Large-Scale Multidisciplinary Optimization of a Small Satellite's Design and Operation

John T. Hwang[1], Dae Young Lee[2], James W. Cutler[3],
and Joaquim R. R. A. Martins[4]
University of Michigan, Ann Arbor, Michigan, 48109, United States

Abstract The design of satellites and their operation is a complex task that involves a large number of variables and multiple engineering disciplines. Thus, it could benefit from the application of multidisciplinary design optimization, but previous efforts have been hindered by the complexity of the modeling and implementation, discontinuities in the design space, and the wide range of time-scales. We address these issues by applying a new mathematical framework for gradient-based multidisciplinary optimization that automatically computes the coupled derivatives of the multidisciplinary system via a generalized form of the adjoint method. The modeled disciplines are orbit dynamics, attitude dynamics, cell illumination, temperature, solar power, energy storage, and communication. Many of these disciplines include functions with discontinuities and nonsmooth regions that are addressed to enable a numerically exact computation of the derivatives for all the modeled variables. The wide-ranging time scales in the design problem, spanning 30 seconds to 1 year, are captured through a combination of multi-point optimization and the use of a small time step in the analyses. Optimizations involving over 25,000 design variables and 2.2 million state variables require 100 hours to converge 3 and 5 orders of magnitude in optimality and feasibility, respectively. The results show that the geometric design variables yield a 40% improvement in the total data downloaded, which is the objective function, and the operational design variables yield another 40% improvement. This demonstrates not only the value in this approach for the design of satellites and their operation, but also promise for its application to the design of other large-scale engineering systems.

Nomenclature

$\vec{(\cdot)}$	=	Vector
$\hat{(\cdot)}$	=	Unit vector
A	=	Area, m^2
B_r	=	Bit rate, Gb/s
$\boldsymbol{c}$	=	Vector of all constraint variables
$\boldsymbol{C}$	=	Vector of all constraint functions
$\boldsymbol{C}_i$	=	Vector of constraint functions corresponding to the ith component
d_s	=	Distance from satellite to Earth–Sun axis, km
d_c	=	Distance from satellite to Earth tangent plane at ground station, km
D	=	Downloaded data, Gb
G_t	=	Transmitter gain
I	=	Current, A
$\boldsymbol{J}$	=	Mass moment of inertia matrix, kg·m^2
$\boldsymbol{L}$	=	Angular momentum vector, kg·m^2/s
LOS_s	=	Satellite-to-Sun line of sight
LOS_c	=	Satellite-to-ground-station line of sight
$\boldsymbol{O}$	=	Orientation matrix
P	=	Power, W
$\dot{Q}$	=	Heat transfer rate, W

r = Position vector norm, km
$\vec{\boldsymbol{r}}$ = Position vector, km
SOC = Battery state of charge
T = Temperature, K
$\vec{\boldsymbol{v}}$ = Velocity vector, m/s
$\boldsymbol{v}$ = Vector of all variables (input, state, output)
$\boldsymbol{v}_i$ = Vector of variables corresponding to the ith component
$\boldsymbol{V}_i$ = Vector of functions corresponding to the ith component
γ = Satellite roll angle, rad
η_s = Sun line of sight transition parameter
τ = Torque, N·m
$\boldsymbol{\omega}$ = Angular velocity vector, 1/s

1 Introduction

Satellites serve a multitude of purposes that range from navigation and scientific research to military applications. Over the past decade, small satellites have gained increasing interest as alternatives to larger satellites because of the low time and cost required to manufacture and launch them. In particular, the CubeSat class of small satellites is becoming a common platform for education and research because it has a set of specifications that facilitates relatively frequent launches as secondary payloads.

CADRE (CubeSat investigating atmospheric density response to extreme driving) is funded by the National Science Foundation and will study the response of the Earth's upper atmosphere to auroral energy inputs [1]. This mission addresses the need for more accurate modeling of space weather effects, motivated in part by the growth of the global space-based infrastructure. To help answer some of the important scientific questions in this area, CADRE will provide critical *in situ* measurements in the ionospheric and thermospheric regions.

CADRE will inherit much of the design of the University of Michigan's Radio Aurora eXplorer (RAX) CubeSat. However, the unique scientific goals of the mission necessitate a detailed design study. Power is a driving factor because the scientific instruments are to run continuously, and large amounts of data must be transmitted to ground stations. Fortunately, there are several geometric and operational design variables whose impact can be captured with relatively inexpensive computational models, and it is possible to use these variables to satisfy the mission requirements while improving the satellite's performance. In the past, this has mostly been done via experience and human intuition aided by computational design tools that work with a relatively small number of design variables.

In the literature, there are many studies in which computational modeling and optimization have been applied to satellite design. For instance, Boudjemai et al. [2] performed topology optimization on the structure of a small satellite using NASTRAN for the finite element analysis. Numerical optimization has been applied to several other disciplines as well. Galski et al. [3] optimized a thermal control system, while Jain and Simon [4] implemented real-time load scheduling optimization of a small satellite's batteries. More recently, Richie et al. [5] and Zhang et al. [6] used optimization to size the energy storage and attitude control system and to design the layout of the satellite's components, respectively. All of these single-discipline optimization studies share a common approach: with the exception of the actuator sizing optimization, they use a genetic algorithm (GA) as a simple solution to deal with the discrete design variables and the discontinuities that are often present in the models.

Other authors considered multiple disciplines simultaneously to better model the overall physical problem. Barnhart et al. [7] implemented SPIDR, a systems-engineering-based framework for satellite design with an artificial-intelligence-based optimization algorithm that incorporates user-defined rules and constraints. Fukunaga et al. [8] developed OASIS, which uses a machine-learning algorithm to adaptively select and configure a metaheuristic optimizer such as a GA to optimize a model in MIDAS [9], a satellite design framework. SCOUT [10] is another framework that uses a GA for optimization, and ATSV [11] uses a shopping paradigm to aid the design process. Recently, Ebrahimi et al. [12] and Jafarsalehi et al. [13] developed multidisciplinary design frameworks that use a particle swarm optimizer (PSO) and a GA, respectively.

With the exception of the last two efforts, all of the computational design tools cited above have graphical

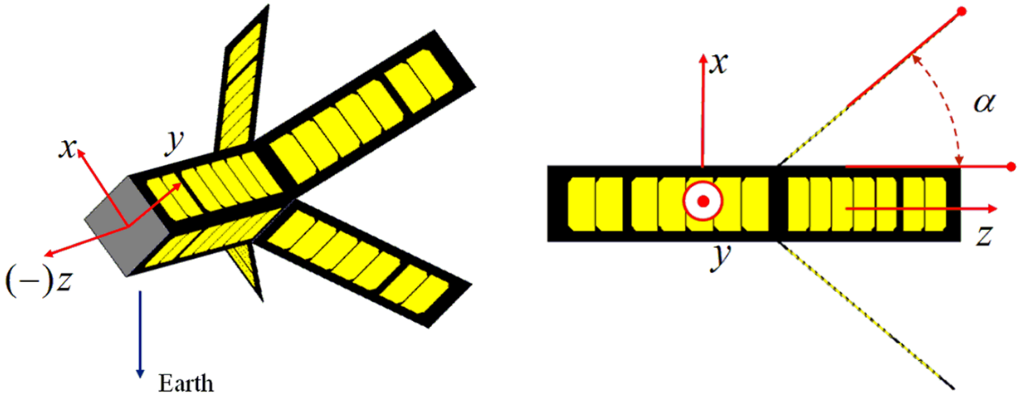


Figure 1: CADRE CubeSat geometry.

user interfaces (GUI) that significantly enhance usability. For these tools, the approach is to make user interaction with the framework as streamlined as possible, allowing the user's knowledge and experience to work together with the framework's optimization capability. However, as was the case with the single-discipline studies, all of these computational design tools use optimizers or design techniques that do not use gradients, which limits the number of design variables that can be considered. Without gradients, algorithms must rely on sampling the design space at a cost that grows exponentially with the number of design variables, and in practice, this becomes prohibitive when there are more than $\mathcal{O}(10)$ variables. Wu et al. [14] used a gradient-based approach to solve a satellite MDO problem with collaborative optimization (CO) [15, 16], but the cost of computing coupled derivatives limited the number of design variables to $\mathcal{O}(10)$ here as well.

Given the existing body of work, this paper seeks to address the question whether multidisciplinary design optimization (MDO) can handle the full set of design variables in the satellite design problem simultaneously, even when there are tens of thousands of them. The high-level approach is gradient-based optimization in combination with adjoint-based derivative computation, with a modular implementation of the disciplinary models in an integrated framework. The full small-satellite design problem is simultaneously considered, including all major disciplines, multiple time scales, and tens of thousands of design variables that parametrize the variation of several quantities over time.

The paper begins with a detailed description of CADRE and the design problem, which explains why such a large number of design variables is necessary. Next, we discuss the approach taken to implement and solve this large-scale MDO problem by listing all the challenges, as well as the measures taken to address them. Having established the background and context, we describe each disciplinary model, emphasizing those models that are original and have been developed specifically for this problem. Finally, we present optimization results that demonstrate the validity of the MDO approach advocated in this paper and its potential for small-satellite design.

2 The CADRE Design Problem

CADRE is a 3U CubeSat [1], meaning its body is a square prism with dimensions of 30 cm × 10 cm × 10 cm. As with other CubeSats, CADRE's dimensions are fixed so that it can be launched as a secondary payload with a larger satellite in order to reduce costs. The satellite has four fins that are initially folded at the sides of the satellite but are permanently deployed after launch in the rear direction to a preset angle. Although the roll angle is flexible, CADRE must always be forward-facing because of the scientific requirements, so the swept-back fins provide passive attitude stabilization through aerodynamic drag. CADRE has 12 solar panels with 7 cells each: 4 panels on the sides of the body, and one on the front and back of each fin. In general, the 84 cells are only partially illuminated because the Earth or another part of the satellite can cast shadows even when a cell is facing the Sun. Since the cells cover most of the satellite, it may be beneficial to install a radiator in place of one or more of the solar cells that are often shaded to provide cooling and to

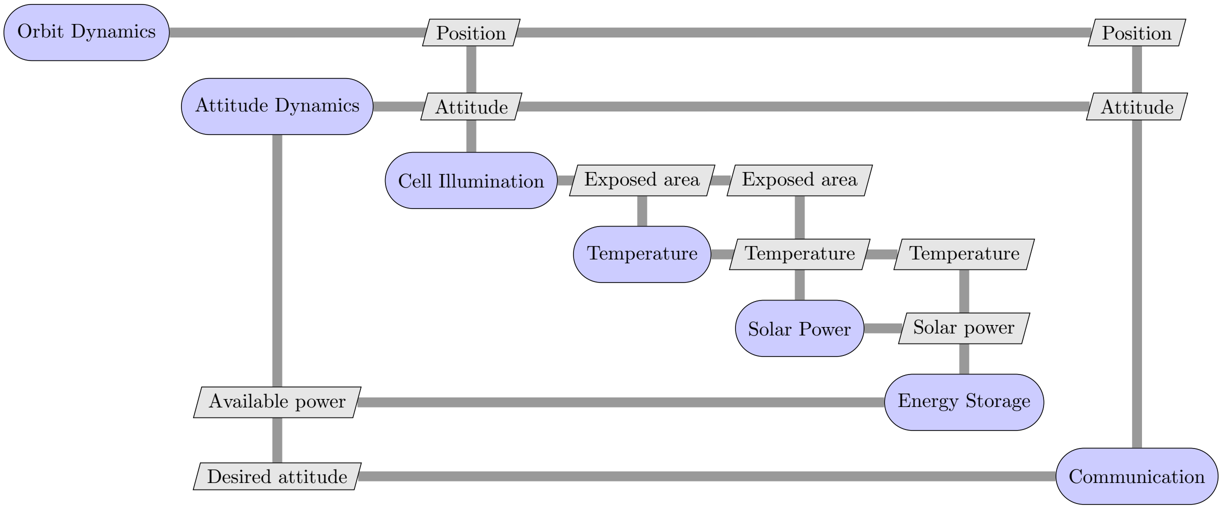


Figure 2: Extended design structure matrix (XDSM) diagram [17] showing all relevant disciplines in the CADRE design problem.

improve the power generation of other cells. A rendering of the CADRE CubeSat is shown in Fig. 1.

Other relevant subsystems include energy storage, communication, and attitude control. Lithium-ion batteries are installed with charge- and discharge-rate constraints, and a depth-of-discharge limit of 20% is enforced to lengthen the battery life. To transmit data to ground stations, an antenna is installed toward the rear of the satellite, and the installation angle is a parameter that can be varied, although it is constrained to be in the vertical plane. For the purposes of this paper, data transmission to ground stations is assumed to use a UHF antenna, although the final design for CADRE may use an S-band antenna for high-speed data download. CADRE uses two types of attitude-control actuators that complement each other: magnetorquers for gross changes, and reaction wheels for more precise control. With the latter, there is potential for the rotation rates of the wheels to accumulate and grow unmanageably large, so the magnetorquers are used to counteract constant torques such as that due to solar pressure. In this paper, only the reaction wheels are modeled to capture the power requirements of the desired attitude profiles, and the cost of counteracting disturbance torques is modeled as a constant background power consumption. Fig. 2 shows the disciplines and how they are coupled through the state variables.

CADRE's mission is to continuously collect data and transmit as much of that data as possible to the ground stations. Therefore, the total data downloaded is the natural objective function for the CADRE design optimization problem, although generating and storing sufficient energy is a driving factor. The fin and antenna angles are important geometric design variables, because they affect the power generation and the data-transmission rate, respectively. CADRE's attitude profile over time can be designed as well, providing further flexibility that can be used to increase power generation, cool panels when necessary, and increase transmission gain during communication with a ground station. The attitude profile must be optimized simultaneously with the geometric design variables because the fin and antenna angles that are optimal for an assumed attitude profile may no longer be optimal for an attitude profile that is optimized separately. The available power must also be optimally distributed between communication and actuation, so the power-distribution profile must be considered simultaneously as well.

Optimizing these profile variables involves manipulating 2-D curves without any *a priori* knowledge of their final optimal shapes. To do this, the curves must be discretized and parametrized, and in the simulation of hours, days, or even months of the satellite's operation, the resulting number of design variables can easily reach tens of thousands. To summarize, the objective of the CADRE design problem is to maximize the total data downloaded subject to constraints on the power and energy available, with respect to the fin angle, the antenna angle, the attitude profile, the communication power profile, and the 84 binary cell-installation variables.

3 Approach

The advantage of the gradient-based MDO approach taken in this paper is that it can handle a problem with many disciplines, design variables, and state variables. As a result, the true design problem can be optimized with few simplifications. This section discusses the approach used in this paper by listing each of the technical challenges and their solutions.

3.1 Large Number of Design Variables: Gradient-Based Optimization

Gradient-free optimizers do not perform well for problems with hundreds or thousands of design variables [18]. Gradient-free methods such as GAs and PSOs work with populations distributed over the design-variable range, but for a constant level of convergence, the number of required function evaluations grows exponentially.

When combined with analytic derivative computation, gradient-based optimization can be a powerful tool, because the adjoint method makes it possible to compute derivatives at a cost that is nearly independent of the number of design variables [19]. In addition, the number of iterations required for the optimization typically scales only linearly with sequential quadratic programming (SQP). In the aerospace field, the adjoint method has been very successful in aerodynamic shape optimization, initially with the approach of Jameson [20] and later with the optimization of full aircraft configurations [21–23]. It has also been successfully applied to the multidisciplinary optimization of aircraft aerodynamics and structures simultaneously using a coupled adjoint approach [24, 25].

SQP methods are among the best for constrained nonlinear optimization problems because an approximate Hessian of the Lagrangian is built from only first derivatives and the method is second-order convergent close to the optimum point. All the optimization problems in this paper are solved using SNOPT [26], through the pyOpt interface [18]. SNOPT is a reduced-Hessian active-set SQP optimizer that efficiently solves large-scale, sparse nonlinear constrained problems.

Gradient-based optimizers can never guarantee convergence to the global optimum. However, the global optimum is not a realistic goal given the number of design variables in this problem, because many local minima are expected. Even as a local optimizer, the gradient-based MDO algorithm is still useful because it is able to find a design for CADRE better than that found using experience and human intuition. The argument is that a local optimum for a problem that closely represents reality may be more useful than the global optimum of a problem based on lower fidelity models.

3.2 Large Number of Disciplines: New MDAO Formulation

From an implementation perspective, the large number of disciplines presents one of the most significant challenges of this work. Each discipline is decomposed into separate computations to simplify and modularize the code, but this results in a large number of components. For instance, the output of the communication discipline is ultimately the total data downloaded, but it is broken down into the computations of the ground station line-of-sight variable, the position vector from the satellite to the ground station, the transmitter gain as a function of this vector, the data-download rate, and the total data downloaded. In total, there are 43 components when those of all the disciplines are combined, and Fig. 3 shows the dependencies to illustrate the scope and complexity of this problem. In the figure, the (i, j)th entry represents the variable(s) computed by the ith component and used as an argument by the jth component. The design structure matrix is upper triangular because the problem has been formulated to remove feedback, as will be discussed in Sec. 5. 5.1.

Programming all of these components, accounting for all the dependencies between variables, and correctly programming their derivatives is a long and error-prone process. We implemented all the components within a framework that uses a new multidisciplinary design, analysis, and optimization (MDAO) formulation [27].

This formulation simplifies the task by defining the multidisciplinary analysis problem mathematically as a nonlinear system of equations. The design, state, intermediate, input, and output variables are a subset of the unknowns in this nonlinear system, and the objective, constraint, residual, and other functions are formulated as equations. This is accomplished by classifying each variable as an *independent* variable whose value can be set, an *explicit* variable, or an *implicit* variable whose value is the root of an equation. Design variables and parameters are independent; behavior variables and other intermediate variables computed by evaluating

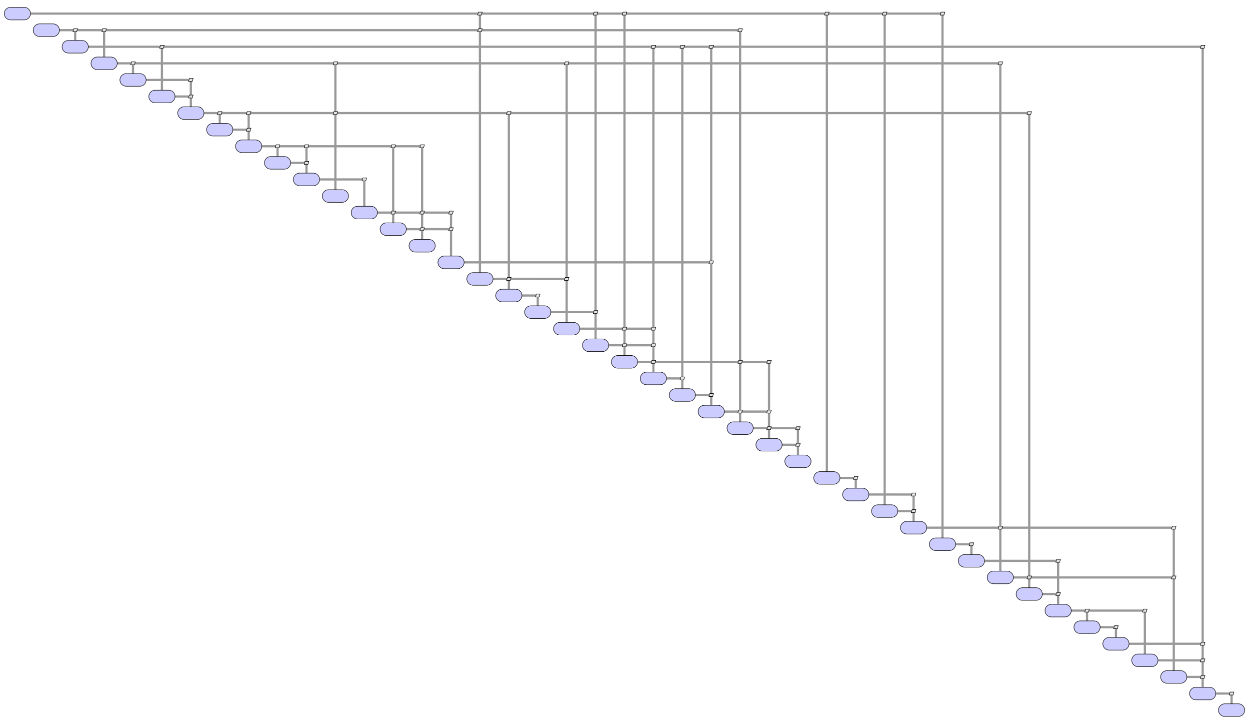


Figure 3: Design structure matrix diagram illustrating the complexity of the problem.

Type	ith Variable	Corresponding constraint
Independent	$v_i = v_i^*$	$C_i(\boldsymbol{v}) = v_i - v_i^*$
Explicit	$v_i = V_i(v_1, \ldots, v_{j \neq i}, \ldots, v_n)$	$C_i(\boldsymbol{v}) = v_i - V_i(v_1, \ldots, v_{j \neq i}, \ldots, v_n)$
Implicit	$v_i : R_i(v_1, \ldots, v_{j \neq i}, \ldots, v_n, v_i) = 0$	$C_i(\boldsymbol{v}) = -R_i(v_1, \ldots, v_{j \neq i}, \ldots, v_n, v_i)$

Table 1: Classification of variables in the MDAO formulation.

an expression (no matter how complicated) are explicit; and state variables that are computed by solving a linear or nonlinear system of equations are implicit. Thus, the input variables for the multidisciplinary analysis are independent, while the output variables, including the optimization objectives and constraints, are either explicit or implicit, typically the former.

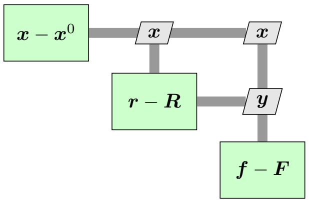
We define for each variable a corresponding constraint function that yields zero when the variable has converged and has the value given by its definition, as shown in Table 1. In the remainder of this section, upper-case symbols designate functions and lower-case symbols designate variables, e.g., $\boldsymbol{c}$ is the vector of the output variables of $\boldsymbol{C}$.

The variables are concatenated into a single vector, $\boldsymbol{v}$, and the constraint vector is denoted $\boldsymbol{C}$. This formulation effectively defines the multidisciplinary analysis problem mathematically as the nonlinear system $\boldsymbol{C}(\boldsymbol{v}) = \boldsymbol{0}$. Since $\partial \boldsymbol{C} / \partial \boldsymbol{v}$ is always invertible in practice, the inverse function theorem states that the inverse of the Jacobian of $\boldsymbol{C}$ is the Jacobian of the inverse function of $\boldsymbol{C}$, yielding the result [19]:

$$\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}} \frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}} = \boldsymbol{\mathcal{I}} = \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}^T \frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}^T . \tag{1}$$

Methods resulting from the left equality are described as forward mode and those from the right equality as reverse mode. For example, the two modes of algorithmic differentiation as well as the direct and adjoint methods correspond to one of these two modes. This equation unifies all known methods for computing numerical derivatives; more details can be found in Martins and Hwang [19]. Given accurate partial derivatives, the equality on the right yields numerically exact partial derivatives at a cost that is nearly independent of

Variables and Constraints


$\boldsymbol{x}-\boldsymbol{x}^0$
$\boldsymbol{x}$
$\boldsymbol{x}$
$\boldsymbol{r}-\boldsymbol{R}$
$\boldsymbol{y}$
$\boldsymbol{f}-\boldsymbol{F}$

$$\boldsymbol{v}=\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\\ \boldsymbol{f}\end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{v})=\begin{bmatrix}\boldsymbol{x}-\boldsymbol{x}^0\\ \boldsymbol{r}-\boldsymbol{R}(\boldsymbol{x},\boldsymbol{y})\\ \boldsymbol{f}-\boldsymbol{F}(\boldsymbol{x},\boldsymbol{y})\end{bmatrix}$$

Derivation

$$\left[\frac{\partial\boldsymbol{C}}{\partial\boldsymbol{v}}\right]\left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right]=\boldsymbol{I}=\left[\frac{\partial\boldsymbol{C}}{\partial\boldsymbol{v}}\right]^T\left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right]^T$$

$$\begin{bmatrix}\frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)}{\partial\boldsymbol{x}} & \frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)}{\partial\boldsymbol{y}} & \frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)}{\partial\boldsymbol{f}}\\ \frac{\partial(\boldsymbol{r}-\boldsymbol{R})}{\partial\boldsymbol{x}} & \frac{\partial(\boldsymbol{r}-\boldsymbol{R})}{\partial\boldsymbol{y}} & \frac{\partial(\boldsymbol{r}-\boldsymbol{R})}{\partial\boldsymbol{f}}\\ \frac{\partial(\boldsymbol{f}-\boldsymbol{F})}{\partial\boldsymbol{x}} & \frac{\partial(\boldsymbol{f}-\boldsymbol{F})}{\partial\boldsymbol{y}} & \frac{\partial(\boldsymbol{f}-\boldsymbol{F})}{\partial\boldsymbol{f}}\end{bmatrix}\begin{bmatrix}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{r}} & \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{f}}\\ \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{r}} & \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{f}}\\ \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}} & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{f}}\end{bmatrix}=\boldsymbol{I}=\begin{bmatrix}\frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)^T}{\partial\boldsymbol{x}} & \frac{\partial(\boldsymbol{r}-\boldsymbol{R})^T}{\partial\boldsymbol{x}} & \frac{\partial(\boldsymbol{f}-\boldsymbol{F})^T}{\partial\boldsymbol{x}}\\ \frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)^T}{\partial\boldsymbol{y}} & \frac{\partial(\boldsymbol{r}-\boldsymbol{R})^T}{\partial\boldsymbol{y}} & \frac{\partial(\boldsymbol{f}-\boldsymbol{F})^T}{\partial\boldsymbol{y}}\\ \frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)^T}{\partial\boldsymbol{f}} & \frac{\partial(\boldsymbol{r}-\boldsymbol{R})^T}{\partial\boldsymbol{f}} & \frac{\partial(\boldsymbol{f}-\boldsymbol{F})^T}{\partial\boldsymbol{f}}\end{bmatrix}\begin{bmatrix}\frac{\mathrm{d}\boldsymbol{x}^T}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{y}^T}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{x}}\\ \frac{\mathrm{d}\boldsymbol{x}^T}{\mathrm{d}\boldsymbol{r}} & \frac{\mathrm{d}\boldsymbol{y}^T}{\mathrm{d}\boldsymbol{r}} & \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{r}}\\ \frac{\mathrm{d}\boldsymbol{x}^T}{\mathrm{d}\boldsymbol{f}} & \frac{\mathrm{d}\boldsymbol{y}^T}{\mathrm{d}\boldsymbol{f}} & \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{f}}\end{bmatrix}$$

$$\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0}\\ -\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{x}} & -\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{y}} & \boldsymbol{0}\\ -\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}} & -\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{y}} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boxed{\begin{matrix}\boldsymbol{I}\\ \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}}\\ \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}\end{matrix}} & \begin{matrix}\boldsymbol{0}\\ \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{r}}\\ \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}}\end{matrix} & \begin{matrix}\boldsymbol{0}\\ \boldsymbol{0}\\ \boldsymbol{I}\end{matrix}\end{bmatrix}=\boldsymbol{I}=\begin{bmatrix}\boldsymbol{I} & -\frac{\partial\boldsymbol{R}^T}{\partial\boldsymbol{x}} & -\frac{\partial\boldsymbol{F}^T}{\partial\boldsymbol{x}}\\ \boldsymbol{0} & -\frac{\partial\boldsymbol{R}^T}{\partial\boldsymbol{y}} & -\frac{\partial\boldsymbol{F}^T}{\partial\boldsymbol{y}}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\begin{matrix}\boldsymbol{I}\\ \boldsymbol{0}\\ \boldsymbol{0}\end{matrix} & \begin{matrix}\frac{\mathrm{d}\boldsymbol{y}^T}{\mathrm{d}\boldsymbol{x}}\\ \frac{\mathrm{d}\boldsymbol{y}^T}{\mathrm{d}\boldsymbol{r}}\\ \boldsymbol{0}\end{matrix} & \boxed{\begin{matrix}\frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{x}}\\ \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{r}}\\ \boldsymbol{I}\end{matrix}}\end{bmatrix}$$

Direct method

$$\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{y}}\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}}=-\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{x}}$$

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}=\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}}+\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{y}}\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}}$$

Adjoint method

$$\frac{\partial\boldsymbol{R}^T}{\partial\boldsymbol{y}}\frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{r}}=-\frac{\partial\boldsymbol{F}^T}{\partial\boldsymbol{y}}$$

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}=\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}}+\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}}\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{x}}$$

Figure 4: Derivation of the adjoint method from Eq. (1) [19].

the number of design variables. As an illustration, Fig. 4 shows how the adjoint method can be derived from Eq. (1) with the appropriate definition of the vector $\boldsymbol{v}$. The nomenclature used in this figure is consistent with the source of the figure [19], but it does not apply to the rest of this paper. In Fig. 4, $\boldsymbol{x}$ is the design or input variable vector, $\boldsymbol{y}$ is the state variable vector, and $\boldsymbol{f}$ represents the output variables.

Figure 4 also helps explain the entries of the $\mathrm{d}\boldsymbol{v}/\mathrm{d}\boldsymbol{c}$ matrix, the interpretation of which may be unclear. This matrix is the Jacobian of the local inverse of the $\boldsymbol{C}$ function at the solution of $\boldsymbol{C}(\boldsymbol{v})=\boldsymbol{0}$, so a column $\mathrm{d}\boldsymbol{v}/\mathrm{d}c_i$ is the vector of perturbations to $\boldsymbol{v}$ that solves the nonlinear system with a small perturbation added to the ith constraint. The figure illustrates the meaning of the derivatives of input variables such as $\mathrm{d}\boldsymbol{x}/\mathrm{d}\boldsymbol{f}$; since $\boldsymbol{f}$ has neither an explicit dependence on $\boldsymbol{x}$ nor an implicit one through $\boldsymbol{y}$, these derivatives are zero. It also illustrates the meaning of the derivatives with respect to the residuals: a column of $\mathrm{d}\boldsymbol{y}/\mathrm{d}\boldsymbol{r}$ represents the vector of perturbations to $\boldsymbol{y}$ that comes about by adding a small perturbation to one of the residuals.

In the second equality of Eq. (1), solving for the column of the transpose of $\mathrm{d}\boldsymbol{v}/\mathrm{d}\boldsymbol{c}$ corresponding to a desired objective or constraint function, for instance f, yields the full vector of the derivatives of f, a subset of which is $\mathrm{d}f/\mathrm{d}\boldsymbol{x}$. Thus, one of the gradients needed during an optimization iteration can be computed at the cost of just a single solution of a linear system.

Based on this formulation of the problem, it is possible to automatically solve the multidisciplinary system and compute the coupled derivatives. To implement a component, we simply define methods for computing $\boldsymbol{C}_i$, $\partial\boldsymbol{C}_i/\partial\boldsymbol{v}$ and its transpose, and the preconditioner for $\partial\boldsymbol{C}_i/\partial\boldsymbol{v}_i$ and its transpose, and we implement a component-level solver that computes $\boldsymbol{v}_i : \boldsymbol{C}_i(\boldsymbol{v})=\boldsymbol{0}$. Since feedback has been removed from the current problem, as shown in Fig. 3, a system-level solver can then solve the nonlinear system by running a single iteration of the nonlinear block Gauss–Seidel method. It can also solve the linear system in Eq. (1) using a single iteration of the linear block Gauss–Seidel method to obtain the desired vector of derivatives. For the optimization problem presented in this paper, we must find 31 linear solutions of the second equality in Eq. (1)—one for each objective and constraint function—where the right-hand side of the linear system is the ith column of the identity matrix, and i is the index of the output variable of interest in the global $\boldsymbol{v}$ vector.

Alhough coupled derivatives are computed in the framework using (1), each component must be able to compute the partial derivatives of any variables it computes with respect to any variables on which it depends. The simplest option is finite-differencing, either column-by-column or as a directional derivative. However, the accuracy of finite differences is limited because subtractive cancellation error dominates at small step sizes, and the error term in the first-order Taylor expansion becomes significant at larger step sizes. The complex-step method avoids this issue since there is no subtraction operation in the formula [19, 28]. However, the cost of differentiation is proportional to the number of variables, which is of the order of thousands in this case. Algorithmic differentiation is another option with an error theoretically on the order of machine precision, but the cost is proportional to either the number of outputs or the number of inputs, which are both large in this problem [19].

For these reasons, each component has been analytically differentiated. This approach yields numerically exact derivatives and can be more efficient than the other methods. Furthermore, many of the Jacobians are large (thousands by thousands) but sparse. These sparse matrices are assembled directly in Fortran very efficiently.

3.3 Multiple Time Scales: Multi-point Optimization

The CADRE design problem involves multiple time scales. Capturing CADRE's power generation and temperature fluctuations requires a time-resolution of $\mathcal{O}(5\,\text{min})$ because its orbit has a period of roughly 90 min. Ground-station passes last $\mathcal{O}(10\,\text{min})$ and energy must be stored between sets of ground-station passes, which occur in patterns that roughly repeat each day. Assuming that the ground stations are close to the Equator, this requires a resolution of $\mathcal{O}(1\,\text{min})$ with a simulation of at least 12 h of the satellite's operation. However, depending on the launch orbit, CADRE's orbit may precess multiple times per year. This, combined with the effect of the seasons, requires a simulation of one year to model one period of oscillations in the satellite's operating conditions.

This multi-scale characteristic, combined with the ambitious scope of the design problem, presents a significant challenge. For a truly unbiased simulation, the year must be simulated with a resolution of 1 min, yielding $262,800$ discrete points. If a shorter period of time is simulated, the resulting design may be optimal in one season but not others. In some seasons the satellite may have difficulty generating sufficient power because the solar cells see much less of the Sun at the chosen fin angle.

The periodic nature of many of the variables suggests a frequency-domain approach for the state and design variables to reduce the size of the model and the optimization problems. Such an approach would capture the oscillatory behavior of the variables with a relatively small number of degrees of freedom. However, we did not adopt this approach for two reasons. First, many of the state variables, such as the Sun line-of-sight variable, have near-discontinuous jumps that cannot be accurately represented with a small number of frequencies. These effects propagate to other variables, such as the temperature, solar power, and battery current, as similar discontinuities or as nonsmoothness. Second, while other state variables do behave smoothly for the most part, they tend to have one or more spikes due to ground-station passes. The transmitter gain, battery current, and temperature are examples of variables that exhibit such spikes, although for temperature these are on a relatively small scale. These high-frequency components could potentially be represented by additional modes, but they would require *a priori* knowledge of where the passes are, and some of the automation in the computational tool would be lost. Nonetheless, it is worth noting for future work that representing some of the state variables in the frequency domain and others in the time domain would no doubt reduce the size of the problem.

Our solution is to simulate six 12-hour blocks with 0.5 min resolution, distribute them uniformly over the year, and weight each one equally in a single optimization problem. The orbit and communication time scales are captured within the 12-hour blocks, and simulating half a day every two months captures the orbit-precession time scale. The optimization constraints are applied separately to each block, and the objective functions computed from the six blocks are averaged. This approach is essentially multi-scale, multi-point optimization; the minute-level time scale is directly simulated, and the objective function for the month-level time scale is numerically integrated using the midpoint rule.

3.4 Large Number of Constraints: Constraint Aggregation

As previously mentioned, the coupled-derivative equations compute the gradients efficiently because they give either the full vector of derivatives with respect to a single variable (forward mode of (1)) or the full gradient of a variable (reverse mode of (1)) using a single solution of a linear system. Since our optimization has a large number of design variables, the reverse mode must be used, but it requires a linear solution for each constraint. Moreover, the battery discipline requires four inequality constraints at each time instance—maximum charge rate, maximum discharge rate, minimum state of charge, and maximum state of charge—resulting in tens of thousands of constraints.

We use constraint aggregation to reduce the number of constraints. The Kreisselmeier–Steinhauser (KS) function [29] aggregates the constraints over all the time instances into a single criterion. Constraint aggregation with KS functions has been shown to work well in combination with the adjoint method in optimization problems, for instance in structural weight minimization [30] and aerostructural optimization [31, 32]. The KS function is given by

$$KS(x) = f_{i_{\max}}(x) + \frac{1}{\rho} \ln \sum_i e^{\rho(f_i(x) - f_{i_{\max}}(x))}, \tag{2}$$

where f_i is the i^{th} function in the vector of functions we wish to aggregate, i_{max} is the index of the function with the largest value at x, and ρ is a parameter that is problem-dependent. In the limit, as ρ approaches infinity, the KS function approaches the maximum function because $e^{\rho \cdot 0}$ dominates in the sum, and $KS(x)$ approaches $f_{i_{\max}}(x)$. For finite ρ, the KS function is a smooth function that is dominated by the f_i with the largest values. Thus, as an inequality constraint, the KS function encourages the optimizer to resolve the largest infeasibilities first and eventually choose a point at which the KS function itself is less than or equal to zero. The optimization problems solved in this paper use $\rho = 50$, a value which was found through numerical tests.

3.5 Nondifferentiable Models: B-spline Interpolant

Often, a discipline has a model that cannot be differentiated. The reason could be that the underlying physical phenomenon is nonsmooth, the computational model is a legacy code without source code access, or only a table of data is available. To address these situations, we implemented in Fortran a tensor-product B-spline interpolant with analytic derivatives. A model with any number of input variables can be fitted with this interpolant given a structured array of data that spans the full range of values for the input variables.

3.6 Derivatives of ODE Variables: Modular Runge–Kutta Solver

Several of the disciplines in the CADRE design problem involve ordinary differential equations (ODEs), which complicates the task of computing partial and total derivatives. In particular, ODEs in time have a natural forward direction, so the unknown variable depends only on those before it in time, but the reverse mode of (1) must compute the total derivatives in the opposite direction. This is not possible if the values from previous time instances are discarded as the algorithm moves forward, so the CADRE MDO algorithm explicitly keeps track of the full time series as a vector and operates on the entries of this vector in sequence. Furthermore, the fourth-order Runge–Kutta method (RK4) has been implemented in Fortran as a modular solver with the time-marching scheme differentiated. For each discipline that uses this modular RK4 solver, only the derivatives of the ODE must be provided; correct indexing and application of the chain rule to combine these with the partial derivatives of the RK4 equations are automatically handled.

4 Discipline Models

This section describes the models for all the disciplines in the CADRE MDO algorithm. For vectors, the nomenclature used in this section is as follows. Upper-case subscripts represent the frames of reference: B, R, E, and I represent the body-fixed frame, rolled body-fixed frame (explained later), Earth-fixed frame, and Earth-centered inertial (ECI) frame, respectively. Lower-case subscripts represent the origins of frames:

b, e, g, and s denote the body (satellite), Earth, ground station, and Sun, respectively. For instance, $\vec{\boldsymbol{r}}_{b/e}$ signifies a vector pointing from the Earth's origin to the satellite's origin. The axes of the body-fixed frame are denoted $\hat{\boldsymbol{\imath}}_B$, $\hat{\boldsymbol{\jmath}}_B$, and $\hat{\boldsymbol{k}}_B$. The orientation matrices are represented by $\boldsymbol{O}$, e.g., $\boldsymbol{O}_{B/I}$ represents the orientation of the body-fixed frame as seen in the ECI frame.

4.1 Orbit Dynamics

The orbit-dynamics discipline computes the Earth-to-body position vector in the ECI frame. In the orbit equation, there are terms that represent the fact that the Earth is not a perfectly spherical and homogeneous mass. These are captured in the J_2, J_3, and J_4 coefficients in the following equation[1]:

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

The values of the coefficients are listed in Table 2.

The J_2, J_3, and J_4 terms must be considered because their effect is to rotate the orbit plane on a scale of months. If they are ignored, a fin angle that may initially increase power generation may no longer be optimal after the orbit plane has changed. This also makes the CADRE design problem a multi-scale problem in time because much of the system's behavior occurs on the scale of minutes and hours, since the period of the satellite's orbit is roughly 90 minutes. The slow rotation of the orbit plane affects the power generation and communication as well, since the satellite's trajectory affects how much data can be transmitted as it passes over ground stations. We solve the orbit equation using the modular RK4 solver described earlier.

4.2 Attitude Dynamics

Because of the requirements for scientific data collection, CADRE must always have a forward-facing orientation. The roll angle, γ, can change provided the maximum rate of 1 rad/min is not exceeded. The optimizer controls the roll-angle profile over time, and all the other attitudes, torques, and related quantities are computed from this. Since all the time instances are modeled simultaneously, this approach is equivalent to determining the control inputs using optimization instead of a controller.

At any given time instance, CADRE's attitude is determined by applying the rotations from the ECI frame to what is referred to here as the *rolled* frame and then to the actual body-fixed frame. The rolled frame is an intermediate frame obtained after ensuring that CADRE is forward-facing but prior to applying the appropriate rotation from the specified roll-angle profile. For this frame, $\hat{\boldsymbol{k}}_B$ must point in the opposite direction to $\hat{\boldsymbol{v}}_{b/e}$, and the chosen convention is that $\hat{\boldsymbol{\jmath}}_B$ is parallel to $\hat{\boldsymbol{r}}_{b/e}$. The orientation matrices that implement these two successive rotations are given by

$$\boldsymbol{O}_{R/I} = \begin{bmatrix}\hat{\boldsymbol{\imath}}_B^T \\ \hat{\boldsymbol{\jmath}}_B^T \\ \hat{\boldsymbol{k}}_B^T\end{bmatrix} = \begin{bmatrix}-(\hat{\boldsymbol{r}}_{b/e}\times\hat{\boldsymbol{v}}_{b/e})^T \\ \hat{\boldsymbol{r}}_{b/e}^T \\ -\hat{\boldsymbol{v}}_{b/e}^T\end{bmatrix} \qquad \text{and} \qquad \boldsymbol{O}_{B/R} = \begin{bmatrix}\cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1\end{bmatrix}. \tag{4}$$

Once the $\boldsymbol{O}_{B/I}$ matrix is known for all time instances, its time derivative can be computed using finite differences, and the angular-velocity vector can be computed using $\vec{\boldsymbol{\omega}}_B^{\times} = \dot{\boldsymbol{O}}_{B/I}\cdot\boldsymbol{O}_{B/I}^T$.

As mentioned previously, we model only the reaction wheel for actuation. The required inputs are computed from the satellite's angular-velocity profile. We do this by applying conservation of angular momentum to the satellite and reaction-wheel system, expressed by setting the time derivative of the total angular momentum to zero:

$$\dot{\vec{\boldsymbol{L}}} = \underbrace{\boldsymbol{J}_B\cdot\dot{\vec{\boldsymbol{\omega}}}_B + \vec{\boldsymbol{\omega}}_B\times(\boldsymbol{J}_B\cdot\vec{\boldsymbol{\omega}}_B)}_{\vec{\boldsymbol{\tau}}_B} + \underbrace{\boldsymbol{J}_{RW}\cdot\dot{\vec{\boldsymbol{\omega}}}_{RW}}_{\vec{\boldsymbol{\tau}}_{RW}} + \vec{\boldsymbol{\omega}}_B\times(\boldsymbol{J}_{RW}\cdot\vec{\boldsymbol{\omega}}_{RW}) = 0. \tag{5}$$

Computing the required reaction-wheel torque is a three-step process. First, we can compute $\vec{\boldsymbol{\tau}}_B$ since $\vec{\boldsymbol{\omega}}_B$ is known and its time derivative can again be computed using finite differences. Next, we solve the

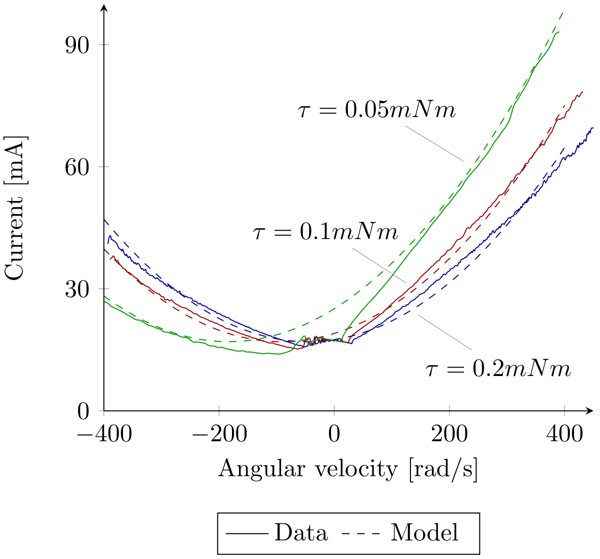


Figure 5: Reaction-wheel model compared with manufacturer-provided data for three torques.

resulting ODE to determine $\vec{\boldsymbol{\omega}}_{RW}$ over time, and finally we can compute $\vec{\boldsymbol{\tau}}_{RW}$ when the reaction wheels' angular-velocity profiles are known. The mass moment of inertia matrices for the satellite and reaction wheels are, respectively,

$$\boldsymbol{J}_B = \begin{bmatrix} 18 & 0 & 0 \\ 0 & 18 & 0 \\ 0 & 0 & 6 \end{bmatrix} \times 10^{-3} \text{ kg m}^2 \qquad \text{and} \qquad \boldsymbol{J}_{RW} = \begin{bmatrix} 28 & 0 & 0 \\ 0 & 28 & 0 \\ 0 & 0 & 28 \end{bmatrix} \times 10^{-6} \text{ kg m}^2. \tag{6}$$

Based on the manufacturer's data[2], we develop a simple equation to model the dependence of the reaction wheel's current draw on its angular velocity and desired torque:

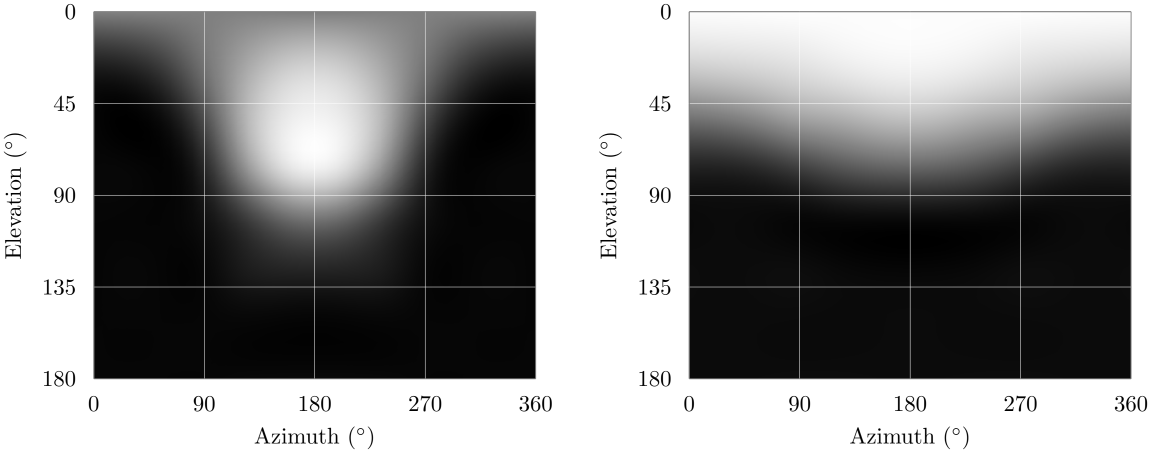
$$I = (a\omega + b\tau)^2 + I_0. \tag{7}$$

Given the right coefficients, this simplified model correctly captures the trends, as shown in Fig. 5. When both the angular velocity and desired torque are zero, there is a constant baseline current draw. As the angular velocity increases in either direction, the current draw increases roughly quadratically with the torque constant. However, the behavior is asymmetric, since we need less power to achieve a torque in the opposite direction of the angular velocity, which amounts to slowing down the wheel with the assistance of friction. This effect is reflected in both the actual data and the model, as shown in Fig. 5. The motor is assumed to run at 4 V.

4.3 Cell Illumination

The cell-illumination discipline models the area of each solar cell that is exposed to the Sun, projected onto the plane normal to the Sun's incidence. The 84 exposed areas depend on the fin angle as well as the azimuth and elevation angles of the Sun in the body-fixed frame. We compute these using an OpenGL model of the geometry, in which the satellite is discretized into small rectangles.

Since this model is both discontinuous and difficult to incorporate into the framework, we generate a table of data, and we use the B-spline multi-dimensional interpolant mentioned in Sec. 3. 3.5. to provide an approximation of the exposed areas in terms of the three parameters. This also has the effect of smoothing the areas since the B-spline interpolant does not have a sufficient number of control points to capture the discontinuous jumps, but it does have the degrees of freedom to follow the general trends. Figure 6 shows



(a) Fin angle: 30°

(b) Fin angle: 80°

Figure 6: Normalized exposed area as a function of relative Sun position for the outermost cell in an inward-facing panel.

the variation in the exposed area as a function of the Sun's position for the outermost cell in one of the inward-facing panels.

The line-of-sight variable, LOS_s, is essentially a multiplier for the exposed areas: it is 0 if the satellite is behind the Earth and 1 otherwise. To smooth this discontinuous jump, we assume that the sunlight does not decrease instantaneously as the satellite moves into the Earth's shadow, but instead, smoothly transitions to zero. This is physically the case to a certain extent because of the umbra and penumbra effects, but it is greatly exaggerated to avoid numerical difficulties in the optimization. The procedure used to compute LOS_s is illustrated in Fig. 7, and it is defined by

$$LOS_s = \left\{ \begin{matrix} 1 & , \vec{r}_{b/e} \cdot \hat{r}_{s/e} \geq 0 \\ \left\{ \begin{matrix} 1 & , & d_s > R_e \\ 3\eta^2 - 2\eta^3 & , & \alpha R_e < d_s < R_e \\ 0 & , & d_s < \alpha R_e \end{matrix} \right\} & , \vec{r}_{b/e} \cdot \hat{r}_{s/e} < 0 \end{matrix} \right\}, \tag{8}$$

where d_s and η are given by

$$d_s = ||\vec{r}_{b/e} \times \hat{r}_{s/e}||_2 \qquad \text{and} \qquad \eta = \frac{d_s - \alpha R_e}{R_e - \alpha R_e}. \tag{9}$$

Mathematically, we construct a simple cubic function between $d_s = \alpha R_e$ and $d_s = R_e$, satisfying C^1 continuity at each end point. The value of α represents how far this smoothing effect extends into the Earth's shadow; a typical value is $\alpha = 0.9$.

4.4 Temperature

Temperature is an important consideration that couples many disciplines: it affects solar power generation and battery performance, while both cell illumination and data transmission generate heat. The temperature is assumed to be uniform within each of the fins and the body, so there are five temperature state variables at each time instance. We use the Stefan–Boltzmann law to model the rate of heat radiation, and we use the area exposed to the Sun to compute each cell's contribution to the heating of its fin. Since communication power amplifies data transmission with an efficiency, η_p, of roughly 20%, we assume that the remaining 80%

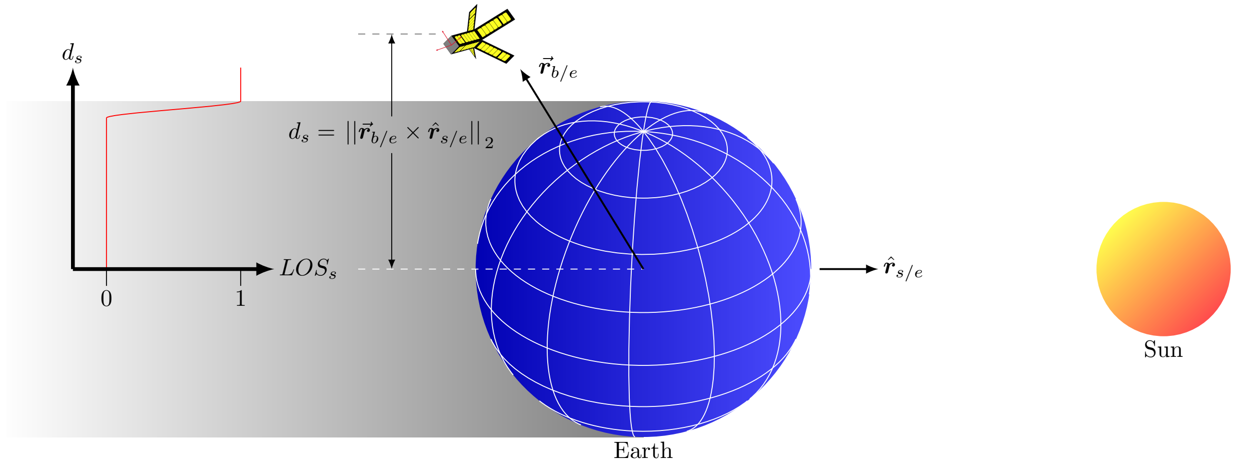

d_s
LOS_s
0
1
$d_s = ||\vec{\boldsymbol{r}}_{b/e} \times \hat{\boldsymbol{r}}_{s/e}||_2$
$\vec{\boldsymbol{r}}_{b/e}$
$\hat{\boldsymbol{r}}_{s/e}$
Earth
Sun

Figure 7: Illustration of the Sun line-of-sight variable.

is converted to heat, which contributes to the temperature ODE for the body. The equations are

$$\dot{T} = \frac{\dot{Q}_{in} - \dot{Q}_{out} + \dot{Q}^*_{comm}}{mc_v} \tag{10}$$

$$\dot{Q}_{in} = \alpha q_{sol} A_{exp} LOS_s \tag{11}$$

$$\dot{Q}_{out} = \epsilon \left(\frac{2\pi^5 k^4}{15c^2 h^3} \right) T^4 A_T \tag{12}$$

$$\dot{Q}_{comm} = (1 - \eta_p) P_{comm}, \tag{13}$$

where A_{exp} is the exposed area of the cell, T is the temperature, and $\dot{Q}$ is the rate of heat transfer. The values for all the constants are listed in Table 2.

4.5 Solar Power

The cells in each solar panel are connected in series, so their output voltages are added to compute the total voltage for the panel. The voltage is set so as to maximize the power output, but the optimal voltage, and thus the optimal current, changes depending on the illumination and temperature of the cells.

Each cell has a unique I-V curve that depends on its exposed area and temperature. Our model is based on one [33] that is a nonlinear implicit equation in I given by

$$I = I_{sc} - I_{sat} \left[\exp\left\{ \frac{V + R_s I}{V_T} \right\} - 1 \right] - \frac{V + R_s I}{R_{sh}}, \tag{14}$$

where the values of the constants are listed in Table 2, and

$$I_{sc} = LOS_s \frac{A_{exp}}{A_T} I_{sc_0} \qquad \text{and} \qquad V_T = \frac{nkT}{q}. \tag{15}$$

Our model has two modifications. First, the series resistance is very small, so the two terms containing R_s can be neglected. Second, a diode is used to limit the voltage in the negative region to $V_0 = -0.6\,\mathrm{V}$, so a shifted hyperbolic tangent function is used to model the I-V curve for negative voltages. We determine the coefficient in the argument of tanh by applying the constraint that the first derivative must be continuous at $V = 0$. Since V is still an implicit function of I in the positive voltage region, we evaluate the model for the full ranges of areas and temperatures, and we fit the B-spline interpolant discussed earlier. The model is plotted in Fig. 8 and the expressions are

$$\left\{ \begin{array}{l} I_{sc} - I_{sat} \left[\exp\left\{ \frac{V}{V_T} \right\} - 1 \right] - \frac{V}{R_{sh}} - I = 0 \,, I \le I_{sc} \\ V(I) = V_0 \tanh\left[\frac{-V_T R_{sh}}{V_0 (I_{sat} R_{sh} + V_T)} (I - I_{sc}) \right] , I > I_{sc} \end{array} \right\}. \tag{16}$$

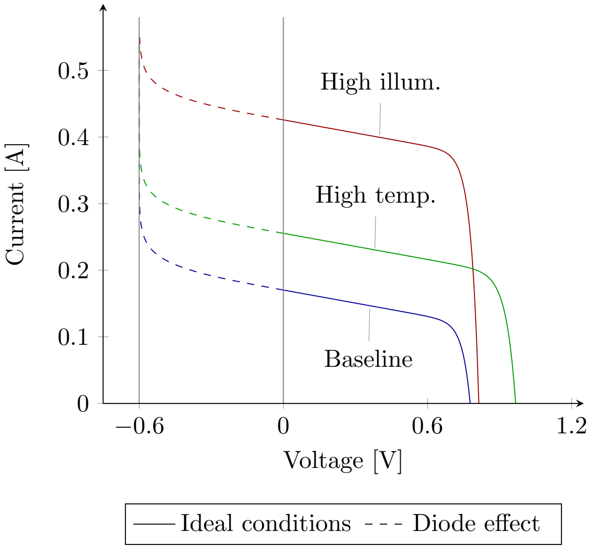


Figure 8: Solar cell I-V curve at different cell temperatures and exposed areas.

4.6 Energy Storage

The energy-storage discipline tracks the state of charge (SOC) of the battery. We can compute the SOC by integrating the nonlinear ODE given by

$$S\dot{O}C = \frac{P_{bat}}{V_{bat}Q}, \tag{17}$$

where Q is the nominal discharge capacity of the battery.

We model the dependence of the voltage on the SOC as an exponential, primarily to ensure that the voltage is always positive. A linear relationship would have been within the scope of this work. However, a battery at a large negative SOC has a negative voltage, and drawing power from the battery would increase its SOC since the current is still positive. Negative states of charge often arise at the initial point in an optimization, when a poor baseline design point uses more power than is available. In these circumstances, the model must provide the optimizer with the correct gradient directions instead of failing. Artificially removing the drop-off in voltage does not lead to inaccuracies that affect our results, since the optimization constrains the SOC to be nonnegative, which ensures that the optimal design is never in this drop-off region.

The dependence of the voltage on the temperature is also exponential, as shown in Fig. 9, which compares the model to the manufacturer's data[3]. The values for the constants are listed in Table 2, and the expression is

$$V_{bat}(SOC) = \left(3 + \frac{e^{SOC} - 1}{e - 1}\right)\left(2 - e^{\lambda \frac{T - T_0}{T_0}}\right). \tag{18}$$

At any given time instance, the battery power is the sum of the loads, i.e.,

$$P_{bat} = P_{sol} - P_{RW} - P_{comm} - P_0, \tag{19}$$

where P_0 is a 2-W constant power usage that accounts for the scientific instruments on the satellite and small actuator inputs in response to disturbance torques.

4.7 Communication

The communication discipline models the data-transfer bit rate as a function of several variables. We fix the signal-to-noise ratio (SNR) to a minimum acceptable value to maintain a reliable connection. A line-of-sight variable, similar to that computed in the Sun-position discipline, is used to account for the times when a link

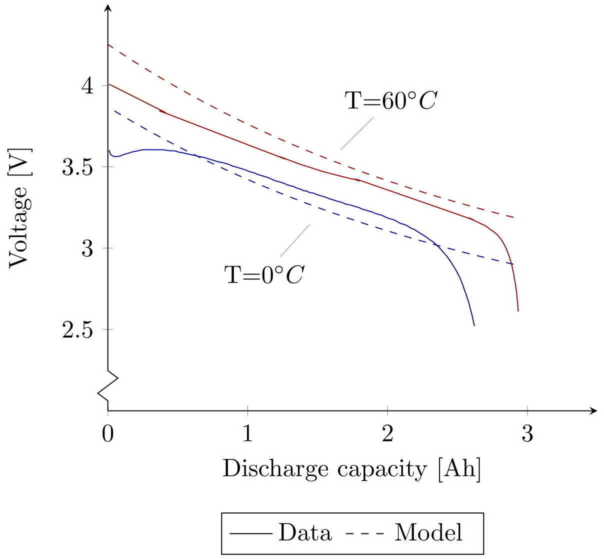


Figure 9: Battery-discharge curve model compared with manufacturer's data at two temperatures.

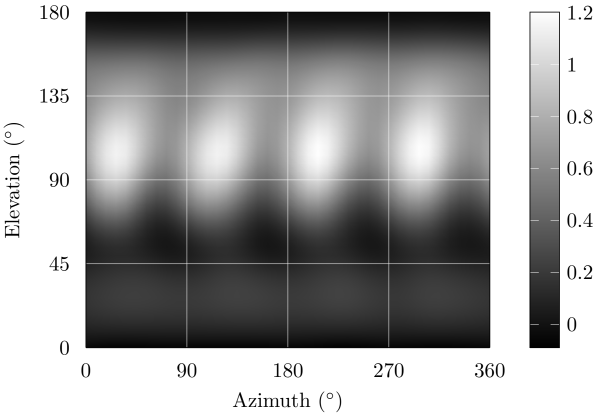


Figure 10: Transmitter gain as a function of the ground-station position in the body-fixed frame.

with the ground station is not possible. We compute the resulting data-download rate using the following equation [34, pp. 550–558]:

$$B_r = \frac{c^2 G_r L_l}{16\pi^2 f^2 k T_s (SNR)} \frac{\eta_p P_{comm} G_t}{S^2} LOS_c, \tag{20}$$

where the constants are listed in Table 2, S is the distance to the ground station, and G_t is the transmitter gain, which is plotted in Fig. 10.

We compute the LOS_c variable based on the dot product between the normalized Earth-to-ground station vector and the Earth-to-body vector in the inertial frame. We again smooth the discontinuous function, in this case by assuming that the line-of-sight variable gradually increases as the satellite comes over the horizon. This is illustrated in Fig. 11.

Orbit Dynamics		
Earth's gravitational parameter	μ	398600.44 km^{3}s^{-2}
Earth's radius	R_e	6378.137 km
Orbit perturbation coefficients	J_2	1.08264×10^{-3}
	J_3	-2.51×10^{-6}
	J_4	-1.60×10^{-6}
Attitude Dynamics		
Model coefficients	a	4.9×10^{-4} A$^{1/2}$ s/rad
	b	4.5×10^{2} A$^{1/2}$/(Nm)
	I_0	0.017 A
Temperature		
Mass	m	0.4 (fin), 2.0 (body) kg
Specific heat capacity	c_v	0.6 (fin), 2.0 (body) kJ/kg K
Absorptivity	α	0.9 (cell), 0.2 (radiator)
Emissivity	ϵ	0.87 (cell), 0.88 (radiator)
Boltzmann constant	k	$1.3806488 \times 10^{-23}$ m^2 kg/(s^2 K)
Speed of light	c	2.99792458×10^{8} m/s
Planck's constant	h	$6.62606957 \times 10^{-34}$ m^2 kg/s
Total cell area	A_T	2.66×10^{-3} m^2
Solar constant	q_{sol}	1.36×10^{3} W/m^2
Communication efficiency	η_p	0.2
Solar Power		
Diode voltage	V_0	-0.6 V
Max. short-circuit current	I_{sc_0}	0.453 A
Saturation current	I_{sat}	2.809×10^{-12} A
Diode factor	n	1.35 V
Charge of an electron	q	$1.60217657 \times 10^{19}$ C
Shunt resistance	R_{sh}	40 Ω
Energy Storage		
Nominal capacity	Q	2900 mAh
Temperature decay coeff.	λ	$ln\left(\frac{1}{1.1^5}\right)$
Reference temperature	T_0	293 K
Max. discharge rate	I_{min}	-10 A
Max. charge rate	I_{max}	5 A
Communication		
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

Table 2: Data for discipline models

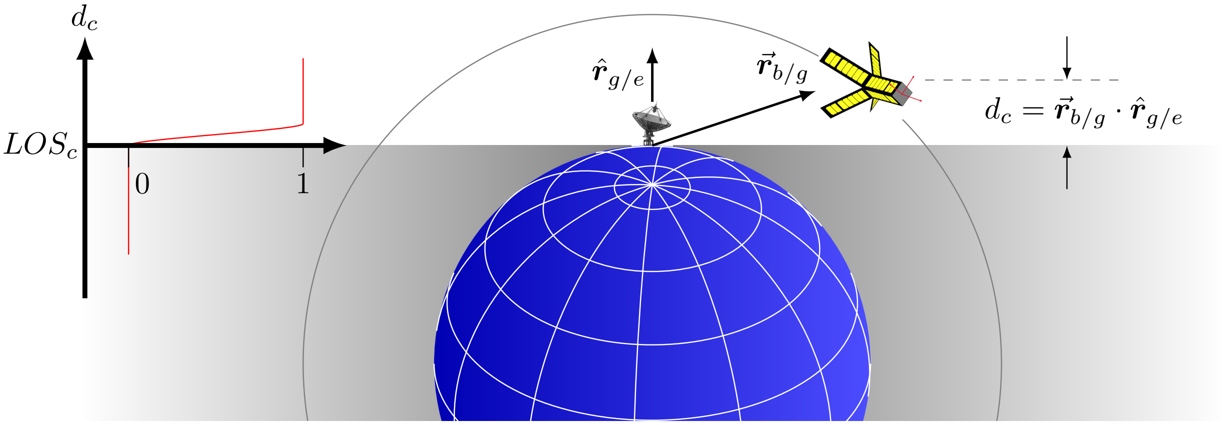

d_c
LOS_c
0
1
$\hat{\boldsymbol{r}}_{g/e}$
$\vec{\boldsymbol{r}}_{b/g}$
$d_c = \vec{\boldsymbol{r}}_{b/g} \cdot \hat{\boldsymbol{r}}_{g/e}$

Figure 11: Illustration of the ground-station line-of-sight variable.

5 Optimization

5.1 MDO Architecture

For an optimization problem involving many disciplines, the choice of the MDO architecture is critical. We use the multidisciplinary feasible (MDF) architecture [16], which solves an MDO problem by fully resolving the coupling between all the disciplines within each optimization iteration, effectively treating the coupled analyses of all the disciplines as one monolithic analysis. The rationale is that taking a restricted path to the optimum, with the interdisciplinary coupling converged at every optimization iteration, yields a robustness that is likely necessary for a problem with such a large number of disciplines. For a review of MDO architectures, the reader is encouraged to refer to Martins and Lambe [16].

However, the approach has elements that resemble the simultaneous analysis and design (SAND) architecture [16, 35] because some of the design variables could also be state variables. The roll-angle design variables could be replaced with reaction-wheel control inputs that are computed using a control law, and the optimal solar panel current at every time instance could be computed using maximum power point tracking (MPPT). Instead, we use nonlinear optimization as the controller in the former case. In the latter case, we compute all the peak power currents simultaneously as a smooth profile over time; our goal is to avoid poor conditioning due to local maxima.

Overall, this SAND-type approach yields three benefits. First, it avoids assumptions that would be required if these design variables were implemented as state variables. For instance, the attitude-control law must assume a desired roll-angle profile based on predetermined weights for the solar panel heating and cooling, cell illumination, and communication signal strength, while the optimization considers the net effect of rolling on the objective function by way of these three criteria. Second, it eliminates the risk that a discipline may not have a feasible solution, such as the attitude controller lacking the power to satisfy the forward-facing orientation constraint for any roll angle. Allowing the optimizer to control the distribution of power and enforce the battery-power and charge-level constraints ensures that all the disciplines are feasible internally, while we allow the battery constraints to be violated during the optimization. Finally, it removes the coupling between disciplines from the multidisciplinary analysis by moving the appropriate state variables to the optimization as design variables. The optimizer resolves the coupling, allowing the MDA to become a sequential problem, as shown in Fig. 12.

To avoid confusion, it is worth restating that the MDO architecture used in this problem is still MDF. The connection to SAND is limited to the fact that certain variables that could have been state variables have been implemented as design variables. However, the remaining state variables are not exposed to the optimizer, and all the variables are converged fully within every optimization iteration, which is consistent with the MDF architecture.

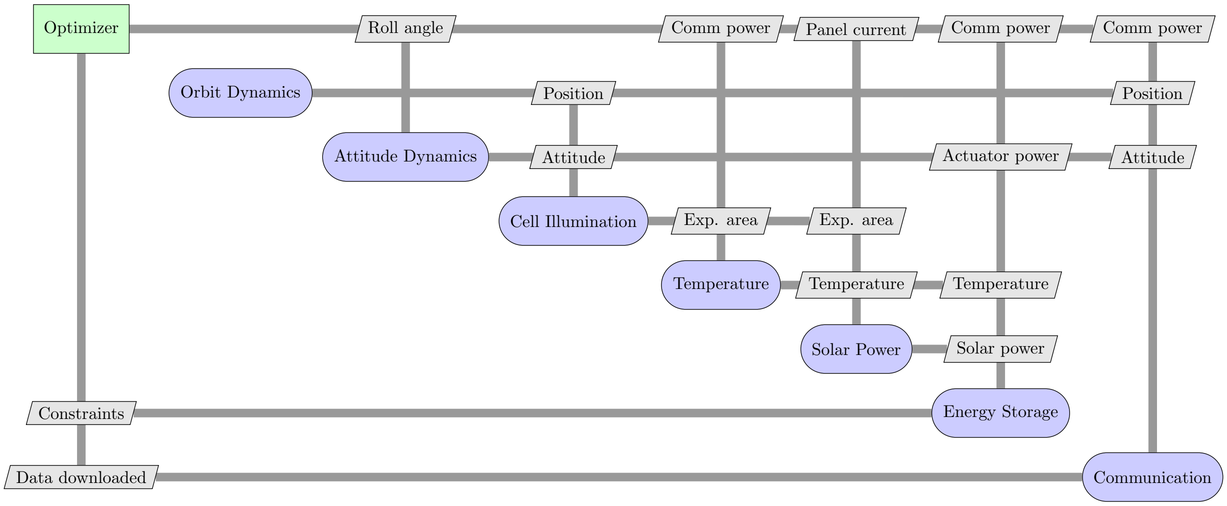


Figure 12: Extended design structure matrix (XDSM) diagram [17] for the MDO problem.

5.2 Optimization Problems

As previously mentioned, the multi-scale nature of the problem requires a multi-point optimization with 6 points, each representing a 12-hour simulation at the midpoint of every 2-month interval. We simulate half a day, 1, 3, 5, 7, 9, and 11 months after launch. This results in a multidisciplinary analysis with a total of 2,204,861 variables. The objective function is the average of the data-downloaded values of the 6 points, which is an estimate of the total annual data downloaded after a scaling factor. The battery charge rate, discharge rate, minimum SOC, maximum SOC, and periodicity constraints are enforced separately for each of the 6 points. The periodicity constraint enforces equality of the SOC at the beginning and end of each 12-hour simulation. The remaining four constraints are KS aggregation functions.

There are two scalar design variables (fin angle and antenna angle) and 84 binary variables that indicate whether or not a cell or radiator is installed. The variables that require the parameterization of their variations over time are the roll angle, the 12 solar panel currents, and the power allotted to the communication discipline for transmission. The current variable has the effect of emulating MPPT for the solar-power module, since the optimizer effectively selects the current, and indirectly the voltage, at which the maximum power can be generated from the cells in a given solar panel. Each profile variable is discretized with 1500 points, which is the number of points used in the time integrations, and they are represented using fourth-order B-splines with 300 control points. The optimization problem is summarized in Table 3.

As previously mentioned, we solve the optimization problem using SNOPT [36], a reduced-Hessian active-set SQP optimizer that solves nonlinear constrained problems very efficiently, particularly when derivatives are provided, as is the case here. We use the pyOpt optimization framework [18]; it provides a common interface to a suite of optimizers, including SNOPT.

Figure 13 plots the convergence history for the optimization. The number of function evaluations roughly corresponds to the number of SQP major iterations, and each takes about 20 min on a single processor, including the derivative computations. Overall, the algorithm requires 100 h to achieve convergence of nearly 5 orders of magnitude in feasibility and 3 orders of magnitude in optimality. Figure 13 shows that by the end of the optimization, nearly half of the design variables are superbasic variables in SNOPT, which are those variables that are truly free to change because they are not fixed by bounds or constraints. This indicates that near this local optimum, the dimension of the feasible design space is large, meaning that there is considerable design freedom with respect to which the design is optimal.

Figure 13 also illustrates the sequence in which the objective function was improved and the battery constraints were satisfied. For all six points, the initial design is clearly infeasible since the SOC curve is mostly negative. The optimizer spends most of the first 100 function evaluations trying to increase the power

	Variable/function	Description	Quantity
maximize	$\sum_{i=1}^{6} D_i$	Data downloaded	
with respect to	$0 \leq I_{\text{setpt}} \leq 0.4$	Solar panel current	$300 \times 12 \times 6$
	$0 \leq \gamma \leq \pi/2$	Roll-angle profile	300×6
	$0 \leq P_{\text{comm}} \leq 25$	Communication power	300×6
	$0 \leq \text{cellInstd} \leq 1$	Cell vs. radiator	84
	$0 \leq \text{finAngle} \leq \pi/2$	Fin angle	1
	$0 \leq \text{antAngle} \leq \pi$	Antenna angle	1
	$0.2 \leq SOC_i \leq 1$	Initial state of charge	6
		Total number of design variables	25292
subject to	$I_{\text{bat}} - 5 \leq 0$	Battery charge	6
	$-10 - I_{\text{bat}} \leq 0$	Battery discharge	6
	$0.2 - SOC \leq 0$	Battery capacity	6
	$SOC - 1 \leq 0$	Battery capacity	6
	$SOC_f - SOC_i = 0$	SOC periodicity	6
		Total number of constraints	30

Table 3: The general optimization problem.

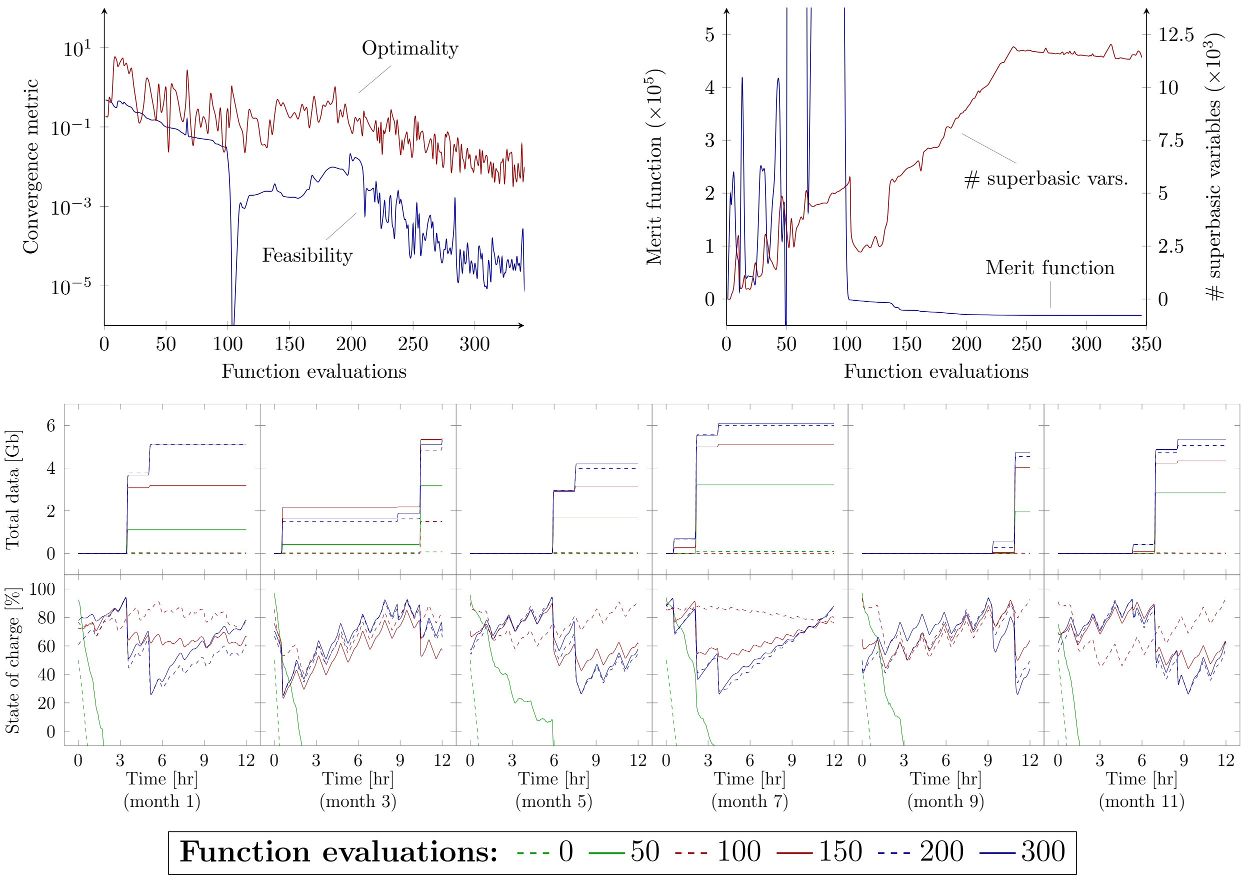


Figure 13: Convergence histories and snapshots of data and SOC at intermediate optimization iterations.

	Fin angle	Antenna angle	Data downloaded
Baseline optimization	45°	0°	2122 Gb/yr
Geometry optimization	63.8°	$-45°$	2991 Gb/yr
Geometry and attitude optimization	64.4°	$-45°$	3758 Gb/yr

Table 4: Optimal design variables for for the three optimization problems.

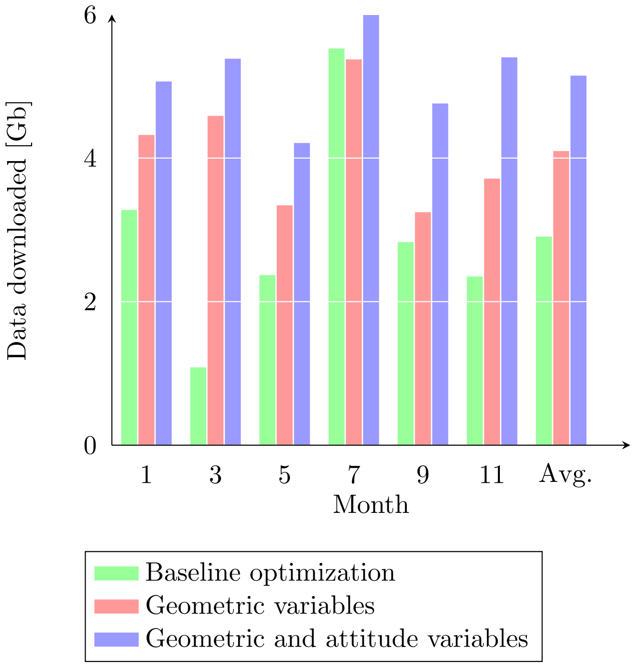


Figure 14: Division of total data downloaded over the six simulations for the three optimization problems.

generation to make the SOC curves feasible. After this, it focuses on increasing the objective function.

5.3 Impact of Optimization

To quantitatively assess the impact of the optimization, we solve three optimization problems. The first is a baseline optimization that is the same as the original optimization problem in Table 3, except that the fin angle, antenna angle, and roll-angle profile are removed from the set of design variables. The remaining design variables are the solar-panel current, communication power, initial SOC, and installation of cell or radiator, which provide a baseline design. The second optimization adds the geometric design variables, which are the fin and antenna angles. The third optimization adds both the geometric and attitude design variables to the baseline optimization, yielding the problem described in Table 3.

Table 4 summarizes the results. Since the constraints are satisfied in all of the optimizations, the objective function alone provides a good metric for comparison. Adding the geometric design variables yields a 40% increase in the estimate of the data downloaded, and adding the roll angle yields an additional 40% increase. Whenever the antenna angle is permitted to vary, it goes to the bound of $-45°$, while the fin angle converges to an interior optimum. For all the problems, the optimizer chooses to install the solar cell instead of the radiator for all 84 cells. Figure 14 shows how the objective-function increases are distributed among the 6 points.

These optimization results can be summarized as follows. The fin angle and roll-angle profile increase the cell illumination to provide more communication power, and the antenna angle increases the gain during the ground-station passes for higher data rates. However, an examination of each variable reveals more insight into how the optimization problems compare, as well as how they satisfy the battery constraints and increase the total downloaded data. Figures 15 and 16 plot several quantities of interest as functions of time, for each point and for each of the three optimization results.

In Fig. 15, the data downloaded plots demonstrate the importance of optimizing both the geometric and attitude design variables. The communication power plots show that the optimizer allocates power to the transmitter only during the ground-station passes, as expected, but the peaks of the spikes are limited by the available SOC and the discharge constraint. The transmitter gain plots show the highest gains for the geometry and attitude optimization, followed by the geometry optimization, then the baseline optimization. This is evidence that the fin angle provides an increase in gain, and the roll-design variables provide a further increase in gain, which translates to higher data rates. An interesting observation regarding the roll-angle profiles is that they are smooth for the most part but exhibit spikes aligned with ground-station passes. Finally, the SOC plots show that the additional power generated by the optimizations is used for a gradual build-up of energy between data transmissions, enabling short and rapid power discharges for high-bit-rate data transmissions.

In Fig. 16, the large increase in the solar power generation from the baseline optimization to the geometry optimization and the smaller increase from the geometry optimization to the geometry and attitude optimization indicate that the fin angle has a large effect, and the attitude profile has a smaller but still definite effect. The solar-panel current curves represent the maxima for each time instance among the twelve panels, and they correctly go to zero when the satellite is in the Earth's shadow to prevent negative voltages, while taking on optimal current values when in the sun to maximize power. As with the solar-power plots, the total-exposed-area plots give a clear indication that the fin angle has the largest effect in increasing cell illumination, and interestingly, the exposed area is sacrificed in months 1 and 7 when the satellite is always in the sun and is not power constrained. The body temperature is weakly dependent on the roll-angle profile and it also has a smaller effect on the solar power, so periodicity constraints are not used to avoid the additional linear solutions required for the associated derivatives. The battery-current plots show that the communication power is limited by the battery-discharge constraint for many of the ground-station passes, while the remainder are energy-limited.

5.4 Comparison Between Launches

One of the strengths of our approach and implementation is that the design optimization algorithm is robust with respect to convergence. To aid decision-making, it is possible to run optimizations for various choices of parameters, such as the launch, ground-station selection, and satellite specifications, and to compare them. To illustrate, we ran two additional optimizations for different launch orbits and dates. These optimizations involve the design variables listed in Table 3.

The results are summarized in Table 5. The antenna angles converge to the same value, and the estimated data downloaded is roughly the same for the three launches. However, there is a large discrepancy in the optimal fin angle, which suggests that it could be sensitive to the launch orbit. It has been consistently observed that the fin angle has a large effect on the potential power generation, and the optimal fin angle varies significantly for different launches as the result for launch 2 shows. This observation points to the importance of carefully selecting the fin angle once the launch orbit is known. Figure 17 shows how the total downloaded data is divided among the points.

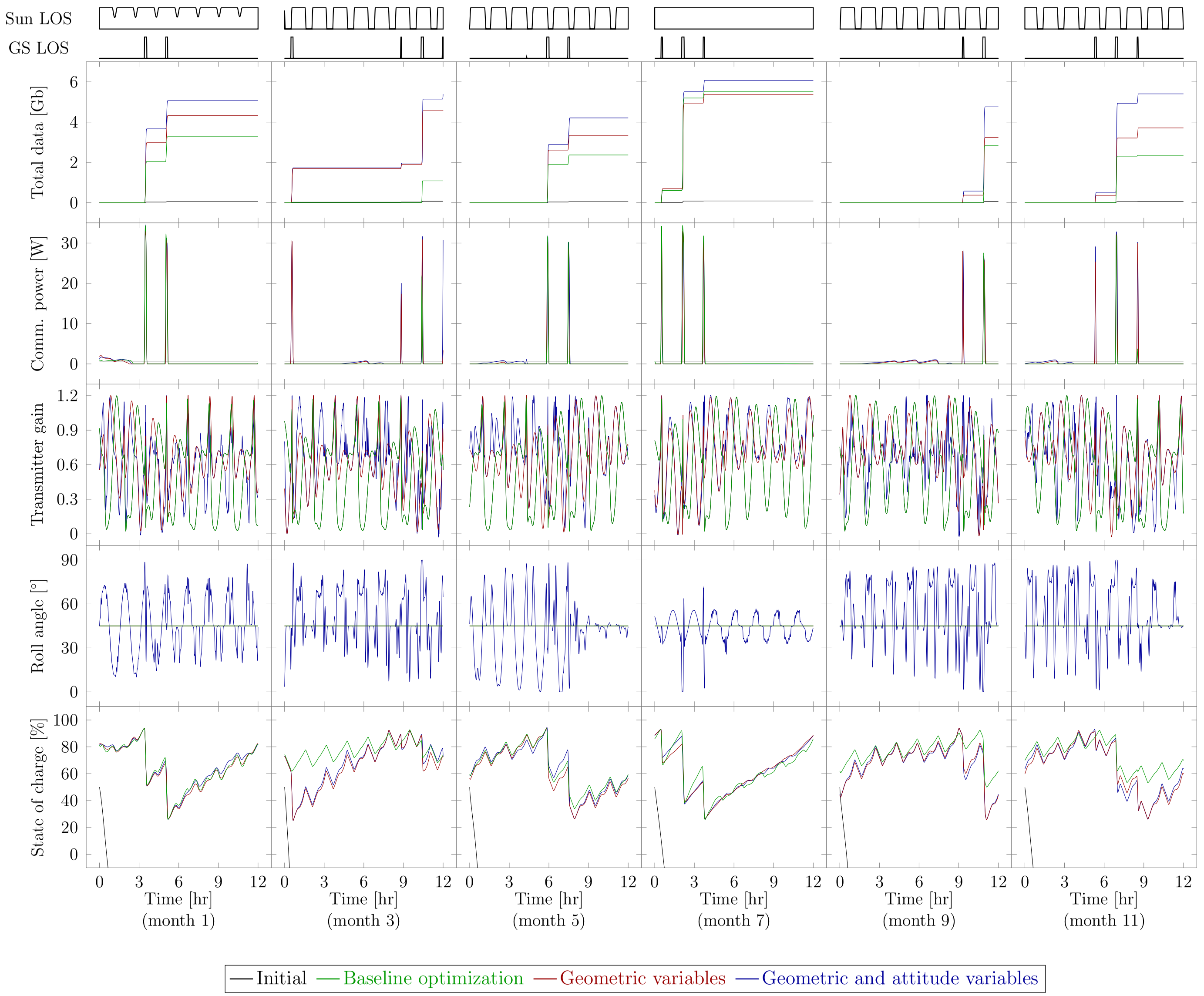


Figure 15: Initial and optimized profiles of quantities of interest for the three optimization problems (continued in Fig. 16).

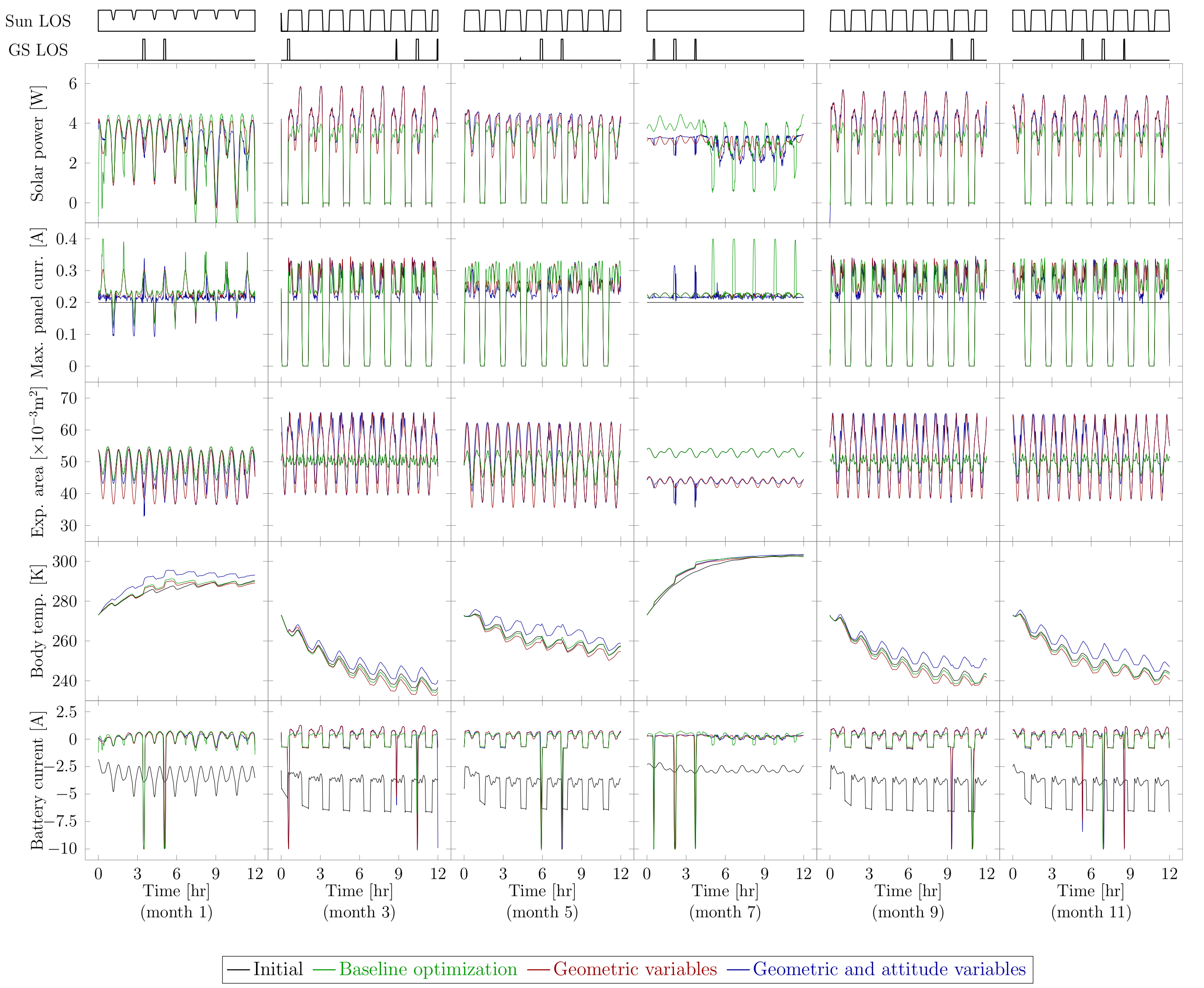


Figure 16: Initial and optimized profiles of quantities of interest for the three optimization problems (continued from Fig. 15).

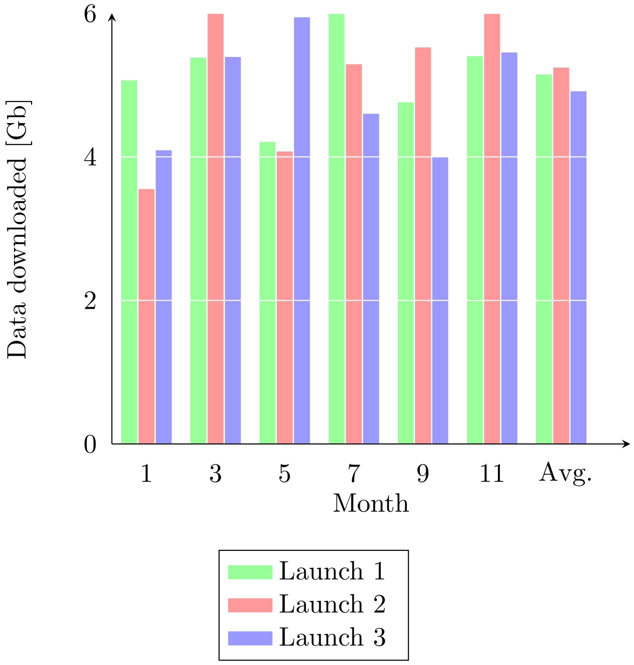


Figure 17: Division of total data downloaded over the six simulations for the three launches.

Launch	Fin angle	Antenna angle	Data downloaded
1	64.4°	−45°	3758 Gb/yr
2	49.9°	−45°	3829 Gb/yr
3	68.5°	−45°	3587 Gb/yr

Table 5: Optimal values for the three launches.

6 Future Research Directions

In the course of this work, it has become clear that combining small-satellite design and MDO is beneficial to both fields. From the MDO point of view, this problem has many disciplines, variables, and tradeoffs, and the fact that much of the behavior and coupling in the system can be captured with relatively inexpensive numerical models allows it to truly leverage MDO. These features also make it a suitable benchmarking problem; in fact, it has been implemented as a test problem in OpenMDAO [37], which is an open-source framework for MDAO. From the satellite-design point of view, our results show that MDO has the potential to shorten the design process and yield performance improvements when CADRE is designed and built.

The most obvious avenue for future work is to improve the models and explore the design problem with more fidelity. In particular, the temperature model and the reaction-wheel power consumption model have room for improvement, and a model for the magnetorquers would be beneficial. These areas do not strongly affect the objective and constraint functions, so they are not expected to significantly change the results of this study. However, they represent weaknesses in the model that could be addressed in the future.

Another potential avenue for future research is the exploration of alternative objective functions. The modularity of the framework that has been developed facilitates the rapid implementation of new objective and constraint functions since the most time-consuming aspects—interfacing new code to the existing code and computing the coupled derivatives—are handled with a high level of automation. For instance, to maximize profit rather than the total data downloaded, we would simply need to implement a computation for profit as a function of the data downloaded and other relevant variables as well as the corresponding partial derivatives. The coupled derivatives that the framework automatically computes would then include derivatives of the profit with respect to any other variable in the MDAO problem.

On the numerical side, parallel computing would significantly reduce the wall time for the analysis and optimization, enabling a faster turnaround time and a more rapid exploration of the design problem. Many of the models can easily be parallelized because the state variables at all time instances are tracked simultaneously. Parallel computing would also result in more available memory, which would allow us to consider larger problems with more resolution or a longer satellite operation time.

Finally, the generalization of our MDAO formulation to seamlessly support problems in other fields is a promising area for future work. A forthcoming paper will present a formal description and detailed exposition of the generalized formulation.

7 Conclusion

The objective of this paper was to apply large-scale multidisciplinary design optimization to a small satellite. We used gradient-based optimization along with the multidisciplinary feasible architecture implemented within a new formulation for multidisciplinary analysis and optimization problems. Other key enabling tools included multi-point optimization, constraint aggregation, multi-dimensional B-spline interpolation, a differentiated fourth-order Runge–Kutta solver, and efficient numerical linear algebra.

We have demonstrated the ability to reliably solve an optimization problem with 7 disciplines, more than 25,000 design variables, and over 2.2 million state variables that represent 12 hours of the satellite's operation at 6 uniformly spaced points over the year. To assess the impact of this tool, we solved three optimization problems with varying sets of design variables. The addition of geometric design variables to the satellite design problem yielded a 40% improvement in the objective function (the total data downloaded), and the addition of operational design variables yielded a further 40% improvement. Furthermore, changing the launch parameters changed the values of the objective function and the design variables, suggesting that this tool could be used to evaluate launch options and to tailor the design to a particular launch opportunity.

In addition to these contributions to satellite design, we have made four broader contributions. First, we have demonstrated that considering all the major disciplines, time scales, and design variables simultaneously for the small-satellite problem is feasible through a rigorous multidisciplinary approach. As an alternative to more detailed single-discipline studies, this approach provides a system-level perspective of the problem with sufficient depth to capture high-level tradeoffs and reveal insights that are perhaps not obvious at the discipline level.

Second, we have demonstrated that a design problem with many discontinuities and discrete data can be solved with gradient-based optimization. The discontinuities were overcome with a combination of multi-dimensional B-spline interpolation of the data and the development of smooth models for various disciplines.

Third, the SAND-like approach of implementing what might more naturally be state variables as design variables simplified the multidisciplinary analysis without a significant sacrifice in the optimization convergence. Instead of using predetermined state variables to satisfy the power constraints, the optimizer had the freedom to optimally distribute the available power among the attitude-control actuator, communication gain, and scientific instruments, while satisfying the battery constraints.

Our final contribution is the successful application of the MDAO formulation. The disciplines were implemented in a modular way so that solving the multidisciplinary analysis problem and computing the coupled derivatives was as automated as possible, and the task of adding more disciplines was greatly simplified. These results demonstrate the promise of this approach and build a strong case for the adoption of this method not only in satellite design but in other engineering design problems as well.

8 Acknowledgments

This work was partially supported by NASA through award No. NNX11AI19A—Technical Monitor: Justin S. Gray. The authors would like to thank Daniel Meinzer, who developed the OpenGL-based exposed-area model; Alyssa Francken, who developed the transmitter-gain model; and John Springmann for sharing his knowledge of small-satellite design.

References

[1] Cutler, J. W., Ridley, A., and Nicholas, A., "Cubesat Investigating Atmospheric Density Response to Extreme Driving (CADRE)," *Proceedings of the 25th Small Satellite Conference*, Logan, UT, August 2011.

[2] Boudjemai, A., Bouanane, M. H., Merad, L., and Si Mohammed, A. M., "Small Satellite Structural Optimisation Using Genetic Algorithm Approach," *Proceedings of the 3rd International Conference on Recent Advances in Space Technologies*, Istanbul, Turkey, 2007, pp. 398–406. doi:10.1109/RAST.2007.4284021.

[3] Galski, R. L., De Sousa, F. L., Ramos, F. M., and Muraoka, I., "Spacecraft thermal design with the Generalized Extremal Optimization Algorithm," *Inverse Problems in Science and Engineering*, Vol. 15, No. 1, 2007, pp. 61–75. doi:10.1080/17415970600573924.

[4] Jain, S. and Simon, D., "Genetic Algorithm Based Charge Optimization of Lithium-Ion Batteries in Small Satellites," *Proceedings of the 19th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, August 2005.

[5] Richie, D. J., Lappas, V. J., and Palmer, P. L., "Sizing/Optimization of a Small Satellite Energy Storage and Attitude Control System," *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, July 2007, pp. 940–952. doi:10.2514/1.25134.

[6] Zhang, B., Teng, H.-F., and Shi, Y.-J., "Layout optimization of satellite module using soft computing techniques," *Appl. Soft Comput.*, Vol. 8, No. 1, Jan. 2008, pp. 507–521. doi:10.1016/j.asoc.2007.03.004.

[7] Barnhart, D. A., Kichkaylo, T., and Hoag, L., "SPIDR: Integrated Systems Engineering Design-to-Simulation Software for Satellite Build," *Proceedings of the 7th Annual Conference on Systems Engineering Research*, Loughborough, UK, 2009.

[8] Fukunaga, A., Chien, S., Mutz, D., Sherwood, R., and Stechert, A., "Automating the Process of Optimization in Spacecraft Design," *Proceedings of the 1997 IEEE Aerospace Conference*, Vol. 4, Aspen, CO, 1997, pp. 411–427. doi:10.1109/AERO.1997.577524.

[9] George, J., Peterson, J., and Southard, S., "Multidisciplinary Integrated Design Assistant for Spacecraft (MIDAS)," *Proceedings of the 36th Structures, Structural Dynamics, and Materials Conference*, New Orleans, LA, 1995. doi:10.2514/6.1995-1372.

[10] Mosher, T., "Spacecraft design using a genetic algorithm optimization approach," *Proceedings of the 1998 IEEE Aerospace Conference*, Vol. 3, Aspen, CO, 1998, pp. 123–134. doi:10.1109/AERO.1998.685783.

[11] Stump, G., Yukish, M., Simpson, T., and O'Hara, J., "Trade space exploration of satellite datasets using a design by shopping paradigm," *Proceedings of the 2004 IEEE Aerospace Conference*, Vol. 6, 2004, pp. 3885–3895. doi:10.1109/AERO.2004.1368206.

[12] Ebrahimi, M., Farmani, M. R., and Roshanian, J., "Multidisciplinary design of a small satellite launch vehicle using particle swarm optimization," *Structural and Multidisciplinary Optimization*, Vol. 44, No. 6, 2011, pp. 773–784. doi:10.1007/s00158-011-0662-7.

[13] Jafarsalehi, A., Zadeh, P. M., and Mirshams, M., "Collaborative Optimization of Remote Sensing Small Satellite Mission using Genetic Algorithms," *Transactions of Mechanical Engineering*, Vol. 36, No. 2, 2012, pp. 117–128.

[14] Wu, W., Huang, H., Chen, S., and Wu, B., "Satellite Multidisciplinary Design Optimization with a High-Fidelity Model," *Journal of Spacecraft and Rockets*, Vol. 50, No. 2, March 2013, pp. 463–466. doi:10.2514/1.A32309.

[15] Braun, R. D. and Kroo, I. M., "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment," *Multidisciplinary Design Optimization: State of the Art*, edited by N. Alexandrov and M. Y. Hussaini, SIAM, 1997, pp. 98–116.

[16] Martins, J. R. R. A. and Lambe, A. B., "Multidisciplinary Design Optimization: A Survey of Architectures," *AIAA Journal*, Vol. 51, No. 9, September 2013, pp. 2049–2075. doi:10.2514/1.J051895.

[17] Lambe, A. B. and Martins, J. R. R. A., "Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes," *Structural and Multidisciplinary Optimization*, Vol. 46, August 2012, pp. 273–284. doi:10.1007/s00158-012-0763-y.

[18] Perez, R. E., Jansen, P. W., and Martins, J. R. R. A., "pyOpt: a Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization," *Structural and Multidisciplinary Optimization*, Vol. 45, No. 1, January 2012, pp. 101–118. doi:10.1007/s00158-011-0666-3.

[19] Martins, J. R. R. A. and Hwang, J. T., "Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models," *AIAA Journal*, Vol. 51, No. 11, November 2013, pp. 2582–2599. doi:10.2514/1.J052184.

[20] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, Sept. 1988, pp. 233–260.

[21] Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J., and Saunders, D., "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 1," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 51–60.

[22] Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J., and Saunders, D., "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 2," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 61–74.

[23] Lyu, Z. and Martins, J. R. R. A., "Aerodynamic Shape Optimization of a Blended-Wing-Body Aircraft," *Proceedings of the 51st AIAA Aerospace Sciences Meeting*, Grapevine, TX, Jan. 2013. doi:10.2514/6.2013-283, AIAA 2013-0283.

[24] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design," *Optimization and Engineering*, Vol. 6, No. 1, March 2005, pp. 33–62. doi:10.1023/B:OPTE.0000048536.47956.62.

[25] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Derivative Computations," *AIAA Journal*, Vol. 52, No. 5, May 2014, pp. 935–951. doi:10.2514/1.J052255.

[26] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Journal of Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006. doi:10.1137/S1052623499350013.

[27] Hwang, J. T., Lee, D. Y., Cutler, J. W., and Martins, J. R. R. A., "Large-Scale MDO of a Small Satellite using a Novel Framework for the Solution of Coupled Systems and their Derivatives," *Proceedings of the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Boston, MA, April 2013. doi:10.2514/6.2013-1599.

[28] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., "The Complex-Step Derivative Approximation," *ACM Transactions on Mathematical Software*, Vol. 29, No. 3, 2003, pp. 245–262. doi:10.1145/838250.838251.

[29] Kreisselmeier, G. and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *International Federation of Active Controls Syposium on Computer-Aided Design of Control Systems, Zurich, Switzerland*, 1979.

[30] Poon, N. M. K. and Martins, J. R. R. A., "An Adaptive Approach to Constraint Aggregation Using Adjoint Sensitivity Analysis," *Structural and Multidisciplinary Optimization*, Vol. 34, No. 1, 2007, pp. 61–73. doi:10.1007/s00158-006-0061-7.

[31] Kenway, G. K. W. and Martins, J. R. R. A., "Multi-Point High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration," *Journal of Aircraft*, Vol. 51, No. 1, January 2014, pp. 144–160. doi:10.2514/1.C032150.

[32] Kennedy, G. J. and Martins, J. R. R. A., "A Comparison of Metallic and Composite Aircraft Wings Using Aerostructural Design Optimization," *14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, IN, Sep. 2012, AIAA-2012-5475.

[33] Kawamura, H., Naka, K., Yonekura, N., Yamanaka, S., Kawamura, H., Ohno, H., and Naito, K., "Simulation of I-V characteristics of a PV module with shaded PV cells," *Solar Energy Materials & Solar Cells*, Vol. 75, 2003, pp. 613–621.

[34] Larson, W. and Wertz, J., *Space mission analysis and design*, Kluwer Academic Publishers, January 1991.

[35] Haftka, R. T., "Simultaneous Analysis and Design," *AIAA Journal*, Vol. 23, No. 7, 1985.

[36] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131. doi:10.1137/S0036144504446096.

[37] Gray, J., Moore, K. T., Hearn, T. A., and Naylor, B. A., "Standard Platform for Benchmarking Multidisciplinary Design Analysis and Optimization Architectures," *AIAA Journal*, Vol. 51, No. 10, 2013, pp. 2380–2394. doi:10.2514/1.J052160.