarms under strict constraints. Based on existing work, the thesis seeks to model a complete physical-based CubeSats swarm operation that can efficiently handle thousands of design variables simultaneously. At the methodology level, the thesis applies gradient-based optimization with adjoint-based derivative computation to the VISORS model and provides a mathematical framework for discipline modeling of CubeSat swarms operation.

The thesis proceeds as follows. Chapter 2 describes the VISORS mission, introducing the VISORS scientific objectives, design of CubeSats swarm, and satellite mission phases; Chapter 3 describes the approaches taken to solve the large-scale multidisciplinary optimization problem; Chapter 4 introduces all the disciplines used in the model, discussing the relationship between disciplines and the mathematical expression of the physical model; Chapter 5 introduces the formation of the optimization problem, mainly how the multidisciplinary design optimization (MDO) method handles multiple disciplines and thousands of variables. Chapter 6-7 present the optimization results and suggests directions for future work respectively.

Chapter 2

Mission Overview

2.1 Mission concept

The VISORS project is intended to study the solar corona in space plasma physics and to test the fundamental theories of coronal heating. [11] [12] Though traditional observation by soft x-ray and extreme ultraviolet (EUV) imagers using the conventional system [13] has provided some clues about the heating nature of coronal heating, the heated regions systems still remain unobserved due to the low resistivity of coronal plasma. The VISORS project will equip a swarm of multiple photon sieve telescopes to observe the solar corona. [14] CubeSat is used as the spacecraft to carry the photon sieve telescopes and the spacecraft will collect nearly diffraction-limited EUV images of coronal heating.

Each VISORS mission unit comprises three 3U CubeSats that form a distributed telescope, with a total volume of $10\,\text{cm} \times 10\,\text{cm} \times 30\,\text{cm}$. The formation flight of VISORS includes three spacecrafts, optics, sun-shade, and detector to form a Sun-pointing distributed telescope. The components of the VISORS mission are listed below:

- Optics spacecraft (OSC) includes a photon sieve to produce an image.
- Detector spacecraft (DSC) includes a detector system to record the image.

- Sunshade spacecraft (SSC) prevents solar radiation from outside the telescope's field of view (FOV) from reaching the detector and dominating the in-FOV signal.

- Ground station (GS) used for data communication, command receiving and control.

Each CubeSat can rotate in two dimensions, roll and pitch in flight. The axes are designated as transverse and longitudinal. The spacecraft is designed to operate in low Earth orbit (LEO) shown in the Figure 2.1, which is near-circular. The detector photon sieve in the spacecraft swarm starts to collect pictures while the three CubeSats are aligned with each other and the alignment vector in the direction of sun light. The alignment constraints will be further explained in the section of Discipline Models.

2.2 Mission phases

In this section, we introduce the major operational phases conceptualized for the VISORS mission. The mission phases include the pointing of the spacecraft, the alignment and separation of spacecraft, and data communication. In our physical modeling, some phases are simplified; some are more detailed to create a better optimization explanation for CubeSats swarm models.

- Initial Deployment: The CubeSats swarm with launch vehicle will be initially deployed into the target orbit by along-track separation. For initial deployment, the spacecrafts should be placed into nearly the same target orbit. Also, separation along the orbit track is needed to avoid the risk of recontact between spacecrafts.

- Standby Formation Assembly: After solar panel deployments and initial spacecraft check-out, each spacecraft will assume a standby non-science formation according to ground-commanded impulsive maneuvers and differential drag.

- Routine Non-Science Standby Operations: While satellites are orbiting in standby formation, routine operation is passively safe (any safety mechanisms the engagement of which requires little power or human control). Collision risk may be negligible without maneuvers. During routine spacecraft operations, CubeSats start to follow data communication command and downlink operations from ground stations, charge the battery and manage the system.

- Scientific Operations: The science operation is implemented by issuing a command to enter the science formation. The alignment positioning and attitude requirements will be explained more detailed in the discipline models chapter. Each formation alignment occurs once per orbit, and the three spacecraft should point in a line toward the Sun. Science images are recorded to onboard memory for later data download, and the satellite returns to the passively safe standby formation for non-science operations and to prepare for data communication following the science operations.

- Contingency Operations: If any spacecraft detects an out-of-limit safety condition, the formation will autonomously return to the passively safe non-science standby formation and await further instructions. If the spacecraft are unable to achieve the standby formation, they will autonomously maneuver to a safe formation.

- End-of-Mission Operations: The scientific objectives of the VISORS mission will be met with the post-processing of a single solar image demonstrating the same resolution as the virtual telescope. Typically, the criterion for success can be met as soon as one month after launch. For higher scientific returns, the mission concept makes multiple attempts to obtain the desired solar image and obtain additional images. The nominal operation period of the VISORS mission is expected to last approximately six months maximum and three months minimum after launch. In conclusion, each spacecraft will be placed into a passively safe orbit and they may continue to be operated individually to conduct additional technology demonstrations. They may be manually de-orbited using any remaining unused propellant.

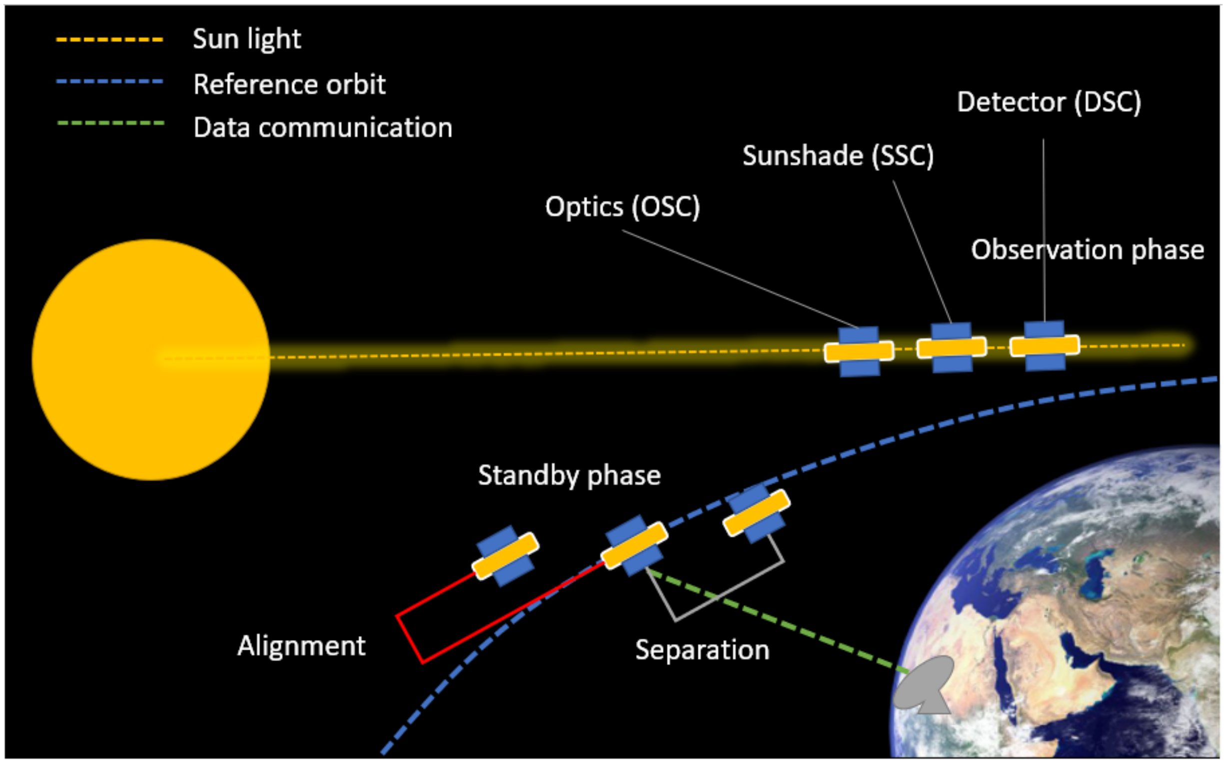


Figure 2.1: VISORS mission phases and 3U CubeSats formation flying.

2.3 CubeSat flight system

As discussed before, the VISORS flight system comprises three 3U CubeSats that form a distributed telescope. Each spacecraft in the swarm is similarly composed of three sections: a bus section (BUS), instrumental section (INST), propulsion, and inter-satellite crosslink section (PXLINK). We do not consider inter-satellite crosslinks for now, and some parameters of the spacecrafts and ground stations are used for physical modeling. The VISORS project intends to use a spacecraft bus designed by Blue Canyon Technologies, who have designed several commercial spacecraft buses, XB1, XB3, XB6, and XB12. As some technical details are not crucial to our optimization model, we do not expand on the commercial data. VISORS intends to use the 3D-printed cold-gas propulsion thruster developed by Georgia Tech, one of the team members for the propulsion system.

2.4 Ground station

The VISORS project intends to use four ground stations operated by team members University of California San Diego (UCSD), Montana St., University of Illinois at Urbana-Champaign (UIUC), and Georgia Tech. Instrument data will be collected in science mode and stored in the onboard memory until downlink. We should emphasize that the observation line of sight (LOS) is independent from the LOS, which means the time spent by the satellite on observation (data collection) is different from the time spent on data communication between the satellite and ground stations. The data accumulation budget is 20 MB/day. The predicted contact time is 35 minutes per day; the entire 20 MB/day data accumulation can be downlinked at the spacecraft transmission rate of 19.2 kbps.

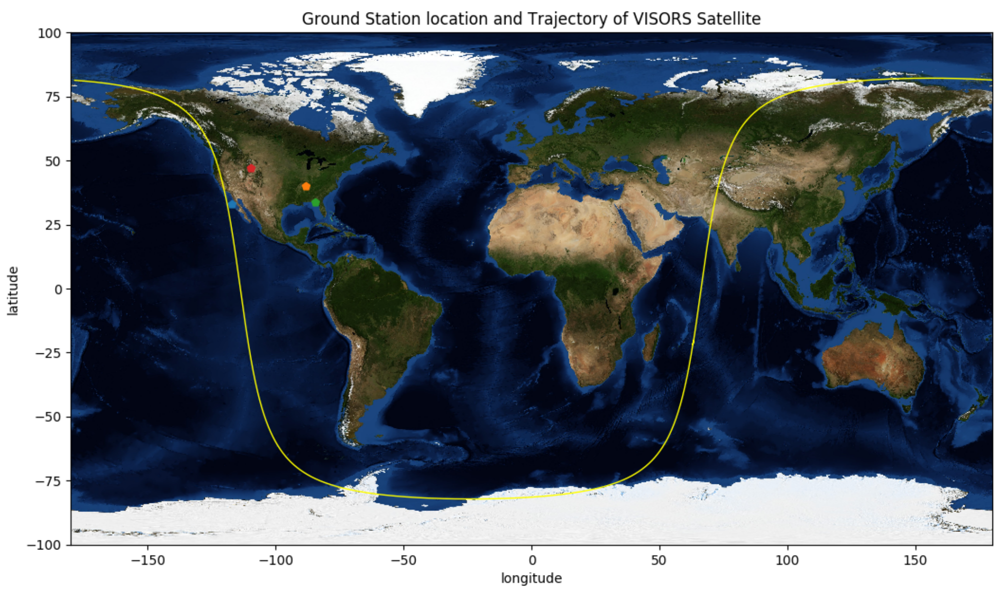


Figure 2.2: Satellite trajectory and ground station locations.

Chapter 3

Methodology

The main advantage of MDO is that it can handle multiple disciplines and thousands of design variables or more. In the VISORS project, around 3000 design variables need to be handled, so the robustness of the model is essential. The MDO approaches provides an efficient method based on the OpenMDAO platform, which can efficiently manage the disciplines relationship and data flow.

3.1 Adjoint method

An important requirement of large-scale optimization is accurate and efficient derivative computation. The accuracy of derivation computation affects the optimization time and the level of convergence possible. The efficiency of a derivative computation method is determined by how fast and or slow it is, which depends on the number of design variables. The accuracy of derivation computation requires accurate sparse matrix operations while we set each component. Hwang et al. created a general adjoint-based method combined with unifying derivatives equation (UDE) [15] for derivative computation. Here, we will review this method, which is important for solving our gradient-based optimization problem. Let us first set the variables and functions as follows, suppose we have input x, states y, and output f, R is the residual, and we can represent the model as a nonlinear equation $R(u) = 0$, where we are evaluating $x = x^*$. Given $r = R(u)$, we have an inverse function.

$$u = R^{-1}(r) \tag{3.1}$$

$$\frac{\partial R}{\partial u}^{-1} = \frac{\partial R^{-1}}{\partial r} \tag{3.2}$$

As it turns out, we get the equation on the left below, and by the definition of the Jacobian matrix inverse, we obtain the equation on the right below.

$$\frac{du}{dr} = \frac{\partial R^{-1}}{\partial r} \qquad\qquad \frac{\partial R}{\partial u} \cdot \frac{\partial R}{\partial u}^{-1} = I \tag{3.3}$$

Then by Eq. 3.2, 3.3, we substitute $\frac{\partial R}{\partial u}^{-1}$ with $\frac{du}{dr}$, and get the unifying derivative equation (UDE) [15].

$$\frac{\partial R}{\partial u} \cdot \frac{du}{dr} = I = \frac{\partial R}{\partial u}^{T} \cdot \frac{du}{dr}^{T} \tag{3.4}$$

The two different derivative computation methods represent two modes of algorithmic differentiation: forward mode and reverse mode. Based on the number of design variables, these two derivative computation methods lead to two different Jacobian matrix dimensions. The size of the Jacobian matrix directly affects the computation cost of the system, which then affects the complexity. Next, we introduce how these two modes drive the direct method and adjoint method. We will also discuss the influence of the number of design variables on the pros and cons of the methods. [16] Using the chain rule, we need to represent the derivative of f with respect to x as shown in Eq. 3.5.

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\partial F}{\partial x} - \frac{\partial F}{\partial y}\left[\frac{\partial R}{\partial y}\right]^{-1}\frac{\partial R}{\partial x} \tag{3.5}$$

Figure 3.1 shows two different ways of calculating $\frac{\mathrm{d}f}{\mathrm{d}x}$, $\frac{\mathrm{d}y}{\mathrm{d}x}$, and $\frac{df}{dr}$ with the same settings as before. x is the design variable, y is the state, and f is the output variable. When the number of design variables is much larger than the number of output variables, the derivation computation

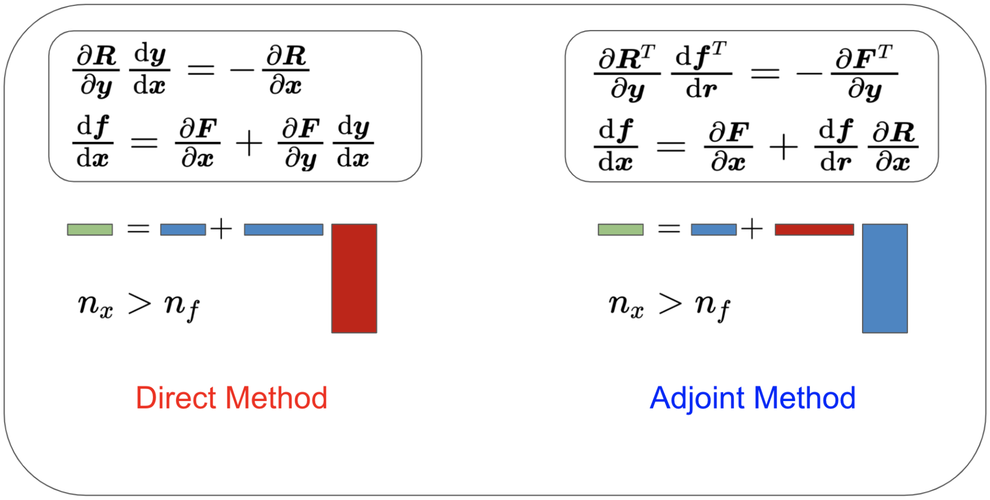


Figure 3.1: Derivation of adjoint and direct methods

of the direct method costs much more than the adjoint method. The red rectangle represents the size of the Jacobian matrix. Since this project involves large-scale design variables, the number of design variables is much larger than that of output variables. In the process of algorithm implementation, we provide OpenMDAO with the clear derivative computation of each component. OpenMDAO will select which method to use according to the number of design variables. The adjoint method made certain achievements in the aerospace field; for instance, Reuther and Jameson have successfully made an aerodynamic shape optimization [17] using adjoint-based methods. [18]

3.2 Gradient-based optimization

The emergence of the adjoint method makes gradient-based optimization a powerful tool to handle large-scale design variables. The gradient method, as an algorithm of optimization, uses as search directions the gradient of the function at the current point. [19] In this thesis, we choose to use Sparse Nonlinear Optimizer (SNOPT) as an optimizer through the pyOpt interface. [20] SNOPT is a software package for solving large-scale nonlinear optimization problems, developed by Philip Gill, Walter Murray, and Michael Saunders [21]. SNOPT is designed mainly for constrained optimization, which minimizes a linear or nonlinear function subject to bounds on the variables and sparse linear or nonlinear constraints. The advantage of SNOPT is its ability to solve large-scale linear and quadratic programming, linear constrained problems and nonlinear programs. With optimization computation, SNOPT finds the locally optimal solutions. It requires users to provide gradients to optimizers, which correspond to the components, derivatives that we wrote during the implementation of the CubeSat operation model. SNOPT employs sparse sequential quadratic programming (SQP) with limited-memory quasi-Newton approximations of the Hessian of the Lagrangian. The nonlinear functions should ideally be smooth but need not be convex. The discontinuity of the nonlinear function gradients can be tolerated in the gradient provided if they are too close to an optimum, and the local optima can be seen as a global solution. The augmented Lagrangian merit function is reduced along each path of the search, which promises convergence from any point of departure. SNOPT is particularly efficient at solving nonlinear problems with equations on a large scale and when gradients are costly. With several disciplines and thousands of design variables to be considered in this project, SNOPT is a good option for solving this large-scale nonlinear problem.

3.3 Multidisciplinary optimization

Multidisciplinary optimization focuses on using numerical modeling for device design spanning a variety of disciplines or subsystems. The performance of a multidisciplinary system is driven by the discipline performance and interdependence. In implementing the MDO in the VISORS project, one of the greatest challenges is the mathematical modeling of each discipline for satellite operation. Each discipline should be decomposed into a set of basic computation components along with the derivative computation. Component implementation is based on the latest version of OpenMDAO 3.0. The OpenMDAO platform helps us divide the problem of optimization into several stages and forms a nonlinear system of problems composed of multiple multidisciplinary tasks. The architecture, state, intermediate, input, and output variables from each basic component are a subset of the unknowns in the nonlinear system, subject to corresponding constraints. OpenMDAO and our modeling ensure the independence of input variables. Also, the goals and constraints for optimization are either explicit or implicit (usually explicit in our model). To clarify how the MDO is implemented in the VISORS CubeSats design operation problem, we will introduce each basic component unit of the optimization problem.

The design variable in the MDO problem is a collection of independent variables under the explicit control of an optimizer, based on our target mission. Some of the design variables might be local and some were shared by multiple disciplines. Usually in an MDO problem, the number of design variables is in the thousands, so we hereby denote the vector of design variables local to discipline i by x_i and the shared variables by x_0.

The model of the disciplines is one of the basic components of formulating of optimization work. To simplify and modularize the code, each discipline is decomposed into multiple stages of computations. For example, the output of the communication discipline is total data download in one of the VISORS project disciplines. We separate the discipline into several computations, including the location vector from the satellite to the ground station, the ground station contact LOS variable, the data download rate, the KS function for the best data download rate selection and the total data download.

By implementing all these components, which includes defining the dependencies between variables and programming their derivatives and components composition. We formulate objective optimization work within a framework that includes the function of multidisciplinary optimization and analysis formulation based on the OpenMDAO platform. The state, design, intermediate, input and output variables, objective, constraints and residuals are defined and formulated in nonlinear systems. The OpenMDAO platform helps us simplify the task of connecting variables and combining disciplines. In OpenMDAO, the properties of each independent variable need to be clarified, the value of variable should be set, if the variables are explicit or implicit, and the value of which is the root of the equation. In this project, the design variables are part of inputs variables, the objectives and constraints are also explicit.

3.4 Kreisselmeier-Steinhauser (KS) function

The KS function, known as KS function, is a widely used constraint aggregation method for gradient-based optimization, which was first presented by G. Kreisselmeier and R. Steinhauser. The function contains an aggregation parameter ρ. It is similar to the penalty factor used in the penalty method used to perform constrained optimization. Eq. 3.6 is an example one KS function [22]:

$$KS\left(g_j(x)\right) = g_{\max}(x) + \frac{1}{\rho}\ln\left[\sum_{j}^{n_g} e^{\rho\left(g_j(x) - g_{\max}(x)\right)}\right] \tag{3.6}$$

where

$g_{\max}(x)$	the maximum of all constraints evaluated at the current design point x
$g_j(x)$	the jth of all constraints evaluated at the current design point x
ρ	determines the difference between the KS function and the maximum value of the constraint
n_g	the number of all constraints evaluation

As ρ approaches infinity, the KS function approaches the value of $g_{\max}$. Also, from the Eq. 4.1 , we can see that the value of the KS function is bounded with these two representations:

$$g_{\max}(x) < KS < g_{\max}(x) + \frac{\ln n_g}{\rho} \tag{3.7}$$

We can calculate the maximum error for a particular ρ value. When ρ is large enough, the machine zero could be achieved. The method may also cause numerical difficulties when ρ

becomes too large. From J R.R.A.Martins and Nicholas M.K.Poon's discussion, [23] $\rho = 50$ is usually a reasonable value with a maximum relative error of 0.03 for constraints. In the VISORS project, another use of the KS function is about the selection of the data download rate. As we want to select the optimal or highest data download rate while the satellite communicates constantly with the ground stations in, the KS function is used to compare the data download rates for the satellites with each ground station. In this way, we ensure the most effective and highest data download for each orbit cycle.

3.5 B-Spline Interpolation

In numerical analysis, B-spline is a spline function with minimal support concerning a given degree, smoothness, and domain partition. Given $n+1$ control points $P_0, P_1, ..., P_n$ and a knot vector $U = \{u_0, u_1, ..., u_m\}$, the B-spline curve of degree p defined by these control points and knot vector U is

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \tag{3.8}$$

In this thesis, the B-spline interpolation is used in design variables. Given 300 control points of specific design variable, we can generate 1500 knots by choosing corresponding p. The variable curve will be represented by knot points. This method can reduce the computation cost of sending large number of design variables to the optimizer and calculating the derivatives. With the B-spline interpolation, we only need to send small number of control points (300) to the optimizer and then represent the variable curve over time with knot points (1500). The effectiveness of computation is much improved, and the cost of optimization time is significantly decreased.

Chapter 4

Discipline Models

This chapter introduces the designs and mathematical models of each discipline in the VISORS multidisciplinary optimization algorithm. The CubeSats swarm optimization method models multiple disciplines, including orbit mechanics, attitude dynamics, communication, and propulsion. Specific modeling issues are discussed in this section.

4.1 Propulsion

There are multiple methods to provide CubeSat's propulsion supply, including chemical, electrical, and propellantless propulsion methods. One method is the cold gas propulsion system, which is used in attitude control and reaction wheel. [24] VISORS spacecraft is equipped with a 0.5U 3D-printed cold-gas propulsion module designed by a team from Georgia Tech. [25] The 3D-printed thruster made full use of the available volume of thruster and has the advantages of low cost and short development time. [26] The performance characteristics of the VISORS 0.5U cold-gas thruster meet requirements in terms of total impulse (67 N.sec) and minimum impulse bit (200 microN.sec). Some of the design parameters of the 0.5U cold-gas propulsion module are listed in Table 4.1. [25]

With the equations below, we introduce the mathematical modeling of propellant mass.

Table 4.1: 3D-printed propulsion module characteristics

Size	0.5U	Printed material	PerFORM
Propellant	R236fa	Operating temperature	[-20,+50] deg C
Empty mass	0.30kg	Specific impulse	45sec
Propellant mass	0.17kg	Total delta-velocity	19m/sec
Filled mass	0.47kg	Total impulse	67N.sec
Non-actuating power	0.25W	Min impulse bit	$200 \times 10^{-6} N.sec$
Actuating power (hold)	1.10W	Min actuation time	3 msec
Actuating power (spike)	7.89W	actuation force	20mN

The average specific impulse across all the actuation can be calculated using impulse measurements from each actuation J_i, standard gravity acceleration g, and mass change across all actuation ΔM. The propellant mass flow rate is calculated from thrust F, specific impulse I_{sp}, and standard gravity acceleration g. Given the initial propellant mass, we can compute the propellant mass state by calculating the thrust's integration over time. In our propulsion model, the discrete thruster scalar profile is set as the design variable. Eq. 4.1 shows the discrete thrust profile's integration to calculate the propellant state over time. With RK4, we can generate the propellant mass in the thruster while the CubeSat is operating

$$\begin{aligned} I_{sp} &= \frac{\sum J_i}{g \cdot \Delta M} \\ m = m_0 - \int \dot{m}\mathrm{dt} &= m_0 - \int \frac{F}{I_{sp} \cdot g}\mathrm{dt} \end{aligned} \tag{4.1}$$

$$m = m_0 - \sum \frac{F_i \cdot dt}{I_{sp} \cdot g} \tag{4.2}$$

4.2 Orbit dynamics

Before discussing the orbit dynamics issue, we should emphasize the atmospheric drag is a significant effect on orbital decay. As VISORS project progresses, team members from Purdue

University will produce more accurate drag models for orbital decay. We temporarily do not consider the effect of drag within the scope of the optimization procedure. The ideal equation for the VISORS research problem should characterize small relative orbits of CubeSats with respect to the large magnitude of the reference orbit radius. Also, because Earth's mass is not perfectly spherical and homogeneous, some coefficients need to be quoted to represent the characters of satellites in LEO. The equation below is referenced[1] to calculate the orbit dynamics of the satellite, where the terms J_1, J_2, J_3 are considered to capture the perturbation effects of rotating the orbit plane on the scale of a month. The design period of CubeSats, in LEO is roughly 90 minutes. The multiscale time problems need to be considered in MDO to capture the perturbation effects across minutes and hours. In the meantime, the slow rotation of the orbit plane will affect data communication as the rotation of the orbit plane will affect the satellite's trajectory, which is connected to the transmission of data to the ground stations.

$$
\begin{aligned}
\ddot{r} = & -\frac{\mu}{r^3} r - \frac{3\mu J_2 R_e^2}{2r^5}\left[\left(1-\frac{5r_z^2}{r^2}\right) r + 2r_z\hat{z}\right] \\
& -\frac{5\mu J_3 R_e^3}{2r^7}\left[\left(3r_z - \frac{7r_z^3}{r^2}\right) r + \left(3r_z - \frac{3r^2}{5r_z}\right) r_z\hat{z}\right] \\
& +\frac{15\mu J_4 R_e^4}{8r^7}\left[\left(1-\frac{14r_z^2}{r^2}+\frac{21r_z^4}{r^4}\right) r + \left(4-\frac{28r_z^2}{3r^2}\right) r_z\hat{z}\right]
\end{aligned}
\tag{4.3}
$$

$$
\mathbf{u}_i \approx \frac{\mathbf{F}_i}{m} \quad \text{where } \mathbf{u}_i = \mathbf{r}_i - \mathbf{r}_0 \tag{4.4}
$$

Based on the Eq. 4.3 4.4, we use the RK4 solver to generate the orbit equation along with time. The VISORS preferred orbit is a low Earth near-circular Sun-synchronous orbit at a range of altitude of $450\,\text{km} - 600\,\text{km}$, which ensures that the differential atmosphere drag effects are within acceptable tolerances and the formation is passively de-orbited. The orbit design requirements for CubeSats are listed in Table 4.2.

[1]Eagle,C.D., "Orbital Mechanics with MATLAB" http://www.cdeagle.com/ommatlab/toolbox.pdf [retrieved Februray 2013].

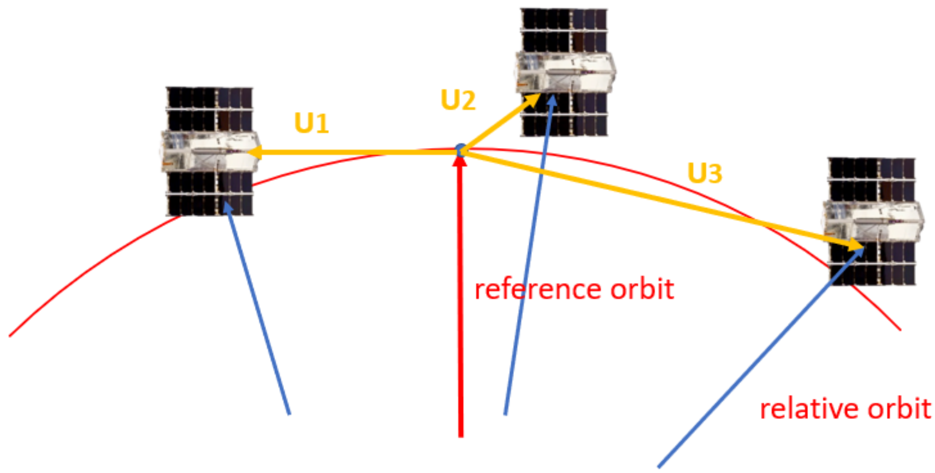


Figure 4.1: Reference orbit and relative orbits of three Cubesats

Table 4.2: VISORS orbit requirements

Orbital Parameter	Requirement	Preferred	Tolerance
Altitude	450 km – 600 km	500 km	±50 km
Inclination	$\geq 30°$	97.4°	±0.2°
Eccentricity	≤ 0.1	0.001	±0.01
Period	$93.5 - 97.8 min$	$94.6 min$	$\pm 1 min$

4.3 Attitude dynamics

The VISORS project's objective is to collect observation data of solar corona, so the spacecraft must always have a forward-facing orientation. Reaction wheels are internal mechanical components of controllable spacecrafts that enable them to reposition while in orbit and are sometimes referred to as momentum wheels. They control the spacecraft's attitude with very high precision, which is critical for our requirements of relative orbit adjustment at very small magnitudes. Reaction wheels store rotational energy, providing spacecrafts with three-axis attitude control. Three-axis control refers to the typical Cartesian system used to specify an object's location in three dimensions. Let us assume three torques working on the spacecraft first, which are roll, pitch and yaw. The deviation angles caused by these three rotations are θ_3, θ_1, and

θ_2, which transform the reference axes, xyz into body axes XYZ. θ_1,θ_2 and θ_3 are rotations about x, y, and z axes. The unit vectors for transformations between two sets of axes as follows:

$$\begin{pmatrix} e_X \\ e_Y \\ e_Z \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} e_x \\ c_y \\ e_z \end{pmatrix} \tag{4.5}$$

where

$$\begin{aligned}
a_{11} &= \cos\theta_2 \cos\theta_3 - \sin\theta_1 \sin\theta_2 \sin\theta_3 \\
a_{12} &= \cos\theta_2 \sin\theta_3 + \sin\theta_1 \sin\theta_2 \cos\theta_3 \\
a_{13} &= -\cos\theta_1 \sin\theta_2 \\
a_{21} &= -\cos\theta_1 \sin\theta_3 \\
a_{22} &= \cos\theta_1 \cos\theta_3 \\
a_{23} &= \sin\theta_1 \\
a_{31} &= \sin\theta_2 \cos\theta_3 + \sin\theta_1 \cos\theta_2 \sin\theta_3 \\
a_{32} &= \sin\theta_2 \sin\theta_3 - \sin\theta_1 \cos\theta_2 \cos\theta_3 \\
a_{33} &= \cos\theta_1 \cos\theta_2
\end{aligned} \tag{4.6}$$

In this optimization problem, we consider the roll and pitch rates as design variables and will add the yaw rate to consider three-axis rotation in the future. So the rotation matrix from body frame to the Earth-centered inertial (ECI) frame represented by roll and pitch is :

$$\mathbf{Rot\_body\_EF} = \begin{bmatrix} \cos\theta & \sin\theta \cdot \cos\phi & \sin\theta \cdot \sin\phi \\ -\sin\theta & \cos\theta \cdot \cos\phi & \cos\theta \cdot \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{4.7}$$

where

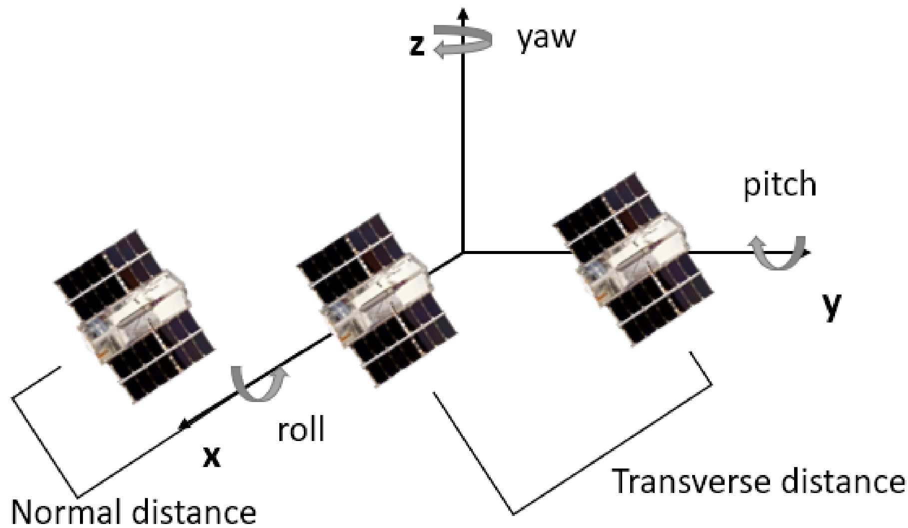


Figure 4.2: CubeSat attitude alignment and separation

θ	roll angle
ϕ	pitch angle

The VISORS spacecraft's attitude is determined by applying the rotations from the ECI frame to the actual body-fixed frame. With the rotation matrix above, we ensure the transforming frame is an intermediate frame obtained after ensuring the VISORS spacecraft is forward-facing and before applying the appropriate rotation from the specified roll angle and pitch angle profile.

4.4 Communication

The communication discipline models the data download rate as a function of several variables. [27] Four ground stations operated by VISORS project team members at UCSD, UIUC, Montana St., Georgia Tech are downlinked with the satellite. Data download is also one of our main optimization objectives in this project. The communication group's independent inputs include the longitude, latitude, altitude of ground station location, antenna angle, transmitter gain, initial data, communication power, and time scale. The orbit state and coordinate system rotation matrix are also used as inputs to compute the satellite and ground station locations simultaneously. Communication power is set as design variables, and total data download is the output of the communication discipline. With the optimization of the whole system, we can provide an energy use optimization strategy to control the maximum data download. According to the design of the VISORS project, one of the three CubeSats is selected as the chief communication spacecraft to keep contact with the four ground stations. Through optimization analysis, we find that there is no obvious difference in the data download rate between the four ground stations. However, to apply the CubeSats swarm toolkit to the global ground station layout, we give the chief communication spacecraft a real-time data communication selecting function. With the selection function, the spacecraft can choose which ground station to communicate with at any second. KS function [28] is used to help the satellite select the maximum data download rate from the connections between the four ground stations. Please refer to Chapter 3 for detailed mathematical formulas. The data-download rate is computed using the following Eq. 4.8 [29]

$$B_r = \frac{c^2 G_r L_l}{16\pi^2 f^2 k T_s (\mathrm{SNR})} \frac{\eta_p P_{\mathrm{comm}} G_t}{S^2} \mathrm{LOS}_c \tag{4.8}$$

The constants are listed in Table 4.3, S is the distance to the ground station, G_t is the transmitter gain, and LOS_c represents the LOS for communication model. In the communication discipline, the power of communication is set as one of the design variables in optimization. The power of communication is also used as the input variables to control if the CubeSat makes the data communication. For example, we have detector, sunshade, and optics CubeSats in one swarm, and according to the requirements of the VISORS project, we can make one CubeSat as chief spacecraft for data communication. Suppose we set sunshade CubeSat (SCS) as the chief CubeSat, we would set the power of communication (P_{comm}) at a nonzero value, and set the P_{comm} of another

Table 4.3: Constant coefficients for communication discipline [1]

Variable	Symbol	Value
Speed of light	c	$299792458 m/s$
Receiver gain	G_r	$12.9 dB$
Line loss factor	L_l	$-2.0 dB$
Transmission frequency	f	$437 MHz$
System noise temperature	T_s	$500 K$
Minimum acceptable SNR	SNR	$5.0 dB$

two other satellites at zero throughout the whole optimization process. In this way, we succeed in controlling the chief CubeSat to make a data connection with the four ground stations.

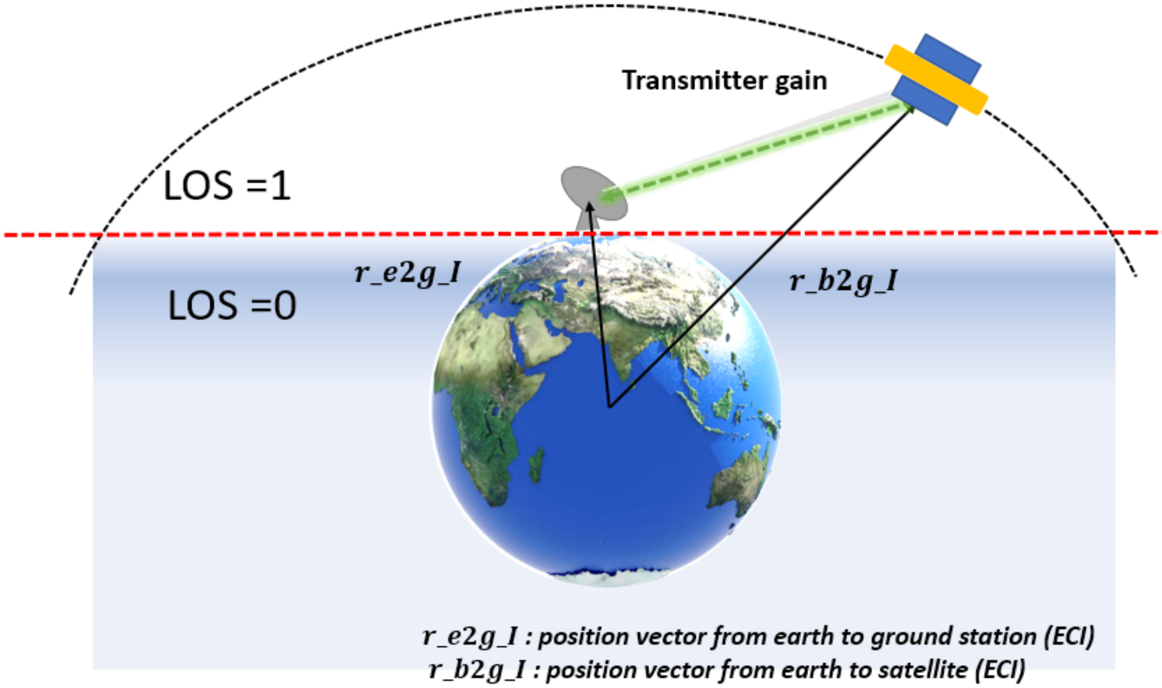
Quaternions provide a simple way to encode the axis-angle representation in four numbers. They can be used to apply the corresponding rotation to a positive vector, representing a point relative to the origin in R^3. In aerospace engineering, we usually use quaternions to represent the position and attitude vector of spatial vehicles. The function of the quaternion of the antenna angle is shown in Eq. 4.9, where θ represents antenna angle. The ECI frames originate at the center of Earth's mass and do not rotate with respect to the stars. Earth-centered, Earth-fixed (ECEF) frames also originate at the center of Earth's mass but remain fixed with respect to Earth surface's rotation. In the communication model, we need to calculate the rotation matrix from the Earth-centered inertial frame to the Earth-fixed frame over time as a tool of transforming the quaternions for potential use. A quaternion rotation $\mathbf{p}' = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$, (with $\mathbf{q} = q_r + q_i\mathbf{i} + q_j\mathbf{j} + q_k\mathbf{k}$), can be algebraically manipulated into a matrix rotation, so the rotation matrix can be calculated through $\mathbf{R} = \mathbf{p}'\mathbf{p}^{-1}$. In the same way, we drive the rotation matrix from ECI to ECEF based on the quaternion of Earth's spin, $s = \|\mathbf{q}\|^{-2}$. The rotation matrix is shown in Eq. 4.10.

$$
\mathbf{q_A} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \frac{\sqrt{2}}{2}\sin\left(\frac{\theta}{2}\right) \\ -\frac{\sqrt{2}}{2}\sin\left(\frac{\theta}{2}\right) \\ 0 \end{bmatrix} \tag{4.9}
$$

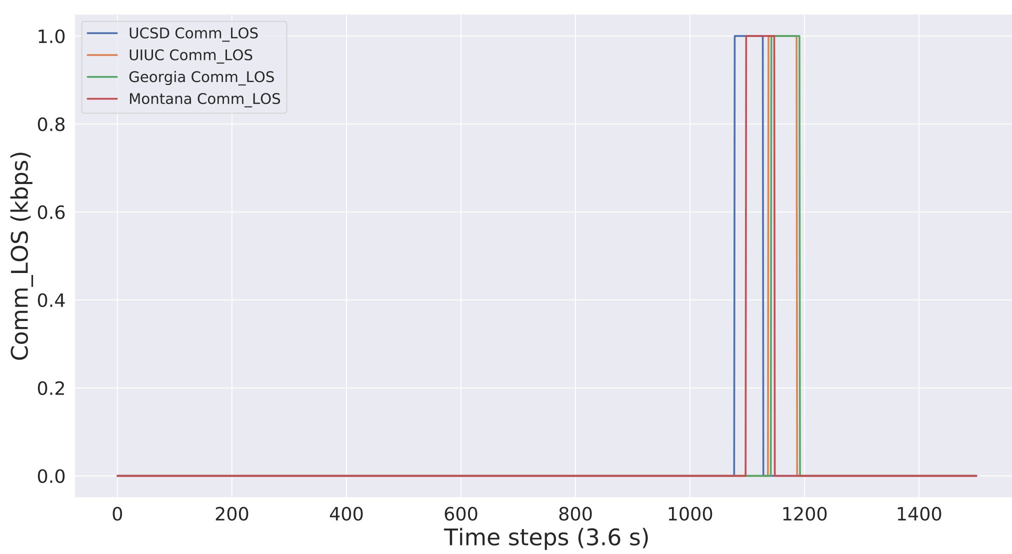
$$
\mathbf{Rot\_ECI\_EF} = \begin{bmatrix} 1-2s\left(q_j^2+q_k^2\right) & 2s\left(q_iq_j-q_kq_r\right) & 2s\left(q_iq_k+q_jq_r\right) \\ 2s\left(q_iq_j+q_kq_r\right) & 1-2s\left(q_i^2+q_k^2\right) & 2s\left(q_jq_k-q_iq_r\right) \\ 2s\left(q_iq_k-q_jq_r\right) & 2s\left(q_jq_k+q_iq_r\right) & 1-2s\left(q_i^2+q_j^2\right) \end{bmatrix} \tag{4.10}
$$

The Communication line of sight variable (Comm_LOS) is a multiplier for the exposed areas. We use the Comm_LOS variable to control the length of the connection time between the satellite and ground stations. When the satellite is behind the Earth, Comm_LOS is 0 and turns to 1 otherwise. In accordance with the requirements of the VISORS project, the contact time of *Comm_LOS* is around 15 seconds for each cycle. We then use the KS function to make a selection mechanism of data download rate between the satellite and each ground station.

The spacecraft bus for each CubeSat generates the power for communication. Each bus is a high-performance XB1 CubeSat bus commercially procured from Blue Canyon Technologies. The XB1 power system is a shunt-regulated direct energy-transfer system that consists of two 2-panel deployed solar arrays, a Li-ion battery, and a fuse assembly. The power system controller provides shunt regulation battery charging and peak power tracking for the solar panels. The spacecraft provides the communication power when data communication happens. The power of the communication supporting the operation of CubeSat is set as the design variables of the communication model. The objective of our research is to present a plan for power management and for time spent on data communication with the ground stations. To reduce computation, we use the control points of power variables as design variables and use a B-spline function



(a) Variable Comm_LOS and position vectors related to the calculated distance between ground station and satellite



(b) Comm_LOS differs with different ground stations location and is used to control the length of data communication time.

Figure 4.3: CubeSat–ground station data communication mechanism

Table 4.4: VISORS CubeSat mass and power design data

CubeSat	Volume(U)	Mass(g)	Average Power (W)	Peak Power (W)
Optics CubeSat total	2.95	3628	13	37.2
Sunshade CubeSat total	2.95	3628	13	37.2
Detector CubeSat total	2.4	3808	14.3	42.7
Budget	3	4000		21

Table 4.5: Geographic coordinates of four ground stations

Ground Station	Altitude (km)	Longitude (degree)	Latitude (degree)
UCSD	0.4849	-117.1611	32.7157
UIUC	0.2329	-88.2272	32.8801
Georgia Tech	0.2969	-84.3963	33.7756
Montana	1.04	-109.5337	33.7756

to generate whole power design variables and data communication time. According to the requirements of the VISORS team, we set SSC as the chief spacecraft for communication. The input P_{comm} of the DSC and OSC are set zero and input P_{comm} of SSC is then set 13 W as initial value and is sent to the optimizer.

The power for data communication mainly supports the data communication between spacecrafts and ground stations, and the 5G-inspired high data-rate inter-CubeSat swarm communication and networking. For the communication model of our design, we temporarily neglect the cross-link data communication and focus on the data communication between satellite and ground stations. Four ground stations are responsible for collecting optics data from the satellite. The data accumulation budget is 20 MB/day, and is predicted to last for 35 minutes of contact time per day. The entire 20 MB/day data accumulation can be downlinked at the spacecraft transmit rate of 19.2 kbps. The instrumental data will be collected in scientific mode and stored in onboard memory until downlinking.

4.5 Spacecraft alignment and separation

According to the requirements of the VISORS project, the observation in each cycle happens approximately ten seconds when the photon sieve, sunshade, detector spacecraft, and Sun are aligned with each other. To meet the alignment requirements, we calculate the relative orbits of three CubeSats. The relative orbits of CubeSats should guarantee three conditions: 1) relative acceleration perpendicular to the LOS is minimized to ensure that scientific images are not degraded by the relative motion of the formation; 2) the alignment of three CubeSats is periodic ally repeated so that observations can be repeated; 3) the three CubeSats do not collide.

There are three vectors related to the alignment modeling: the satellite position vector, velocity vector, and sun direction vector. Figure. 4.4 shows the vectors in relation to satellites alignment. First, we calculate the cross product of the position vector and velocity vector and the cross product of the Sun direction vector and position vector called the normal cross-product and observation cross-product separately. When the satellite flies into the observation phase, the velocity vector is aligned with the Sun direction vector. The directions of the observation cross product vector and normal cross product vector are opposite. Then we set a cross threshold value as an observation filter; the mask vector equals 1 when the dot product of the observation cross product vector and normal cross product vector is less than the cross threshold and it returns to 0 when it is greater than the cross threshold. Both normal distance and transverse distance can be calculated from the satellite's orbit state, which contains the position vector and velocity vector. By decomposing the position vector into the coordinate system whose z axis is aligned with the orbit radius vector, we can calculate the normal distance and transverse distance between satellites. In our modeling, alignment is the highest priority to consider, and is constraints are very strict.

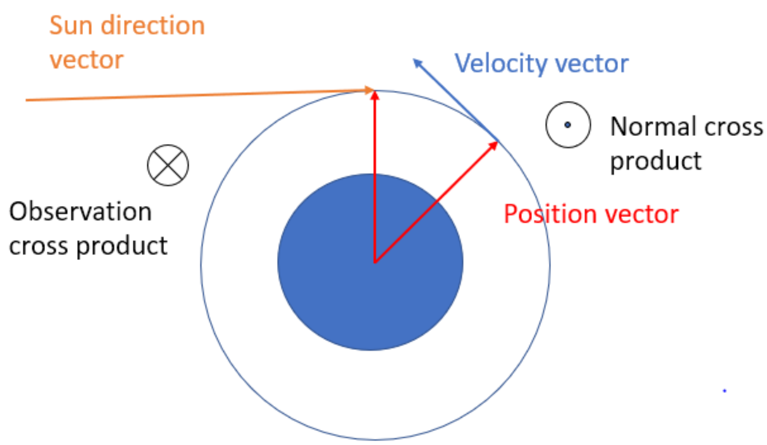


Figure 4.4: Mask vector judgement of observation phases

Chapter 5

Optimization

5.1 Multidisciplinary optimization architecture

The model architecture is critical for solving a multidisciplinary optimization problem, especially for explaining the relationship between variables. For instance, the outputs of one discipline analysis might be the inputs of another discipline; some inputs should be taken as a primitive variables and sent to all disciplines. Also, the objective function and constraints depends on the design variables; the precise expression of each variable's interdependence would lead to a significantly more accurate representation of the system's behavior. Good architecture can manage the system by coupling disciplines efficiently and clearly expressing complex interdependence among disciplines. Multidisciplinary optimization architecture offers a clear, systematic framework for the management of interdependent relationships in optimization problem-solving.

Martins and Lambe [30] discussed several specific architectures based on various optimization problems, including the all-at-once (AAO) problem, simultaneous analysis, and design (SAND), and individual discipline feasible (IDF). The architecture we use in this project is the multidisciplinary feasible (MDF) architecture. The main advantage of MDF is that its design variables, objective function, and constraints are under the direct control of the optimizer. MDF returns a system design that always satisfies the consistency constraints, even if the optimization process is terminated early. For each iteration of optimization, MDF solves a multidisciplinary optimization problem with all disciplines, and effectively analyzes all disciplines in one monolithic analysis.

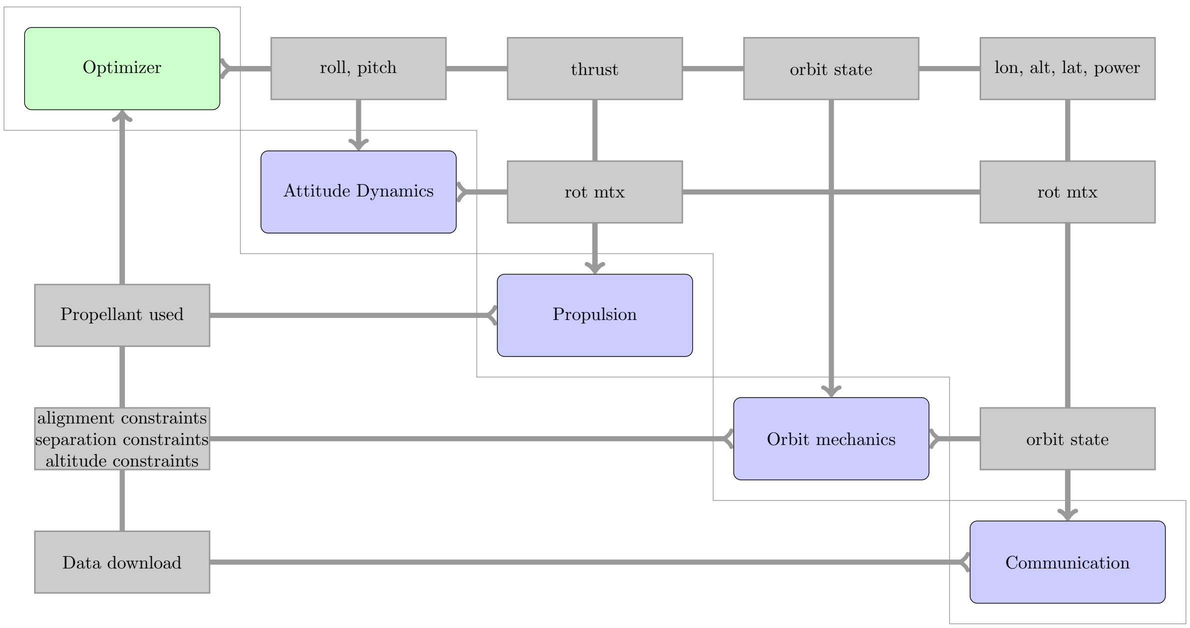


Figure 5.1: Multi-discipline architecture of CubeSat swarm optimization

5.2 Optimization Problem

In this stage of optimization, we focus on driving good optimization for one orbit cycle of the CubeSat swarm, so the satellite flight lasts 90 minutes. The optimization objective is to minimize the total propellant used and to maximize the total data download. We form the objective function using a linear combination of total propellant used and total data downloaded along with the alignment and separation distance between CubeSats. The linear combination of normal distance between CubeSats and relative orbit states of each CubeSats are added to

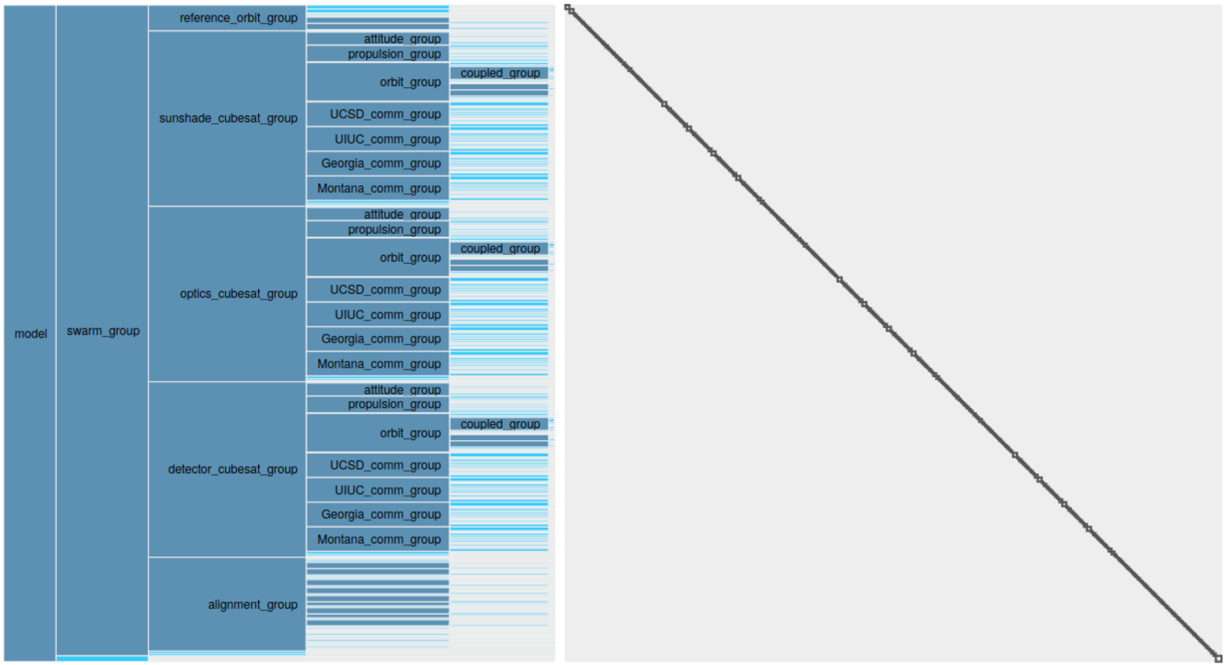


Figure 5.2: OpenMDAO N2 structure of CubeSats swarm

the objective function. We also add the alignment and separation constraints to the objective function as a penalty on the optimization objective. As table **??** shows below, the design variables are pitch and roll rate from reaction wheels of Cubesats in order to control the forward facing orientation to the sun light; the magnitude of thrust from the propulsion system, which is directly connected with the optimizing objective of propellant used; and the communication power, which controls the power supply of data communication and which spacecraft can communicate as the communication power is directly connected with the data download rate and affects the total data download. The constraints are bounds on roll rate and pitch rate, the minimum altitude to control spacecraft in low Earth orbit, alignment between sunshade and detector, alignment between optics and detector, and separation between optics and detector, separation between sunshade and optics. Each profile variable is discretized with 1501 points, which means we divide the 90-minute satellite period into 1501 time intervals. As mentioned before in Chapter 4 Methodology, for reducing computation efficiency, 1501 points are represented using fourth-order B-splines with 300 control points. The total number of design variables is 3000, and the total

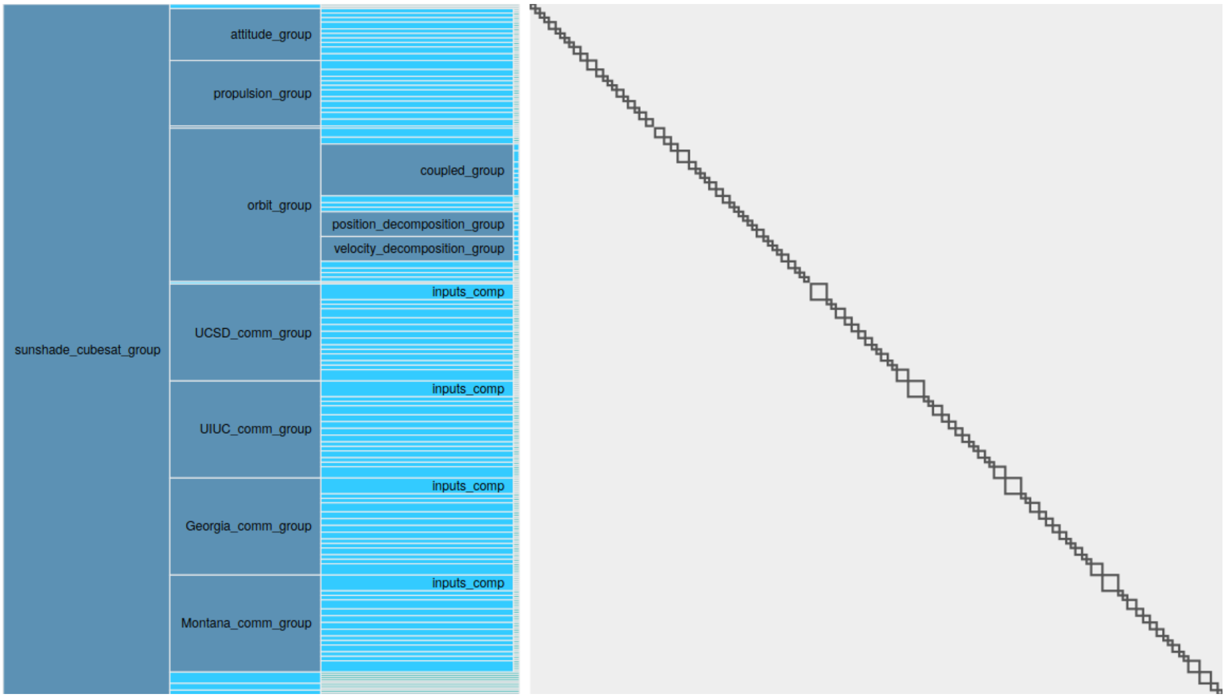


Figure 5.3: OpenMDAO N2 structure of sunshade CubeSat

number of variables is 9009, which relates to the multidisciplinary optimization analysis.

As discussed in Chapter 4, we solve the optimization problem using SNOPT, a reduced-Hessian active-set SQP optimizer that solves nonlinear constrained problems very efficiently. SNOPT requires that derivatives of each component be provided. OpenMDAO can cooperate with SNOPT and provides derivative computation for disciplines. The pyOpt optimization framework is used to interface with a suite of optimizations, including SNOPT.

Table 5.1: Optimization problem of CubeSat swarm operation

	Variable	Size
Objective	Total propellant used	
	Total data download	
Design variables	Roll	3×300
	Pitch	3×300
	Thrust magnitude	3×300
	Communication power	1×300
	Total	3000
Constraints	Roll rate	3×1501
	Pitch rate	3×1501
	Minimum altitude	3×1
	Alignment (sunshade-detector)	
	Alignment (optics-detector)	
	Separation (optics-detector)	
	Separation (sunshade-optics)	
	Total	9009

Chapter 6

Results and Discussion

The optimization problem focuses on the operation of CubeSat swarms over one orbit cycle. This chapter presents the optimization results of the CubeSat swarm multidisciplinary problem, the approach feasibility analysis and evaluation of mathematical modeling. The analysis focuses two aspects: determining whether the mathematical modeling is reasonable and evaluating the optimization results.

6.1 Alignment and separation for CubeSats

The alignment constraints require the satellites to be aligned with each other for at least 10 seconds, which means three spacecrafts need to be located on a line parallel to the sunlight. The alignment constraints are strict because the detector on a CubeSat has very high requirements for the LOS angle formed by the light passing through the optics and sunshade satellite. The normal distance between CubeSats is used to evaluate their alignment. Figure. 6.1 shows the optimized result of the normal distance between the sunshade and optics CubeSats and the normal distance between the optics and detector CubeSats. We set 300 mm as the tolerance bound of the alignments, which means at the observation time, the normal distance between CubeSats should be no greater than 1000 mm. The yellow bar in the picture characterizes the time range for

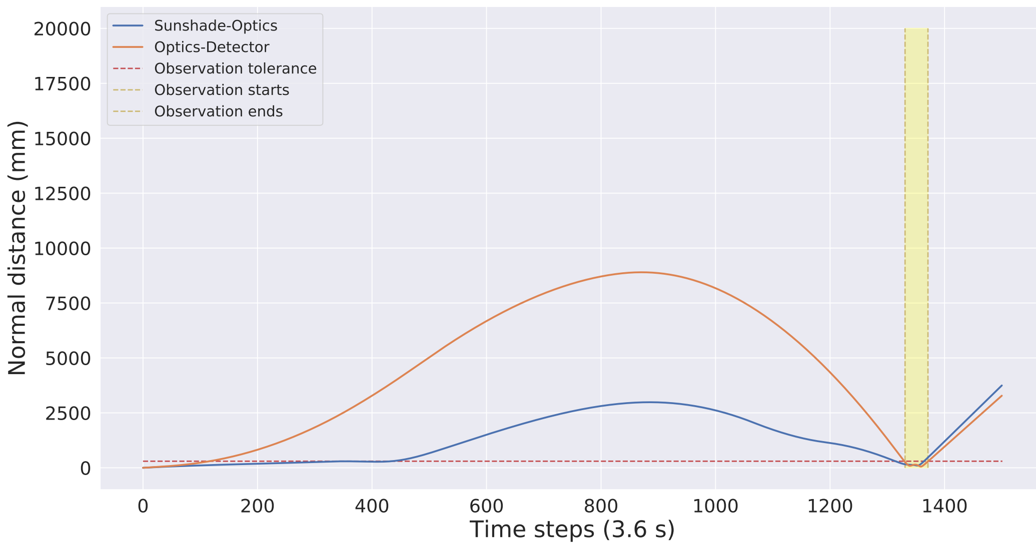


Figure 6.1: The optimized normal distance between spacecrafts

CubeSat observation. Through calculation, the observation time range is around 120 s. The length of observation time satisfies the requirement of the observation time should attain at least 10s.

6.2 Communication

For the communication discipline, we first compute the data download rate between the chief communication CubeSat and four ground stations. Then, the KS function is used to linearize four data download rate curves and to drive the most massive data download rate over time. In this way, we ensure the chief communication CubeSat can proceed with the communication at the maximum data download rate in every second. Figure. 4.3(b) shows the data download rate linearized by the KS function. The KS function will play a more important role when the layout of the ground stations becomes more scattered.

Based on the design requirements of VISORS, the data accumulation budget is 20 MB per day and is downlinked with four ground stations. The satellite maintains contact with the ground stations for 35 minutes per day. The entire data budget is 20 MB per day. Data accumulation

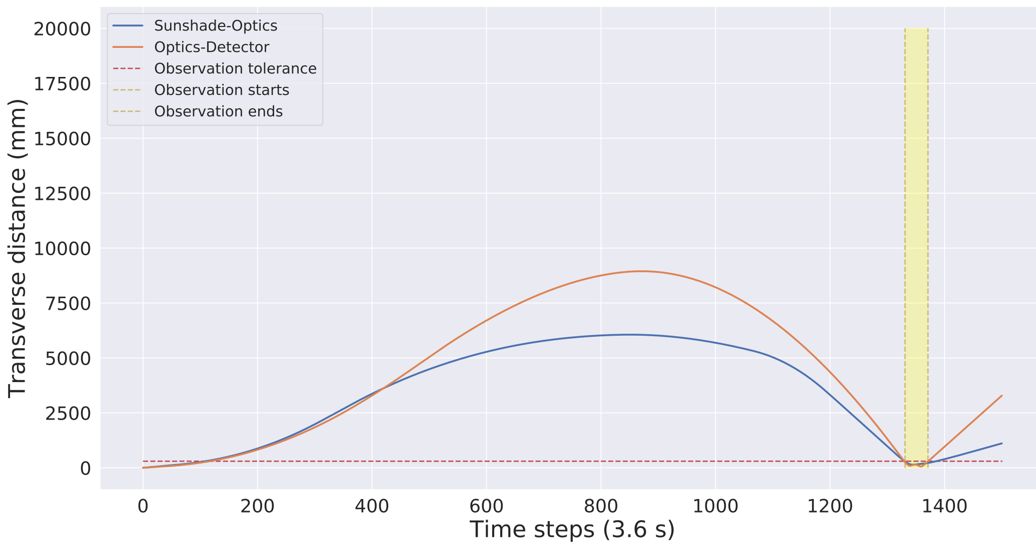


Figure 6.2: The optimized transverse distance between spacecrafts

should be achieved with a suitable data download rate. Since the satellite revolves around the Earth in a period of 90 minutes, the amount of data downloaded when the satellite orbits the Earth for one cycle is 1.25 MB. The total data download per day in our optimized result reaches around 1.25 MB. The communication power yields 52.9 % improvements on total data download. The optimized data download satisfies the requirement of 1.25 MB data downloaded per cycle while the satellite is orbiting Earth.

6.3 Attitude control and propellant used

The value of the roll and pitch angles represents the rotation state of CubeSats. Figure. 6.6 shows the change in roll and pitch angles for three CubeSats over time. To maintain forward-oriented flight, in the observation phase, the satellite makes noticeably visible adjustments on the longitudinal axis (roll) and makes a minor adjustment on the transverse axis (pitch). When the observation time starts, the satellites begin to make significant rotational movements to ensure they are aligned with each other. They revert to standby state after the observation ends. The

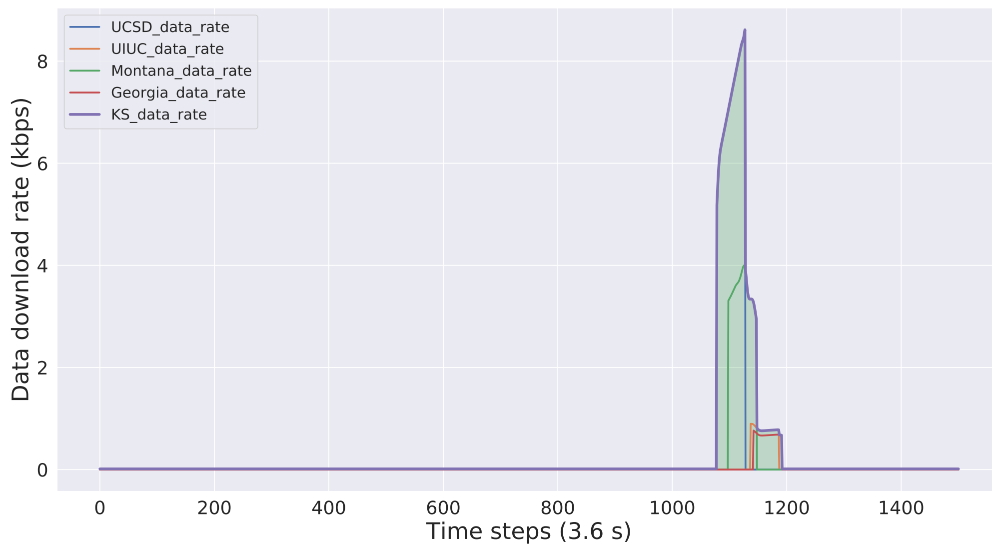


Figure 6.3: Chief communication CubeSat data download rate linked with four ground stations and total KS data download rate

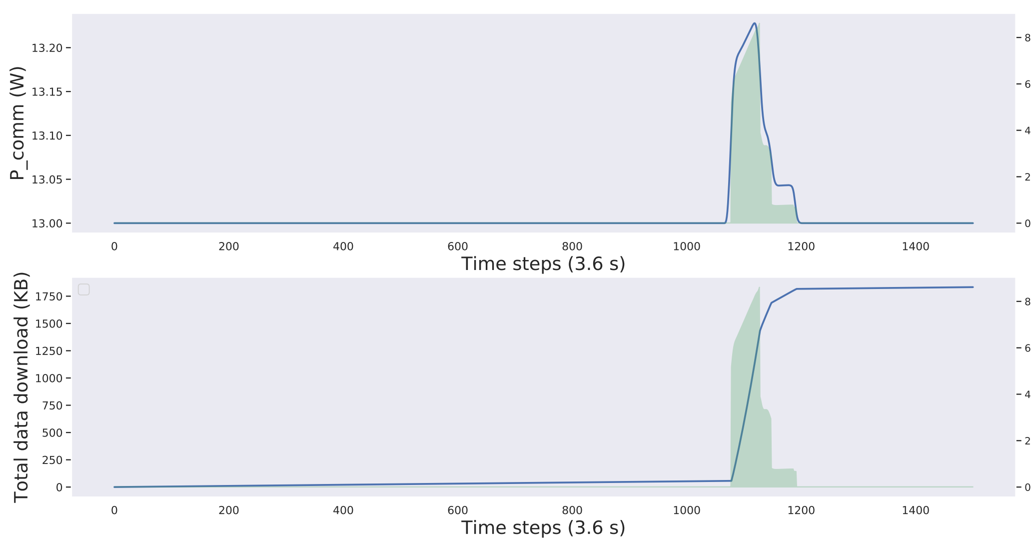


Figure 6.4: P_comm (design variable) and total data download (objective), green shade represents data communication interval.

roll and pitch angle plots show the reasonable use of power by satellites and the satellites' rapid response to the scientific observation phase. To better express the flight status of the satellites and judge the force direction, we introduce the variable delta velocity Δv. Δv is a scalar measure for the amount of "effort" needed to carry out an orbital maneuver, which is typically provided by the thrust of a rocket engine [31]. The mathematical definition of delta-v is from the Tsiolkovsky rocket equation [32] shown in Eq. 6.2. In our model, we use this equation to calculate the discretized delta-v in discrete time, which is equal to the thrust divided by the mass state. In the ideal rocket function, delta-v is related to acceleration, initial and final engine mass, and the thrust profile of the spacecraft. m_0 is the spacecraft's initial mass and m_e is the spacecraft's exhausted mass with fuel used up (the final mass).

$$\Delta v = \int_{t_0}^{t_1} \frac{|T(t)|}{m(t)} dt \tag{6.1}$$

$$\Delta v = v_{\mathrm{e}} \ln \frac{m_0}{m_e} = I_{\mathrm{sp}} g \ln \frac{m_0}{m_e} \tag{6.2}$$

Here we introduce the delta-v variable as an evaluation of the propulsion optimization result. As we model the CubeSats operation for one Earth orbit, the amount of propellant used by CubeSat is very small, and we do not have data from an existing unfinished model to compare the propellant use with our optimization result. Based on the ideal rocket function, we generate an evaluation plan for propellant use based on delta-v. Different from the former Δv, which is calculated from the initial and final mass of the spacecraft, we introduce a new variable Δv_t, which is calculated from the initial mass m_0 and current mass state m_{ft} of spacecraft (suppose I_{sp} and g are constants). We have the following equation:

$$
\begin{aligned}
\mathrm{du} &= -\mathrm{I}_{sp} g \frac{dM}{M} \\
\Delta v_t &= -\, I_{sp} \ln(\mathrm{M})\Big|_{\mathrm{m_{ft}}}^{\mathrm{m_e}} \\
\Delta v_{\mathrm{t}} &= I_{\mathrm{sp}} g \ln \frac{m_0}{m_0 - m_{ft}}
\end{aligned}
\tag{6.3}
$$

where Δv_t divided by Δv, and we have a proportional equation whose unique variable is m_{ft}:

$$
\begin{aligned}
\frac{\Delta v_t}{\Delta v} &= \frac{\ln \frac{m_0}{m_0 - m_{ft}}}{\ln \frac{m_0}{m_e}} \\
&= \frac{\ln m_0 - \ln \left(m_0 - m_{ft}\right)}{\ln m_0 - \ln \left(m_0 - m_f\right)}
\end{aligned}
\tag{6.4}
$$

$$
\Delta v = \frac{\ln m_0 - \ln \left(m_0 - m_f\right)}{\ln m_0 - \ln \left(m_0 - m_{ft}\right)} \cdot \Delta v_t \tag{6.5}
$$

Based on Eq. 6.5, we generate an expression of Δv represented by the Δv_t and m_{ft} variables. Δv_t, as we discussed before, is calculated by discrete function integration. Given the propellant mass state optimized by our model, we can calculate the total delta-v budget as an evaluation of propulsion discipline. From our calculation, the predicted total delta-v budget is 22.0183 m/sec, which is larger than the design value of 19 m/sec of a 0.5U cold-gas thruster. As for the analysis of Δv, the total propellant mass is the same for the optimization and design. Our optimization results provide a better thrust strategy for the propulsion system than the original plan and have greater impact with the same propellant consumption. We evaluate the propulsion system using the ratio of two total Δv values and finds it drive around improvement of around 15%.

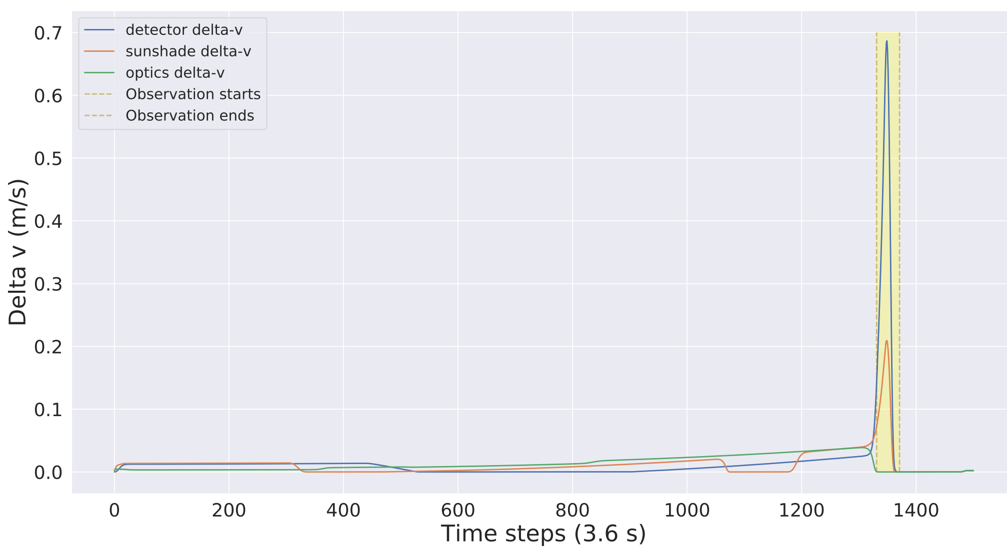
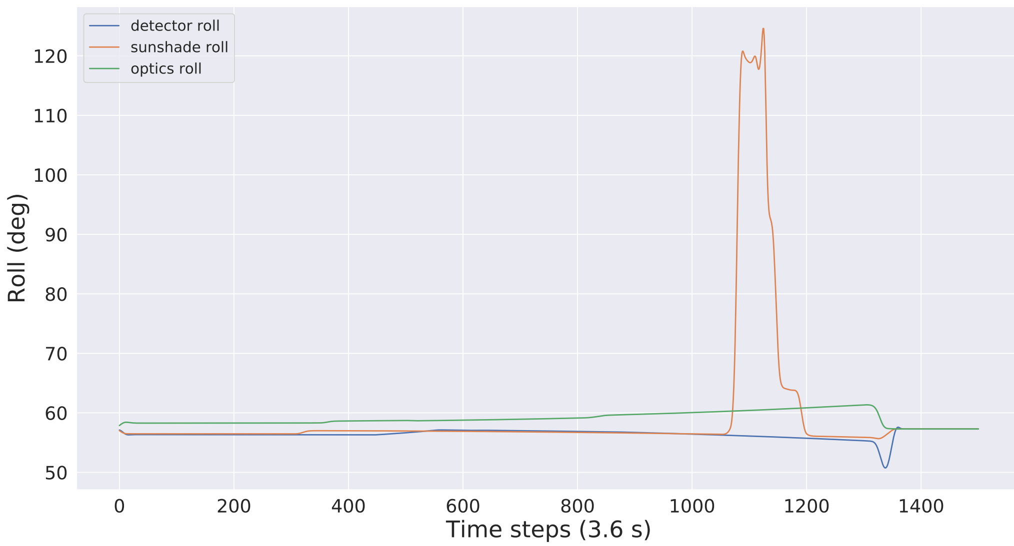
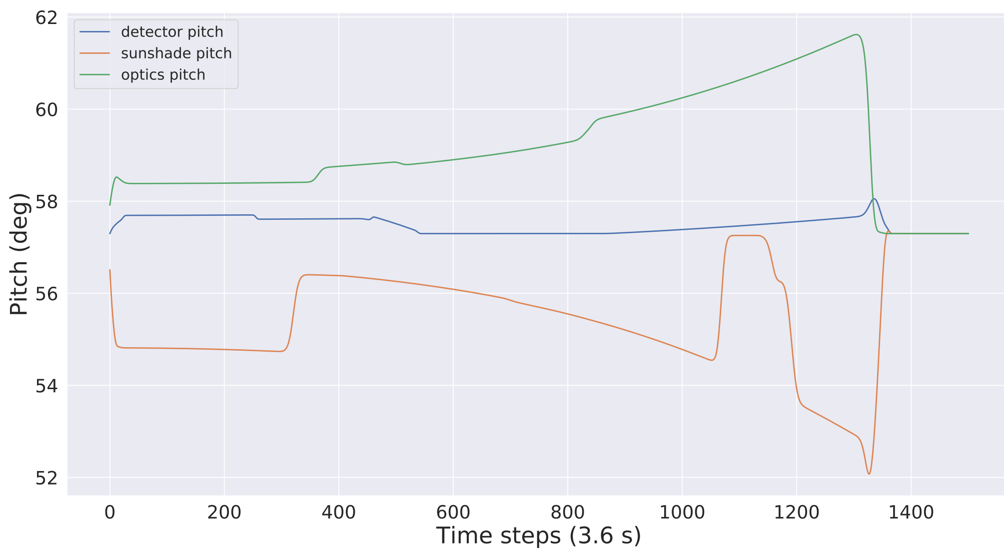


Figure 6.5: Delta-v of spacecrafts over time

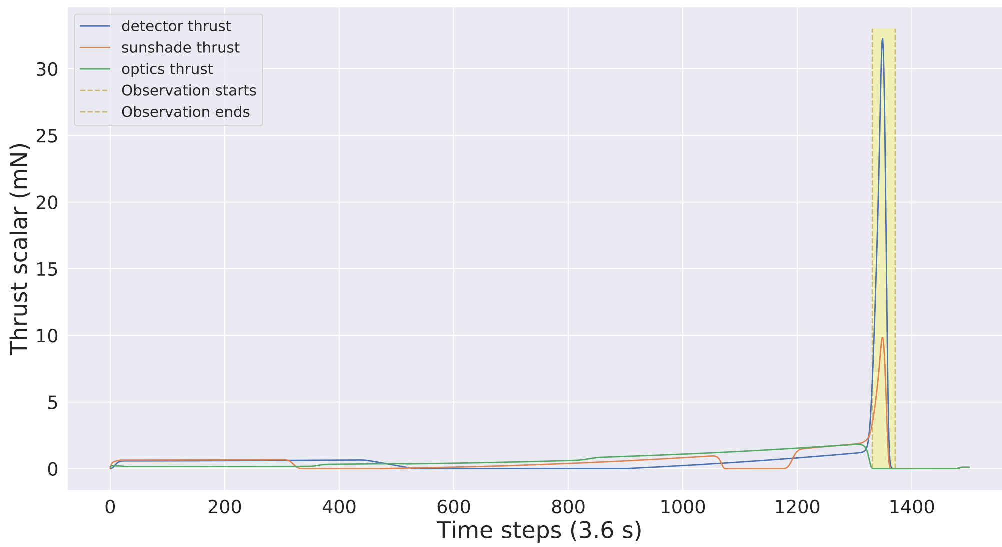


(a) Roll angle of three CubeSats over time

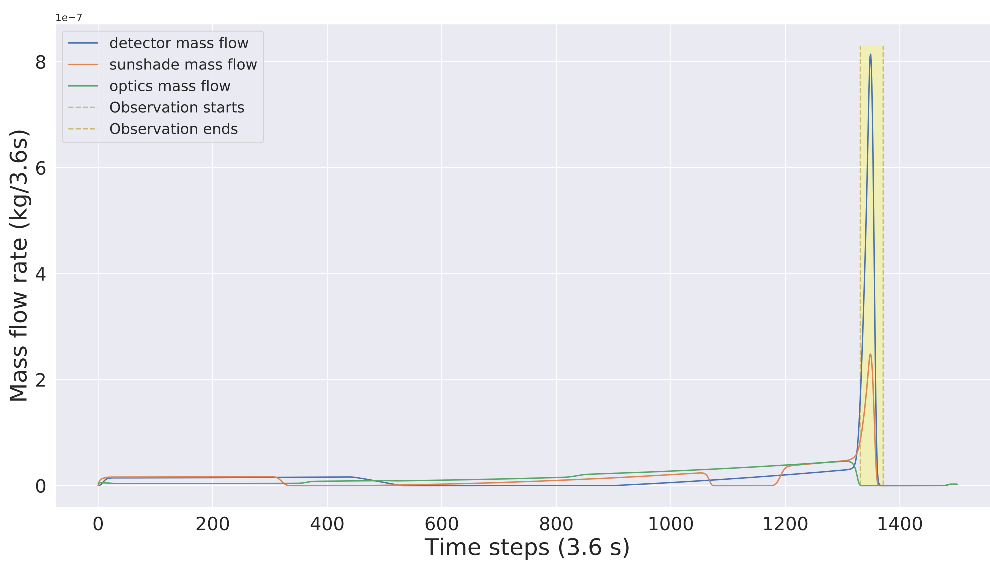


(b) Pitch angle of three CubeSats over time

Figure 6.6: CubeSats attitude control, roll and pitch angle change over time



(a) Thrust scalar of three CubeSats



(b) Propellant mass flow rate over time

Figure 6.7: CubeSats thrust propellant mass flow rate variable over time

Chapter 7

Conclusion and Future work

7.1 Conclusion

With the implementation of VISORS project, the CubeSat swarm increases the total data download of the chief communication spacecraft by 52.9 % compared to that of the original design. For the propulsion model, based on our evaluation of the delta-v variable, the thrust distribution plan improves by around 15 % compared to the original thrust design scheme. With the same amount of propellant, the thrust distribution plan provided by our model can generate a larger delta-v over time than the original design. As for the mission concept, the method models the multiple formation flight phases of the CubeSat swarm and controls the alignment constraints between CubeSats within a 300 mm tolerance threshold. The CubeSats rotate and attain forward-orientation flight in the observation phases. The observation time of solar corona is planned as long as 120 s, which meets the VISORS design requirement of a length of 10 s. The KS function introduces the data filtering mechanism for the CubeSat swarm model, which will have a more significant effect when the layout of the ground stations becomes wider. In summary, we have generated a physical-based model, implemented MDO algorithm on CubeSats swarm and well optimized the objective function.

The optimization method of solving the CubeSat swarms operation problem efficiently handles thousands of design variables and combines multiple disciplines, including orbital dynamics, attitude control, propulsion, and communication. In terms of the robustness of the CubeSat swarm model, the convergence time is around 1-3 hours. The cooperation of MDO algorithm and the SNOPT optimizer provides an efficient solution for handling thousands of variables, multiple disciplines, and their derivatives. The control parameter setting and model relationship representation help SNOPT find the local minimum and global solution in a relatively short time.

In conclusion, this thesis generates a new physics-based mathematical model of CubeSat swarm operation, provides a new algorithm for optimizing the CubeSat swarm multidisciplinary problem, and achieves the implementation of a multidisciplinary optimization algorithm on practical engineering design.

7.2 Future work

Although we aimed to include as many disciplines as possible in the operation of the CubeSats swarm, some new disciplines need to be further considered, such as solar panel and energy storage. Also, some technical details are temporarily simplified in the first generation of the CubeSats swarm toolkit. The method is intended to be developed as an open-source toolkit. First, the inter-satellite crosslink between CubeSats should be considered in the communication discipline. In accordance with VISORS's mission design, the satellites have a high inter-satellite crosslink with each other. Secondly, the transmitter gain is a part of the design variables we should not ignore. The antenna gain is a key performance number that combines the antenna's directivity and electrical efficiency in electromagnetics. Thirdly, we aim to achieve the attitude control of the satellite in three axes.

As for the macro mission concept of small-satellite operation design, a new flying formation of CubeSats, like two 6U CubeSats will be discussed by team members. The design will achieve more complex work and be much easier for batch cooperation than the previous mission concept of using three 3U CubeSats. Also, we could set one CubeSats swarm as one mission unit and send multiple units to the orbit to conduct the scientific work. The flying formation of multiple CubeSats would be like the Starlink concept, which means that in one Earth orbit, we would have tens or hundreds of CubeSats swarms to conduct scientific work and achieve more complex mission goals. Finally, we can image something even greater, such as building a satellite network. We would design the optimization platform used for a multi-agent spacecraft operation. Then more multi-agent swarm control optimization problem would be generated, and the relationships between spacecrafts would become more complicated. The toolkit developed by that time will be feasible for much more practical problems and could be used not only as an academic optimization tool but also to solve problems with simulation, optimization, design, and analysis.

Bibliography

[1] John T Hwang, Dae Young Lee, James W Cutler, and Joaquim RRA Martins. Large-scale multidisciplinary optimization of a small satellite's design and operation. *Journal of Spacecraft and Rockets*, 51(5):1648–1663, 2014.

[2] Franco Davoli, Charilaos Kourogiorgas, Mario Marchese, Athanasios Panagopoulos, and Fabio Patrone. Small satellites and cubesats: Survey of structures, architectures, and protocols. *International Journal of Satellite Communications and Networking*, 37(4):343–359, 2019.

[3] Daniel Selva and David Krejci. A survey and assessment of the capabilities of cubesats for earth observation. *Acta Astronautica*, 74:50–68, 2012.

[4] Sun Zhaowei, Xu Guodong, Lin Xiaohui, and Cao Xibin. The integrated system for design, analysis, system simulation and evaluation of the small satellite. *Advances in Engineering Software*, 31(7):437–443, 2000.

[5] Wenrui Wu, Hai Huang, Shenyan Chen, and Beibei Wu. Satellite multidisciplinary design optimization with a high-fidelity model. *Journal of Spacecraft and Rockets*, 50(2):463–466, 2013.

[6] Katherine Alston and Tyler Winter. Balancing high fidelity mdao with robust system design. In *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 884, 2011.

[7] James Cutler, Aaron Ridley, and Andrew Nicholas. Cubesat investigating atmospheric density response to extreme driving (cadre). 2011.

[8] Scott Perry, James W Cutler, Aaron Ridley, Tom Heine, and Andrew Nicholas. Cubesat investigating atmospheric density response to extreme driving (cadre).

[9] Vincent Giralo, Michelle Chernick, and Simone D'Amico. Guidance, navigation, and control for the dwarf mission.

[10] Christopher A Carter and E Glenn Lightsey. Cubesat slosh dynamics and its effect on precision attitude control for the visors distributed spacecraft telescope mission.

[11] Manuel Güdel. The sun in time: Activity and environment. *Living Reviews in Solar Physics*, 4(1):3, 2007.

[12] Ignasi Ribas, Edward F Guinan, Manuel Güdel, and Marc Audard. Evolution of the solar activity over time and effects on planetary atmospheres. i. high-energy irradiances (1-1700 å). *The Astrophysical Journal*, 622(1):680, 2005.

[13] Gerald R Smith, Darrell F Strobel, AL Broadfoot, BR Sandel, DE Shemansky, and JB Holberg. Titan's upper atmosphere: Composition and temperature from the euv solar occultation results. *Journal of Geophysical Research: Space Physics*, 87(A3):1351–1359, 1982.

[14] L Kipp, M Skibowski, RL Johnson, R Berndt, R Adelung, S Harm, and R Seemann. Sharper images by focusing soft x-rays with photon sieves. *Nature*, 414(6860):184–188, 2001.

[15] John T Hwang and Joaquim RRA Martins. A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives. *ACM Transactions on Mathematical Software (TOMS)*, 44(4):1–39, 2018.

[16] Joaquim RRA Martins, Peter Sturdza, and Juan J Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software (TOMS)*, 29(3):245–262, 2003.

[17] James J Reuther, Antony Jameson, Juan J Alonso, Mark J Rimllnger, and David Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 2. *Journal of aircraft*, 36(1):61–74, 1999.

[18] Antony Jameson. Aerodynamic design via control theory. *Journal of scientific computing*, 3(3):233–260, 1988.

[19] Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900, 2000.

[20] Ruben E Perez, Peter W Jansen, and Joaquim RRA Martins. pyopt: a python-based object-oriented framework for nonlinear constrained optimization. *Structural and Multidisciplinary Optimization*, 45(1):101–118, 2012.

[21] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

[22] Nicholas MK Poon and Joaquim RRA Martins. An adaptive approach to constraint aggregation using adjoint sensitivity analysis. *Structural and Multidisciplinary Optimization*, 34(1):61–73, 2007.

[23] JRRA Martins and Nicholas MK Poon. On structural optimization using constraint aggregation. In *VI World Congress on Structural and Multidisciplinary Optimization WCSMO6, Rio de Janeiro, Brasil*. Citeseer, 2005.

[24] Kristina Lemmer. Propulsion for cubesats. *Acta Astronautica*, 134:231–243, 2017.

[25] Travis K Imken, Terry H Stevenson, and E Glenn Lightsey. Design and testing of a cold gas thruster for an interplanetary cubesat mission. *Journal of Small Satellites*, 4(2):371–386, 2015.

[26] Matt Sorgenfrei, Terry Stevenson, and Glenn Lightsey. Performance characterization of a cold gas propulsion system for a deep space cubesat. In *40th Annual AAS Guidance and Control Conference, Breckenridge, CO*, 2017.

[27] Gian B Alaria and Roberto Preti. Method of and system for communication via satellite, October 30 1984. US Patent 4,480,328.

[28] Igor Splawski, Martin Tristani-Firouzi, Michael H Lehmann, Michael C Sanguinetti, and Mark T Keating. Mutations in the hmink gene cause long qt syndrome and suppress l ks function. *Nature genetics*, 17(3):338–340, 1997.

[29] Wiley J Larson and James Richard Wertz. Space mission analysis and design. Technical report, Torrance, CA (United States); Microcosm, Inc., 1992.

[30] Joaquim RRA Martins and Andrew B Lambe. Multidisciplinary design optimization: a survey of architectures. *AIAA journal*, 51(9):2049–2075, 2013.

[31] K Tsiolkovsky and W Moore. Academy, rm: Tsiolkovsky rocket equation. *Energy*, pages 1–8, 1903.

[32] V Dvornychenko. The generalized tsiolkovsky equation. In *NASA Conference Publication*, volume 3102, page 449. National Aeronautics and Space Administration, Scientific and Technical . . . , 1990.