71st International Astronautical Congress (IAC) - The CyberSpace Edition, 12-14 October 2020.


IAC–20–D1.4B.6

# Comparison of Multidisciplinary Design Optimization Architectures for the design of Distributed Space Systems

**Raja Pandi Perumal**
University of Luxembourg, Luxembourg, raja.pandiperumal@uni.lu

**Holger Voos**
University of Luxembourg, Luxembourg, Holger.Voos@uni.lu

**Florio Dalla Vedova**
LuxSpace Sarl, Luxembourg, dallavedova@luxspace.lu

**Hubert Anton Moser**
LuxSpace Sarl, Luxembourg, moser@luxspace.lu

Advancement in satellite technology, and the ability to mass-produce cost-effective small satellites has created a compelling interest in Distributed Space System (DSS), such as Low Earth Orbit (LEO) satellite constellations. Optimization of DSS is a complex Multidisciplinary Design Optimization (MDO) problem involving a large number of variables and coupling relations. This paper focuses on comparing three different MDO architectures for a DSS design problem. Initially, an overview of the constellation model, the subsystems model, and the coupling relationships between the subsystems and the constellation are provided. The modelling of the subsystems and the constellation configuration are carried out in OpenMDAO. Later, three monolithic MDO architectures, namely, Individual Discipline Feasible (IDF), Simultaneous Analysis and Design (SAND) and Multidisciplinary Feasible (MDF) are compared by implementing them to the developed DSS model. The results indicate IDF outperforms the rest of the architectures for the conceptual design of DSS. The optimum objective function obtained by IDF is 1% lower than SAND and 7% lower than MDF. While the functional evaluation required for IDF is 50% lower than SAND and 90% lower than MDF.
**Keywords:** MDO Architecture, Satellite Design, Satellite System Engineering, Distributed Space Systems

## 1. Introduction

In the emerging NewSpace industry, driven by the advancements and miniaturization of electronics, the capabilities and application of small satellites are growing tremendously. Small satellites offer unique benefits such as shorter development time, lower cost, relatively simple maintenance, and mass producibility. As a result, small satellites are currently considered for almost every space applications.[1] Small satellites are predominant in Low Earth Orbits (LEO).[24] The ground coverage capacity of a small satellite operating in LEO is limited and often operate in a group to accomplish a commercial mission. Distributed Space Systems (DSS) is a system in which several satellites work together to achieve a common goal which is not feasible with a single small satellite. There is always a growing demand for cost-effective DSS operable in LEO. Previous researches on DSS[8,9,12,26] were focused on optimizing the geometric configuration of the satellite constellation to improve coverage and reduce the overall cost. However, the satellite subsystems and their parameters were not considered for the optimization. The long-term success of a DSS hinges on substantial cost reduction. This is possible only when the connection between the constellation configuration and satellite subsystems are fully exploited. Therefore, design of DSS considering all the disciplines with interdisciplinary coupling needs to be optimized. Because of its disciplinary boundaries, DSS cannot be optimized as a standard constrained nonlinear programming problem and has to be considered as a Multidisciplinary design problem.

The optimum solution to a multidisciplinary design problem can be achieved not only by considering the design of individual disciplines in the system but also their interactions. Multidisciplinary Design Optimization (MDO) is a field of engineering that is used to solve optimization problems involving more than one discipline. Like any other optimization, the primary focus of MDO is to identify an optimal solution in a specified design space subject to constraints. Optimization of individual subsystems can conflict with each other when assembled. Therefore, the entire system has to be optimized holistically. Modelling of subsystem in-

teractions with each other in a DSS complicates the optimization problem as the subsystem compatibility has to be maintained along with the objective function minimization. There are several ways to overcome this challenge with problem formulation procedure called MDO architecture.

Martins and Lambe[21] presented 14 different MDO architectures that are found in the literature. MDO architectures are broadly classified as hierarchical and non-hierarchical architectures. In hierarchical architecture, each child element has a parent element with which it exclusively interacts.[7] Whereas, in the non-hierarchical architectures, there are substantial interactions among the child elements, as well as the parent element. Non-hierarchical approach is needed for our case as there are strong interactions among the child elements. Based on the problem formulation, non-hierarchical architectures are further classified into two types. When the problem is formulated as a single optimization problem, it is called monolithic architecture. Whereas, when the problem is decomposed into sub-problems and reassembled to have a combined solution, it is called Distributed architecture.

There has been numerous efforts to compare different architectures in the past.[4,7,10,13–15,22,28,33] When optimizing a complex system, Hulme and Bloebaum[13] arrived at a result favouring Multiple Design Feasible (MDF) over Individual Design Feasible (IDF) and All-At-Once (AAO). Another study by Marriage and Martins,[20] concluded that Collaborative Optimization (CO) outperformed MDF when few subsystems were highly-coupled while the others were not. On the contrary, benchmarking by Tedford and Martins[28] showed that IDF and Simultaneous Analysis and Design (SAND) had the best performance over MDF and CO for their problems. A recurring point made from these studies is that the performance of architecture is problem-specific, and there is no one superior architecture that is suitable for all types of problems. Diverse results obtained from the above studies show that the selection of MDO architecture affects both the optimality of the solution as well as the computational resources required. Very few researchers[27] have optimized the DSS design problem using MDO approaches but the comparisons of MDO architectures have not been done for this specific case. The goal of this paper is to compare three different Monolithic architectures: MDF, IDF and SAND for the DSS design problem. The comparison of Distributed architecture for the DSS design problem is foreseen in the future.

The various disciplines and their interactions of DSS are modelled in OpenMDAO,[11] which is a specialized framework for MDO optimization. The usage of such framework removes most of the human-factors in programming of the architectures. Therefore, the results remain unbiased and depend only on the nature of the problem.

The paper begins with a detailed description of the DSS design problem, which explains the objectives and the intricate connections between the disciplines. Next, we describe each subsystem and its developed analytical model. Having established the background for the DSS design problem, we proceed to examine MDF, IDF and SAND architectures. Finally, we present the optimization results and compare all the MDO architectures considered in the paper.

## 2. Problem Description

In this section, we describe a DSS model to compare the selected architectures. A DSS is a complex multidisciplinary system consists of following subsystems: (i) Constellation model, (ii) Payload, (iii) Power, (iv) Thermal, (v) Structure, (vi) Attitude Determination and Control System (ADCS), (vii) The on-board data handling (OBDH), (viii) Telemetry, Tracking and Command Systems (TT&C), and (ix) Propulsion. For the initial study, we consider only Constellation, Payload, Power, Thermal and structure subsystems. We assume that ADCS, OBDH, TT&C, and Propulsion subsystems are readily available and their mass and power budgets are estimated based on design estimation relationships.[2,6,31] The multidisciplinary design optimization problem requires a set of design variables and subsystem inputs to generate subsystem states by solving the respective analysis model. The calculated subsystem states are either needed to compute the objective/constraints or needed by other subsystems (coupling). The connections between the subsystems of modelled DSS is illustrated in Fig. 1. The set of design variables used in this problem, and their permissible ranges are presented in section 4.

The difference between the MDO architectures lies in their handling of coupling variables. Most of the design variables in this problem are continuous while some of them are discrete. If we consider both types of variables then the optimization problem becomes a Mixed Integer Non-Linear Problem (MINLP) which are very hard to solve. The discrete variables such as the number of satellites and orbital planes are important for reducing the launch cost but they are not directly coupled to any other subsystem. While these discrete variables will impact the resultant satellite design and the computation time, it will not affect the comparison between the architectures. Therefore, in this paper we fix the number of satellites and orbital planes to reduce

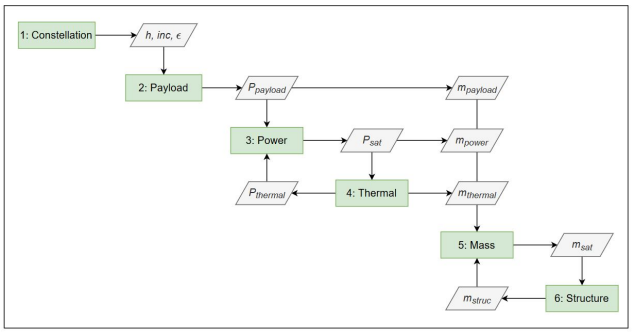


Fig. 1: Coupling of DSS (MDO) disciplines

MINLP to Non-Linear Problem (NLP), and optimize only the Altitude, Inclination and Elevation angle in the constellation discipline.

The DSS considered in this paper is an earth observation constellation operating in LEO. The constellation consists of 25 satellites evenly distributed over five orbital planes. The satellites in the constellation are assumed to be identical to each other and have the same optical payload. The primary constraint enforced in this DSS optimization problem is to provide at least 70% coverage over Luxembourg, Belgium, and Germany. The objective of the optimization problem is to minimize the overall mass of the DSS system subject to various constraints. The mass of DSS $m_{sys}$ is calculated using equation (1).

$$m_{sys} = N_s \times (m_{struc} + m_{payload} + m_{power} + m_{thermal} + m_{remaining}) \quad [1]$$

where $N_s$ is the number of satellites, $m_{struc}$ is the mass of satellite structure, $m_{payload}$ is the mass of payload, $m_{power}$ is the mass of power subsystem, $m_{thermal}$ is the mass of thermal subsystem and $m_{remaining}$ is the mass of remaining subsystems.

## 3. DSS Subsystems

In this section, the mathematical model for all the considered disciplines in the Distributed Space System is detailed. Here the parent element is DSS and the children elements are as follows: Constellation, Payload, Power, Thermal and Structure.

### *3.1 Constellation Model*

Coverage goal of the DSS can be achieved by numerous constellation patterns. However, we consider Walker Delta pattern for its simplicity and cost-effectiveness.[30,32] The constellation parameters $i$, $N_s$, $p$, and $f$ define the distribution of the satellites in space, where $i$ is the inclination, $N_s$ is the total number of satellites, $p$ is the number of orbital planes, and $f$ is the phase difference between satellites in the adjacent plane. Walker Delta constellation pattern is denoted by $i : N_s/p/f$. The number of satellite in each orbit is given by $s = \frac{N_s}{p}$ where $p \mid N_s$. In this paper, to avoid collisions between satellites, the phase difference between adjacent satellites in a single plane is calculated by $f \times \frac{360^\circ}{N_s}$, where $f$ is an integer between 0 to $(p-1)$.

The orbital parameters of a satellite in the three-dimensional space include the six Keplerian elements: semi-major axis ($a$), eccentricity ($e$), inclination ($i$), the longitude of ascending node ($\Omega$), the argument of perigee ($\omega$), and true anomaly ($\nu$). Since the Walker Delta constellation consists of circular orbits, $e = 0$ and $\omega = 0$. Therefore, $a$ is equal to the radius of the orbit, and $\nu$ becomes the angle from the ascending node to the satellite's position vector. The right ascension of ascending node is given by $\Omega = \frac{360^\circ}{p}$. At epoch $\nu_n = f_n$ where $n = 1, 2, ..., N_s$. The optimization design variables for the constellation discipline are the Altitude ($h$), $i$ and elevation angle ($\epsilon$). The constraints set for this problem is to achieve total temporal coverage of at least 70%.

At first, the satellite's initial state is determined from the orbital parameters. Then, the satellite state is propagated around the Earth over a defined time, 12 days in our case. Finally, the coverage is computed and updated for each satellite. Detailed descriptions of these steps are as follows:

**Satellite State Determination** The state ($\vec{Y}$) of the satellite at any a given point in space is determined by the position and velocity vectors of the satellite. Equation (2-4) represent the state ($\vec{Y}_{PQW}$), position ($\vec{r}_{PQW}$), and velocity ($\vec{v}_{PQW}$) vectors in the Perifocal coordinate system, $PQW$, where $\mu$ is the standard gravitational parameter. Here, $P$ axis is towards perigee ($\omega$), $Q$ axis is 90° from $P$ in the direction of satellite motion, and $W$ axis is normal to the orbit.

$$\vec{Y}_{PQW} = \begin{bmatrix} \vec{r}_{PQW} \\ \vec{v}_{PQW} \end{bmatrix} \quad [2]$$

$$\vec{r}_{PQW} = \begin{bmatrix} \frac{acos(\nu)}{1+ecos(\nu)} \\ \frac{asin(\nu)}{1+ecos(\nu)} \\ 0 \end{bmatrix} \quad [3]$$

$$\vec{v}_{PQW} = \begin{bmatrix} -\sqrt{\frac{\mu}{p}} sin(\nu) \\ \sqrt{\frac{\mu}{p}}(e + cos(\nu)) \\ 0 \end{bmatrix} \quad [4]$$

Since $\omega = 0$, the perifocal frame of reference becomes obsolete for further calculation. Therefore, using coordinate transformation, the state variables are transformed into the Earth-Centred Inertial (ECI) system, $IJK$, where $I$ axis points towards the vernal equinox direction, $J$ axis is 90° towards the east in the equatorial plane and $K$ axis goes through the north pole.

**Satellite State Propagation** Due to the gravitational attractions between the satellite and the Earth, the movement of a satellite in an orbit around the earth is considered as a two-body problem. In an ideal case, this is represented by simple equations of motion. However, in an actual case, there are perturbations due to the following: (i) Non-homogeneity and oblateness of Earth, (ii) Third-body effects, (iii) atmospheric drag, and (iv) solar pressure. The effect of the perturbations in the satellite cannot be neglected in a real-world scenario. Cowell's Formulation[3] accounts for these effects by adding the perturbing accelerations to the two-body equation of motion, as shown in equation (5).

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} + \vec{a}_p \quad [5]$$

where perturbing acceleration, $\vec{a}_p$, is the total acceleration caused by all other forces acting on the satellite and $\ddot{\vec{r}}$ is the resultant satellite acceleration. The specific form of $\vec{a}_p$ depends on the number of perturbation sources considered in the problem. In this paper, we assume a simplified acceleration model that includes perturbations due to the non-spherical central body. This perturbing acceleration on the satellite is obtained from the gradient of the gravitational potential of the non-spherical Earth that is modelled using spherical harmonics.[16] The perturbations arising from the second (J2), third (J3) and fourth (J4) harmonics are considered for our calculation. The components of the perturbing acceleration vector due to J2, J3, and J4 harmonics used for the calculation are found in[29] and are added linearly to equation (5).

Combining the state vector (2) and Cowell's second order equation of motion, the state of the satellite is reformulated into a first-order system as follows:

$$\dot{\vec{Y}} = \begin{bmatrix} \vec{v} \\ -\frac{\mu\vec{r}}{r^3} + \vec{a}_p \end{bmatrix} \quad [6]$$

Equation (6) is known as the variation of Cowell's formulation and is solved by using numerical integration methods.

**Coverage Analysis** The satellite observational area is the field of view from the satellite that projects a circular or rectangular footprint on the Earth. Access between the satellite and a target point in the footprint area at a given time represents the instantaneous coverage of the satellite. With the latitudes and longitudes of the sub-satellite point ($\Theta_s$, $\Lambda_s$), and the latitudes and longitudes of the target ($\Theta_t$, $\Lambda_t$), the value of Earth Central angle $\lambda$ is calculated using equation (7).[19]

$$cos\lambda = sin\Theta_s sin\Theta_t + cos\Theta_s cos\Theta_t cos|\Lambda_s - \Lambda_t| \quad [7]$$

Then the nadir $\eta$ is calculated using the design variable $\epsilon$ from equation (8) which is used to calculate the maximum earth central angle $\lambda_{max}$.

$$sin\ \eta_{max} = cos\ \epsilon_{min}\frac{R_E}{h + R_E} \quad [8]$$

$$\lambda_{max} = 90° - \epsilon_{min} - \eta_{min} \quad [9]$$

A total of three hundred grid points, evenly distributed among the intended region (Luxembourg, Belgium and Germany) are selected as targets for this study. The condition for the coverage, $\lambda < \lambda_{max}$ is checked for each grid point. The total temporal coverage is determined by equation (10).

$$C = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} T_{ij}}{nm} \quad [10]$$

where $C$ indicates the coverage performance of the constellation, $n$ is the number of time points considered, $m$ is the number of grid points and $T_{ij}$ is the coverage matrix. $C$ should be at least 70% as per the constraints set for the problem.

### *3.2 Payload*

Payload, being the most important subsystem of a satellite, drives the system design. Therefore, payload parameters such as size, weight and power requirement are needed at the initial stage of the satellite design. Usually for constellation missions, the payload is well defined and needs to be populated properly in orbit. However, in our case, the payload design is also optimized. It is unlikely to know the exact value of the payload parameters at the early stage. Therefore, we use viable estimation techniques to find its approximate value.[31] For earth observation mission, we consider an optical payload operating in a spectral range of $10.8\mu m$. Given the satellite altitude, the size of the payload is estimated using the relation in the following equations.

$$f = \frac{hd_x}{X/N_{samp}} \quad [11]$$

$$D = \frac{Bf}{Qd_x} \quad [12]$$

where h is the altitude, $f$ is the focal length, $D$ is the aperture diameter, $d_x$ is the width of cross-track detectors, $X$ is the cross-track ground pixel resolution, $N_{samp}$ is the cross-track detector samples in one pixel, $B$ is the operating wavelength and $Q$ is the quality factor for imaging. Based on the estimated aperture diameter $D_{payload}$, the mass $m_{payload}$, and power $P_{payload}$ of the payload is determined by sub-scaling from a reference payload using following relations.[31]

$$R = D_{payload}/D_0 \quad [13]$$

$$m_{payload} \approx KR^3W_0 \quad [14]$$

$$P_{payload} \approx KR^3P_0 \quad [15]$$

where $R$ is the aperture ratio, $D_0$, $W_0$, and $P_0$ are the aperture diameter, mass, and power of the reference payload respectively and $K$ is the scaling factor which is 2 when $R$ is less than 0.5, and 1 otherwise.

To increase the overall system coverage, we must ensure that the satellite footprints overlap. Therefore, the swath of a satellite must be greater than the successive node crossings at the Equator. This implies that the orbits maintain substantial margins at higher latitudes which ultimately increases the coverage. The swath of the satellite is given by $2\lambda_{max}$. Successive node crossings are determined from the perpendicular separation between the orbits as given in equation below.

$$S = sin^{-1}(sin(\Delta L)sin(i)) \quad [16]$$

where $\Delta L$ is the longitudinal shift per orbit and $i$ is the orbit inclination angle.

### *3.3 Power Subsystem*

The electrical power required to operate the satellite is generated by the Power subsystem with the help of solar panels. In addition to power generation, the power subsystem is also responsible for storing, distributing, and regulating the electrical power to each subsystem as needed. The area of solar panels influences the power generation as well as the size of the satellite. The size and capacity of the rechargeable battery to store the generated power depends on the duration of the satellite in eclipse.

**Solar Panel Sizing** The solar arrays must be sized such that the power generated by it is greater than the power required by the satellite. The area of the solar panel is determined by the amount of power required by the satellite. The power required by the satellite is calculated as follows:

$$P_{req} = \frac{P_{payload}T_{payload} + P_{thermal}T_e + P_{batt}T_e + P_{others}T}{T - T_e} \quad [17]$$

where $P_{payload}$ is the power required by the payload in time $T_{payload}$ , $P_{thermal}$ is the power required by the thermal subsystem operated during eclipse $T_e$, $P_{batt}$ is the power required to charge the battery, $P_{others}$ is the combined power required by the remaining for the total orbital period $T$. The power generated by the solar panel[27] depends on various factors as given in the following equation.

$$P_{gen} = S_0X_iX_sX_eX_0A_s\eta F_c(\beta_p\Delta T + 1)cos(\chi) \quad [18]$$

where $S_0 = 1367W/m^2$ is the solar constant, $X_i = 0.95$, $X_s = 0.9637$, $X_e = 1$ and $X_0 = 0.98$ are the correction factors, $A_s$ is the area of solar panel, $\eta$ is the photoelectric conversion efficiency, $F_c$ is the solar array loss coefficient, $\beta_p$ is the power temperature coefficient, $\chi$ is the worst-case sun vector deviation from the solar panel normal. Using required and generated power, the surplus power[27] is calculated as follows:

$$P_{surplus} = (1 - d_y)^{L_t}P_{gen} - (1 + 5\%)P_{req} \quad [19]$$

where $d_y$ is the annual power degradation of the solar panels and $L_t$ is the mission lifetime.

**Battery Sizing** During the eclipse, the power generated by the solar panel is zero. A rechargeable battery is required to maintain the supply of power to the satellite. The discharge capacity, $C$ of the battery depends on the duration of the eclipse and the power required during the eclipse. The Depth-of-Discharge, DOD of the battery, is taken as 80% of the rated capacity $C_{rated}$.

With the area of the solar panels as design variable and the estimated battery capacity, the mass of the power subsystem, $m_{power}$ is calculated from equation below.

$$m_{power} = \rho_sA_s + C_{rated}.V_{DB}/\mu_b \quad [20]$$

where $\rho_s$ is the areal density of solar array, $V_{DB}$ is the battery voltage and $\mu_b$ is the specific energy of the battery.

### *3.4 Thermal Subsystem*

The satellite in an orbit is subject to radiation from the sun, thermal radiation from the earth, and albedo. The temperature within the satellite must be maintained to keep the electronics in their operational range.

Excess heat collected inside the satellite is ejected to the outer space by the radiators located in sun-facing direction. The common heat sources are external environment and internal heat generation. While the common heat sinks are controlled heat rejection from the radiator and heat leaks from the insulation. Initially, the satellite is assumed to in steady-state equilibrium. The heat balance equation for the satellite is given by equation (21).[2,19]

$$Q_{source} = Q_{sink}$$
$$Q_{external} + Q_{internal} = Q_{Radiator} + Q_{MLI} \quad [21]$$

We assume that the predominant source of radiation is the sun and other external radiations are negligible. The entire satellite is encapsulated by Multi-Layer Insulation (MLI) except the faces where the radiators are mounted. The heat leaks from MLI are insignificant and is neglected in the calculation. Therefore, using Stefan-Boltzmann law the heat balance equation is rewritten as follows:

$$\alpha S_0 A_R + Q_{internal} = \varepsilon A_R \sigma T^4 \quad [22]$$

where $\alpha$ is the Absorbity of the material, $\varepsilon$ is the Emissivity of the material, $S_0$ is the solar constant, $\sigma$ is the Stefan-Boltzmann constant, $T$ is the satellite temperature, $A_R$ is the radiator area and $Q_{internal}$ is the internal heat generation. With equation (22) and $A_R$ as the design variable, the temperature $T$ is calculated for hot and cold cases. $Q_{internal}$ is assumed as 60% of the satellite power during hot case and 40% of the satellite power during cold case. The temperature during the hot case must not exceed 340K and the temperature during the cold case must not be less than 263K. Finally, the mass and power of the thermal subsystem are estimated by equations (23,24) respectively.

$$m_{thermal} = A_R \rho_R \quad [23]$$
$$p_{thermal} = \varepsilon \sigma A_R T^4 \quad [24]$$

where $\rho_R$ is the areal radiator density.

### *3.5 Structure Subsystem*

The structure of the satellite is the mechanical enclosure enveloping the satellite subsystems to protect them from the launch and space environments. The structural elements are treated as a separate subsystem for design and analysis purposes. This subsystem stays in contact with a launch vehicle, and it experiences severe static and dynamic loads. The load-carrying capacity of a satellite depends on the strength and stiffness of the structure subsystem that can be improved by careful selection of materials, and adequate reinforcement. However, it is necessary to keep the mass of the satellite as light as possible to reduce the launch cost. The shape of the satellite has significant importance, especially when solar panels, radiators or any other elements are mounted on its surfaces. Satellite exits not only in regular shapes such as cylinder, cuboid, sphere but also in irregular shapes.

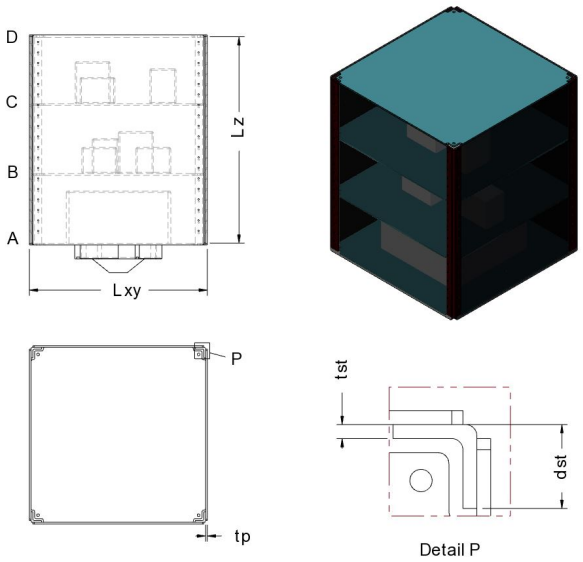


Fig. 2: Satellite cross section

The satellite considered here is a semi-monocoque cuboid whose length in X and Y directions are equal as shown in Fig. 2. It has four side panels, four stiffeners at the corners, and four trays/frames perpendicular to the stiffeners. The launch adapter in the launcher is attached to the outside of the bottom tray, A. The trays B and C hold payload and other subsystems, while the tray D covers the cuboid. The number of elements in our structural design is fixed and its dimensions are optimized. The space-grade aluminium alloy - AL7075 T6 is selected as the material for our design. Its material properties is indicated in Table 1. The structural model presented in this paper is based on the analytical structural design methodologies provided in the literature.[2,5,23,25] The design variables and constraints for structural optimization are the geometric dimensions and launch loads, respectively. The magnitude of launch loads varies with the launchers. We consider a launcher with severe launch loads shown in Table 2.

**Static Model** A static model or time-invariant satellite model is used to evaluate the structure under quasi-static limit loads imposed by the launcher. For pre-

Table 1: AL7075 T6 material properties

| $E$ (GPa) | $\nu$ | $G$ (GPa) | $\rho$ (kg/m$^3$) | $\sigma$ (MPa) | $\tau$ (MPa) |
|---|---|---|---|---|---|
| 71.7 | 0.33 | 26.9 | 2810 | 503 | 331 |

Table 2: Launch loads considered

| Launch load | Longitudinal | Lateral |
|---|---|---|
| Acceleration (g) | ± 10g | ± 7.5g |
| Frequency (Hz) | ≥ 90Hz | ≥ 60Hz |

liminary calculations, the model assumes the satellite is considered as a cantilever beam fixed at the base through the launch adapter. The satellite experiences the maximum axial load of 10g and a uniform lateral load of 7.5g. Then the maximum normal stress, $\sigma_{max}$ and maximum shear stress, $\tau_{max}$ are calculated from following equations.

$$A_{sat} = 4[A_b + t_p(L_{XY} - t_p)] \quad [25]$$

$$\sigma_{max} = \frac{M_{max} h_p}{2I_x} + \frac{F_{long}}{A_{sat}} \quad [26]$$

$$\tau_{max} = \frac{V_{max} Q}{I_x L_{XY}} \quad [27]$$

where $A_{sat}$ is the cross sectional area of the satellite, $L_{XY}$ is the satellite dimension in X and Y direction, $t_p$ is the thickness of the side panels, $I_x$ is the satellite moment of inertia, $M_{max}$ is the maximum bending moment and $V_{max}$ is the maximum shear force. The Lateral load, $F_{lat}$ and longitudinal load, $F_{long}$ is obtained by substituting lateral acceleration $a_{lat}$ and longitudinal accelerations $a_{long}$ in $F = m_{sat} \times a$, respectively. The calculated stress should be less than the yield strength of the material indicated in Table 1.

**Dynamic Model** The satellite should withstand both static and dynamic load applied to it. The classical methodology is to design for static loads and verify the design for dynamic loads or vice versa. However, in our optimization, a set of design variables is iteratively checked against both static and dynamic loads. A dynamic model of the satellite predicts the natural frequency of the satellite in the given load conditions. The dynamic model here is a four degrees-of-freedom spring-mass system. The mass $m_1$, $m_2$, $m_3$, and $m_4$ represent the lumped masses for trays A, B, C, and D respectively. The launch adapters and structural elements between the trays act like spring. The equations of motion in longitudinal and lateral directions are given by equations (28,29) respectively.

$$\begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix} \begin{bmatrix} \ddot{z}_1 \\ \ddot{z}_2 \\ \ddot{z}_3 \\ \ddot{z}_4 \end{bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 & 0 & 0 \\ -k_2 & k_2 + k_3 & -k_3 & 0 \\ 0 & -k_3 & k_3 + k_4 & -k_4 \\ 0 & 0 & -k_4 & k_4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad [28]$$

$$\begin{bmatrix} I_m & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix} \begin{bmatrix} \ddot{\varphi} \\ \ddot{x}_2 \\ \ddot{x}_3 \\ \ddot{x}_4 \end{bmatrix} + \begin{bmatrix} k_\varphi & 0 & 0 & 0 \\ 0 & k_5 + k_6 & -k_6 & 0 \\ 0 & -k_6 & k_6 + k_7 & -k_7 \\ 0 & 0 & -k_7 & k_7 \end{bmatrix} \begin{bmatrix} \varphi \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad [29]$$

$k_1$, $k_\varphi$ are the longitudinal and lateral stiffness of the launch adapter. $k_{2-4}$ are the longitudinal stiffness, and $k_{5-7}$ are the lateral stiffness of the structural elements between the trays A-B, B-C and C-D, respectively. $I_m$ is the mass moment of inertia of the satellite.

With the mass and stiffness matrices, the angular frequency of the satellite is obtained by solving the eigenvalue problem, $([K] - \omega_n^2[M]) = 0$. The first natural frequency is calculated by $f_n = \omega_n / 2\pi$. The calculated first lateral frequency and the first longitudinal frequency must be greater than launcher constraints shown in Table 2.

## 4. MDO Architectures

The MDO problem of the DSS is formulated to have a single objective function (to minimize the mass of DSS) with continuous design variables described in Table 3. The problem is solved using three different architectures MDF, IDF, and SAND. The main difference between the architecture lies in the handling of interdisciplinary coupling. For a fair comparison, all the architectures use the Sequential Least-Squares

Quadratic Programming (SLSQP) optimizer[17] and the convergence tolerance of the optimizer set to $1 \times 10^{-3}$.

To describe the sequence of operation and data interactions within the architecture, we use an extended Design Structure Matrix (XDSM) proposed by Martin and Lambe.[18] The conventions used in XDSM are as follows: (i) Rounded rectangle represents the optimizer that controls the entire optimization. (ii) Rectangular-shaped nodes represent discipline modules and are placed along the diagonal. (iii) Parallelogram shaped nodes represent data and results. (iv) Thick grey lines indicate data flow and thin black lines indicate process flow. (v) Data flow in vertical direction represents the input to the module while the data flow in horizontal direction represents the output from the module. (vi) In addition to the thin black lines, a numbering system is used to indicate the process flow. (vii) The process flows from module-0 and continues in sequential order up to module-n. (viii) $i \rightarrow j$ represents loops executed within the architecture such that a process $i$ is followed by process $j$ until a specified condition is met. (ix) Data external to the optimizer, such as design variables initial guesses $x^{(0)}$, variables at their optimum $x^*$, and discipline-specific variables are placed in the outer nodes.

IDF and SAND are decoupled while, MDF requires a solver to handle the coupling.The following sub-sections present a brief description of problem formulations in MDF, IFD, and SAND architectures and their respective XDSM diagrams for the given DSS model.

### *4.1 Simultaneous Analysis and Design (SAND)*

As the name suggests, the SAND architecture simultaneously analyses and designs the system. This is performed by including state variables ($\bar{y}$) and coupling variables ($y$) from each discipline to the set of design variables ($x$). The analysis models are reformulated to provide residuals for each disciplinary equations. These residuals are treated as equality constraints for this architecture. The general mathematical formulation of SAND architecture[21] is shown below.

$$\begin{aligned} \min \quad & f(x, y) \\ \text{w.r.t.} \quad & x, y, \bar{y} \\ \text{s.t.} \quad & g_i(x_0, x_i, y_i) \geq 0 \quad \text{for } i = 1, ..., N \\ & R_i(x_0, x_i, y, \bar{y}_i) = 0 \quad \text{for } i = 1, ..., N \end{aligned} \tag{30}$$

The XDSM of the DSS problem implementation in SAND architecture is shown in Fig. 3. The design variables and their acceptable range are given in Table 3.

### *4.2 Multidisciplinary Feasible (MDF)*

MDF architecture consists of Multidisciplinary analysis (MDA) modules over which the optimizer is placed. This implies that at each iteration of MDF, a multidisciplinary feasible solution is present. The set of design variables ($x$) are passed into the MDA modules which iterate over the discipline analysis models until a consistent set of coupling variables ($y$) is generated. Then the design variables and the resultant coupling variables are used to compute the objective and constraints. Typical iterative solvers such as block Gauss-Seidel, Newton solver are used to solve the MDA. The general mathematical formulation of MDF architecture[21] is shown below.

$$\begin{aligned} \min \quad & f(x, y(x, y)) \\ \text{w.r.t.} \quad & x \\ \text{s.t.} \quad & g_i(x_0, x_i, y_i(x_0, x_i, y_i)) \geq 0 \quad \text{for } i = 1, ..., N \end{aligned} \tag{31}$$

The XDSM of the DSS problem implementation in MDF architecture is shown in Fig. 4. Nonlinear Block Gauss Seidel iterative solver (NLBGS) and Newton solver is used to solve the MDA of the DSS.

### *4.3 Individual Discipline Feasible (IDF)*

Decoupling the MDF architecture results in IDF architecture. To decouple the MDF, copies of coupling variables ($\hat{y}$) are added to the design variable set ($x$). This design variable set is then used to compute the objective and constraints at each iteration. Multidisciplinary feasibility is ensured by the usage of consistency constraints, $g_c = \hat{y} - y$. The general mathematical formulation of IDF architecture[21] is given by:

$$\begin{aligned} \min \quad & f(x, y(x, \hat{y})) \\ \text{w.r.t.} \quad & x, \hat{y} \\ \text{s.t.} \quad & g_i(x_0, x_i, y_i(x_0, x_i, \hat{y}_i)) \geq 0 \quad \text{for } i = 1, ..., N \\ & g_c = \hat{y}_i - y_i(x_0, x_i, \hat{y}_i)) = 0 \quad \text{for } i = 1, ..., N \end{aligned} \tag{32}$$

The XDSM of the DSS problem implementation in IDF architecture is shown in Fig. 5.

Table 3: Design Variables of Optimization

| Variable | Symbol | Unit | Range | Initial Guess |
|---|---|---|---|---|
| Altitude | $h$ | km | [800,1000] | 900 |
| Inclination | $i$ | deg | [53, 55] | 53.5 |
| Elevation Angle | $\epsilon$ | deg | [15, 25] | 15 |
| Length in X & Y direction | $L_{xy}$ | m | [0.5, 1.2] | 1 |
| Length in Z direction | $L_z$ | m | [0.5, 1.2] | 1 |
| Thickness of panel | $t_p$ | m | [0.001, 0.005] | 0.005 |
| L-bar width | $d_{st}$ | m | [0.02, 0.05] | 0.03 |
| L-bar Thickness | $t_{st}$ | m | [0.001, 0.005] | 0.005 |
| Length Ratio between plate A and B | $AB$ | - | [0.3, 0.5] | 0.325 |
| Length Ratio between plate B and C | $BC$ | - | [0.25, 0.375] | 0.25 |
| Area of Solar Panel | $A_s$ | $m^2$ | [1, 5] | 2 |
| Area of Radiator | $A_r$ | $m^2$ | [0.1, 2] | 1.06 |
| Mass of Structure* | $m_{struc}$ | kg | [4, 50] | 15 |
| Mass of Satellite* | $m_{sat}$ | kg | [65, 150] | 100 |
| Power required by Satellite* | $P_{sat}$ | W | [300, 750] | 400 |
| Power required by Thermal Subsystem* | $P_{thermal}$ | W | [30, 200] | 90 |
| Power required by Payload** | $P_{payload}$ | W | [20, 200] | 40 |

* Needed by IDF and SAND
** Needed by SAND

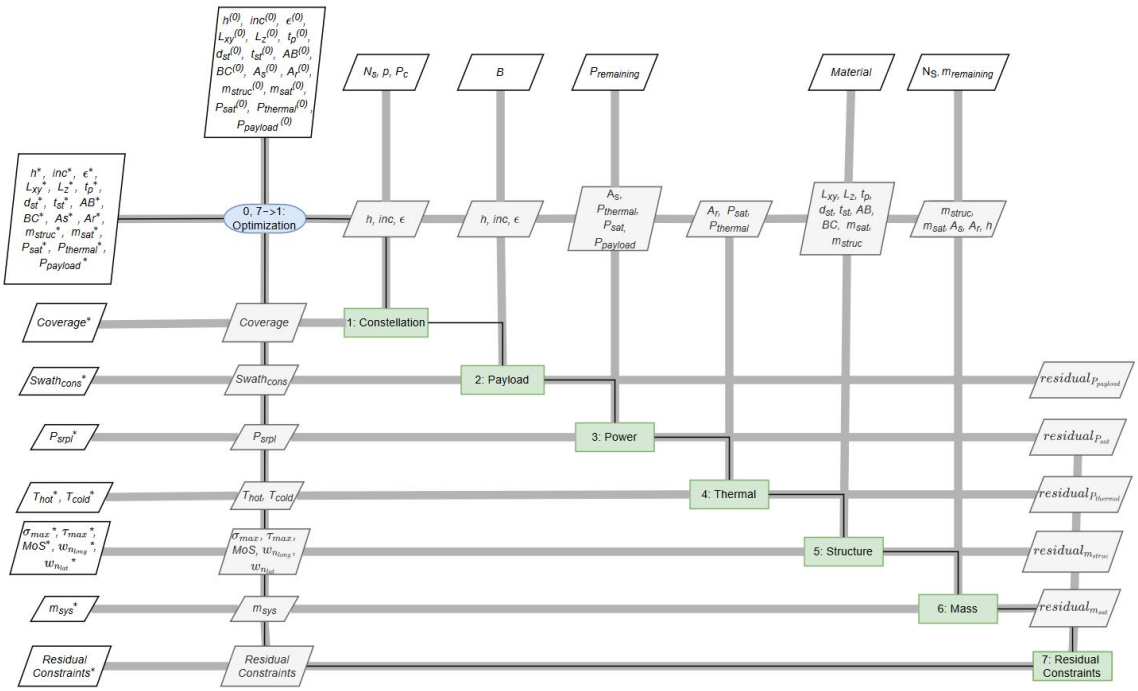


Fig. 3: DSS XDSM diagram for SAND

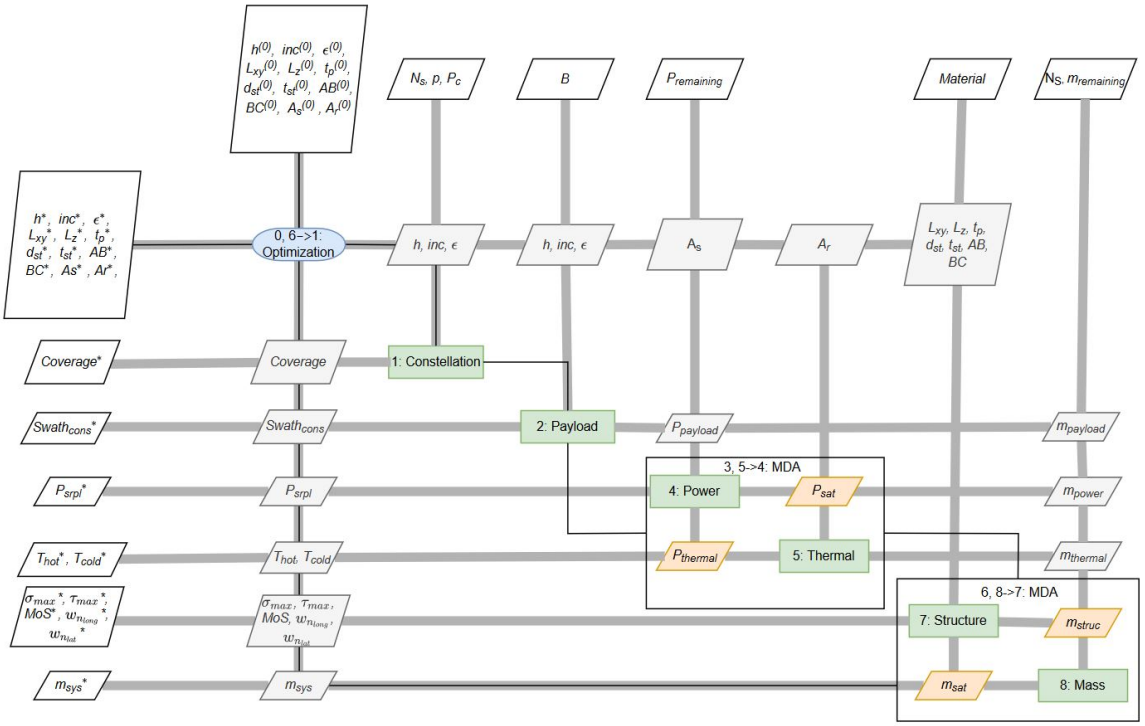


Fig. 4: DSS XDSM diagram for MDF

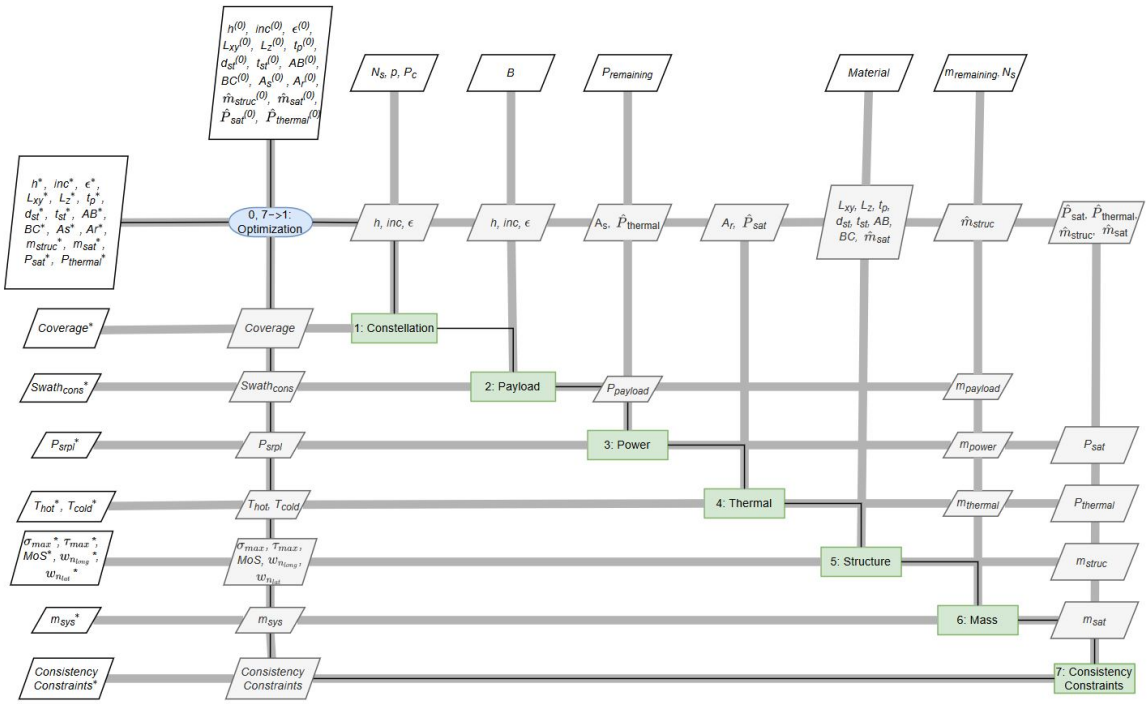


Fig. 5: DSS XDSM diagram for IDF

## 5. Results and Discussions

The DSS design problem was solved successfully in all three architectures. IDF, SAND, and MDF are optimized using SLSQP optimizer. For MDF, a solver is needed to handle the coupling between the disciplines. For completeness, the MDF architecture is optimized for two cases: (i) using a gradient-free Nonlinear Block Gauss Seidel (NLBGS) solver and (ii) using a gradient-based Newton solver. Additionally, a linear Direct solver is needed for the Newton solver to compute to the derivatives. Comparison between the solvers demonstrates the differing performance of the same problem within the architecture. The optimization results obtained from each architecture are shown in Table 4.

There are multiple ways to measure the effectiveness of an architecture. The most obvious way is to compare the optimum objective values attained by each architecture. However, other metrics such as Total Function Evaluations and Convergence Characteristics are also important for useful comparison.

### *5.1 Objective Value*

IDF provided the minimum objective value, i.e. $m_{sys}$. However, there are no significant changes in $m_{sat}$ between the architectures. The nonlinear solvers used in MDF introduces a noise during convergence which causes the variations of the results. Since the resultant mass of DSS is only an estimation, minor variations do not significantly impact the DSS design. However, the accumulation of error grows with the addition of subsystems. It is possible to minimize the error by using analytical derivatives instead of numerical approximation, in the later stages.

### *5.2 Total Function Calls*

The computation power required by the architecture is indicated by the number of functions evaluated/called while optimization. The calls to calculate the derivatives are also included in the functional call counts recorded for each subsystem. The number of function calls to each subsystem for every architecture is shown in Table 5. IDF supersedes the other architectures with the lowest function calls. Since MDF converges to a multidisciplinary feasible design at each iteration it has the most function calls. Additionally, the presence of Direct solver induces a spike in function calls of MDF$_{Newton}$.

The optimizations were run on an intel COREi7 7th Gen processor. The computational time required for each architecture is given in Table 4. Due to careful selection of design variable ranges combined with the calculation of DSS coverage for a period of 12 days, the computation time required is significantly lower than the time generally required to solve a problem of this size. In the actual scenario, the design variable have wide ranges and the temporal coverage is calculated for the entire mission lifetime. This exponentially increases the computational time. However, the results obtained are sufficient for the comparison of architectures. The execution times given in Table 4 are consistent with the number of function calls. As expected, the MDF$_{Newton}$ had the highest execution time and IDF had the lowest.

### *5.3 Convergence Characteristics*

The path taken for the optimization problem to arrive at the optimum value is given by the convergence characteristics of the problem. Due to the complexity, the exact optimal solution for the DSS problem is not known. Therefore, the lowest objective value calculated is considered as the exact optimal solution for this comparison. The relative error is calculated using the following equation.

$$\text{Relative Error} = \frac{|f - f_{min}|}{f_{min}} \quad [33]$$

where $f$ is the objective of the targeted architecture and $f_{min}$ is the corresponding objective from the architecture with the lowest objective value. The convergence rate of optimization for each architecture is shown in Figure 6. IDF architecture exhibits a clear convergence trend while MDF$_{Newton}$ required more iterations to converge. The convergence behaviour will be different when analytical derivatives are used instead of numerical approximation.[10]

Table 4: Optimization results

| | Unit | IDF | SAND | MDF$_{Newton}$ | MDF$_{NLBGS}$ |
|---|---|---|---|---|---|
| $m_{sys}$ | kg | 2477 | 2502 | 2560 | 2660 |
| $m_{sat}$ | kg | 99.11 | 100.09 | 102.4 | 106.43 |
| h | km | 917.2 | 913.3 | 918.92 | 964.2 |
| $i$ | deg | 55 | 55 | 53.57 | 53.5 |
| $\epsilon$ | deg | 18.64 | 18.56 | 17.83 | 18.88 |
| $L_{xy}$ | m | 0.54 | 0.66 | 0.55 | 0.56 |
| $L_z$ | m | 0.8 | 0.66 | 0.8 | 0.79 |
| $t_p$ | m | 0.001 | 0.001 | 0.001 | 0.001 |
| $d_{st}$ | m | 0.05 | 0.05 | 0.05 | 0.049 |
| $t_{st}$ | m | 0.001 | 0.001 | 0.001 | 0.003 |
| $AB$ | - | 0.25 | 0.25 | 0.25 | 0.366 |
| $BC$ | - | 0.25 | 0.25 | 0.25 | 0.316 |
| $A_s$ | m | 2.38 | 2.364 | 2.364 | 2.38 |
| $A_r$ | m | 0.876 | 0.877 | 0.878 | 0.887 |
| $m_{struc}$ | kg | 9.374 | 10.59 | 9.47 | 11.763 |
| $P_{thermal}$ | W | 55.75 | 55.78 | 55.89 | 56.43 |
| $P_{payload}$ | W | 99 | 98.608 | 106 | 109.6 |
| $P_{sat}$ | W | 566.96 | 567.25 | 568.37 | 565.84 |
| $Coverage$ | % | 70 | 70 | 70.1 | 0.7 |
| $P_{srpl}$ | W | 5.48 | 1.13 | 0 | 6.8 |
| $T_{hot}$ | K | 340 | 340 | 340 | 340 |
| $T_{cold}$ | K | 276.55 | 276.55 | 276.56 | 275 |
| MoS | - | 1.8 | 2.17 | 1.86 | 3.6 |
| $w_{n_{long}}$ | Hz | 74 | 73.76 | 72.8 | 71 |
| $w_{n_{lat}}$ | Hz | 60 | 72.41 | 60 | 62 |
| Execution time | h | $\leq 1$ | $\sim 1$ | $\sim 6$ | $\sim 2$ |

Table 5: Function evaluation counts

| Architecture | Constellation | Payload | Power | Thermal | Structure | Mass | Total |
|---|---|---|---|---|---|---|---|
| IDF | 122 | 82 | 162 | 64 | 122 | 122 | 674 |
| SAND | 226 | 198 | 310 | 170 | 254 | 254 | 1412 |
| MDF$_{Newton}$ | 3020 | 3020 | 17362 | 11072 | 23813 | 23813 | 82100 |
| MDF$_{NLBGS}$ | 964 | 964 | 2181 | 2181 | 2380 | 2380 | 11050 |

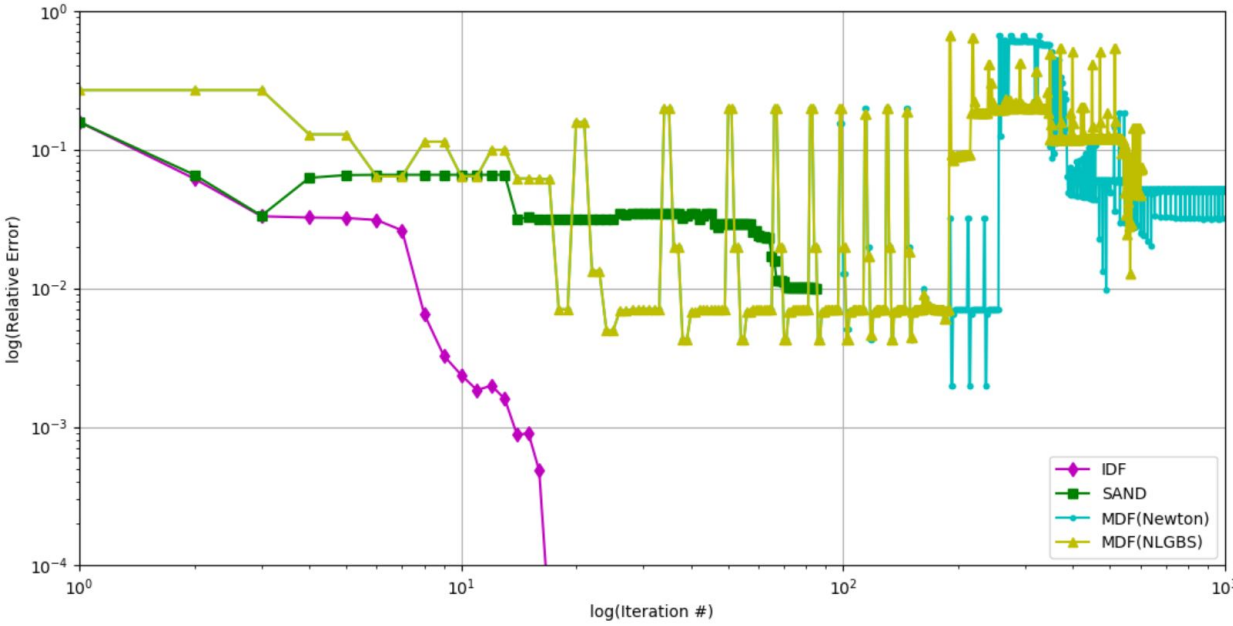


Fig. 6: Objective Convergence

## 6. Conclusion and Future Work

In this paper, a Distributed Space System was analytically modelled. The three MDO architectures MDF, IDF and SAND implementation of the DSS problem was then optimized. The results from the optimization showed that MDF needed the highest computational resources compared to other architectures. MDF solved using NLBGS introduced noise in the objective value while MDF solved using Newton solver had highest function calls. SAND had better results compared to MDF, in terms of optimality, computation, and convergence. However, IDF outperformed the all the architectures in terms of optimality, computational efficiency, and convergence rate. Based on the comparison it can be concluded that IDF is preferable for the conceptual design of DSS followed by SAND. At later stages of the DSS design, the simple analytical models are not sufficient. More detailed design has to be developed considering all the subsystems and using high fidelity simulation models. This further complicates the optimization and makes it more expensive to compute. From an engineering design context, it might be beneficial to get an improved design that need not be an optimal one in a strict mathematical sense. In such cases, MDF architecture is advantageous, provided that the optimization maintains the feasible design path, as it offers a multidisciplinary feasible solution at each iteration.

Extension of this research work shall include additional disciplinary models such as ADCS, OBDH, Propulsion and TT&C to enhance the existing DSS model. From the experience gained on their behaviour, the problem shall be translated into a mixed-integer non-linear problem that also considers the discrete variables for optimization. Comparison of Distributed architectures shall be performed for this problem in the future. Possible additional research includes the wrapping of gradient-based optimizer under a gradient-free caller to further explore the design space.

## Acknowledgements

This work is supported by FNR "Fonds national de la Recherche" (Luxembourg) through Industrial Fellowship Ph.D. grant (ref.12687511)

## References

[1] Jonathan R Behrens and Bhavya Lal. Exploring trends in the global small satellite ecosystem. *New Space*, 7(3):126–136, 2019.

[2] Peter Berlin. *Satellite platform design*. Kiruna Space and Environment Campus [Kiruna rymd- och miljöcampus], 2004.

[3] Leon Blitzer. Handbook of orbital perturbations. *University of Arizona*, 1970.

[4] Nichols F Brown and John R Olds. Evaluation of multidisciplinary optimization techniques applied to a reusable launch vehicle. *Journal of Spacecraft and Rockets*, 43(6):1289–1300, 2006.

[5] Elmer Franklin Bruhn. *Analysis and design of flight vehicle structures*. Tri-State Offset Co., 1965.

[6] Young-Keun Chang, Ki-Lyong Hwang, and Suk-Jin Kang. Sedt (system engineering design tool) development and its application to small satellite conceptual design. *Acta Astronautica*, 61(7-8):676–690, 2007.

[7] Brian W Chell, Steven Hoffenson, and Mark R Blackburn. A comparison of multidisciplinary design optimization architectures with an aircraft case study. In *AIAA Scitech 2019 Forum*, page 0700, 2019.

[8] WILLIAM A CROSSLEY and Edwin A Williams. Simulated annealing and genetic algorithm approaches for discontinuous coverage satellite constellation design. *Engineering Optimization+ A35*, 32(3):353–371, 2000.

[9] Eric Frayssinhes. Investigating new satellite constellation geometries with genetic algorithms. In *Astrodynamics Conference*, page 3636, 1996.

[10] Justin Gray, Kenneth T Moore, Tristan A Hearn, and Bret A Naylor. Standard platform for benchmarking multidisciplinary design analysis and optimization architectures. *AIAA journal*, 51(10):2380–2394, 2013.

[11] Justin S Gray, John T Hwang, Joaquim RRA Martins, Kenneth T Moore, and Bret A Naylor. Openmdao: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104, 2019.

[12] Quan He and Chao Han. Satellite constellation design with adaptively continuous ant system algorithm. *Chinese Journal of Aeronautics*, 20(4):297–303, 2007.

[13] KF Hulme and CL Bloebaum. A simulation-based comparison of multidisciplinary design optimization solution strategies using cascade. *Structural and Multidisciplinary Optimization*, 19(1):17–35, 2000.

[14] Srinivas Kodiyalam. Evaluation of methods for multidisciplinary design optimization, phase 1. *NASA CR-1998-208716*, 1998.

[15] Srinivas Kodiyalam, Charles Yuan, and Jaroslaw Sobieski. Evaluation of methods for multidisciplinary design optimization, part 2. *NASA CR-2000-210313*, 2000.

[16] Yoshihide Kozai. The motion of a close earth satellite. *The Astronomical Journal*, 64:367, 1959.

[17] D Kraft. A software package for sequential quadratic programming. *DLR German Aerospace Research Center Institute for Flight Mechanics TR, Koln, Germany*, 1998.

[18] Andrew B Lambe and Joaquim RRA Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46(2):273–284, 2012.

[19] Wiley J Larson and James Richard Wertz. *Space mission analysis and design*. Microcosm, Inc., 1992.

[20] Christopher Marriage and Joaquim Martins. Reconfigurable semi-analytic sensitivity methods and mdo architectures within the pimdo framework. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 5956, 2008.

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

[22] Ruben Perez, Hugh Liu, and Kamran Behdinan. Evaluation of multidisciplinary optimization approaches for aircraft conceptual design. In *10th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 4537, 2004.

[23] Ali Ravanbakhsh and Sebastian Franchini. Multi-objective optimization applied to structural sizing of low cost university-class microsatellite projects. *Acta Astronautica*, 79:212–220, 2012.

[24] Tyler GR Reid, Andrew M Neish, Todd F Walter, and Per K Enge. Leveraging commercial broadband leo constellations for navigation. In *Proceedings of the ION GNSS*, pages 2300–2314, 2016.

[25] Thomas P Sarafin and Wiley J Larson. *Spacecraft structures and mechanisms–from concept to launch*. Springer, 1995.

[26] Tania Savitri, Youngjoo Kim, Sujang Jo, and Hyochoong Bang. Satellite constellation orbit design optimization with combined genetic algorithm and

semianalytical approach. *International Journal of Aerospace Engineering*, 2017, 2017.

[27] Renhe Shi, Li Liu, Teng Long, Yufei Wu, and G Gary Wang. Multidisciplinary modeling and surrogate assisted optimization for satellite constellation systems. *Structural and Multidisciplinary Optimization*, 58(5):2173–2188, 2018.

[28] Nathan P Tedford and Joaquim RRA Martins. Benchmarking multidisciplinary design optimization algorithms. *Optimization and Engineering*, 11(1):159–183, 2010.

[29] David A Vallado. *Fundamentals of astrodynamics and applications*, volume 12. Springer Science & Business Media, 2001.

[30] John G Walker. Satellite constellations. *Journal of the British Interplanetary Society*, 37:559–572, 1984.

[31] James R Wertz, David F Everett, and Jeffery J Puschell. *Space mission engineering: the new SMAD*. Microcosm Press, 2011.

[32] James Richard Wertz. *Mission Geometry: Orbit and Constellation Design and Management*. Springer, 2001.

[33] Sang-Il Yi, Jung-Kyu Shin, and GJ Park. Comparison of mdo methods with mathematical examples. *Structural and Multidisciplinary Optimization*, 35(5):391–402, 2008.