3rd European Conference for AeroSpace Science (EUCASS 2009)
July 6 – 9th 2009 Versailles, Paris

VEGA Launcher Trajectory/Guidance Loop: Overall Performance Optimisation.

A. Espinosa Ramos[1], P. D. Resta[2], D. Barbagallo[2], L. Perrot[3], M. Baudin[3], S Hervouet[3]
[1]CNES-Launcher directorate, France. [2]VEGA IPT ESA/ESRIN, Italy. [3]ALTRAN, France.

The particularity of the VEGA launcher is its large domain of required missions, covering Low Earth Orbit from equatorial up to SSO inclinations with altitudes going from 300km up to 1500 km. In addition, due to the suitability of the launcher performance to ESA scientific payloads needs, several specific and exotic missions are foreseen for VEGA launcher. Each mission analysis has its own specifics. The standard trajectory/guidance loop in which the guidance is conformed to the optimal trajectory doesn't seem to be adapted to VEGA. As the guidance manoeuvre performed at the Z9 burnout to control Z9 fall-down penalises the optimal trajectory profile, several loops are needed to optimise the mission scenario. Consequently, a dedicated method has been developed for VEGA launcher to perform and optimise this loop; the aim is to facilitate and prepare for the VEGA exploitation phase.

In this communication, a new method to optimise the Launcher performance is presented. This method allows for meeting statistical requirements such as stage impact control, upper stage consumption and more generally the guidance performance requirements and mission constraints.

1. INTRODUCTION

Some launchers like VEGA (Small European New ESA Launcher) are developed to cope with a large domain of mission requirement. As each mission has its own specifics, classical trajectory/guidance loop in which guidance fits optimal trajectory seems little adapted, as could lead to strong penalization (de-optimisation) of the guided trajectory.

CNES Launchers Directorate (DLA) must maintain a high level of technical expertise with respect to the exploitation of European Launchers and in preparing for the future. Applied Research and Technology activities are aimed on this purpose.

The new approach, which consists of managing payload optimization coping with statistical constraints in a stochastic mode (evaluated with closed loop guidance simulations), seems the most adequate to reduce payload loss due to guidance coping with mission requirements. The method presented in this paper has been developed by the CNES/DLA with the collaboration of ALTRAN.

CNES Launcher Directorate ensures technical support to ESA/IPT for the VEGA program in several fields, and in particular, in trajectory/performance and guidance activities. In this frame, the testing of this methodology on the VEGA Launcher mission was proposed to ESA to evaluate its interest. The aim is to facilitate and to prepare the VEGA exploitation phase.

2. CLASSICAL METHOD DESCRIPTION

2.1. Limitations of the classical approach for the Launcher performance optimisation.

The classical scheme to optimize the Launcher performance is based on the adjustment of the Launcher payload and flight reserve usually allocated to the upper stage.

The Launcher performance is first optimized to maximize the payload according to a predetermined flight reserve. This first step determines an "open-loop" optimized trajectory in nominal conditions (without dispersions). The following step consists of adjusting the guidance parameters in order to retrieve the main characteristics and the performances of the open-loop optimized trajectory. The guidance parameters are usually optimized to minimize the propellant consumption.

Random dispersions are generated with Monte-Carlo methods and guided trajectories simulations are computed in order to check statistical constraints (stage impact locations, propellant mass consumption to compensate dispersions, final and intermediary orbital parameters accuracy, re-entry stage conditions). If those statistical constraints are not satisfied, the nominal parameters are modified (propellant nominal mass is corrected) by, for example, decreasing the payload or by increasing the flight reserve. The process is then repeated until satisfaction of all constraints. It is an iterative process.

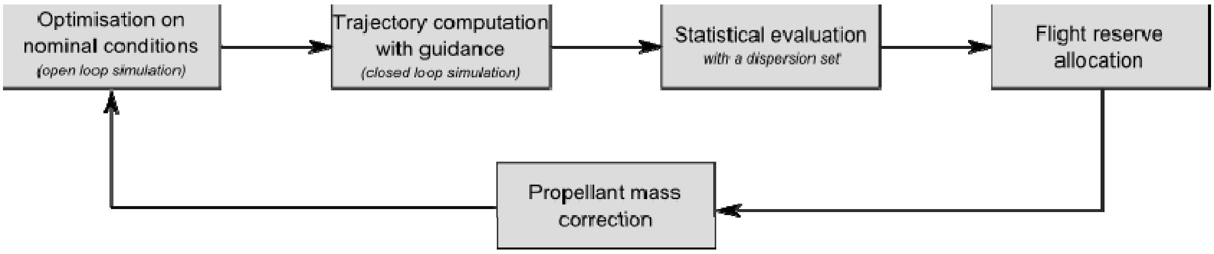

Figure 1 : Classical performance Optimisation process

Following this method allows to find a flight reserve and a payload satisfying the statistical constraints. However, the launcher performance is not guaranteed to be optimal in terms of payload and statistical constraints. The reason is that those constraints are not taken into account directly in the trajectory-guidance loop optimisation.

3. NEW METHOD DESCRIPTION

We suggest a new approach coupling the statistical trajectory characteristics evaluation, the guidance and the performance optimization and solving a global optimisation problem. The problem formulation is a stochastic optimization problem and it is presented hereafter.

3.1. Problem Formulation: Stochastic optimization problem

Preliminary notations:

D: Dispersions set

X: the vector of "optimization parameters"

D represents the whole set of dispersions. The dispersions are the random deviations impacting the propulsion law (thrust, mass flow rate,..), the masses, the aerodynamics and the atmosphere model. These dispersions can be encountered by the Launcher on a certain level of probability. They can be modelled by random variables and D can be generated by a random numerical process representative of the given probabilities characteristics.
X represents the main optimization parameters (which can be related to the open loop optimization but also to the guidance adjustment), such as:

- The initial pitch over after lift-off,
- The launcher attitude (pitch and yaw) in the outer atmospheric phase,
- The boosts and coasting phase durations,
- The orbital parameters (intermediary and final orbits), aimed by the guidance throughout the flight sequential,
- The guidance parameters during particular manoeuvres.

Complete formulation of the Launcher performance optimization problem

P_0 Problem

$$\underset{X/D}{Max}\ (Payload)$$

With respect to the following constraints:

- Usual guidance constraints on the final and intermediate orbits,
- Physical constraints like flux or visibility constraints,
- Statistical constraints like propellant consumption and stages impact points:

$$C_0(X) = \underset{di \in D}{\text{Probability}}\ [\text{Consumption of stage n° n } (X, di) > Consum_{max}] < 1 - P_0$$

$$C_1(X) = \underset{di \in D}{\text{Probability}}\ [\text{Overshoot impact point of stage n° n } (X, di) > d_1] < 1 - P_1$$

$$C_2(X) = \underset{di \in D}{\text{Probability}}\ [\text{Undershoot impact point of stage n° n } (X, di) > d_2] < 1 - P_2$$

The difficulty is the formulation of stochastic constraints and the optimization of the launcher performance. The constraints here are linked to the vector optimization parameter X and must be estimated for a defined occurrence probability (P_0, P_1, P_2).

There are three main pitfalls encountered while solving this problem:

1. Trajectory –guidance loop: the process (Trajectory optimization in open loop → guidance in closed loop → statistical evaluation) has to be automated

2. The statistical constraints are difficult to take into account: It is not immediate to estimate these constraints, and they have no reason to be continuous and differentiable. Treating them directly by conventional numerical optimization methods is not possible and specific statistical estimators should be developed. To make the optimization process effective, they must be sufficiently accurate to evaluate values at low probability and also stay inexpensive.

3. The coding of the whole trajectory – guidance optimization process itself is complex, mainly because:
   - It is a sequence of several complex sub-processes, each of whom requires the solution of an optimization problem.
   - The statistical estimation of low probability can be very expensive when using usual methods (like Monte-Carlo), and is needed at least at the beginning of the computations.
   - The classical optimization methods can fail due to the non-differentiability of the problem.

3.2. Choice of the stochastic optimization method

3.2.1. Review of methods of stochastic optimization

In a stochastic optimization problem, some parameters can be regarded as unknown or uncertain and are modeled by random variables. More generally, the uncertainty is represented by random experiments whose result is noted ω. The set of all possible random experiments is the fundamental set noted Ω. For instance, if the elements of Ω are the weather conditions, they allow the description of random variables such as electric consumption. The formulation of such stochastic optimization problems can be formulated as:

$$\begin{cases} \min Ef_0(x,\omega) \\ Ef_i(x,\omega) \le 0, i = 1,....,k \\ Ef_i(x,\omega) = 0, i = k+1,....,m \\ x \in X \end{cases}$$

E is a statistical estimator associated with the probability measure (any statistical estimator: estimation of a value at a given probability, mathematical expected value), X is a closed non-empty set representing the set of optimization parameters.

$$\begin{matrix} f_0 \Re^n \times \Omega \rightarrow \Re \cup \{+\infty\} \\ f_i \Re^n \times \Omega \rightarrow \Re \cup \{+\infty\} \quad 1 \le i \le m \end{matrix}$$ are given functions (process models).

Two large classes of stochastic optimization problems can be considered:

1. The problems of decision with uncertain future: we must optimize the parameters of the system thanks to the knowledge of the system at the time of the decision (for example by taking action) based on probabilistic models of the system behavior or its environment (prediction). The decision parameters are similar to system controls.

2. The stochastic optimization problems, so-called "open loop" problems where it is possible to repeat the parameter adjustment by measuring the response of the system. The optimization process is similar to the optimum system setting in order to maximize robustness and performance.
   We discuss here the second type of problem.

3.2.2. Evaluation Process and optimization

The "objective" function and the probabilistic constraints of the problem are evaluated by a calculation process at two levels:

- The “real” model (or any representative simulated system) in which samples are injected representing the system uncertainties and providing the system output response,
- The probabilistic estimation model based on the system responses in order to provide the numerical optimization solver with the values of the objective and constraints.

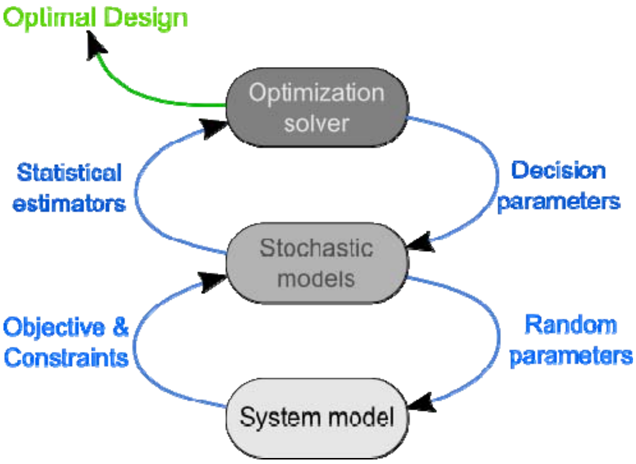

Figure 2: Resolution structure of a stochastic optimization problem (optimal design)

3.2.3. Choice of the stochastic optimization method

a. The stochastic gradient method

The goal is to minimize a stochastic function with respect to a vector of optimization parameters x

$$\bar{f}(x) = Ef(x) = \int f(x,\omega)\mu(d\omega)$$

The stochastic gradient algorithm consists of changing the variables to optimize (x) according to the following recurrence formula:

$$x_{n+1} = x_n - a_n \nabla_x f(x_n, \omega_n)$$

under certain conditions on the series $a_n \in \Re^+$ to insure the convergence (see ref [R6])

Drawbacks of the method:

- The convergence is slow and requires a large number of iterations (the known results are relative to asymptotic convergence),
- Difficulties in taking into account stochastic constraints. The constraints can be taken into account by penalty techniques (augmented Lagrangian).

b. The non differentiable methods

In the most common case, where the process simulation and the statistical values of the objective function and constraints are estimated on the basis of a sampling of the fundamental set Ω, the statistical estimators have no longer the qualities of numerical continuity and differentiability.

The most well-known methods are the simplex method (Nelder and Mead method) and the methods based on a random process of generation of candidate solutions (like Genetic algorithms, simulated annealing, ants colony or the subgradient methods).

These methods are interesting only if the evaluation processes of the objective function and the constraints are inexpensive. The difficulty is also to translate the problem in the formalism of the optimization method.

c. Optimization methods for continuous functions

The optimization methods of continuous functions are powerful, but they generally can not apply directly to the statistical estimators in general non-continuous and not differentiable. Nevertheless, to cope with this difficulty, the non-differentiable functions can be approximated by continuous models as: polynomial approximation models based on design of experiments (DOE) and the response surface techniques, the multi-quadrics, techniques for mesh or techniques of "kriging" as well.

Finally, the final selection of the optimization method is oriented towards a polynomial approximation method. It is the most adequate method taking into account the characteristics of the Launcher performance optimization problem (P_0 problem):

- The evaluation of the objective and constraints with classical methods (like Monte Carlo or general sampling method) are very expensive ,
- The constraints are stochastic,
- The problem is slightly non differentiable, mainly because the stochastic constraints are computed with numerical random methods for very low probability level (10^{-3}, 10^{-4}).

Among the optimization methods described above, we have eliminated the stochastic optimization methods and the non-differentiable optimization methods. These methods are too expensive (they need several thousand evaluations of the objective function and constraints) and present also great difficulties to properly manage the constraints. As they are weakly non-differentiable, a polynomial approximation

method should be accurate enough, and the classical methods of continuous functions optimization may apply.
The last item required to get an efficient optimization method is to develop an inexpensive method to evaluate values (the constraints) at low probability level.

3.2.4. Inexpensive method to evaluate values at low probability level: Bootstrap method

Probabilities are usually estimated from Monte-Carlo runs. To estimate the probabilities at a 10^{-4} level, at least a hundred thousand simulations are required. At this probability level, a Monte Carlo run to simulate trajectories of a guided launcher requires at least one day of computing time (cpu time). To avoid this drawback, we propose an inexpensive approach for estimating low-probability results.

We will try to define an inexpensive estimator based from a limited sample that is representative of the required probability level. This estimator should be unbiased, convergent, efficient and robust.

The Monte-Carlo method is used to generate random events (or dispersions) (ω_1, ω_2... ω_{nrun}) from which the response of the models is calculated with a numerical simulator representative of the real system. After having classified the values of the model response by increasing order we can construct the cumulative distribution function as shown in Figure 3.

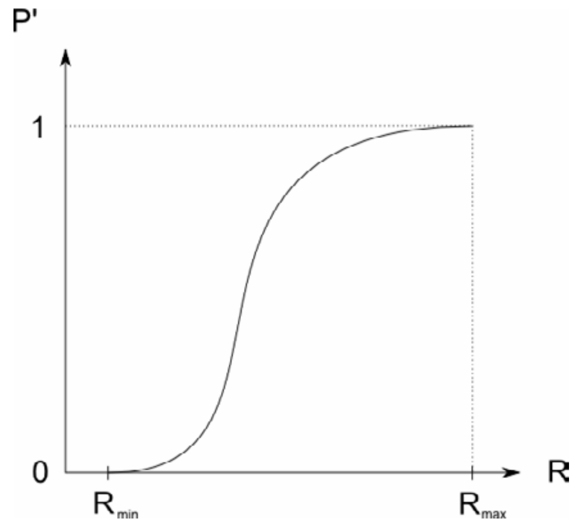

Figure 3 : Cumulative distribution of the model response

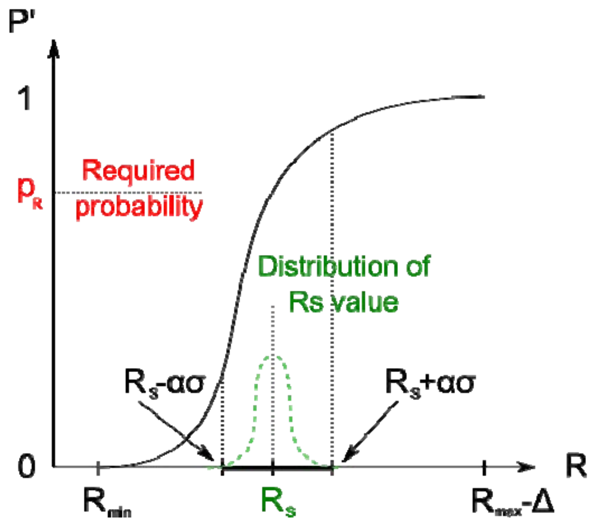

Figure 4 : Statistical analysis of the value to estimate

The statistic result *Rs* for a given probability p_R is obtained after constructing the cumulative probability function and can be obtained by inverse interpolation $R_S = (P')^{-1} p_R$. The idea is to identify dispersions leading to the model responses in the neighborhood of the search value at the required probability.

The Monte Carlo method provides a R_S value calculated on a sample computed randomly. The estimated value R_S is also a random variable with an expected value $E(R_S)$ and variance σ^2 (cf. Fig. 4).

The parameter σ represents the standard deviation of the *Rs* value. The parameter α is a confidence parameter to determine the events (the dispersions) leading most probably to the value to estimate (R_S).
To determine the standard deviation of the *Rs* value (σ), we propose to use the bootstrap method applied on a full Monte Carlo sample.
Bootstrapping is a method to estimate the properties of an estimator (such as variance) by measuring those properties when sampling from an approximated distribution. The bootstrap method can be applied when a set of observations is assumed to be extracted from an independent and identically

distributed population. It samples randomly the original dataset, allowing repetitions, and produces another dataset of the same size on which is computed the value to estimate (variance, average …). Repeating this sampling n times produces a 'boostrapped' distribution of size n of the value to estimate. Taking the average and deviation of this distribution gives an estimator and a confidence neighborhood for this value.

The two parameters (σ, α) allow selecting a dispersion set. After simulation of guided trajectories on this set, we obtain a set of values ($R_{s,k}$) which are used to estimate inexpensively Rs (for instance by averaging). The parameter α can be experimentally adjusted (see Figure 5).

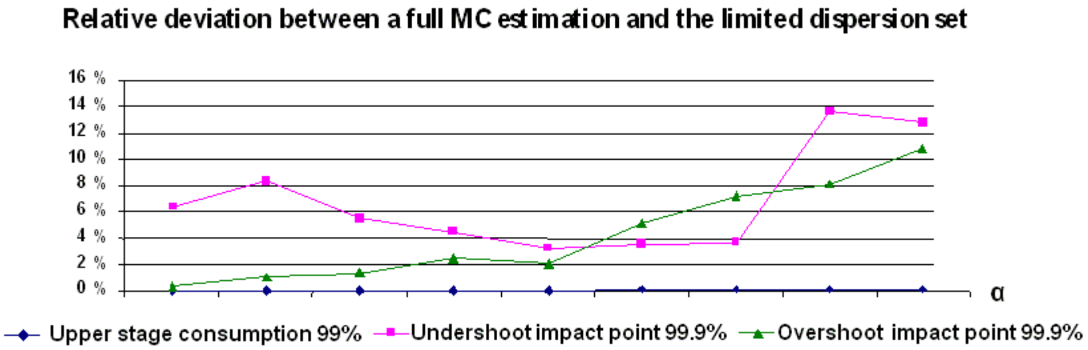

Figure 5 : Adjustment of the α parameter

3.2.5. Approximation by a response surface –choice the Design Of Experiments (DOE) scheme

Assuming that we have a system ("black box" or "transfer function") providing a response to a set of inputs, the Design Of Experiments method consists in approximating the response by a polynomial (of order depending on the required accuracy). The response model $y \in \mathrm{R}$ is a polynomial of the optimization inputs $\mathbf{x} \in \mathrm{R}^{N_{param}}$. The polynomial coefficients $(a_i)_{i=1,N_{coeff}} \in \mathrm{R}^{N_{coeff}}$ are the model parameters to be determined. The calculation of these coefficients requires a number of experiments at least equal to the number of coefficients. The DOE scheme specifies the experiments points to use (i. e. input values).

Once the model coefficients are determined, the model provides an approximate response for a cheap time computation cost. Moreover, as the model is polynomial, the response is continuous and differentiable, which is very interesting when using the model in the context of an optimization method. However, we can take into account the accuracy of the model so that the optimum determined by the optimization algorithm using the model is close to the original system response.

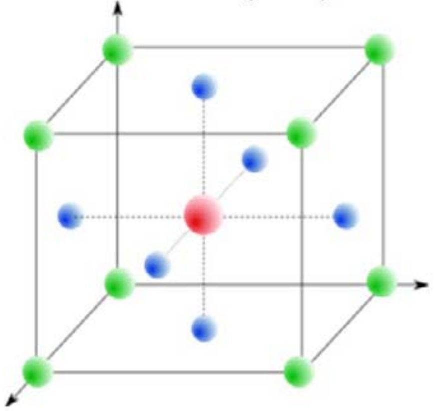

Figure 6 : composite design (order 2) and 3 parameters

In consequence, we have selected an incomplete central composite design of order 2 which presents a good compromise between accuracy and the computation cost.

3.3. Driving the optimization process

3.3.1. Optimization process scheme

The complete process to optimize the trajectory/guidance loop is the following:

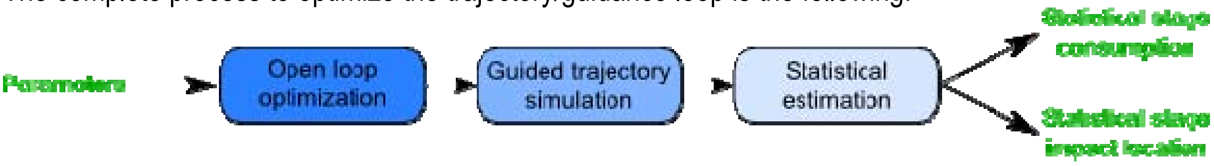

Figure 7 : Complete optimization process

1. Open Loop Optimization

The open-loop optimization implies that all data acting on the trajectory are known, by definition the trajectories are simulated without dispersions.

The optimization parameters are:

- The initial pitch over,
- The payload mass,
- The control parameters (to define the launcher attitude),
- The durations of coasting and propelled phases.

The open loop trajectory is also called nominal trajectory.

2. Guided Trajectory

To carry out the simulation of the closed loop guidance trajectory, we have to define:

- the Launcher attitude during the atmospheric phase (data from open loop optimization),
- the guidance parameters and data to simulate closed loop guided trajectories with dispersions (considering the durations of the coasting and propelled phases and the orbital parameters at the end of each guidance segment),
- the guidance parameters in order to respect the statistical constraints (propellant consumption, stages fall down area...).

3. Statistical estimation

The real launcher trajectories are subjected to all kinds of dispersions (atmosphere, aerodynamic, propulsion, mass distribution...) which affect the performance and the characteristics of the nominal trajectory. The Launcher trajectories must meet certain specifications expressed in statistical form like orbital parameters accuracy, maximal stages consumption, stages fall down area expressed at given probabilities. The purpose of statistical estimation is to verify the compliance with these specifications.

The statistical estimation may be computed with direct Monte Carlo methods, or, as we have explained, with an inexpensive statistical estimator.

3.3.2. Optimization process resolution

The global vector optimization is built by parameter contributors taking into account both nominal trajectory definition and the closed-loop guidance. The parameters involved in the nominal trajectory define the main characteristics of the trajectory (initial pitch over maneuver, transfer orbits, nominal consumption, nominal stages impacts points). The guidance parameters influence the statistical characteristics of the trajectories (flight reserve, statistical stage re-entry conditions and fall down area).

- Global optimization Process

The effort is located upstream of the optimization itself since the optimization is almost immediate and does not present real difficulties because the models are based on polynomials. The global optimization process consists in the following steps:

1) Nominal trajectory optimization and guidance
2) Massive Monte Carlo simulation
3) Elaboration of inexpensive statistical estimators (accuracy check)
4) Elaboration of polynomial models (result of a DOE method)
   Accuracy check, limitations, validity domain
5) Optimization limited to the validity domain of the models
6) Solution verification, validity check
7) Return to step 1) 2) 3) or 4) under certain conditions

Determining the validity of the response surfaces and the accuracy of the statistical estimators is capital. The statistical estimators can be compared to a full Monte Carlo simulation, this verification is generally costly. The accuracy of the polynomial models can be checked against the inexpensive statistical estimator responses.

The model accuracy limits the variations of the optimization vector (X). So to converge surely towards the optimal solution, we must apply a sliding DOE technique by limiting at step 5) the possible parameters variations (validity domain).

- Sliding DOE technique

At iteration “n” of the global optimization process, the optimization can be performed effectively on a limited area of the optimization parameters on which we have verified the accuracy of the approximation models. If the optimizer sends the parameters to the frontier of the area, the optimization continues by sliding the DOE model in the direction of the optimal solution. If the solution found by the optimizer is in the domain, we can perform a "zoom" to refine the solution.

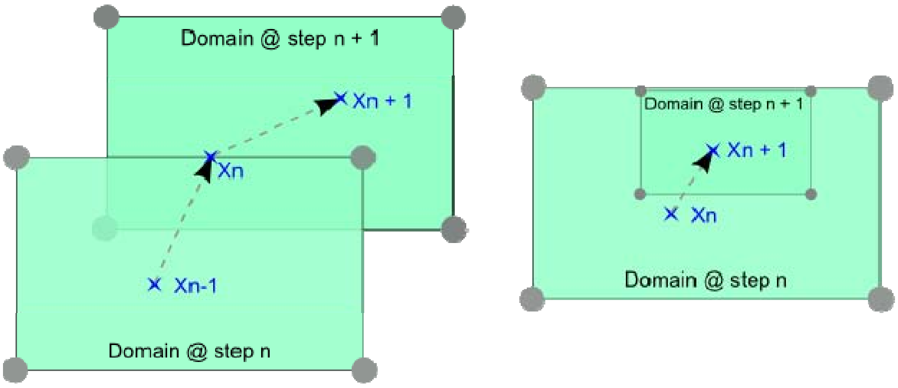

Figure 8 : DOE sliding and zoom technique

This method is based on the following principles:

- Local approximation of the process by DOE (polynomial approximation),
- Optimization in a limited area around the initial point,
- Verification of the approximation and periodic models update,
- Strategy for reducing or increasing the area of research.

4. APPLICATION TO VEGA LAUNCHER

The method was tested on the VEGA launcher to evaluate its interest by a real case of application. The mission are the reference mission for VEGA, a LEO orbit at 700 km altitude inclined at 90° and the Aeolus mission at 400km/97°.
Important note: *the following chapter aims at giving a practical numerical application of the methodology but in no way represents a Safeguard VEGA computation.*

The VEGA LV, described in figure 9, consists of a lower composite composed by three solid propellant stages (P80, Z23 and Z9) and a restartable upper stage named AVUM (Attitude and Vernier Upper Module, N_2O_4/UDMH). The development is under the responsibility of ESA/IPT (European Space Agency).

In the case of the Vega launcher, the impact points of first two stages fall down in the Atlantic Ocean far from the coast of Kourou. The impact location of the third stage may show large fluctuations due to the dispersion of the previous stages. A specific guidance strategy before Z9 burn-out has also been considered to limit the fall down area. This maneuver relies on several parameters to control the location and the length of the Z9 fall down area (for this mission, it is located in the Arctic sea between Greenland and Siberia). In what follows, this maneuver will be called “impact control maneuver” (ICM). The upper stage (AVUM) should be able to compensate the dispersions of the previous three stages. A propellant reserve (flight reserve) is allocated for this purpose.

Figure 9 : VEGA LV

4.1. Implementation of the method

OPTAX is the CNES main optimization tool for Ariane and all expendable launchers’ ascent trajectories including VEGA. OPTAX is based on a direct application of the Maximum Principle of Pontryagin. Due to this, optimal control is applicable only outside atmosphere when aerodynamic forces are negligible. The atmospheric arc can be optimized parametrically. Guidance simulator has been implemented as an option for trajectory optimization. The complete computation chain (Monte Carlo runs, statistical estimators, polynomial models by DOE, optimization, accuracy verification and computation validity at all levels) has also been fully automated.

For the overall optimization of the guidance-trajectory loop, eight main parameters have been selected: four parameters linked to the trajectory optimization (including payload mass to be maximized and Z9 impact point constraint) and four parameters linked to guidance (two for Z9 Impact Control Maneuver and two guidance parameters acting mainly on the AVUM consumption).

Four main statistical constraints are imposed for the overall optimization process for specified probabilities (numerical values mentioned here are given only as examples):

- Flux at fairing jettisoning < Fluxmax at a probability of 0.999
- Statistical consumption < Cmax at a probability level of 0.99
- Distances between the nearest coast and Z9 impact > 300 km at a probability level of 0.999 for the two extremes of the fall down area (undershoot and overshoot Z9 impacts).

The initial set of parameters has been obtained by the classical method of Launcher performance optimization. For these initial values, the statistical constraints are almost respected (only the limits of the undershoot constraint is slightly exceeded).
Starting from this initialization, we apply the overall optimization of the guidance-trajectory loop. To build the inexpensive estimators (for AVUM consumption, undershoot and overshoot Z9 impacts) 10000 Monte Carlo runs are computed (a 10000 MC runs ~ 3 hours CPU).

The statistical analysis on the 10000 Monte Carlo runs gives:

	Estimated value (Rs)	Bootstrap standard deviation σ	Confidence parameter (α)	Number of selected dispersions for statistical estimator	Observed accuracy between estimator & 10000 MC runs
AVUM consumption (*) P = 0.99	96.5%	0.2%	2.5	35	< 1kg
Overshoot Z9 impact point (**) (Distance/coast) P=0.999	310% (OK)	10 km	3.	33	20 km
Undershoot Z9 impact point (**) (Distance/coast) P=0.999	83% (KO)	20 km	3.	58	45 km

** : The AVUM consumption is given here as percentage of the maximal authorized value.*
***: The Z9 impact distance is given here as percentage of the minimal authorized value. Values mentioned here are given as examples and do not represent a safeguard VEGA computation.*

The errors due to the polynomial models obtained with DOE depend mainly on the domain size and the non linearity with respect to the optimization parameters. In our problem, it is necessary to strongly limit the variations of initial pitch over (highly influential and non linear parameter), the Z9 final orbit main parameter and the ICM parameters (guidance). The domain size is at first limited to obtain model errors less than 3kg (AVUM consumption) and 100 km (Z9 impact point). The domain size is then greatly reduced near the optimal solution to obtain accuracies under 1 kg and 50 km.

The optimization process needs two main sliding iterations with a progressive domain reduction. A complete 10000 Monte Carlo run is performed after each main iteration in order to update the statistical estimators.

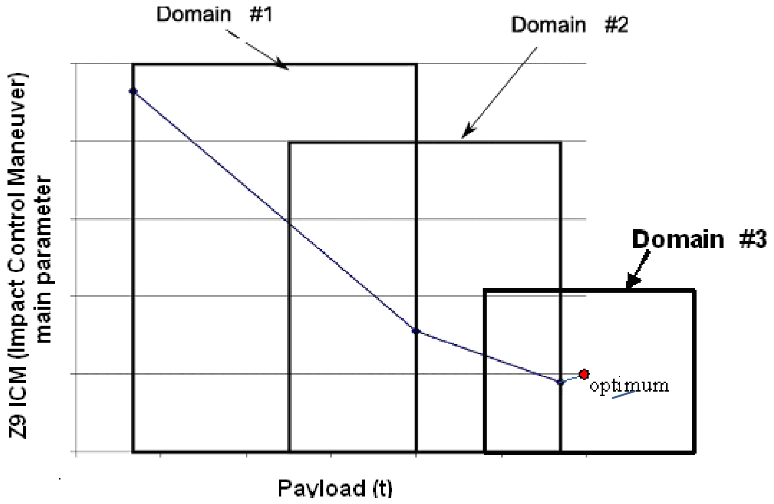

Figure 10 Main sliding iterations and domain reduction (wrt an ICM main parameter)

The optimal solution is limited by the consumption constraint and the overshoot distance constraint.

Statistical constraints evolution versus payload at optimum				
Payload (delta in %*)	AVUM consumption (% of max) P=0.99	Distance undershoot to nearest coast (km) P=0.999**	Distance overshoot to nearest coast (km) P=0.999**	
+4.51	99.6	534	314	
+4.57	99.7	539	292	←Limit for Distance overshoot
+4.63	99.8	544	271	
+4.69	99.9	549	250	
+4.75	100	555	229	←Limit for Avum consumption

**: The payload gain is here in percentage of the nominal payload computed with the classical method.*
***: Values mentioned here are given as examples and do not represent a safeguard VEGA computation.*

The solution lies in the interior of the last domain used to construct the polynomial models: the solution is certainly optimal. We can compare in the following figure the distributions of the constrained variables (AVUM consumption and Z9 impact points distances to nearest coast) between initial and final parameterization :

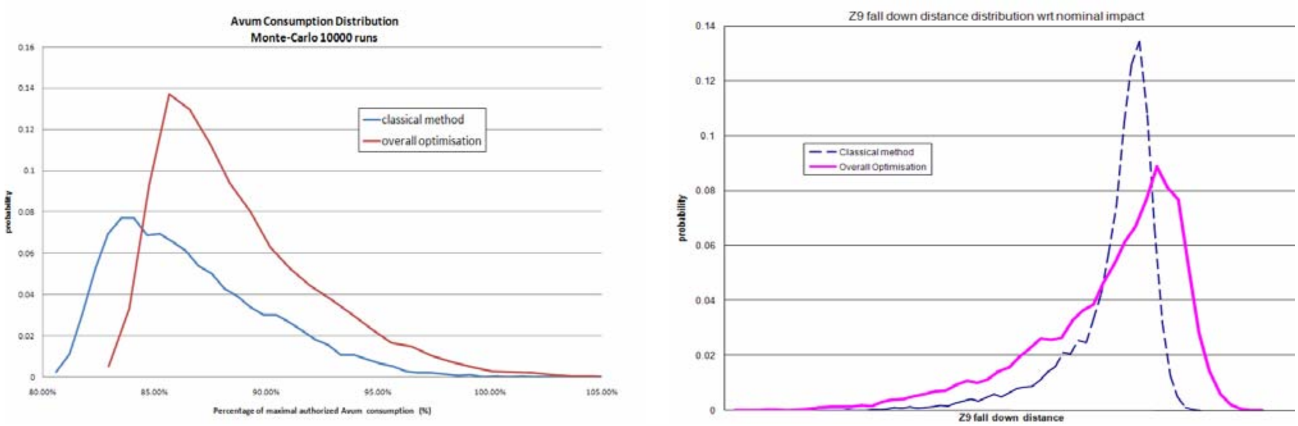

Figure 11 AVUM consumption distribution and Z9 fall down distance distribution

There are significant changes in the distributions shapes between the initial solution and the optimized solution. The distribution function for consumption has a much higher maximum and is shifted more to the right. The optimal solution takes full advantage of the AVUM consumption capacity. The distribution function of the impact distance is shifted to the right but it is also more flattened: the optimal solution distributes in a better way the overshoot/undershoot impact point on all the available space.

Below, the fall-down areas for the reference polar mission (700 km x 700 km / 90°):

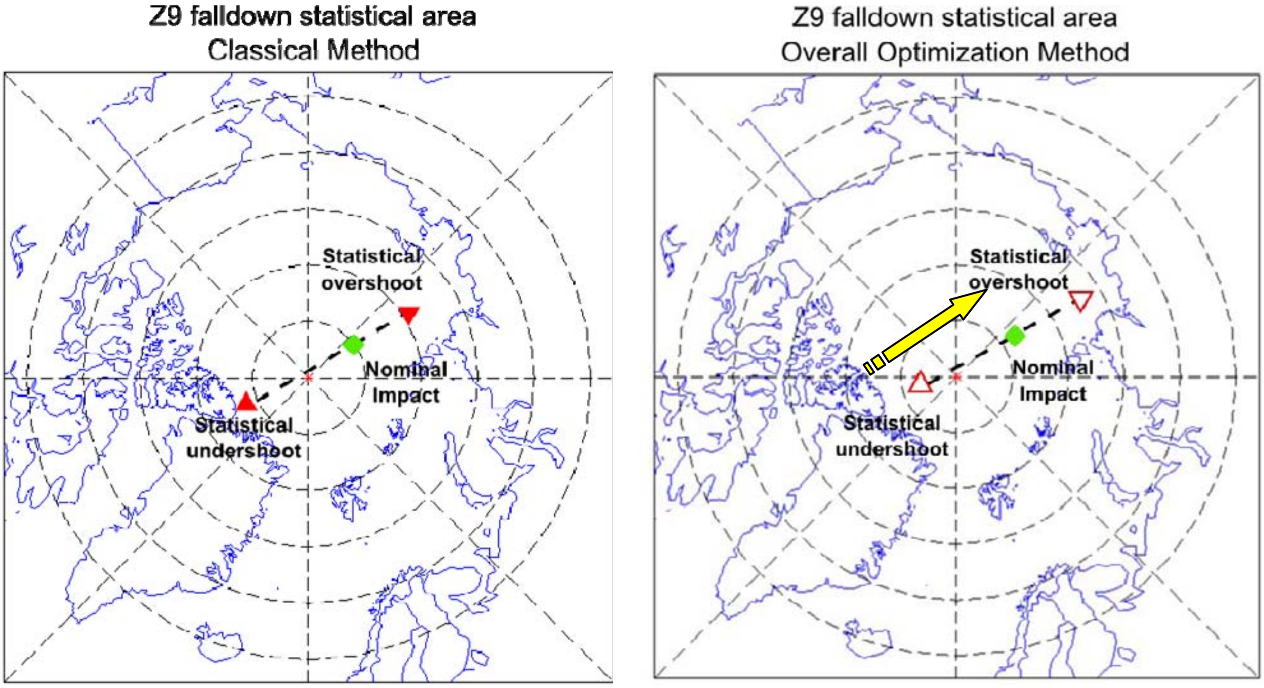

Figure 12: Z9 fall down area in the Artic sea for both methods (10000 Monte Carlo runs)
(It does not represent a safeguard VEGA computation)

This method has also been applied for Aeolus mission. Full results are not cited here but the fall down areas before and after computations are reproduced below:

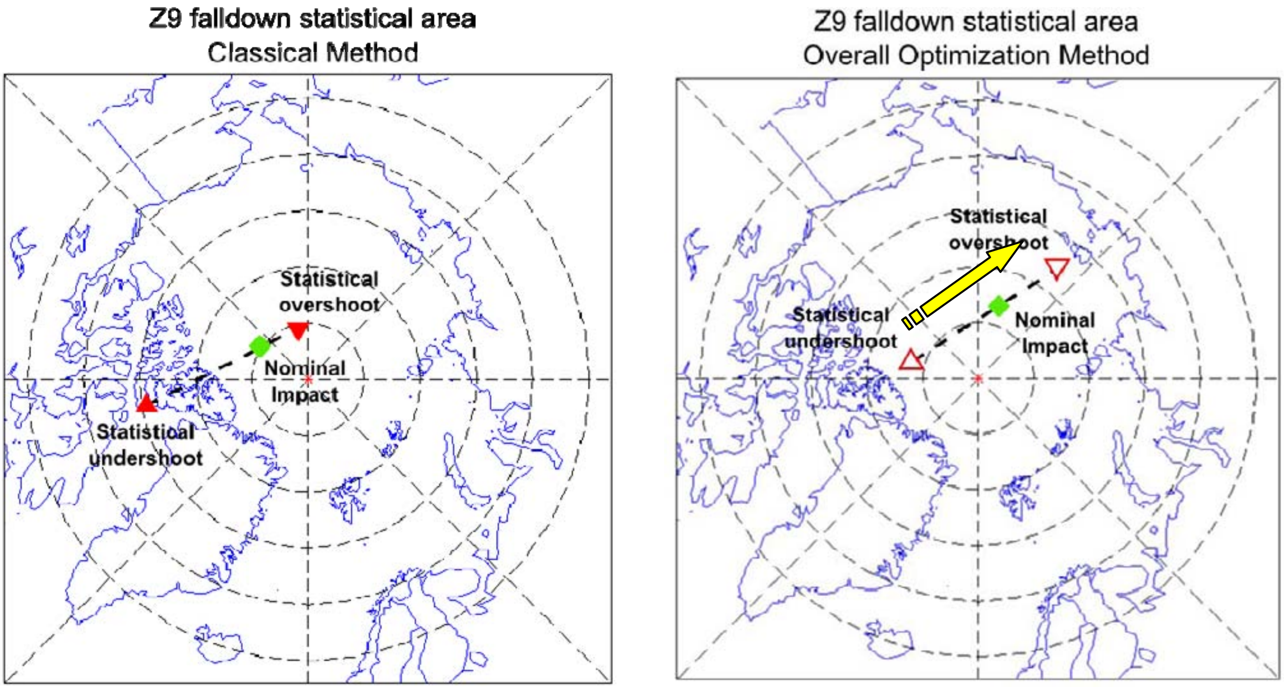

Figure 13 : Z9 fall down area in the Artic sea for both methods (10000 Monte Carlo runs)
(It does not represent a safeguard VEGA computation)

The following table gives the results of the overall optimization method for both mission configurations:

Mission	AVUM consumption(*) P=0.99	Z9 undershoot distance(**) P=0.999	Z9 overshoot distance(**) P=0.999	P/L gain(***)
700km / 90° Reference	99.6%	178%	105%	+4.5%
400km / 97° Aeolus	99.8%	110%	133%	+2.3%

** : The AVUM consumption is given here as percentage of the maximal authorized value.*

***: The Z9 impact point distance to the nearest coast is given here as percentage of the minimal authorized value. Values mentioned here are given as examples and do not represent a safeguard computation.*

**** : The payload gain is here in percentage of the nominal payload computed with <u>the classical method.</u>*

5. CONCLUSION

A new dedicated method has been developed to perform and optimise the trajectory/guidance loop. This method combines the optimal trajectory and the guidance in the performance optimization process. The launch vehicle performance remains the optimization criteria (but could be modified). More specifically, it can directly take into account mission specifications expressed in terms of probability. Finally the solution computed with this method fully meets the conditions of robustness (sensitivity to dispersions during the flight) and the various constraints of statistical success of the mission or safeguard).

To perform this overall trajectory-guidance optimization, several computational aspects have been improved:

- Statistical estimation: Due to the low probability level required (10^{-3}, 10^{-4}) the intensive use of traditional methods of statistical estimation (like Monte Carlo) was prohibited. So a particular inexpensive method has been developed to estimate values at low probability level.

- Approximated models: To improve the efficiency of the overall optimization loop, the method of design of experiments (DOE) was used to construct polynomial models representing the statistical response of the system. By using these polynomial models, the traditional methods of continuous optimization (SQP, interior point, reduced gradient...) can be used to solve the performance optimization problem under statistical constraints.

- Sliding DOE technique: the range of validity of the models is limited (due to strong non linearity). To converge surely with sufficient accuracy towards the optimal solution, we have to narrow the approximation domain of the polynomial models and also limit the optimization parameters variations. The optimization can be performed effectively on a limited area which is shifted for the next optimization step after having reconstructed the polynomial models.

Finally, application to VEGA launcher has been successfully performed for two missions (reference polar orbit and Aeolus mission) driving to very promising preliminary results. In spite of a very low statistical probability level and the complexity of the problem (very strong nonlinearity of the fall down constraints), the method was able to converge and to adequately characterize the solution in a reasonable time. The solution is optimal and justified, reducing the number of loops necessary when considering classical method and avoiding over constraining the Launcher trajectory.

As it takes into account the statistical constraints of the problem from the start, the solution meets precisely the mission specifications with an implicit guidance validation.

By greatly facilitating the overall performance optimization, the method seems appropriate, even though complex, for mission analysis and of full merit to prepare for the VEGA exploitation phase.

6. **ACRONYMS**

AVUM:	Attitude and Vernier Upper Module
CNES :	Centre National d'Etudes Spatiales
DLA:	Direction des Lanceurs du CNES
DOE:	Design Of Experiments
ESA:	European Space Agency
ICM:	Impact Control Maneuver
Isp :	Specific impulse
IPT:	Integrated Project Team
MC:	Monte Carlo
SQP:	Sequential quadratic programming
SSO:	Sun Synchronous Orbit
Z9:	Zefiro 9 (VEGA third stage)

7. BIBLIOGRAPHY

[R1] P. Larrañaga et J.A. Lozano (éditeurs), *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation (Genetic Algorithms and Evolutionary Computation)*, Éd. Kluwer Academic Publishers

[R2] J. Dréo, A. Petrowski, E. Taillard et P. Siarry : *Métaheuristiques pour l'optimisation difficile*, Éd. Eyrolles

[R3] Lecture by L. Bottou for the Machine Learning Summer School 2003 in Tübingen: *Stochastic Learning*

[R4] H. Robbins and S. Monro: *A Stochastic Approximation Method.* Annals of Mathematical Statistics 22, #3 (September 1951), pp. 400–407.

[R5] J.C. Dodu, M. Goursat, A. Hertz, J.P. Quadrat, M. Viot : *Méthodes de gradient stochastique pour l'optimisation des investissements dans un réseau électrique*. EDF Bulletin de la Direction des Etudes et Recherches Série C, Mathématique, Informatique, n° 2, p.133-167, 1981.

[R6] Polyak: C*onvergence and convergence rate of iterative stochastic algorithms*. Automatica i Telamekhanika Vol 12 1976

[R7] S. Vivier, Thèse Ecole centrale de Lille (laboratoire L2EP) : *Stratégie d'optimisation par la méthode des plans d'expériences et applications aux dispositifs électrotechniques modélisés par éléments finis*

[R8] Th. Ebner, Ch. Magele, B. R. Brandstätter, M. Luschin: *Approximation of the objective function: multiquadrics versus neural networks.* COMPEL, Vol. 18 N°3, 1999

[R9] A. C. Davison, D. V. Hinkley: *Bootstrap Methods and Their Application*, Cambridge University Press, 28 October 1997