MULTIDISCIPLINARY DESIGN OPTIMISATION OF EXPANDABLE LAUNCHERS IN ASTOS

Christian Möllmann(1), **Andreas Wiegand**(2), **Michael Dalheimer**(3), **Alvaro Martinez Barrio**(4), **Jens Kauffmann**(5)

(1) Astos Solutions GmbH, Meitnerstr. 10, 70563 Stuttgart, Germany, Email: christian.moellmann@astos.de
(2) Astos Solutions GmbH, Meitnerstr. 10, 70563 Stuttgart, Germany, Email: andreas.wiegand@astos.de
(3) Astos Solutions GmbH, Meitnerstr. 10, 70563 Stuttgart, Germany, Email: michael.dalheimer@astos.de
(4) ESA/ESTEC, Noordwijk, Netherlands, Email: alvaro.martinez.barrio@esa.int
(5) ESA/HQ, Paris, France, Email: jens.kauffmann@esa.int

ABSTRACT

The trajectory optimisation software ASTOS has been enhanced for Multidisciplinary Design Optimisation (MDO) of expandable launchers in preliminary design. Engineering models for the disciplines weights and sizing, aerodynamics, structural analysis and propulsion have been selected and integrated into ASTOS. The trajectory optimisation has been used as the integrating basis. The vehicle design parameters are optimised considering all disciplines simultaneously using a gradient based optimisation algorithm. The launch vehicle model can be built up interactively using predefined building blocks to define all required data. The model may then be simulated and optimised and results may be analysed within the ASTOS program environment. The implemented MDO capabilities added to ASTOS improve the preliminary design process of expandable launchers by giving the optimizer a more complete and more detailed picture of the system to be designed.

1. INTRODUCTION

The optimisation of an expandable launcher involves important interactions between the various disciplines involved like weights and sizing, aerodynamics, propulsion or structures. Even in preliminary design it is advantageous to reflect the most important interactions to get the best possible design which is at the same time reliable so that the risk of design corrections in later design phases is reduced.

If such interactions are reflected in the optimisation they are either very simple or neglected or each discipline is optimised separately and then the global solution is found by iterating between all disciplines. The approach shown here is different: the trajectory and the vehicle design are optimised together in one optimisation problem with all disciplines being computed whenever required.

Two requirements have to be met to use this approach: firstly all discipline models have to be automized so that they can reliably run without any human interaction. And secondly the models must be sufficiently fast.

The approach has been realized with the trajectory optimisation software ASTOS. ASTOS has been developed for the last 20 years and is a reference tool for space trajectory optimisation at ESA/ESTEC.

The various disciplines integrated are trajectory, weights and sizing, propulsion, aerodynamics and structures. The engineering models which have been selected and integrated or implemented are fast and accurate enough. Special requirements for optimisation have to be followed when choosing or implementing these models like mathematical differentiability, computational efficiency and support of multiple successive calls. The integration has been done using the trajectory as the basis and integrating all disciplines into one large optimisation problem. This approach is called All-At-Once (AAO, sometimes also SAND, i.e. Simultaneous Analysis and Design) opposed to other approaches where disciplines are optimised separately [1].

The enhanced ASTOS software can be used for analysis and design of expandable launchers using ASTOS specific capabilities in model definition (simple building blocks), setup of the optimisation problem, NLP algorithms and result analysis.

2. TRAJECTORY OPTIMISATION IN ASTOS

The optimisation software ASTOS can be used to optimize ascent trajectories of launchers and reentry vehicles as well as interplanetary trajectories. Before giving more details on trajectory optimisation in ASTOS it is worth to briefly explain the essence of a trajectory optimisation problem.

Trajectory optimisation is the process of finding a trajectory that minimizes or maximizes a specified objective function while fulfilling prescribed constraints. The optimisation problem consists of the equations of motion (EoM), the controls, the cost function, initial, final and path constraints and optimisable parameters. It is essentially an optimal control problem.

The EoM describe the physical behaviour and are ordinary differential equations (ODE) that lead to the time-varying position and velocity when integrated in time. The controls are the means to influence the EoM

directly; they are to be optimised (e.g. the aerodynamic angles). The constraints put certain restrictions on the states or on model parameters and may be defined at the beginning (initial boundary constraint), the end (final boundary constraint) or along the entire trajectory (path constraint). Typical constraints in launcher ascent optimisation are minimum altitude of vertical lift-off, maximum dynamic pressure and heat flux, splash down of stages or the final orbit. Optimisable parameters are model parameters which are optimised together with the time-varying controls (e.g. initial propellant mass).
In ASTOS the user can define a launcher trajectory optimisation problem in an easy-to-use software environment. A complete optimisation problem definition includes the vehicle information with the stages and their properties, the propulsions and the aerodynamics, plus the environment definition including the planet and its shape and gravity field, the atmosphere and possibly the wind. The various top level objects are depicted in Figure 1.

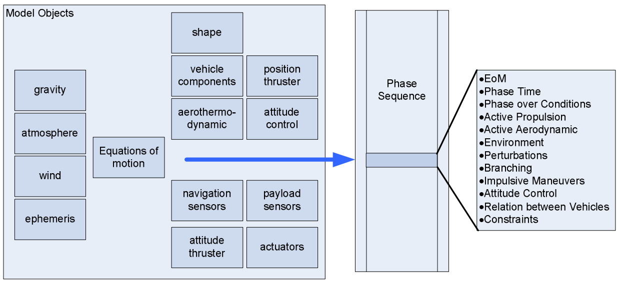

Figure 1. Scenario Builder of ASTOS

The user can choose from several well-established optimisers. By far most of these optimisers use a gradient-based method to numerically solve the optimal control problem. Two approaches can be distinguished here: the multiple shooting method and the direct collocation method. The multiple shooting method essentially discretizes the control while integrating the equations of motion (EoM) with a general-purpose ODE integrator whereas the direct collocation method discretizes both the control and the EoM.

3. MDO EXTENSIONS OF ASTOS

The ASTOS capability of trajectory optimisation has been enhanced to include launch vehicle design optimisation. Therefore the disciplines aerodynamics, propulsion and weights have been improved for optimisability and higher fidelity and the disciplines geometry and structures have been added. The data exchange between these disciplines is considerable as many disciplines depend on input from many others as can be seen in Fig. 2.

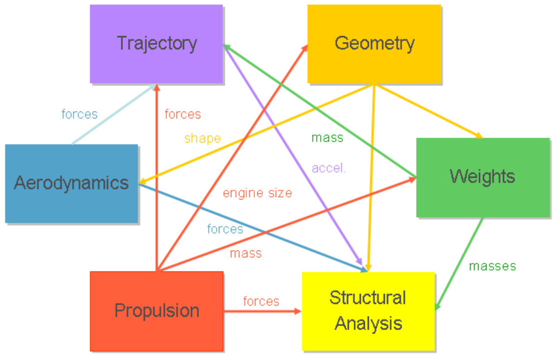

Figure 2. Data exchange between disciplines

4. DISCIPLINES

4.1. Trajectory

The trajectory is computed by integrating the EoM. In the MDO enhancement of ASTOS three degrees of freedom have been considered (6 DOF are possible as well, but are computationally expensive for use in optimisation). In that case six states are used in the EoM, three for the vehicle position and three for the velocity. The states may be chosen from a predefined set within ASTOS, e.g. inertial Cartesian or flight path velocity.
The equations of motion for inertial Cartesian states are then:

$$\dot{x} = \frac{d}{dt}\begin{pmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ v_z \\ F_x/m \\ F_y/m \\ F_z/m \end{pmatrix} \tag{1}$$

This is a system of ODE and is as simple as informative. Essentially the controls and the environment influence the ODEs via the sum of all forces F including the aerodynamic force, the propulsive forces and the gravitational forces. The force divided by the total vehicle mass m gives the total acceleration. This leads us directly to the other disciplines.

4.2. Aerodynamics

For MDO an engineering level computation of the aerodynamics has been integrated into ASTOS. It computes the aerodynamic coefficients for varying geometries (and of course flight conditions) and can thereby reflect changes in the vehicle design. In contrast the aerodynamics used in pure trajectory optimisation are not optimisable but statically defined by a table of

e.g. the drag and lift coefficient depending on the Mach number and the angle of attack.
The engineering level software code chosen was Missile DATCOM, a widely used code by the U.S. Air Force which comprises empirical, semi-empirical and analytical methods [2]. It is integrated via a dynamic library and therefore can be easily replaced by a more preferable aerodynamics code.

4.3. Propulsion

Besides the fixed engines available in ASTOS there are also optimisable engines. An optimisable liquid engine has been enhanced for MDO and an optimisable solid rocket motor has been newly created.
For both engine types estimation relations are used to estimate the engine masses as well as the engine dimensions (length and diameter) based on the thrust and the nozzle area respectively.

Liquid Engine

The liquid design engine's optimisable parameters are the engine sizing factor to vary the overall engine size (and with it the engine thrust), the chamber pressure, the mixture ratio, the throat area and the expansion ratio in order to find an engine that fits best to a given trajectory or set of trajectories.
The exhaust velocity and the characteristic velocity are either specified by the user with a constant value or a profile or they are computed by CEA (Chemical Equilibrium with Applications), a software developed by the NASA Glenn Research Center [3].

The engine's efficiency can be specified by a fixed factor that is multiplied by the engine's specific impulse or it is estimated based on the engine cycle, the oxidizer/fuel combination, the projected technology level (low cost, base line or high-performance) and the use in a lower or upper stage. The engine mass can be computed by a regression function depending on the maximum thrust of the engine and additional characteristics as used for the efficiency calculation. These estimations have been developed by the DLR Space Launcher Systems Analysis group (DLR-SART) based on an extensive review of existing liquid rocket engines.

Solid Rocket Motor

The solid rocket motor implementation enables the definition of various mass flow profiles. The user can select between predefined profiles modelling a tubular or star grain geometry or a generic profile that can model any grain geometry.

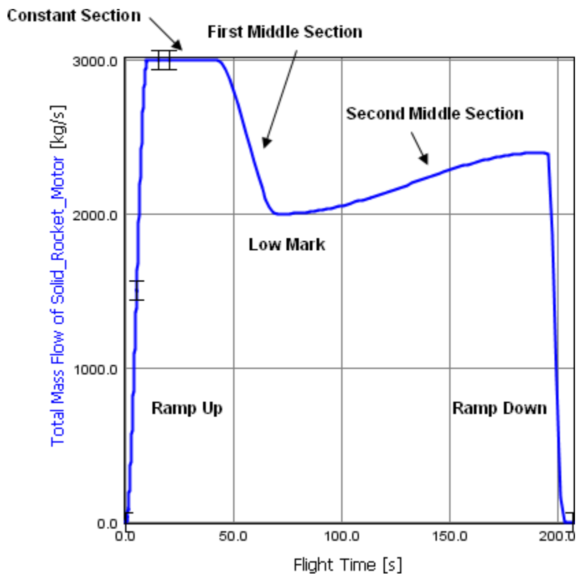

Figure 3. Solid propulsion mass flow for a star grain geometry

In case of the advanced star profile several parameters are optimisable (see Fig. 3), e.g. the duration of the solid propellant burning, the maximum mass flow or the initial and final burn behaviour. The local minimum of the mass flow that is used in mid-flight to reduce the maximum dynamic pressure can be optimised as well. In a similar way there are predefined optimisable parameters within the tubular profile.
The generic profile can respect any characteristics of grain geometries as the profile is defined completely freely via table data. The table data is then interpolated internally. The user may choose from several interpolation methods. Furthermore the table data is normalized with respect to both mass flow and time so that during optimisation the optimizer can scale the profile to the desired burn time and integrated mass flow.

4.4. Geometry

The geometry defines the sizing of the vehicle and its parts, like stage diameters and tank lengths. This data is very basic and is used by several disciplines: aerodynamics, weights and structures.
The launch vehicle is defined by its stages, the payload fairing, the payload itself and optionally the boosters. Each stage is defined by all its subcomponents: tanks, forward skirt, aft skirt, interstage, intertank, thrust frame and the propulsion being used. There are two stage types to choose from: mono- and dual-propellant. For dual-propellant stages three different tank configurations are available: separate tanks, common bulkhead and enclosed tank (Fig. 4).

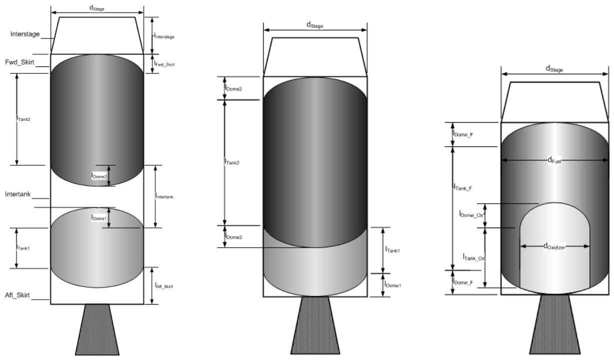

Figure 4. Tank configurations: separate tanks, common bulkhead and enclosed tank

The dimensions of all components have to be specified. The stage diameter and the tank lengths can be made optimisable to find the best stage design. The fairing dimensions and the boosters can be optimised as well. Constraints can be used to restrict the design, so that e.g. two stages will have the same diameter.

4.5. Weights

The weights are estimated in a detailed way summing up the propellant masses and the structural masses of all stages plus the payload fairing and the payload itself (and possibly the boosters).
The structural masses of the stages are computed with mass estimation relations (MER) using the detailed stage definition of the geometry discipline. The MERs can relate the mass of a stage component like the tank to various parameters like the propellant mass or the tank surface or even a combination of several. Linear or exponential relations can be used. Changes of the vehicle design during optimisation will be reflected immediately in the vehicle mass by direct data exchange between the geometry and the weights disciplines.
Optionally the weights can be estimated based on the wall thicknesses computed by the structural analysis shown in the subsequent section.

4.6. Structures

A structural analysis will compute the internal stresses and minimum wall thicknesses of all stage components. A simple beam approximation model has been selected that is fast and sufficiently accurate [4]. The launch vehicle is modelled as a bending beam so that the running loads can be computed from all external forces (aerodynamic and propulsive forces, gravity forces, inertial forces), The components are assumed to be thin-walled cylinders so that the running loads can be translated into minimum thicknesses [5].
Path constraints along the trajectory will then ensure that the optimisable wall thicknesses of all components are greater or equal then the minimum thicknesses.
The structural analysis can thereby counteract the aerodynamics' tendency to choose a long and slender launch vehicle which would minimize the aerodynamic drag.

5. INTEGRATION INTO ASTOS GUI

The complete model of the launcher and its environment is defined in a standard GUI display. The use of predefined building blocks that the user can choose from and quickly modify makes it easy and straight forward to define a complete launch vehicle.
The vehicle will be built up with vehicle stages (mono- or dual-propellant), propulsions, the payload fairing and the payload. The stage order can be arranged freely and all settings for a vehicle part are done in a window as shown for a stage in Fig. 5.
The environment including the planet, the atmosphere and optionally the wind are defined in the same way as a vehicle part using predefined models that can be easily modified and adapted. The optimisation specific features like cost functions, constraints or initial control laws are specified in the same GUI as well. Optimisable parameters are usually defined within the model part they belong to, e.g. the stage diameter and its bounds are defined within the stage definition.

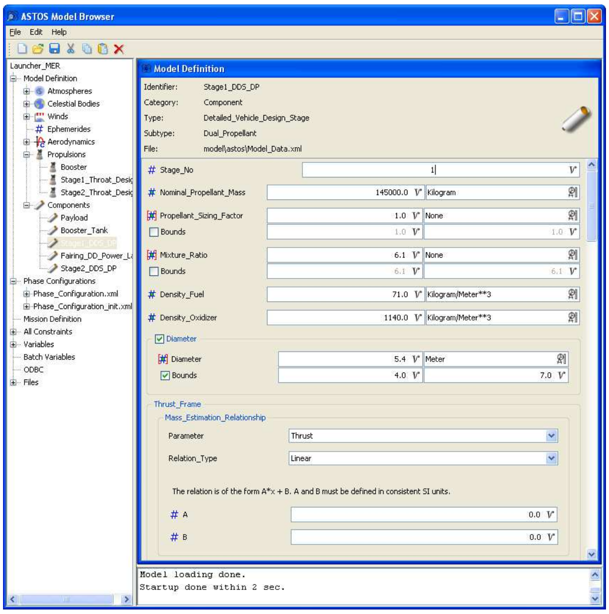

Figure 5. Editing a dual-propellant stage with the ASTOS Model Browser

The initial launch vehicle design as well as any optimised one can be visualized with a 3D viewer.

6. APPLICATION

The ASTOS capabilities for trajectory optimisation have been enriched to perform multidisciplinary design optimisation of expandable launchers. The new disciplines and the complex vehicle parts have been

integrated into the ASTOS core and the GUI seamlessly. The connection of trajectory and design optimisation performed at the same time is a promising approach for preliminary design where the physical models used are still fast enough for such an approach.
The new models allow detailed stage optimization of launchers. It is possible to fix various systems of the launcher on existing, e.g. available solid propulsion boosters, upper stage engines or whole stages. Beside that it is possible to define constraints, which helps to optimise a low cost design. Simple constraints are e.g. the maximum chamber pressure. But also more complex constraints, like the same size of two stages or the same basic engine in two different stages, can be easily considered.

7. AKNOWLEDGEMENT

The activities summarised in this paper have been performed by Astos Solutions under contract with ESA/ESTEC. The liquid rocket engine models have been developed in cooperation with Martin Sippel and Farid Gamgami of the DLR Space Launcher Systems Analysis group (SART).

8. REFERENCES

1. Tedford, N. P., Martins, J. R. R. A., On the common structure of MDO problems: A comparison of architectures, *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2006-7080, 2006.

2. Blake, W.B., *Missile Datcom User's Manual - 1997 Fortran 90 Revision*, AFRL-VA-WP-TR-1998-3009, Air Force Research Laboratory, 1998.

3. McBride, B.J., *Computer Program for Calculation of Complex Equilibrium Composition and Applications, I. Analysis*, NASA Reference Publication 1311, 1994.

4. Hutchinson, V., Olds, J., *Estimation of Launch Vehicle Propellant Tank Structural Weight Using Simplified Beam Approximation*, AIAA 2004-3661, 40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, 2004.

5. Graham, J. B., Luz, P. L., *Preliminary In-Flight Loads Analysis of In-Line Launch Vehicles using the VLOADS 1.4 Program*, Tech. Rep., Marshall Space Flight Center, NASA/TM-1998-208472, 1998.