

Multidisciplinary Design Optimization of Multi-Stage Launch Vehicle using Flight Phases Decomposition

Mathieu Balesdent^{1,2,3,a}, Nicolas Bérend¹, Philippe Dépincé³, Abdelhamid Chriette³

¹ System Design and Performance Evaluation Department, Onera, 91120 Palaiseau, France

² CNES, Launchers Directorate, 91023 Evry, France

³ Institut de Recherche en Communications et Cybernétique de Nantes, 44321 Nantes, France

Received 20 April 2010, Accepted 20 June 2010

Abstract – Optimal design of launch vehicles is a complex multidisciplinary problem. The traditional way to optimize such vehicles is to decompose the problem into the different disciplines and to combine specific solvers and a global optimizer. This paper presents a new MDO method to optimize the configuration of an expendable launch vehicle and describes the application of this method to the design optimization of a small liquid fueled multi-stage launch vehicle. The objective is the payload mass maximization. The proposed bi-level method splits up the original MDO problem into the flight phases and optimizes concurrently the different stages of the launch vehicle. This transversal decomposition allows to reduce the search domain at system-level because the optimization problem at this level is analog to a coordination one. Two formulations of the method are presented and the results are compared to the ones obtained by a standard MDO method.

Keywords: Multidisciplinary Design Optimization, MDO, Launch Vehicles Design, Optimal control, Multi-level optimization

1 Introduction

Optimal design of launch vehicles is a complex task which requires adapted tools to handle the numerous disciplines (*e.g.* propulsion, aerodynamics, structure, mass budget, sizing or trajectory optimization) present in the design problem. These disciplines (which can be coupled) may have antagonistic objectives (*e.g.* structure and mass budget) and the global optimal design might not correspond to the optimum in each discipline. Therefore, specific methods (called Multidisciplinary Design Optimization - MDO - methods) have to be employed in order to find the optimal launch vehicle design.

Optimal design of launch vehicle is a specific MDO problem because it combines optimizations of design and trajectory parameters. Indeed, trajectory optimization is responsible for the launch vehicle performance calculation and the optimal design is strongly dependent of the optimal trajectory found. The most used MDO method in literature to solve the launch vehicle design problem consists in combining disciplinary analyzers and a global optimizer (Multi Discipline Feasible method - MDF - [1], [3], [9]). In this method, the trajectory is considered as a standard discipline and is optimized in its whole. In this method, the global optimizer has to handle all of the design parameters and is subject to a great number of constraints that may induce several problems during the optimization (scaling, important dimension of search domain, lack of robustness with respect to trajectory parameters or initialization, *etc.*).

In order to improve the optimization process behavior and to reduce the dimension of the search space, a new method, based on a transversal stage-wise decomposition of the

launch vehicle design problem, is proposed. Each stage gathers all the disciplines and is optimized independently from the others. The trajectory is decomposed into elementary flight phases and each one of them is optimized during the different stages optimizations. The optimization method is composed of two levels of optimization. The subsystem-level is composed of the different stages optimizations and the system-level is responsible for the coordination of the optimization process. Each subsystem optimization is analog to the optimization of a single stage launch vehicle, which is easier to solve.

The decomposition into the flight phases has been applied to optimize the trajectory of reusable launch vehicles [6], [7], [8]. This process consists in optimizing separately the different flight phases (Ascent, Orbiter and Return To Launch Site phases) in order to split up the original optimization problem into small ones to improve the optimization process efficiency. In this paper, we propose an extension of this method to the complete multidisciplinary design optimization of expendable launch vehicles.

This paper is organized as follows: in Section 2, we describe the MDO formulations and we present the different MDO methods used in launch vehicle design. In Section 3, we present the application case. In this section, we define the objective, the parameters, the models and the MDO methods formulations. Two formulations of the proposed method will be explained. Section 4 presents the results obtained and compares the different methods described in the previous sections.

^a Corresponding author : mathieu.balesdent@onera.fr

2 MDO formulations and design methods

A MDO problem can be resumed as follows:

$$\begin{aligned}
 &\text{minimize} && f(y, z) \\
 &\text{with respect to} && z \\
 &&& g(y, z) \leq 0 \quad (1) \\
 &\text{subject to} && h(y, z) = 0 \quad (2)
 \end{aligned}$$

where :

- $f(y, z)$: objective function.
- z : design variables. They can be used in one or several subsystems $z = \{z_{sh}, \bar{z}_k\}$. The subscript sh stands for the variables which are shared between different subsystems (global variables) and \bar{z}_k denotes the variables which are specific to one subsystem (local variables). Moreover, we use the notation z_k to describe the variables which refer to the k^{th} subsystem.
- y : coupling variables. These variables are used to link the different subsystems and to evaluate the consistency of the design with regard to the coupling.
- g : inequality constraints.
- h : equality constraints.

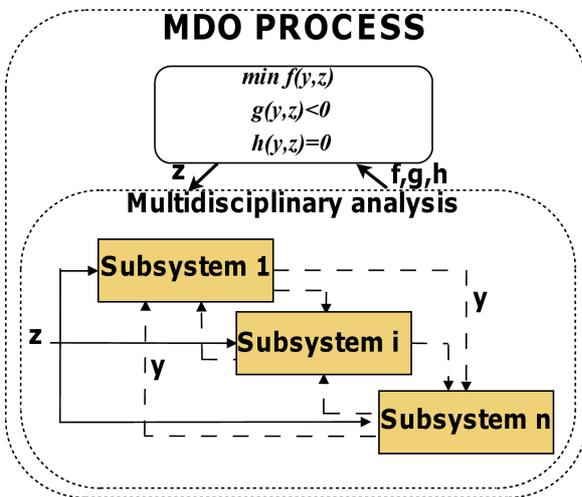


Fig. 1. MDO problem

2.1 Classical design methods

Launch vehicle design is a specific MDO problem because it combines the optimizations of design parameters and a control law. The classical engineering method to solve this problem consists in an interlacing of design parameters and trajectory optimizations. It generally ensues a loop between the design parameters optimization and the control law calculation (analogy with fixed point method). This process is very time consuming.

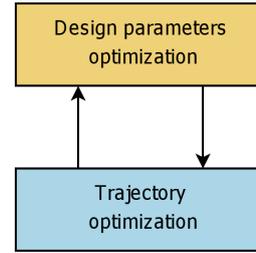


Fig. 2. Classical engineering method

The classical MDO method used in launch vehicle design consists in decomposing the process architecture into the different disciplines and using a MultiDiscipline Feasible (MDF) method. Several other methods (Collaborative Optimization, Bi-Level Systems Synthesis, All-At Once ...) have been tested in launch vehicle design but their use is sporadic. The MDF method consists in a MultiDisciplinary Analysis (MDA) and a global optimizer. The MDA is a process which calls disciplinary analyzers and ensures the consistency of the process with regard to the disciplinary feasibility and the coupling. A global optimizer is used to handle the optimization parameters and to satisfy the constraints. When the disciplines are coupled, the MDA generally consists in a loop between the disciplines. The MDA is performed at each iteration of the global optimizer. In the MDF method, the optimizer has to handle a large number of variables, especially when trajectory parameters are considered at the same level as the design parameters [3].

Therefore, the important size of the optimizer search domain and the use of MDA at each iteration generate several drawbacks as an important calculation time, a well tuned initialization requirement, an accurate determination of design parameters variation domain (especially for the trajectory ones) or a particular sensitivity to multiple optima phenomena.

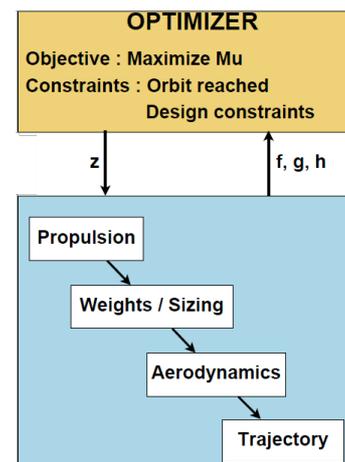


Fig. 3. MDF method

2.2 Stage-wise decomposition

Principle

Design problem decomposition

In order to have an efficient optimization process, a new decomposition of the design problem is proposed. The principle of this method consists in decomposing the initial

optimization problem into smaller ones which can be solved more easily. To this end, the optimization process considers each stage independently and transforms the initial optimization problem to a coordination of elementary ones.

The proposed method is a bi-level one in which the subsystem-level is composed of the different stages optimizations and the system-level is in charge of coordinating the subsystems optimizations (Fig. 4). In that way, each stage optimization is equivalent to the one of a single-stage launch vehicle for an elementary trajectory part, which is a less complex problem and can be quickly solved. Each stage optimization gathers all the different considered disciplines.

In this manner, for a n -stage launch vehicle, the stage-wise decomposition allows to split up the initial problem into n coordinated smaller one.

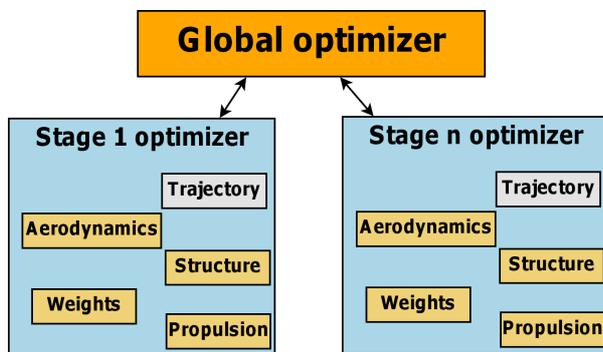


Fig. 4. Stage-wise decomposition

Trajectory handling

The global trajectory is decomposed into n elementary parts and the trajectory parameters corresponding to the different stages flight phases are optimized with the corresponding design parameters in the subsystems optimizations.

In order to take the couplings between the different stages into account, additional variables and constraints are introduced into the initial problem. These ones relate to the state vector at stages separations and the estimated mass of the intermediary and final stages. These variables are optimized by the system-level optimizer at the same time as the shared design parameters z_{sh} .

Thus, the trajectory optimization is performed at both system (state vectors at stages separations) and subsystem (elementary flight phases) levels.

Algorithms used

In order to have an efficient optimization process, we have chosen to implement an hybrid method which associates exploratory algorithm and local search method (Sequential Quadratic Programming) at system-level. First, the exploratory search is performed. The best found design (satisfying the constraints) is used to initialize the local SQP method. Thus, the optimization process takes advantage of the global search algorithm particularity to explore large solutions space and the local method one to efficiently converge to an optimum.

At subsystem-level, the same SQP algorithm is used at both phases I and II, because the optimization problems to solve have small dimension and can be correctly initialized.

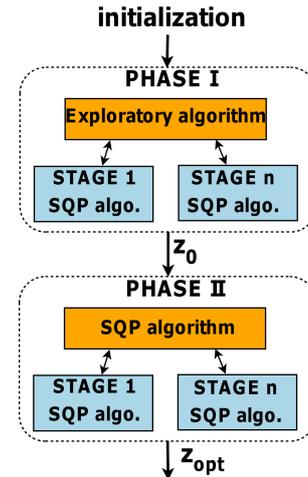


Fig. 5. Optimization process

3 Optimization of a micro liquid fueled launch vehicle

3.1 Presentation of the problem

The problem to solve consists in the optimization of a three-stage-to-orbit launch vehicle for the payload mass maximization. The Gross Lift-Off Weight (GLOW) is constrained to 10 tons. The orbit reached is a 250km altitude Low Earth Orbit. The disciplines considered are propulsion, aerodynamics, weights, and trajectory optimization. The launch vehicle is composed of three liquid fueled Liquid Oxygen/Kerosene (LOX/RP1) stages.

3.2 MDO formulation of the optimization problem

The design parameters used in the optimization process are resumed in Table 1.

Table 1. Design parameters

Disciplines	Stage 1	Stage 2	Stage 3	Payload fairing	Description
Weights	M_{p1}	M_{p2}	M_{p3}	M_u	Propellant mass
					Payload mass
Propulsion	Rm_1	Rm_2	Rm_3		Mixture ratio
	Pc_1	Pc_2	Pc_3		Chamber pressure
	Pe_1	Pe_2	Pe_3		Nozzle exit pressure
	$\left(\frac{T}{W}\right)_1$	$\left(\frac{T}{W}\right)_2$	$\left(\frac{T}{W}\right)_3$		Thrust to weight ratio
Aerodynamics	D_1	D_2	D_3	D_c Al_c	Diameter, Length to diameter ratio
Trajectory	u_1	u_2	u_3		Control law parameter vector

Using the MDO formulation presented in the previous section, the launch vehicle optimization problem can be formulated as follows :

Maximize $f(z) = Mu$

With respect to $z = \{Me_i, \frac{T}{W_i}, Rm_i, Pc_i, Pe_i, D_i, D_c, Alc, u_i, Mu\}$

Subject to

$$g_1(z) : Mu - Mu_{available\ under\ f\ airing} \leq 0 \quad (3)$$

$$g_{2..4}(z) : 0.4 - \frac{Pe_i}{Pa(r)} \leq 0 \quad (4)$$

$$g_{5..7}(z) : Dne_i - 0,8 D_i \leq 0 \quad (5)$$

$$h_1(z) : GLOW - 10000 = 0 \quad (6)$$

$$h_2(z) : r_f - r_{orbit} = 0 \quad (7)$$

$$h_4(z) : \gamma_f - \gamma_{orbit} = 0 \quad (8)$$

$$h_3(z) : v_f - v_{orbit} = 0 \quad (9)$$

where:

- Dne : nozzle exit diameter
- Pa : local atmospheric pressure
- r : radius
- v : velocity
- γ : flight path angle

The inequality constraint g_1 ensures the consistency of the fairing dimensions with regard to the payload, $g_{2..4}$ are flow separation constraints (Summerfield criterion), $g_{5..7}$ are geometry constraints which are introduced to avoid non-realistic designs despite the lack of structural calculation model. The equality constraint h_1 determines the total lift-off weight and $h_{2..4}$ ensure the satisfaction of the mission requirements.

3.3 Models

Propulsion

We have chosen to use liquid fueled stages LOX/RP1. The propulsion module aims to calculate the specific impulse (Isp) from Pc , Pe and Rm , by using standard propulsion equations [5].

Trajectory

The model chosen for the trajectory simulation is a bi-dimensional model with a non-rotating round Earth. The trajectory optimization is realized by using a direct method. The parametric control law (pitch angle) is composed of different waypoints θ_i which are optimized all along the trajectory. The pitch angle is calculated by piecewise linear function on θ_i .

$$\dot{r} = V \cdot \sin \gamma \quad (10)$$

$$\dot{V} = -\frac{1}{2} \rho(r) \frac{S_{ref} \cdot C_X(Mach)}{m} V^2 - g(r) \cdot \sin \gamma + \frac{F \cdot \cos(\theta - \gamma)}{m} \quad (11)$$

$$\dot{\gamma} = \left(\frac{V}{r} - \frac{g(r)}{V} \right) \cos \gamma + \frac{F \cdot \sin(\theta - \gamma)}{m \cdot V} \quad (12)$$

$$\dot{\phi} = -\frac{V \cdot \cos \gamma}{r} \quad (13)$$

$$\dot{m} = -q \quad (14)$$

with ρ : atmospheric density, S_{ref} : surface of reference, C_X : drag coefficient, $g(r)$: gravity acceleration, θ : pitch angle,

m : launch vehicle mass, F : thrust ($F = q \cdot g_0 \cdot Isp$) and q : mass flow rate (constant during the flight phase of each stage).

The axial load factor (n_f) is given by the following relationship:

$$n_f = \frac{g_0 \cdot q \cdot Isp(r, Pc, Pe, Rm) - \frac{1}{2} \rho S_{ref} V^2 C_x \cos(\theta - \gamma)}{m \cdot g_0} \quad (15)$$

Aerodynamics

We use a zero-lift aerodynamic. The drag coefficient is interpolated from the Mach during the trajectory evaluation. This model is generally sufficient in launch vehicles early phase studies.

Mass budget and geometry sizing

For simplification reasons, we do not perform a complete structural calculation but we use simplified models to determine the mass budget. For each stage, the different considered mass are:

- propellant mass
- dry mass

The upper composite mass is the sum of:

- payload mass
- adaptor mass
- fairing mass

From the propellant mass and the mixture ratio, the volumes and the surfaces of propellant and oxidizer tanks are defined. Then, we compute the dry mass (M_d) by the following simplified models:

$$M_d = (M_{tank} + M_{motor}) \cdot c_f \quad (16)$$

$$M_{tank} = k_r \cdot S_{tank} \quad (17)$$

$$M_{motor} = k_m \cdot q \quad (18)$$

with $S_{tank} = S_{LOX} + S_{RP1}$: the total surface of the tanks and c_f : a coupling coefficient between mass budget estimation and trajectory. This coefficient is linearly dependent of the maximal load factor endured by the stage (Table 2). In order to compute the dry mass response surface, the coefficients k_r et k_m have been determined by least-squares estimation on small liquid-fueled stages registered in ESA launch vehicles catalog[4].

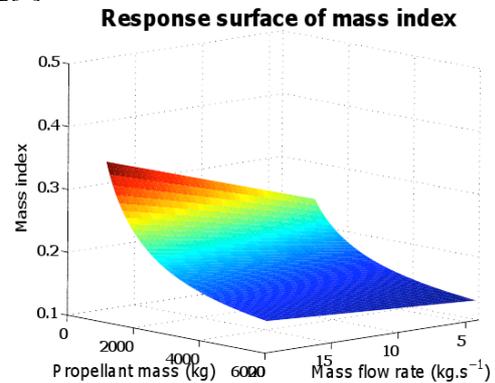


Fig. 6. Response surface of mass index (diameter and mixture ratio are fixed respectively to 1.4m and 2.4)

The GLOW is given by the formula:

$$GLOW = \sum_{i=1}^3 (M_{p_i} + M_{d_i}) + M_{f\ airing} + Mu + M_{adaptor} \quad (19)$$

Table 2. Variations of c_f with respect to the load factor

n_f	0	1	2	3	4	5
c_f	1	1.02	1.04	1.06	1.08	1.1

3.4 Optimization by MDF method

The optimization problem formulation of MDF method is the same as the original problem one. Indeed, a single optimizer has to handle all of the optimization variables and has to satisfy all of the constraints.

$$\begin{aligned}
\text{Maximize} \quad & f(z) = Mu \\
\text{With respect to} \quad & z = \{M\bar{e}_i, \frac{T}{W_i}, Rm_i, Pc_i, Pe_i, D_i, D_c, Alc, \\
& u_i, n_{f_{max\ estimated}}, Mu\} \\
g_1(z) : & Mu - Mu_{available\ under\ f\ airing} \leq 0 \quad (20) \\
g_{2...4}(z) : & 0.4 - \frac{Pe_i}{Pa(r)} \leq 0 \quad (21) \\
g_{5...7}(z) : & Dne_i - 0.8 D_i \leq 0 \quad (22) \\
h_1(z) : & GLOW - 10000 = 0 \quad (23) \\
h_2(z) : & r_f - r_{orbit} = 0 \quad (24) \\
h_3(z) : & v_f - v_{orbit} = 0 \quad (25) \\
h_4(z) : & \gamma_f - \gamma_{orbit} = 0 \quad (26) \\
h_5(z) : & n_{f_{max\ estimated}} - n_{f_{max\ calculated}} = 0 \quad (27)
\end{aligned}$$

In order to have a sequential MDA, an additional parameter $n_{f_{max\ estimated}}$ (and its equality constraint h_5) is introduced into the optimization problem. This parameter, used to compute the dry mass (Eq. 16), allows to decouple trajectory and mass budget and to avoid the use of a coupling loop between these disciplines during the MDA.

3.5 Optimization by decomposition into the flight phases

In this section, we present two formulations of the stage-wise decomposition method. The main difference between these two formulations concerns the state vector junction at stages separations.

First formulation

System-level optimization

In this formulation, the system-level optimizer handles the shared design variables z_{sh} and the coupling ones y . This optimizer attempts to maximize the payload mass Mu and to satisfy the junction constraints concerning the mass of second and third stages, which are respectively used by the first and second stages optimizers in order to optimize their configuration. These junction constraints have to be satisfied only at the convergence of phase II. In phase I, these equality constraints are replaced by inequality ones $Mu_{optimized} - Mu_{estimated} \leq 10kg$ in order to improve the exploratory search flexibility.

$$\begin{aligned}
\text{Maximize} \quad & f(z) = Mu \\
\text{With respect to} \quad & z_{sh} = \{D_1, D_2, D_3, D_c, Alc\}, \\
& y = \{r_{f1}, v_{f1}, \gamma_{f1}, \theta_{f1}, r_{f2}, v_{f2}, \gamma_{f2}, \theta_{f2}, \\
& Mu_{stage1\ estimated}, Mu_{stage2\ estimated}, n_{f_{max\ estimated}}\} \\
\text{Subject to} \quad & h_1 : Mu_{stage1\ estimated} - Mu_{stage1\ optimized} = 0 \quad (28) \\
& h_2 : Mu_{stage2\ estimated} - Mu_{stage2\ optimized} = 0 \quad (29)
\end{aligned}$$

Subsystem-level optimization

The optimizations at subsystem-level consist in optimizing the payload mass Mu_{stage_i} of the different stages, when the total mass of the stages are defined by the system-level optimizer and have to be satisfied at each iteration (h_{i5}). The payload mass of a stage is equivalent to the sum of the upper stages and the upper composite mass. In this formulation, at each iteration, the junction points at stages separations are consistent since the equality constraints $h_{i1...i5}$ are satisfied at subsystem-level.

Second formulation

$$\begin{aligned}
\text{Maximize} \quad & f_i(z_{sh}, \bar{z}_i, y) = Mu_{stage_i} \\
\text{With respect to} \quad & \bar{z}_i = \{Mp_i, \left(\frac{T}{W}\right)_i, Rm_i, Pc_i, Pe_i, \theta_i, Mu_{stage_i}\} \\
g_{i1} : & n_{f_{max\ stage_i\ calculated}} - n_{f_{max}} \leq 0 \quad (30) \\
g_{i2} : & 0.4 - \frac{Pe_i}{Pa(r)} \leq 0 \quad (31) \\
g_{i3} : & Dne_i - 0.8 D_i \leq 0 \quad (32) \\
g_{i4} : & Mu - Mu_{available\ under\ f\ airing} \leq 0 \quad (b) \quad (33) \\
\text{Subject to} \quad & h_{i1} : r_{f_i} - r_{t=f_i} = 0 \quad (34) \\
& h_{i2} : v_{f_i} - v_{t=f_i} = 0 \quad (35) \\
& h_{i3} : \gamma_{f_i} - \gamma_{t=f_i} = 0 \quad (36) \\
& h_{i4} : \theta_{f_i} - \theta_{t=f_i} = 0 \quad (c) \quad (37) \\
h_{i5} : & Mp_i + Md_i + Mu_{stage_i} - Mu_{stage_{i-1}\ estimated} = 0 \quad (d) \quad (38)
\end{aligned}$$

System-level optimization

In this formulation, the system-level optimizer has to optimize the same objective as the first formulation. On the other hand, the two equality constraints handling the mass junction are replaced by three ones which characterize the convergence of the different subsystems. In comparison with the first formulation, an additional parameter Mu is introduced to the system-level and the inequality constraint relating to the fairing consistency is transferred from the 3rd stage optimizer to the system-level one. In the same way as the first

(b) only for stage 3

(c) only for stages 1 and 2

(d) $Mu_{stage_0\ estimated} = GLOW$

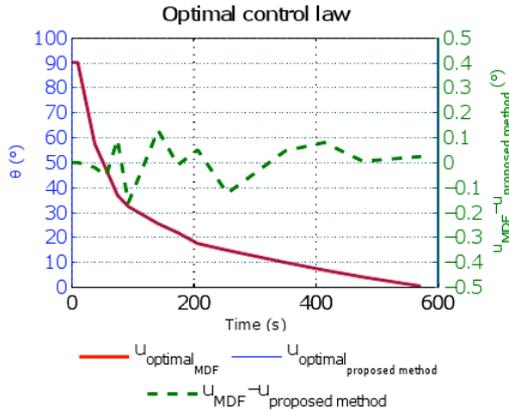
(b) Control law (u)

Fig. 7. Trajectory of the optimal design

4.2 Comparison of MDF and stage-wise decomposition formulations

Dimension of search spaces and numbers of constraints

By decomposing the optimization problem, the search domain of each optimization process at system and subsystem levels is considerably reduced, that makes the proposed method less sensitive to local optima phenomena than MDF method (in which the single optimizer has to handle a great number of variables).

Table 4 recapitulates the number of constraints and parameters handled by the MDF and stage-wise decomposition system-level optimizers. The stage-wise decomposition method allows to distribute constraints on system and subsystem levels.

Moreover, since the search domain at system-level is reduced, an exploratory algorithm can be used in order to initialize a local-search one. The joint use of these two kinds of algorithms avoids resorting to manual initialization. In the MDF method, the use of exploratory algorithm implies an important calculation time because the search domain is too large.

Table 4. Comparison of MDF and stage-wise decomposition formulations

	MDF	System-level, 1 st formulation	System-level, 2 nd formulation
Dimension of search space	34	16	17
Equality constraints	5	2	3
Inequality constraints	7	0	1

Numerical analysis of search space dimensions and number of constraints

In this section, we express the search space dimension and the number of constraints with respect to the problem characteristics, in order to compare the different formulations.

Dimension of search space

Let us consider:

- n_e : number of stages,
- n_x : dimension of the state vector,
- n_c : dimension of the control law parameters vector,
- n_{pp} : number of control law waypoints per stage,
- n_{psh} : number of design parameters per stage which play a part in the flight phases of several stages,
- $n_{f\text{airing}}$: number of parameters which are used to size the fairing,
- n_{pk} : number of design parameters per stage which only play a part in the k^{th} stage flight phase.

The number of parameters n_{MDF} handled by the MDF optimizer can be expressed as follows:

$$n_{MDF} = n_e(n_{pk} + n_c n_{pp} + n_{psh}) + n_{f\text{airing}} + 2 \quad (46)$$

where "2" stands for the payload mass and the estimated maximal axial load factor.

The numbers of parameters n_{f_1} , n_{f_2} handled by the system-level optimizers in case of the first and the second formulations of the proposed method are:

$$n_{f_1} = (n_e - 1)(n_x - 2) + (n_e - 1)n_c + n_e n_{psh} + (n_e - 1) + n_{f\text{airing}} + 1 \quad (47)$$

$$n_{f_2} = (n_e - 1)(n_x - 2) + (n_e - 1)n_c + n_e n_{psh} + n_e + n_{f\text{airing}} + 1 \quad (48)$$

where:

- $(n_e - 1)(n_x - 2)$: variables of state vector junctions (without mass and longitude)
- $(n_e - 1)n_c$: variables of control law junction (not required for the last stage)
- $n_e n_{psh}$: design parameters which are shared at the system-level (stages diameters)
- $n_e - 1^{(g)}$, $n_e^{(h)}$: estimated payload mass of the different stages
- 1 : estimated maximal load factor

Dimension of search space

The number of constraints $n_{c_{MDF}}$ for the MDF problem is:

$$n_{c_{MDF}} = 2n_e + (n_x - 1) + 2 \quad (49)$$

where:

- $2n_e$: constraints about the pressures and the nozzle exit diameters (Eq. 21, 22)
- $(n_x - 1)$: sum of numbers of constraints about the state at the final time $(n_x - 2)$ (Eq. 24-26) and the GLOW (1) (Eq. 23)
- 2 : constraints about the consistency of the fairing and the load factor (Eq. 20, 27)

The numbers of constraints $n_{c_{f_1}}$, $n_{c_{f_2}}$ handled by the proposed method system-level optimizers are given by:

$$n_{c_{f_1}} = n_e - 1 \quad (50)$$

^(g) 1st formulation

^(h) 2nd formulation

$$n_{c_{f_2}} = n_e + 1 \quad (51)$$

The evolutions of the search spaces dimensions and the number of constraints with respect to the number of stages are given in figure 8. We can see that the proposed method is all the more beneficial as the number of stages grows.

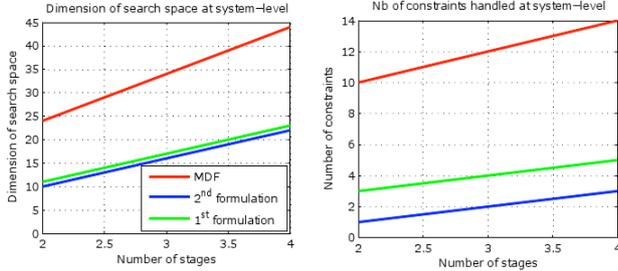


Fig. 8. Variations of search space dimension and number of constraints with respect to the number of stages

Modularity

The flight-phases decomposition method is very modular. Indeed, we can add or subtract a stage without modifying the design process architecture. Moreover, this architecture remains unchanged if the disciplinary codes are modified, which is not the case with the classical method.

Since the optimization problem is decomposed in two levels and the optimization problem at system-level is resumed to a coordination one, the proposed method allows to parallelize the optimizations of the different launch vehicle stages, in order to improve the efficiency of the optimization process.

Finally, the proposed method allows to freeze one or several stages while optimizing the others, which is useful when we want to reuse existing stages.

4.3 Comparison of first and second stage-wise decomposition formulations

The second formulation aims to improve convergence properties at the subsystem and system levels. Indeed, in this formulation, each stage does not optimize its payload Mu_{stage_i}

and is not subject to equality constraints anymore, but minimizes a quadratic sum of differences with respect to system-level instructions. This formulation associates the proposed stage-wise decomposition and the Collaborative Optimization [2].

At the system-level, the two mass consistency constraints are replaced by three equality constraints, which ensure the convergence of each stage optimization. In comparison with the first formulation, the system-level design problem includes one parameter, one equality and one inequality constraint. The objective Mu is directly optimized by the system-level optimizer. The equality constraints at system-level consist in the objectives of the different subsystems. In that way, we can expect a better behavior of the optimization process by using the second formulation (particularly in phase I) because the subsystem-level objectives minimization aims to satisfy the system-level equality constraints.

At the subsystem-level, all the equality constraints are removed and each subproblem dimension is reduced by one (Table 5). The relaxation of the equality constraints in the objective function aims to improve the subsystem-level optimizers behaviors and the robustness with respect to the system-level parameters. Indeed, the strict satisfaction of the coupling constraints (particularly the continuity of the state vector at stages separations) has not to be satisfied at each iteration of the system-level, but only at the convergence (Eq. 40-42), which is not the case through the first formulation.

Table 5. Comparison of stage-wise decomposition formulations at system and subsystem levels

	System-level	Stage 1	Stage 2	Stage 3	Total
Dimension of search space					
<i>First formulation</i>	16	10	10	10	46
<i>Second formulation</i>	17	9	9	9	44
Equality constraints					
<i>First formulation</i>	2	5	5	4	16
<i>Second formulation</i>	3	0	0	0	3
Inequality constraints					
<i>First formulation</i>	0	3	3	4	10
<i>Second formulation</i>	1	3	3	3	10

Comparison of phase 1 system-level optimizer (GA) for 1st and 2nd formulations

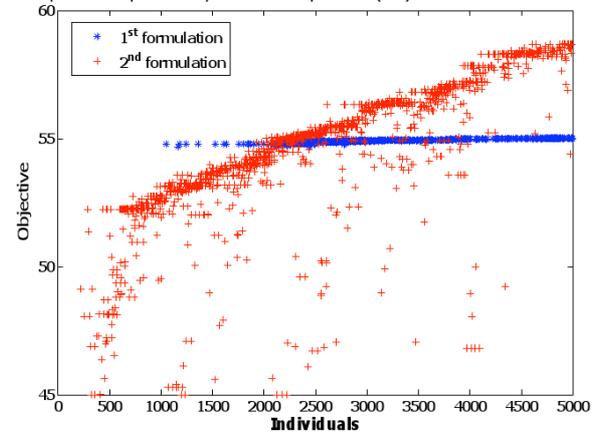


Fig. 9. Phase I optimization with first and second formulations (feasible points)

In order to quantify the improvements of the second formulation with respect to the first one, a comparative study has been realized. In this study, a genetic algorithm (100 generations of 50 individuals) with the same initialization and the same bounds of design parameters has been performed in case of both formulations. The results of the exploratory phase (Fig. 9) show that the second formulation is more adapted than the first one to the optimization problem and converges to the optimum while the first formulation limits the search around the first found design which satisfies the constraints. Study reveals that with the first formulation, 15% of the design points are non feasible (subsystem-level does not converge) but there are only 0.3% with the second formulation (Table 6). In that way, with the relaxation of the equality constraints to the objective function, the robustness

of the subsystem-level with regard to the system-level parameters is improved and the 2nd formulation leads to a better efficiency of the design process in its whole.

Table 6. Summary of formulations tests

	1 st formulation	2 nd formulation
Nb. of individuals for which subsystems converge	4257	4987
Nb. of individuals for which subsystems do not converge	743	13
Total of individuals	5000	5000

5 Conclusion and future works

In this paper, we have presented a new MDO formulation adapted to the expendable launch vehicle design problem. The proposed bi-level method allows to turn the initial optimization problem into a coordination one which can be solved easier. By decomposing the problem, the search domain at system-level is considerably reduced and the behavior of the optimization process is improved. Two formulations of the method have been proposed and compared. Results show that the collaborative formulation of the method is more adapted to the design problem and allows to improve the robustness of the subsystem-level with respect to system-level parameters.

Future works will consist in improving the models (particularly the mass budget models) in order to study the benefits of the proposed method in more complex problems. Moreover, different formulations and optimization algorithms will be tested in order to determine which are the most adapted to the optimization problem. Finally, other trajectory optimization methods (*e.g.* indirect method) may be more efficient than the currently used one and are worth studying in future works.

Acknowledgement

The work presented in this paper is part of a PhD thesis cofunded by CNES and Onera.

References

1. D. Bayley, R. Hartfield, J. Burkhalter and R. Jenkins, *Design Optimization of a Space Launch Vehicle Using a Genetic Algorithm*, Journal of Spacecraft and Rockets, **45**(4), 733-740, (2008).
2. R. Braun and I. Kroo. *Development and application of the Collaborative Optimization architecture in a multidisciplinary design environment*, Multidisciplinary Design Optimization : State of the Art, SIAM, 98-116, (1996).
3. N. Duranté, A. Dufour and V. Pain. *Multidisciplinary Analysis and Optimisation Approach for the Design of Expendable Launchers*, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New-York, USA (2004)
4. ESA Directorate of launchers, ESA Launch Vehicle Catalogue (2004)
5. R.W. Humble, G.N. Henry and W.J. Larson. *Space propulsion analysis and design*, The McGraw-Hill Companies, Inc. (1995)
6. L.A. Ledsinger and J.R. Olds *Optimized Solutions for Kistler K-1 Branching Trajectories Using Multidisciplinary Design Optimization Techniques*, Journal of Spacecraft and Rockets, **39**(3), 420-429, (2002).
7. L. Perrot. *Optimisation of reusable launchers by decomposition and coordination of trajectory optimization problems*, 5th International Conference on Launcher Technology, Space Launchers: Missions, Control and Avionics. Madrid, Spain (November 2003)
8. M. Rahn and U.M. Schöttle *Decomposition Algorithm for Flight Performance and System Optimization of an Air-breathing Launch Vehicle*, 45th Congress of International Astronautical Federation. Jerusalem, Israel, (October 1994)
9. T. Tsuchiya and T. Mori *Optimal Conceptual Design of Two-Stage Reusable Rocket Vehicles Including Trajectory Optimization*. Journal of Spacecraft and Rockets, **41**(5), 770-778, (2004).