Chapter 12

Multidisciplinary Optimization in the Design of Future Space Launchers

As industrial prime contractor ASTRIUM Space Transportation is responsible for the design of space launchers. A space launcher is a complex system, requiring contributions from many disciplines. In this chapter we will endeavour to limit ourselves to the launch vehicle design problem, which proves to be a multidisciplinary optimization problem.

12.1. The space launcher problem

The mission of a space launcher is to place a payload in a given orbit expressed as a speed at a given point. The payloads are classically near Earth artificial satellites, solar system exploration probes or spacecraft able to transport people.

The orbit required depends on the payload's missions. Low orbits are often used for Earth observation missions, geostationary orbits are the preferred orbits for communication satellites. By the term launcher, we therefore understand the vehicle that places a payload in orbit.

A launcher's objective is to communicate high energy to the payload with a gain in altitude (potential energy) and a gain in speed (kinetic energy). To obtain high energy in both the atmosphere and in vacuum, the technology adopted is the rocket motor, which is based on the action-reaction principle with the continuous ejection of fuel.

Chapter written by Guillaume COLLANGE and Nathalie DELATTRE and Nikolaus HANSEN and Isabelle QUINQUIS and Marc SCHOENAUER.

At present, a single body vehicle cannot be used to launch a satellite payload due to the available technologies; the principle used is to progressively eliminate the masses of structures that have ceased to be useful. This elimination is done progressively by a multistage launcher design, where a stage is understood to mean a motor and tank assembly allowing the provision of a certain speed increment. The energy gain must be done in a limited time whilst ensuring the precision of injection into orbit and the integrity of the payload.

The problem thus posed results in a certain number of constraints on the launcher design to guarantee the success of the mission. The payload, whose main purpose is to survive in orbit for several years, is designed to withstand moderate environments. The level of acceleration, acoustic and mechanical vibrations and thermal fluxes applied to the payload must in consequence be limited during the launch phase. These constraints must be taken into account from the start of design, during the preliminary design phase.

In addition to these constraints, due to the payload, there are limitations for launcher reliability. Forces on the launcher, such as the maximum dynamic pressure and the maximum heat flux, are constraints for problems of the strength of structures and the controllability of the launcher. Finally operational constraints must also be considered. Among others we can cite the falling of stages into uninhabited areas, the visibility of the launcher from a network of stations during the flight and the safety constraints close to the launch site.

12.2. Launcher design

Looking at this description of the problem, it is evident that the design of new launchers, from the preliminary design phases to the end of launcher development, involves a large number of disciplines that have greater or lesser interactions and coupled effects between them. These disciplines also have greater or lesser effects on high level objectives and on the global cost and performance of the launcher.

In an ideal world the problem of designing a new launcher would simply be posed as follows: given a customer requirement (specified by type of payload that should be placed in a type of orbit and complying with a certain number of satellite and operational constraints) and a high level criterion (minimising the recurrent cost of a launch, for example), find the best compromise for the system.

In practice the problem is more complicated, because the specifications can include requirements to reuse technologies or entire subsystems such as motors or complete existing stages, which means that the global optimization process must handle discrete and continuous parameters at the same time. In addition, there can be more than one high level criterion to manage (for example the development cost, reuse of the system in other orbits, robustness,. . .).

However, even if we only consider the basic problem presented above, the system design loop as currently implemented is relatively slow taking into account the use of different services associated with different disciplines. The current method can be roughly described as below:

1) a first selection of technologies is made covering the type of fuels, motors, pressurisation, etc. based on a priori ideas of costs or industrial, political or operational constraints;

2) from these choices, a preliminary staging is made by evaluating the speed increment to be provided by the launcher. This simplified calculation takes into account strong assumptions on the energy loss levels, the fuel mass to inert mass ratio and the specific impulse [1] of the motors;

3) secondly, a simplified propulsion force calculation provides more realistic motor parameters (mass-flow rate, thrust level,. . .) and a summary description of the motor subsystem;

4) next, a simplified stage layout exercise based on local force calculations provides a first estimate of the launcher mass budget;

5) these four steps are sufficient to provide inputs for the trajectory and performance optimization;

6) The reference trajectory from the calculation will start the dimensioning loop that includes the general force calculations, thermal, aerodynamic, control and transition phase studies. The payload mass optimization also allows the fuel mass and thrust level to be refined. These studies aim to arrive at a consolidated mass budget and an update of the constraints.

The complete loop is performed a number of times, with increasingly precise and computing time greedy tools, until convergence on a stable concept, if possible. This approach has disadvantages. The process is sequential and not integrated, which means that there is no guarantee that the launcher finally obtained achieves the best compromise, in particular in the case where disciplines interact strongly. In addition, the final result can be heavily dependent on the initial simplification assumptions. Finally, a dead end can be reached, which is the most serious case. The process is long, and in consequence onerous (a large number of loops are performed). This approach does not allow for easy traceability and justification of the final design is difficult.

---

1. The specific impulse is a value in seconds representing the effectiveness of a motor is equal to the ejection speed of the gases divided by the gravitational acceleration on the ground $Isv = \frac{v_e}{g0}$

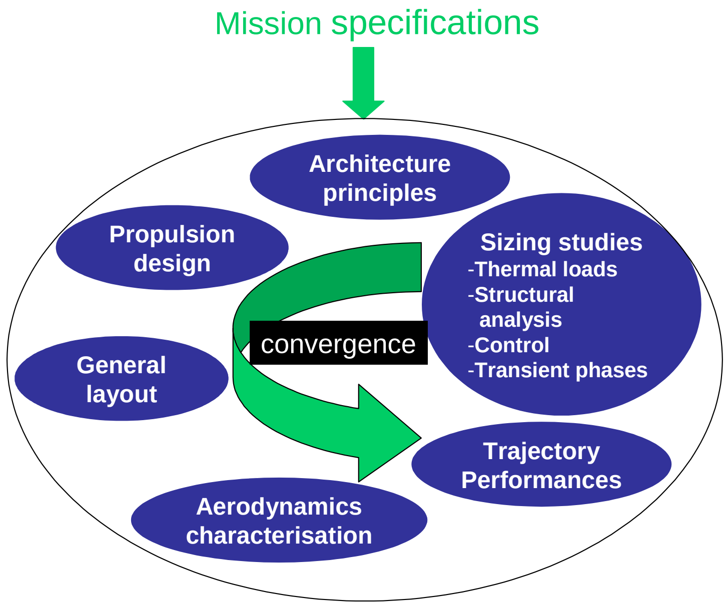

Figure 12.1. *Launcher design loop*

12.3. Multidisciplinary optimization in the launcher preliminary design phase

An alternative to the classic approach is to call on methods belonging to the family generally described by the term "multidisciplinary optimization".

For preliminary designs developed at ASTRIUM Space Transportation, the use of a multidisciplinary approach has resulted in the development of a platform [DUR 04] combining a certain number of essential disciplines considered to be sufficient in the first stages of system engineering activities. The approach developed is to integrate the different disciplines into a single environment using technical skill toolboxes containing simplified models and to optimize the high level launcher parameters for a given payload and a desired orbit, with respect to global criteria and complying with constraints applying to the outputs from these toolboxes. The disciplines represented for the preliminary design phases are:

– mission analysis (optimization of the trajectory and staging under constraints,. . .);

– the design office (stage geometry and dimensions, structure, materials, MCI, parallel or linear implementation, layout,. . .);

– propulsion: motor definitions and technologies (cryogenic or storable propellant, flow rate, thrust law, output section, pressurisation,. . .);

– control (controllability and maximum deflection of nozzles providing flight control,. . .);

– aerodynamic (drag and lift coefficient,. . .);

– simplified cost evaluation.

For each discipline, the simplified models are designed by teams dedicated to the field, with the aim of taking their main first order effects on global system design into account. Then the main parameters for each discipline are identified to participate in the global optimization process.

The principle of multidisciplinary optimization is to combine architecture optimization and trajectory optimization into an integrated process.

The optimization of the launcher architecture or staging uses the criterion of minimising the launcher cost or mass with launcher high level characteristics like motor type, launcher configuration, stage mass and size as parameters and the required energy increment to be communicated to the payload as constraint.

Traditionally, trajectory optimization is done for a fixed launcher architecture [JEA 06]. The criterion is the maximization of payload mass of the minimization of fuel consumption for a fixed payload for a required orbit. The trajectory calculation is done by solving the following simplified dynamic system:

$$
\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{v} \\
\dot{\mathbf{v}} &= \boldsymbol{\gamma} \\
\dot{m} &= -Q \\
\boldsymbol{\gamma} &= \mathbf{g} + \frac{\mathbf{f}_{aero} + \mathbf{f}_{prop}}{m} \\
\mathbf{f}_{aero} &= -\frac{1}{2}\rho v_r^2 C_D S_{ref} \frac{\mathbf{v}_r}{v_r} \\
\mathbf{f}_{prop} &= t\,(\cos\theta\cos\psi\mathbf{i} + \cos\theta\sin\psi\mathbf{j} + \sin\theta\mathbf{k})
\end{aligned}
$$

At each instant the orientation of the launcher acceleration is completely defined by the control law expressed by the two angles θ and ψ, which are the parameters for the trajectory optimization problem.

Multidisciplinary optimization must thus take discrete and continuous parameters into account. The discrete parameters are typically the architecture type (number of stages, parallel or linear configuration) and the technology type (solid or liquid propulsion) The continuous parameters to be optimized are configuration related (fuel mass of each stage, thrust laws, control laws, tank pressure,. . .).

The aim of the multidisciplinary approach developed at ASTRIUM Space Transportation is to find consistent solutions with regard to the different disciplines to initiate the new launcher design process, limit the risks of finishing in a dead end and obtain an optimum solution taking a larger number of system constraints into account. Justification of the selected solution is then easier. The platform is operational and is already used in the preliminary design phase to guide designs of future civil launchers.

Managing uncertainties is a constant preoccupation during the development of a new launcher and must be considered from the preliminary design stage. The methodologies presented in this work are to be implemented in the multidisciplinary tools that are used in preliminary design by ASTRIUM.

Because of both the approach adopted until now to solve the multidisciplinary problem and the complexity of the system studied, ASTRIUM Space Transportation is extremely interested in the scientific advances described in preceeding chapters, in particular the reduction of the model and multi-level optimization. As previously stated, the ASTRIUM multidisciplinary optimization approach allows a global start to the launcher design process. The developments presented in this book could allow for the continued use of multi-disciplinary tools in later preliminary design phases, through the integration of more complex models coupled to metamodels.

12.4. Evolutionary Optimization for Space Launcher Design: An Example

We give an example for a launcher with $n = 23$ continuous parameters to be optimized. Nine of the parameters are architecture parameters (stage diameters, stage masses, and mass flows) and 14 parameters characterizing the trajectory command law for the launcher. The optimization criterion, i.e. the *cost function*, is the launcher recurrent cost that can be computed with a Fortran code within a few seconds. The code admits additionally 18 inequality and 4 equality constraint values. For this reason we introduce in the following section an adaptive constraint handling technique which addresses both inequality and equality constraints in a unified manner.

12.4.1. *Constraint Handling*

The constraint handling algorithm computes first *normalized* constraint values γ_i, for each $i = 1, \ldots, m$, either from an equality constraint $h_i(x) = 0$ or from an inequality constraint $g_i(x) \leq 0$. The normalization relies on a user-specified external parameter ϵ_i. The constraint values γ_i are used for penalization of the cost function within a weighted sum, where the weights are adapted mainly based on the ratio of feasible solutions. We will denote the ratio of feasible solutions in constraint i in the recent iteration as r_i^{feas}. Additionally, we use $\overline{r}_i^{\text{feas}}$ for the ratio of feasible solutions in constraint i averaged over the last $n+2$ iterations, and $\overline{r}_{i[j,k]}^{\text{feas}}$ averages only between the j-th and k-th last iterations.

All the computations are specified in the following.

12.4.1.1. *Epsilon-normalized constraint values*

For each given constraint, indexed with $i = 1, \ldots, m$, and for a given solution x we compute the *epsilon-normalized constraint value* to

$$\gamma_i(x) \stackrel{\text{def}}{=} \frac{1}{\epsilon_i} \times \begin{cases} g_i(x) + \epsilon_i & \text{for inequality constraints } g_i(x) \leq 0 \\ |h_i(x)| & \text{for equality constraints } h_i(x) = 0 \end{cases} , \tag{12.1}$$

where ϵ_i are strictly positive user-defined constants. We call a constraint *active*, when $\gamma_i > 0$, i.e. respectively $g_i > -\epsilon_i$ or $|h_i| > 0$. Active constraints get penalized. We call a constraint *infeasible* when $\gamma_i > 1$, i.e. respectively $g_i > 0$ or $|h_i| > \epsilon_i$. By definition infeasible constraints are also active.

For each constraint a user-defined ϵ_i must be provided. For inequality constraints the value decides when the constraint becomes active and therefore changes the cost function via penalization. For equality constraints the ϵ-value decides when the constraint becomes infeasible (in the continuous search domain an algorithm cannot, in general, satisfy an arbitrary equality constraint exactly).

The objective of the constraint handling is to provide *feasible solutions*, that is single solutions x where $\gamma_i(x) \leq 1$ for all i.

12.4.1.2. *Penalty for constraint violations*

For a solution vector x the penalization due to active constraints reads

$$f_{\text{pen}}(x) = \sum_{i=1}^{m} w_i \gamma_i^+(x)^2 , \tag{12.2}$$

where $\gamma_i^+(x) = \begin{cases} \gamma_i(x) & \text{if } \gamma_i(x) > 0 \\ 0 & \text{otherwise} \end{cases}$ is the positive part of γ_i and $w_i > 0$ denote adaptive weights for $i = 1, \ldots, m$. The penalization value is added to the original launcher recurrent cost (the cost function value) of the solution x.

12.4.1.3. *Target probability*

The desired probability for each γ_i to be feasible, i.e. $\gamma_i \leq 1$ is in general

$$p_{\text{target}} = 0.5 \tag{12.3}$$

and never smaller. When at most one solution of the recent population of size λ is feasible, the target probability is computed to the maximum of 0.5 and

$$p_{\text{target}} = \left(\frac{1}{\lambda n}\right)^{1/(|\{i=1,\ldots,m|\overline{r}_i^{\text{feas}}<1\}|+10^{-6})} . \tag{12.4}$$

The target value is chosen such that at least one solution within n iteration steps should remain feasible, under the assumption that the constraints are independent.

12.4.1.4. *Adaptive Weights*

The penalization weights w_i in (12.2) are adaptive and updated after each iteration step of the underlying search algorithm. The weight i is updated only if the respective constraint appeared to be active (i.e. $\gamma_i > 0$) at least once within the last $n+2$ iterations and additionally either

$$r_i^{\text{feas}} < p_{\text{target}} \wedge \overline{r}_i^{\text{feas}} < p_{\text{target}} \wedge \overline{r}_{i[1,2]}^{\text{feas}} \leq \overline{r}_{i[3,4]}^{\text{feas}} \wedge \text{IQR}(\gamma_i) < 15 \times \text{IQR}(f),$$

where IQR(.) denotes the inter-quartile range from the value of the recent iteration and f denotes the recurrent cost value (cost function) in the recent iteration (we assume that several solutions are evaluated in each iteration), or

$$r_i^{\text{feas}} > p_{\text{target}} \wedge \overline{r}_{i[1,2]}^{\text{feas}} \geq \overline{r}_{i[3,4]}^{\text{feas}} .$$

In the first case the weight i increases, in the second case the weight decreases, because the update for each selected weight $i \in \{1, \ldots, m\}$ reads

$$w_i \leftarrow w_i \exp\left(p_{\text{target}} - r_i^{\text{feas}}\right)^{\frac{1}{n}} . \tag{12.5}$$

The update equation aims in changing the weight w_i in that r_i^{feas} comes closer to p_{target} in the following iterations.

12.4.2. *Optimization*

The evolutionary optimization was carried out with the *covariance matrix adaptation evolution strategy* (CMA-ES) [HAN 01, HAN 06] implemented in Scilab. The Scilab code provides an unconstrained stochastic optimization routine which was coupled with an implementation of the constraint-handling as described above. The

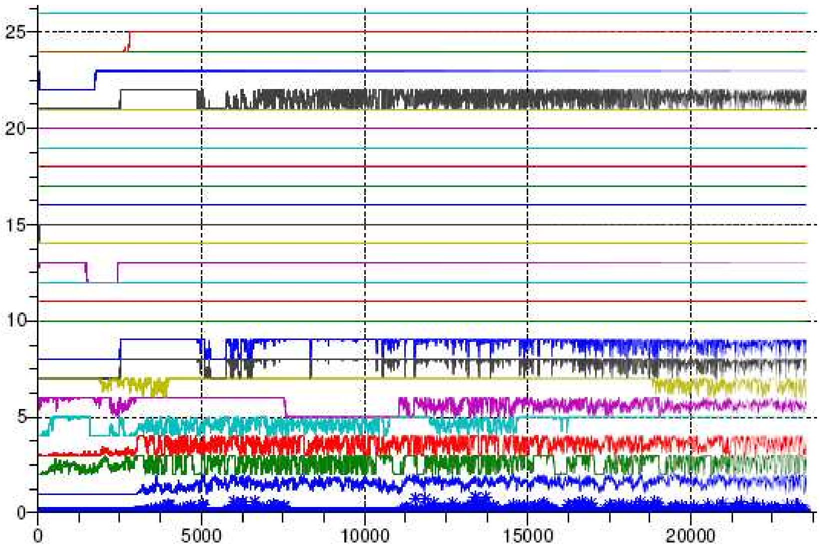

Figure 12.2. *Time evolution of the ratio of feasible solutions overall (lowest line) and for each constraint, where the abscissa shows number of cost function evaluations. Each line i varies between i and i* $+ 1$ *for* $i = 0, \dots, m$*. The variation depicts the ratio of feasible solutions in the recent iteration, ranging between zero and one. The lowest line denotes the ratio of overall feasible solutions (*$\gamma_i \leq 1$ *for all i), the second to forth and the tenth lowest lines are the equality constraints.*

implementation details of CMA-ES in an object-oriented manner are discussed in section 14.7.

The CMA-ES is a stochastic search method which is not only robust in rugged search landscapes, but can also address ill-conditioned, non-separable cost functions effectively. In a comprehensive benchmarking study, the CMA-ES has revealed excellent performance not only, but in particular, on noisy functions [HAN 09a, FIN 09]. In our experiments we applied the CMA-ES with all default values and without taking special measures to address noise in the cost function values (like a large population size or a decreased learning rate for the covariance matrix or a specific noise handling routine [HAN 09b]).

In Figure 12.2 the time evolution of the ratio of feasible constraints (i.e. of $\gamma_i \leq 1$) is shown. During the first thousand iterations (each iteration conducts 13 cost function evaluations) no feasible solution was evaluated. Later on, another period of similar length without any feasible solution can be observed, due to infeasibility of γ_5. During the remaining optimization feasible solutions are found regularly.

The overall optimization procedure was conducted several times. Depending on the chosen initial solution, not always a feasible solution could be found. The best obtained feasible solution could reduce the cost of the reference case (1331.7) by more than 10%. The acquired covariance matrix, which gives some insight into the optimization problem revealed a final condition number between 10^5 and 10^{10}, indicating a considerably ill-conditioned cost function.

Future work will include systematic studies of the influence of population size and initial solution on the success probability to find at least one feasible solution and consider different policies *before* and *after* the first feasible solution were found.

12.5. Bibliography

[DUR 04] DURANTÉ N., DUFOUR A., PAIN V., "Multidisciplinary Analysis and Optimization Approach for the Design of Expandable Launchers", *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, num. AIAA 2004-4441, Albany, New York, 30 August 1 September 2004.

[FIN 09] FINCK S., HANSEN N., ROS R., AUGER A., Real-Parameter Black-Box Optimization Benchmarking 2009: Presentation of the Noisy Functions, Report num. 2009/21, Research Center PPE, 2009.

[HAN 01] HANSEN N., OSTERMEIER A., "Completely derandomized self-adaptation in evolution strategies", *Evolutionary Computation*, vol. 9, num. 2, p. 159–195, MIT Press, 2001.

[HAN 06] HANSEN N., "The CMA evolution strategy: a comparing review", LOZANO J., LARRANAGA P., INZA I., BENGOETXEA E., Eds., *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, p. 75–102, Springer, 2006.

[HAN 09a] HANSEN N., "Benchmarking a BI-Population CMA-ES on the BBOB-2009 Noisy Testbed", *Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference*, ACM, July 2009.

[HAN 09b] HANSEN N., NIEDERBERGER S., GUZZELLA L., KOUMOUTSAKOS P., "A Method for Handling Uncertainty in Evolutionary Optimization with an Application to Feedback Control of Combustion", *IEEE Transactions on Evolutionary Computation*, vol. 13, num. 1, p. 180–197, IEEE, 2009.

[JEA 06] JEAN-MARIUS T., "Multidisciplinary Optimization for Early System Design of Expandable Launchers", *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, num. AIAA 2006-7125, Porthmouth, Virginia, 6-8 September 2006.