Flight Simulation for Advanced Launchers

January 1994

Irv. E. Mooij

3D VIEW of a TSTO SPACEPLANE
Flight Simulation for Advanced Launchers

Ir. E. Mooij
Title: Flight Simulation for Advanced Launchers

Author(s): E. Mooij

Abstract: The design of an advanced launcher, such as an aerospace plane, is strongly dependent on flight simulation. The performance of the propulsion system, highly integrated with the airframe construction, is, for instance, largely influenced by flight conditions. Within the framework of the Space-Plane Project of the Faculty of Aerospace Engineering (Delft University of Technology), a proposal is made for the conceptual design of space planes, with the emphasis on trajectory simulation. The layout of a design package is presented. This package consists of several modules, e.g., the design of the aerothermodynamic configuration, the (air-breathing) propulsion system and the geometry and mass properties. To compute the optimal trajectory for a given design (based on, for instance, minimal fuel consumption), two possible Trajectory Optimizers are reviewed, namely ALTOS and ASCENT. The optimized trajectory will serve as an input for a six-degrees-of-freedom flight-dynamics software. This software is based on the re-entry simulation tool START. Extensions w.r.t. the current version are presented to meet the requirements for proving the steerability and controllability of the space plane. The main extensions are related to variable mass properties, guidance and control, and propulsion.

Keyword(s): space plane, flight simulation, trajectory optimization, conceptual design, ALTOS, ASCENT, START

<table>
<thead>
<tr>
<th>Issue</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>26-01-94</td>
</tr>
<tr>
<td>Prepared</td>
<td>E. Mooij</td>
</tr>
<tr>
<td>Verified</td>
<td>P.Ph. vd Broek</td>
</tr>
<tr>
<td>Approved</td>
<td>B.T.C. Zandbergen</td>
</tr>
</tbody>
</table>
## CHANGE RECORD

<table>
<thead>
<tr>
<th>Issue No.</th>
<th>Rev. No.</th>
<th>Date</th>
<th>Pages Changed/ Added/ Deleted</th>
<th>Topics Introduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>01</td>
<td>08-02-93</td>
<td>All</td>
<td>First version</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>02-11-93</td>
<td>54, 55</td>
<td>Comments processed</td>
</tr>
<tr>
<td></td>
<td>03</td>
<td>26-01-94</td>
<td></td>
<td>Tool development adjusted</td>
</tr>
</tbody>
</table>
Table of contents.

Notations. .................................................................................. iv

1. Introduction. .......................................................................... 1
   1.1. Background. ................................................................. 1
   1.2. Space-plane research in the Netherlands. ......................... 4
   1.3. The scope of the work. .................................................. 6

2. Conceptual layout of a space-plane design model. ..................... 8
   2.1. Introduction. ............................................................... 8
   2.2. Elements of the design model. ...................................... 10
       2.2.1. Space-plane related modules. .............................. 10
       2.2.2. Flight-simulation model. .................................... 13
       2.2.3. Space-plane database. ........................................ 16
   2.3. The Taguchi method and trajectory optimization. .............. 17

3. Flight-simulation model. ....................................................... 21
   3.1. Introduction. ............................................................... 21
   3.2. Trajectory Optimizer. ................................................. 23
       3.2.1. ALTOS. ............................................................ 23
       3.2.2. ASCENT. ........................................................ 27
   3.3. Flight-dynamics software. ............................................ 31
       3.3.1. START Version 1.0. ........................................... 31
       3.3.2. START Version 2.0. ........................................... 34
       3.3.3. START Version 2.1. ........................................... 38
       3.3.4. START Version 3.0. ........................................... 40
       3.3.5. START Version Flight Simulator? ......................... 52
       3.3.6. Current status of START. ................................... 53
       3.3.7. Future development. ........................................... 54

References. ............................................................................... 56

Appendix A - Work packages AEOLUS Project
Notations.

Roman

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF</td>
<td>aerodynamic coefficient</td>
<td>[unit]</td>
</tr>
<tr>
<td>D</td>
<td>drag force</td>
<td>N</td>
</tr>
<tr>
<td>h</td>
<td>height (geometric altitude)</td>
<td>m</td>
</tr>
<tr>
<td>J</td>
<td>zonal harmonic</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>lift force</td>
<td>N</td>
</tr>
<tr>
<td>L</td>
<td>rolling moment</td>
<td>Nm</td>
</tr>
<tr>
<td>m</td>
<td>mass</td>
<td>kg</td>
</tr>
<tr>
<td>M</td>
<td>Mach number</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>pitching moment</td>
<td>Nm</td>
</tr>
<tr>
<td>na</td>
<td>axial acceleration (in Earth g)</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>yawing moment</td>
<td>Nm</td>
</tr>
<tr>
<td>qdm</td>
<td>dynamic pressure</td>
<td>N/m²</td>
</tr>
<tr>
<td>r</td>
<td>distance to the Centre of Mass of the central body</td>
<td>m</td>
</tr>
<tr>
<td>S</td>
<td>side force</td>
<td>N</td>
</tr>
<tr>
<td>t</td>
<td>time</td>
<td>s</td>
</tr>
<tr>
<td>V</td>
<td>relative flow velocity</td>
<td>m/s</td>
</tr>
<tr>
<td>x</td>
<td>x-position</td>
<td>m</td>
</tr>
<tr>
<td>Xi</td>
<td>derivation variable i</td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>y-position</td>
<td>m</td>
</tr>
</tbody>
</table>

Greek

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>angle of attack</td>
<td>rad</td>
</tr>
<tr>
<td>β</td>
<td>angle of sideslip</td>
<td>rad</td>
</tr>
<tr>
<td>γ</td>
<td>flight-path angle</td>
<td>rad</td>
</tr>
<tr>
<td>δ</td>
<td>planetocentric latitude</td>
<td>rad</td>
</tr>
<tr>
<td>δₜ</td>
<td>throttle setting</td>
<td>rad</td>
</tr>
<tr>
<td>εₜ</td>
<td>thrust vector angle</td>
<td>rad</td>
</tr>
<tr>
<td>σ</td>
<td>bank angle</td>
<td>rad</td>
</tr>
<tr>
<td>τ</td>
<td>planetocentric longitude</td>
<td>rad</td>
</tr>
<tr>
<td>χ</td>
<td>heading</td>
<td>rad</td>
</tr>
<tr>
<td>ω</td>
<td>angular rate expressed in the body frame</td>
<td>rad/s</td>
</tr>
</tbody>
</table>

Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACS</td>
<td>Attitude Control System</td>
</tr>
<tr>
<td>AEOLUS</td>
<td>Advanced Earth to Orbit Launcher Upgrade Studies</td>
</tr>
<tr>
<td>ALTAP</td>
<td>Advanced Launcher Trajectory Analysis Programme</td>
</tr>
<tr>
<td>ALTOS</td>
<td>Advanced Launcher Trajectory Optimization Software</td>
</tr>
<tr>
<td>AMLS</td>
<td>Advanced Manned Launch System</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>APAS</td>
<td>Aerodynamic Preliminary Analysis System</td>
</tr>
<tr>
<td>CAPS</td>
<td>Computer Aided Project Study</td>
</tr>
<tr>
<td>CFD</td>
<td>Computational Fluid Dynamics</td>
</tr>
<tr>
<td>CIRA</td>
<td>COSPAR International Reference Atmosphere</td>
</tr>
<tr>
<td>CoM</td>
<td>Centre of Mass</td>
</tr>
<tr>
<td>CONSIZ</td>
<td>CONfiguration SIzIng</td>
</tr>
<tr>
<td>dof</td>
<td>degrees of freedom</td>
</tr>
<tr>
<td>DUT</td>
<td>Delft University of Technology</td>
</tr>
<tr>
<td>ESA</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>ESTEC</td>
<td>European Space research and TElchnology Centre</td>
</tr>
<tr>
<td>FAE</td>
<td>Faculty of Aerospace Engineering</td>
</tr>
<tr>
<td>FDS</td>
<td>Flight-Dynamics Software</td>
</tr>
<tr>
<td>FSS</td>
<td>Fokker Space and Systems</td>
</tr>
<tr>
<td>HOTOL</td>
<td>HOrizontal Take-Off and Landing</td>
</tr>
<tr>
<td>LEO</td>
<td>Low Earth Orbit</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>NASP</td>
<td>National AeroSpace Plane</td>
</tr>
<tr>
<td>NIVR</td>
<td>Netherlands Agency for Aerospace Programs (Dutch abbreviation)</td>
</tr>
<tr>
<td>NLP</td>
<td>Non-Linear Programming</td>
</tr>
<tr>
<td>NLR</td>
<td>Dutch Aerospace Laboratory (Dutch abbreviation)</td>
</tr>
<tr>
<td>POST</td>
<td>Program to Optimize Simulated Trajectories</td>
</tr>
<tr>
<td>SLSQP</td>
<td>Sequential Least Squares Programming</td>
</tr>
<tr>
<td>SMART</td>
<td>Solid Modelling Aerospace Research Tool</td>
</tr>
<tr>
<td>SPE</td>
<td>Stork Product Engineering</td>
</tr>
<tr>
<td>SPP</td>
<td>Space-Plane Project</td>
</tr>
<tr>
<td>SSTO</td>
<td>Single-Stage-To-Orbit</td>
</tr>
<tr>
<td>START</td>
<td>Simulation Tool for Atmospheric Re-entry Trajectories</td>
</tr>
<tr>
<td>TO</td>
<td>Trajectory Optimizer</td>
</tr>
<tr>
<td>TSTO</td>
<td>Two-Stage-To-Orbit</td>
</tr>
<tr>
<td>TVC</td>
<td>Thrust-Vector Control</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>UK</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>US</td>
<td>United States</td>
</tr>
</tbody>
</table>
This page has been intentionally left blank
1. Introduction.

1.1. Background.

The launch of a payload into an Earth orbit is nowadays still highly inflexible and expensive. Every launch of a (conventional) expendable launcher like Ariane 4, Proton, Delta, Titan/Centaur represents a considerable waste of materials, and manufacturing and processing capabilities. Besides, the launch is restricted to a limited number of specially designed launch sites; weather conditions at these sites have often caused intolerable delays for time-depending launches.

A first attempt in increasing the flexibility and decreasing the cost came with the American Space Shuttle. Although the Space Shuttle represents a major technological achievement in the space transportation capabilities of the United States (US), one of the goals of the Space Shuttle, namely low-cost space transportation, has not been met. These higher than anticipated costs can be traced to lower than expected flight rates, labour-intensive ground and flight operations, and recurring hardware costs (the big external tank is still not reusable). The next generation of launch vehicles must reduce the costs of delivering payloads to space, or mission planners will continue to be faced with the prospect of restricted space activities in an era of budget constraints (Powell et al., 1990).

It is nowadays believed that to significantly reduce the costs of space transportation, an advanced space launcher or even a range of advanced space launchers has to be developed (Branscome et al. (1990), Darwin et al. (1991), Feustel-Büechl et al. (1991)). Two categories can be identified. On one hand, the design and manufacturing of conventional launchers can be improved and optimized, with attention to modular design (e.g., the family of Ariane launchers), high reliability, ease of manufacturing and safety, reusability, and reduced dry mass (Zandbergen, 1992). On the other hand, one can think of fully reusable vehicles, which take off and land horizontally like aircraft, operate from airports and have a short turn-around time. These vehicles, called space planes from now on, form the main interest of this report.

Potential economic benefits from the space plane will be large, mainly due to three fundamental aspects (AEOLUS Team, 1992):

- re-use of the space plane; this eliminates the large cost of building a new launcher for each mission,
- improvement of performance by using air-breathing engines\(^1\),
- reduction of operational cost, because of the use of existing, although modified, airports.

\(^1\) We are going forward with research on a new Orient Express that could, by the end of the next
decade, take off from Dulles Airport and accelerate up to 25 times the speed of sound, attaining low Earth orbit or flying to Tokyo within two hours.' With these words, the former president Reagan of the US, in his 1986 State of the Union message, announced the beginning of a multi-billion effort known as the National AeroSpace Plane (NASP) program (Heppenheimer, 1987). The NASP is perhaps the most ambitious concept of all space-plane related studies. It is a Single-Stage-To-Orbit (SSTO) vehicle (Fig. 1.1), propelled by both RAM-jet and SCRAM-jet propulsion in the air-breathing phase, possibly up to Mach 15, followed by a rapid pull-up and using rocket propulsion to obtain orbital velocity.

Fig. 1.1 - The National AeroSpace Plane (Stanley et al., 1992c).

Of course, also the rest of the world is engaged with all kinds of studies and research leading to a feasible design of a space plane. As a combined European effort, the European Space Agency (ESA) is performing a so-called Winged Launcher Configuration study, aiming at the development of air-breathing propulsion systems to be used by either SSTO or TSTO (Two-Stage-To-Orbit) space planes (Grallert et al., 1991). Several ESA-member states are studying varying concepts independently as well.

The United Kingdom (UK) has been working on HOTOL (an acronym for HOrizontal Take Off and Landing) for quite some time now. HOTOL is an unmanned SSTO vehicle, capable of bringing 7000 kg into Low Earth Orbit (LEO) (Conchle, 1985). One of the earlier designs used a rocket-powered trolley as an undercarriage during take off. Lack of support from the UK government and lack of interest by the other European countries to participate, forced a change in the original concept. Nowadays the so-called interim HOTOL is being studied, being a version using rocket propulsion only and launched from atop an Antonov AN-225 transport aircraft (Fig. 1.2).
Fig. 1.2 - Basis of the AN-225/HOTOL proposal.

Fig. 1.3 - Sänger system configuration with the twin upper stages HORUS-C and HORUS-M (Koelle et al., 1990).
In France, three concepts of a winged launcher are under study: TARANIS, a TSTO concept with a rocket-propelled first and second stage, STAR-H, a two-stage carrier, of which the second stage is based on Hermes with an additional propulsion stage, and STS-2000 (Wagner et al., 1991), an SSTO space plane, like the NASP. Sänger is the German concept of a TSTO space transportation system (Koelle et al., 1989). During the atmospheric flight, the first stage is propelled by hydrogen-fuelled turbo-ram-jets. At Mach 7, the upper limit for ram-jet propulsion, the two stages separate. The second stage, either the manned HORUS-M or the unmanned HORUS-C (Fig. 1.3), is rocket-powered accelerated to orbital velocity for LEO, while the first stage flies back to Earth to land on a conventional airport.

Although most countries are still working on their own, it has become obvious that the cost of developing a space plane largely exceeds the national space budgets. This is indicated by, for instance, Germany, which is looking for cooperation on the Sänger project throughout Europe. It is therefore necessary that the several countries combine their efforts, in order to come to one single design. One can think of a European effort only (possibly in ESA context), but even then it is unlikely that the budgets can be cleared. The general opinion at the AIAA Fourth National AeroSpace Plane Conference in Orlando, Florida (December 1992) was, that cooperation should be found world-wide, basically meaning a combined effort of the US and ESA, although other nations, like Japan and the former Soviet Union, should not be excluded beforehand.

1.2. Space-plane research in the Netherlands.

It is crystal clear that any space-plane activity of The Netherlands must be found in the framework of the ESA-programmes. Now, the Dutch aerospace industry has a major involvement in the development of components for Europe's next generation launch vehicle, the Ariane 5 (as it also participated in the previous versions of Ariane). At present, it is not yet clear in which direction the European future launcher development is heading. There are, broadly speaking, two alternatives namely expendable launchers and reusable launchers, of which winged air-breathers are the most evident example.

To ensure work on future launchers in ESA programmes, in 1988 TNO-PML, Stork Product Engineering (SPE) and NLR, supported by the Netherlands Agency for Aerospace Programs (NIVR), started a joint project aiming at the development of key technology for future air-breathing space launchers (Zandbergen, 1992). In 1992, Fokker Space and Systems (FSS) joined the team, as did Delft University of Technology, Faculty of Aerospace Engineering (DUT/FAE).

Realization of future ESA participation entails two main aspects (AEOLUS Team, 1992): technology and business development. Technologically and with respect to engineering, Dutch capabilities have to be upgraded to a level allowing The Netherlands to be considered a serious partner. With respect to business development, Dutch intentions and capabilities have to be marketed to fit within the European effort. The chosen approach is a separate national technology and business development programme, called Advanced Earth to Orbit Launcher Upgrade Studies (AEOLUS). This programme is a broad technological covering of a strategically chosen representative subsystem, namely the engine inlet subsystem of a winged air-breathing launch
vehicle. This subsystem appears in principle to be within Dutch reach, both technologically, commercially and politically. It has been selected from a list of alternatives, based on a number of criteria, although it is not the law of Medes and Persians that it should be pursued at all cost. The list is given below, in descending order of attractiveness (AEOLUS Team, 1992):

- inlet subsystem, including supporting structure, moving parts of a controllable geometry, actuators, mechanisms, thermal control system and interface electronics,
- body flap subsystem of a space-plane booster (e.g., Sänger) or orbiter (e.g., Horus), including actuators, mechanisms and interface electronics,
- elevon subsystem of launchers as under body flap,
- fin/rudder subsystem of launchers as under body flap, and
- canard wing subsystem (if applicable) of a booster (Ariane 5, Sänger), including supporting structure, actuators, mechanisms, thermal control system and interface electronics.

At DUT/FAE, two main space-related research topics have been indentified, namely Earth Observation and Space-Plane Design and Flight Dynamics. To work on the latter topic, the so-called Space-Plane Project (SPP) group has been set up, including the disciplinary groups/sections: Aerodynamics/High-Speed Aerodynamics, Aerospace Design and Flight Mechanics/Space Research and Technology, and Stability and Control (Zandbergen, 1992). The aim of this project group is twofold:

```
to develop a computerized simulation model, which can be used for the analysis of the flight trajectory and of the (real time) flight behaviour, thereby taking into account the vehicle characteristics (architecture, geometry, size, gross mass, payload mass and dry mass), geophysics, aerodynamics, aeroheating and propulsion (Zandbergen, 1992),
```

![Diagram]

**Fig. 1.4 - The Space-Plane-Project schedule (Zandbergen, 1992).**
and
to develop a vehicle scaling model, which can be used for an optimal conceptual design
of several types of advanced launchers.

The planned duration of this project is 5 years, as has been schematically indicated in Fig. 1.4.
It is clear, that this project provides a good basis to participate in the AEOLUS Project.

1.3. The scope of the work.

The design proposal, presented in this report, is strongly related with the SPP, as mentioned
in the previous Section. This proposal will be the heart of a PhD study of the author, performed
at DUT/FAE, and will be explained in the following Chapters.

The performance of, for instance, the propulsion system of a space plane, is strongly dependent
on the flight dynamics. Since the propulsion system is an integral part of the airframe, a change
in this system affects the complete design of a space plane. For this reason, flight simulation
must be included in an early phase of the design (Schmidt, 1992). It is possible to 'decompose'
the space plane into a number of subsystems. It is convenient to have these subsystems stored
in a database in the conceptual design phase, so that a new configuration can be easily com-
posed and tested.

In Chapter 2, the conceptual layout of a space-plane design model is presented, consisting of
these three parts: space-plane related design modules, a flight-simulation model and a space-
plane database. Furthermore, some remarks are made concerning an efficient design and simu-
lation scheme, the Taguchi method, and in particular the relation with trajectory optimization.
We stress here, that the development of detailed space-plane design modules is not considered
to be part of the study, although it may be possible that some simple models will be derived.

The actual work to be done in the coming three years is presented in Chapter 3. This Chapter
deals with the development of a flight-simulation model. With this model it must be possible to
prove the steerability and controllability of a space plane during both ascent and descent flights.
The model consists of two major parts, a trajectory optimizer and flight-dynamics software. The
trajectory optimizer must be able to generate an optimized reference trajectory. For these kinds
of performance analyses, it is sufficient that this software is a so-called three-degrees-of-
freedom (3 dof) code. The flight-dynamics software must be capable of simulating 6 dof, i.e.,
the space plane is treated as a rigid body. In that way, the dynamics and stability of the vehicle
can be simulated. The generated optimized trajectory serves as an input for the 6-dof code, so
that the plane can be flown along this trajectory and the responses to perturbations (e.g., the
influence of wind or design uncertainties) can be studied. All simulations to be performed are
thought to be part of one programme, from now on called the Advanced Launcher Trajectory
Analysis Programme (ALTAP).

As can be seen, part of the work to be conducted for the AEOLUS programme fits well within
the currently proposed work. For this reason, in Appendix A two related work-package descrip-
tions are included. WP-151 asks for the inventory of existing tools, including their availability.
A review of these tools leads to requirements concerning the modules to be developed. In fact, the proposal in this report is partly a result of this review. WP-212 encompasses an investigation into typical space-plane concepts and trajectories, trajectory constraints, and guidance and control laws. Viewed in this light, it is important to refer to Mooij et al. (1993) and Marée et al. (1993), who report on WP-151 and WP-212, respectively.

This report is more than just an overview of the work to be done in the coming three years. It also presents many references, which can serve as a basis for this work. Where possible, the references have been arranged per subject in the text, so that it is easy to have access to these sources. It must be stressed, that this list does not exhaustively cover all aspects of the conceptual design of space planes nor the field of flight simulation. Numerous papers and reports have been written on the corresponding topics, and although the list is long, it is only a (personal) selection.
2. Conceptual layout of a space-plane design model.

2.1. Introduction.

In the predesign phase of any transport vehicle, several concepts are usually compared to come to the most efficient design. In the field of launchers, one can, for instance, compare rocket and air-breathing vehicle concepts for Earth-to-orbit transportation (Dorrington, 1990). Once the choice has been made to continue developing an advanced air-breathing launcher, it becomes interesting to compare single-stage and two-stage vehicles (Dorrington, 1987 and Stanley et al., 1992c). If an Ssto vehicle could be built that is similar in size and weight to a comparable Tsto vehicle, it would enjoy many advantages because only one vehicle would have to be developed, manufactured and operated.

Focusing on one design, it is possible to do a more detailed analysis (Gord et al., 1990 and Schmidt, 1992). The problems to be encountered in the design of systems for the precise control of hypersonic vehicles pose significant research challenges if successful guidance and control systems are to be developed for this new class of vehicle (Schmidt, 1992). These challenges arise due to the stringent mission requirements on the vehicle, and the highly integrated configuration designs being considered. The presence of critical coupling between several vehicle systems can be clearly exposed, as well as the dynamic interactions between these sub-systems, while considering a selected generic hypersonic configuration. Schmidt shows that this vehicle configuration, similar to the NASP, is an unstable (in particular in pitch motion), highly coupled, aeropropulsive/aeroelastic system, with large variations in attitude-dynamic characteristics over its extensive flight envelope. The centre of pressure, for instance, will move radically aft as the angle of attack increases and the rudder will be shielded and ineffective. The pitch control surface inputs will give rise to large disturbances in the engine inlet conditions as well as the thrust response. Furthermore, when the engine control inputs, like fuel flow rate and effective diffuser area ratio, are considered, they give rise to significant attitude disturbances. The strong coupling necessitates a highly integrated airframe-engine control system.

Since the issues mentioned above may ultimately limit the feasibility of some configurations, the flight-dynamic modelling and analysis must be performed early in the design cycle such that critical dynamics and control issues associated with a candidate configuration may be exposed. Therefore, such dynamics and control analyses cannot be postponed until detailed numerical models are available, because configuration changes may then be too expensive (Schmidt, 1992).

From the above discussion, it has been become clear that in the conceptual design phase of a space plane, there is a strong coupling between the actual design and performance analysis. A justified design methodology should therefore include not only a design package w.r.t. the shape, mass properties, aerodynamics, propulsion, etc., and also trajectory and performance analysis. These tools can be simplified for a first design (Chaput, 1992), but later on more detailed methodologies are required (Townend, 1991). The design becomes an interactive process, which must finally lead to an optimal configuration (Hattis, 1981, Hattis et al., 1989, Schoettle et al., 1989 and Furniss et al., 1990).
As an example, the global design process of an SSTO vehicle, as described by Martel (1990) is shown in Fig. 2.1. Martel describes that preliminary studies showed the high sensitivity of the maximum allowed payload to the aerodynamic drag coefficient and to the mass of the structure, and the strong influence of forebody interactions on propulsion performances. As a good evaluation of these parameters is necessary, the aerodynamic design was not included in the on-line optimization, but in an external highly interactive design loop with some main evaluations at certain steps of the study.

Fig. 2.1 - The global design process of a SSTO vehicle (Martel, 1990).

In the remainder of this Chapter, we will introduce the layout of a conceptual design model. Section 2.2 describes the elements of the design model. Our model will consist of three main modules, i.e., the actual design module (including mass and geometry definition, aerothermodynamics computation, propulsion-system definition), a flight-dynamics analysis tool (including trajectory optimization and 6-dof flight simulation) and a space-plane database, to store several concepts and/or subsystems. Of course, there is a two-way relation between each of the modules: the design is verified with flight simulation, which may impose some constraints on the design, when it becomes obvious that the mission cannot be fulfilled. The design, which can be used during both the design process and the flight-dynamics analysis, can be stored in and retrieved from the database. The layout of the described design model has been depicted in Fig. 2.2.
In Fig. 2.2, we have made a distinction between the vehicle description and the flight simulation. It must be stressed that the computation of the trajectory is an essential part of the (conceptual) design, so this distinction is purely artificial, and has been introduced to indicate that we will focus on the flight-simulation module without paying too much attention to the actual design of the space plane. It does not mean that the trajectory computation is more important than the design.

Additional information on design models of hypersonic air-breathing vehicles can be found in, for instance, Alberico (1992), Bowcutt (1992) and Stanley et al. (1992a). Alberico gives an overview of the development of an interactive computer tool for synthesis and optimization of these vehicles at the McDonnel Douglas Corporation and NASA. Bowcutt describes a method for optimizing the aeropulsive performance of a hypersonic cruise or accelerator vehicle accounting for the impacts and requirements of propulsion integration and trim. The design model of Stanley et al. will come up for discussion in the next Section.

In Section 2.3, we introduce the Taguchi method, which can serve as an efficient simulation scheme, and which has been used to indicate the need for optimization.

2.2. Elements of the design model.

2.2.1. Space-plane related modules.

Stanley et al. (1992a) describe the conceptual design of a rocket-powered, two-stage fully reusable Advanced Manned Launch System (AMLS). They present the various tools, which have been used in the design process (Fig. 2.3). As can be seen in this Figure, tools which take the effects of the trajectory, weights/sizing, geometry, aerodynamics and aeroheating into account, are considered. Nota bene: a similar approach has been used by Stanley et al. (1992b) to compare SSTO and TSTO launch vehicles.

All of the trajectory analysis is performed using the 3-dof Program to Optimize Simulated
Trajectories (POST), see also Brauer et al. (1975a-b and 1977). The weights and sizing analysis is performed using the NASA-developed CONfiguration SIZing (CONSIZ) weights/sizing package, using mass-estimation relations based on historical regression. All of the geometry and subsystem packaging is performed with the NASA-developed Solid Modelling Aerospace Research Tool (SMART). This program generates three-dimensional Bezier surface representations of aerospace vehicles for use in aerodynamic and structural analysis. The Aerodynamic Preliminary Analysis System (APAS) was used to determine vehicle aerodynamics, see also Cruz et al. (1989). An aeroheating analysis is performed with the empirical, Space-Shuttle-correlated Miniver aeroheating package. No information on the propulsion-system package could be found.

Fig. 2.3 - The AMLS vehicle design process (Stanley et al., 1992a).

Another example of the generation of an advanced-launcher configuration is given by Shaughnessy et al. (1990). They describe the models used to generate mass, aerodynamic and propulsion properties of this so-called SSTO wing-coned reference vehicle. Besides, all tables with numerical values are presented. This vehicle has been extensively used in several analyses (e.g., Powell et al., 1991). Besides, this vehicle was defined to be one of the reference vehicles for the simulation package ASCENT (Section 3.2.2).

Furniss et al. (1990) state that, since the performance of a space plane is extremely sensitive to design and mission changes, it is essential to have a detailed model of the so-called Total System (see Fig. 2.4). The use of simple analytical representations is totally inadequate. They introduce a project study integrated design tool called CAPS (Computer Aided Project Study), which has been extensively used to perform design trade-off studies. Due to the interactions between trajectory and design, the only conclusion can be that the optimum ascent trajectory can only be derived when its impact on propulsion system sizing and mass, vehicle control requirements (and hence actuation and power supply system mass), wing aeroelastics, wing design, and fuselage shape have all been addressed. It must combine minimum propulsion mass, optimum fuselage and wing shapes, with minimum ascent propellant. The interrelations
are graphically depicted in Fig. 2.5.

![Diagram](image)

Fig. 2.4 - Schematic of Total System Model (Furniss et al., 1990).

![Diagram](image)

Fig. 2.5 - Configuration optimum (Furniss et al., 1990).
Based on the above discussion, we come to a number of necessary modules, in order to design an advanced launcher. They will be summarized below. We already indicated more than once, that trajectory analysis forms an integral part of the design process. This module will be introduced in the next Section, and discussed in great detail in the next Chapter.

The modelling modules, which can be identified, are:

- aerodynamics (see, for instance, Cruz et al., 1989 and Battrick, 1991); for computation of the aerodynamic force and moment coefficients,
- geometry; the output of this module must have such a format, that it can easily be read by a code, which can compute the aerodynamic properties (Computational Fluid Dynamics (CFD) codes),
- mass; this module must be able to compute the location of the CoM and the corresponding inertia properties; possibly, a (linear) scaling algorithm must be included for an easy computation of similar shapes and mass distributions,
- propulsion; this module must be capable of deriving the properties of both air-breathing turbo- and (SC)RAM-jets, and rocket engines,
- thermodynamics (see, for instance, Battrick, 1991 and Serrano Martinez, 1989),
- trajectory; for the evaluation of space-plane performance (see the next Section).

At this moment, the modules dealing with structural analysis and aeroelastics are considered to be beyond the scope of the current design model. The guidance-and-control requirements to fly a predefined trajectory result in output from the geometry and aerodynamics module (w.r.t. the control surfaces). Because the guidance and control system will be defined in combination with the trajectory analysis, a separate module has not been defined.

2.2.2. Flight-simulation model.

The design of a space plane is an interactive process, where trajectory simulation is involved at a very early stage. Once a conceptual design of a space plane has been completed, it is interesting to see whether it can actually fulfil its mission under any circumstance. Fulfilment of the mission relates on one hand to the ability to fly a nominal (usually optimal) trajectory. On the other hand, the optimal conditions, e.g., no wind, will never be completely encountered so it is necessary to see how the space plane responds to all kinds of disturbances. In this respect, we mean with disturbances not only the influence of wind (steady-state wind, gusts, wind shear, turbulence), but also uncertainties in the design of the space plane, resulting in deviations in mass properties, aerodynamic coefficients and propulsion parameters. All these disturbances and uncertainties may pose additional requirements to the guidance and control system. In the remainder of this Section, we will elaborate on this idea.

The motion of the space plane under the influence of external forces can be split into two parts, namely the motion of the centre of mass (com) of the vehicle and the motion of the space plane around its com. For a non-elastic body, we can distinguish six components of motion, also called degrees of freedom (dof). These consist of three translations along the axes of a reference frame (the variation with time of the position coordinates) and three rotations around
these axes (the variation with time of the attitude angles).

Studies of the motion of the space plane can be separated into two parts. On one hand, we can discern the study of the translations of the com only, i.e., the space plane is considered to be a mass point. The corresponding motion is the so-called 3-dof motion. On the other hand, the combined translations and rotations of the space plane, which is now considered to be a (non-elastic) body with finite dimensions, can be examined, requiring so-called 6-dof simulations. Both 3-dof and 6-dof simulations are used in special areas of interest. Nota bene: when the longitudinal stability is a special area of interest, it is also possible to enhance the 3-dof motion with the pitching motion only. In that case, we have an intermediate form: the 4-dof motion.

In the first place, 3-dof simulations are an integral part of the conceptual design process. The trajectory is considered to be a subsystem. The task is, in this case, to generate an optimal trajectory for a given design and operations requirements in order to get an overview of the performance of the vehicle and to serve as input for the design (and performance) of thermal protection systems, the construction, the propulsion system, the attitude control system and others.

In the second place, these kind of simulations can give insight in the steering capabilities of a given design. Is it possible, for instance, that for a given payload (e.g., 7000 kg) and mission (e.g., the space plane should arrive in a Low Earth Orbit (LEO)), the space plane can still fulfil its mission despite uncertainties in the design parameters (i.e., a variation in mass, geometry, aerodynamics, and/or propulsion properties)? What will be the influence of a variation in mission constraints (e.g., a maximum acceleration, maximum dynamic pressure, or maximum heat load) for a given design? The (manned) space plane should always be able to abort its mission in case of failures. Can that be achieved? And last but not least, should either Thrust Vector Control (TVC), aerodynamic control, or a combination of both be used for guidance?

6-dof simulations are used when the attitude and inertia of the vehicle are of importance. Since the results of the 3-dof simulations are only meaningful if the assumptions, which were made, can be proven to be correct. For instance, when guidance of a space plane is studied, it is assumed that the control system can generate the required torques to change the attitude of the vehicle towards the on-coming flow. Besides, it is often assumed that these attitude changes take place instantaneously. So what should be verified is the steerability and controllability for (at least) the nominal mission and payload, including possible aborts. This includes the verification of the size and location of the control surfaces, the location of attitude-control thrusters, and the range of thrust-vector angles. Besides, when aerodynamic control is used, the deflected control surfaces give rise to an additional aerodynamic drag, the so-called trim drag. This effect should be taken into account when the effectiveness of the control system or the performance of the space plane is studied.

A second study topic is the robustness of a given design. What is, for example, the influence of wind on the angles of attack and sideslip (and thus the aerodynamic properties)? How will the mass properties (total mass, location com, inertia tensor) change due to fuel consumption and the (alternative) location of fuel tanks? Sensors, which are responsible to give information about the angle of attack, for instance, show a certain delay. How will this delay influence the
controllability of the space plane? What will happen when a control surface fails and cannot be moved any more? Can, in that case, the control function be taken over by another system, either backup or alternative? Besides, uncertainties in the design (w.r.t. mass properties, aerodynamics and propulsion) might create corresponding problems for the control system.

Apart from the above mentioned study topics, also the following are of interest:

- how is the static stability of the space plane with control system for the complete trajectory;
- what is the influence of different control systems (or algorithms) on the performance of a given design;
- what will be the influence of perturbations (e.g., a steady-state wind, horizontal or vertical gusts, an offset of the location of the com), resulting in changes in the flow conditions (angle of attack, angle of sideslip, flow velocity) near the inlet of the propulsion system, on the performance of the propulsion system;
- can off-nominal values of the aerodynamic angles be compensated for, by flying with the nose 'in the wind', taking into account the resulting trim drag of the deflected control surfaces;
- w.r.t. the flexibility of the given design, what will be the steerability and controllability of other than the nominal mission and payload.

As can be concluded from the above (incomplete) list of study topics, flight simulation is an indispensable part of the design process.

The flight-simulation model, as proposed here, consists of two modules. On one hand there must be a program that can generate an optimized (3-dof) trajectory, using a complete, given design of a space plane. The optimization process can be based on, for instance, mass optimization, i.e., maximize the payload for a given fuel mass, or minimize the fuel mass for a given payload. Typical output can be a mission profile in the form of velocity versus altitude, with a complete flight-path-angle history and switching conditions for the different propulsion systems. This program will be called the Trajectory Optimizer (TO) from now on (in Section 2.3, we will discuss the need for optimization in more detail, hereby using the Taguchi method). Nota bene: one can also think of cost optimization (Arend, 1989 and Koelle, 1992), which includes design, development and operational cost. This kind of optimization is considered to be beyond the scope of the present study.

On the other hand, we must also have a program with which the (6-dof) flight dynamics of the space plane can be examined, especially in combination with disturbances. This program will be called the Flight-Dynamics Software (FDS). The output of the TO will serve as an input for the FDS. Of course, there are some requirements, which the FDS has to meet. It must be capable of simulating ascent and descent flights, flying along a predefined trajectory and responding to disturbances.
2.2.3. Space-plane database.

In the early design phase, the configuration of the space plane keeps on changing. For each developed configuration, we have to check whether the space plane can actually fulfil its mission, thereby finally coming to an optimal design. One can understand that changes in the design deal with the geometry, mass properties, the aerodynamic configuration, the type of propulsion system with the related parameters, the location, shape and size of control surfaces. It is evident that each of these design changes do not have to occur simultaneously.

To keep track of the different designs, good data management is of vital importance. All configurations must be stored somewhere, in such a way that they are easily accessible. However, it may occur, that for a given design (i.e., geometry, mass and aerodynamic properties are specified) we want to see the influence of a different RAM-jet engine, or how a change in the shape of a control surface affects the control of the space plane. In this case, we only want to change a small part of the design, and it may be true that a particular engine has already been implemented in a previous design. It would then be convenient if we could extract the related data from that design and incorporate them in the new design.

Due to the modular nature of the complete data set, this can be accomplished in a relatively easy manner. A possible solution is the use of a database, in which all separate data modules are stored. In this way, we can 'compose' a space plane, by simply selecting the related modules. The data set can be divided into, for instance, the following modules:

- mass and geometry (incl. control surfaces)
- aerodynamics
- engines
- guidance and control systems

The database may also include somewhat more 'mathematical' designs like

- guidance and control laws
- mission profiles

It is obvious that this database can become very large, so it is stressed again that good data management is very important. The database should be managed centrally, and good care should be taken to see whether a data module should be stored or not. If, for instance, only one parameter has changed with respect to an already stored data module, it is not really necessary to store the new module as well. However, this module may very well be stored locally (outside the database), so that the user can always retrieve it when necessary. In that case it is not part of the recognized configurations in the database.

There is, of course, a strong dependency between the database on one hand and the TO and FDS on the other. These programs must be able to read the data, so the data format must be specified. Since it is not likely that the format is the same for both programs, a kind of conversion routine should be developed.
At the moment, a space-plane database is not available. We see this database as an essential part of the design model, so for this reason it should be developed. Due to the specific nature of the development of such a database, it is advised that it is going to be developed in cooperation with (a) software engineer(s).

2.3. The Taguchi method and trajectory optimization.

In the (conceptual) design phase of the space plane, a change in a subsystem results in variations of the configuration as a whole due to the high level of system integration. A change in the propulsion system, for instance, can have its effect on not only the mass and inertia of the vehicle, but also on the shape of the fore and aft body. A shape variation influences the aerothermodynamic properties and, as a result, the whole trajectory and the stability of the vehicle flying this trajectory can be different.

In order to compare the performance of several vehicle configurations, we must have a measure which serves as some kind of performance parameter. As we will see later (Table 3.1 in Section 3.3.4), the payload is only some 2% of the total mass of the vehicle. The fuel mass, on the other hand, can amount to 50%. So a small saving on fuel mass could result in a large increase of payload mass. For this reason, it is important that as little fuel as possible is used during the mission. In that way, the payload mass can be maximized, which can then serve as the performance criterion. When we have the payload maximized, the trajectory which has to be flown can be considered to be optimal w.r.t. the payload. One can easily understand that this trajectory will be different for various vehicle configurations. Suppose that in a certain Mach range the aerodynamic properties of a vehicle are lower or higher compared with some basic configuration, then it might save fuel if for that Mach range the flight will be performed at other altitudes.

A minimum fuel-mass trajectory for a given vehicle configuration, however, is depending on a lot of parameters. Examples are the maximum normal load during take-off, the maximum climb angle and axial acceleration, cruise altitude and Mach number, separation Mach number, and the maximum allowable dynamic pressure and heat load. Each of these parameters has its influence on the fuel mass, and trying to find the optimal combination can take an enormous amount of simulations. For instance, when we consider 13 parameters with 3 possible values only (nota bene: this is a discrete variation of parameters), doing a sensitivity analysis including every combination of parameters would take us $13^5$ (over 1.5 million) simulations.

There is, however, a mathematical method called optimization, which can give us the optimal trajectory in terms of fuel mass or payload. But optimization is a complex method, which can take a long time to implement (even if the optimization module is readily available), so we have to ask ourselves first, whether we will save that much fuel (or increase the payload that much), which makes it worth while to invest the time. There is a way of proving the need for optimization, with a method which has become known as the Taguchi method.

In the late fifties, Dr. G. Taguchi published for the first time his Design of Experiments Method, which concentrated on the process of laying out calculations through orthogonal arrays. It lasted
until 1980 that these methods were introduced into the US. Taguchi's major contribution has involved the combining of engineering and statistical methods to achieve rapid improvements in costs and quality by optimizing product design and production processes. He defined the quality-loss function and the signal-to-noise ratio, which tells us where we are in early product development, when we still have time to make improvements at the lowest cost. His use of orthogonal arrays enables a rapid search through millions of design options to find the design with the best signal-to-noise ratio, i.e., the design that is furthest away from all potential problems (Taguchi, 1988).

The basic idea of the design method is, that a number of design parameters is studied at a restricted number of levels only. For a full factorial, i.e., varying one parameter at a time, design of 4 parameters at three levels (low, medium and high values), 81 ($3^4$) experiments are required. Use of orthogonal matrices reduces this number to 9 (Stanley et al., 1992). It is assumed that each parameter is independent of the levels of the other factors. When this is not the case, an interaction is said to exist, which requires additional experiments.

Stanley et al. (1992) apply the Taguchi design method to propulsion system optimization for SSTO vehicles. They perform a series of parametric trade studies - varying lift-off thrust-to-weight ratio, engine mode transition Mach number, mixture ratios, area ratios, and chamber pressure values - to optimize both a dual mixture ratio engine and a single mixture ratio engine of similar design and technology level. Seven parameters were studied at three levels, which normally would have taken $3^7$ (2187) trajectory and vehicle sizing runs. The methodology developed by Stanley et al. required only 50 runs. These runs included the interactions between several parameters.

In his book, Taguchi (1988) gives an example of (sub)optimizing an electrical design by varying 13 parameters at 3 levels, a low, medium and high value. Instead of the $3^{13}$ simulations to do a full sensitivity analysis, he arrived at a total of 36 (!) combinations, which give virtually the same information. Marée et al. (1993) have applied the same combinations of parameters to the optimal-trajectory problem of a 360-ton space plane, to determine the influence of the trajectory parameters on the payload mass$^2$. The outcome of this study was, that the payload mass varied from -1.9 ton to 5.8 ton, clearly indicating the importance of flying a well-defined trajectory.

This variation in payload mass is, of course, essential if we do studies at space-plane level. But even if we concentrate on the inlet, which is currently the focus point of the AEOLUS Project, we can see the need for optimization arise. To illustrate this, we have included Figs. 2.6 and 2.7.

In these Figures, some results of the simulations based on the Taguchi method as executed by Marée et al. (1993) are given. The two trajectories, which are considered, are the minimum-fuel-for-a-maximum-payload trajectory, and the maximum-fuel trajectory (the worst result of the 36 simulations). The difference in fuel mass (space-plane level) is significant. Besides, we see

---

$^2$ This study has been performed within the framework of WP212 of the AEOLUS Project.
that the different trajectories result in variations in the related parameters for the inlet (subsystem level). The minimum-fuel case has a large part of its trajectory a higher dynamic pressure, which means a higher heat load. On the other hand, the duration of the flight is much shorter, so that the integrated heat load is relatively smaller.

![Graph showing altitude vs. velocity](image)

**Fig. 2.6 -** The altitude versus the velocity of the first stage of a TSTO launcher for two trajectory configurations (Maree et al., 1993).

![Graph showing angle of attack vs. time](image)

**Fig. 2.7 -** The angle-of-attack profile of the first stage of a TSTO launcher, for two trajectory configurations (Maree et al., 1993).

The height versus Mach number shows similar differences as the height versus velocity. Maree in AEOLUS WP-2233 (1993), has indicated that the thermodynamic and mechanical loads (given by pressure and temperature) on the inlet vary significantly with the Mach number. It is
therefore essential that an accurate Mach profile is available to define the loads on the inlet.

The angle-of-attack profile shows larger differences. In the first place, the cruise flight for the minimum-fuel case is shorter, and besides the angle of attack is between 2° and 2.5° smaller. Marée (1993) showed that the aerothermodynamic loads in and the performance (mass flow, pressure recovery) of the inlet are influenced by the variation of the angle of attack with time, although the differences were small (for a $\Delta\alpha$ of 1°). In the second place, we notice large differences at the end of both profiles, the pull-up manoeuvre. How these differences will influence the performance of and the loads on the inlet, cannot be discussed without further study. It is necessary that it becomes clear how the loads and performance depend on the related parameters. Questions such as 'What is the design point for the inlet' and 'What margins can be assumed' should be answered.

Due to the large variations in the flight parameters - and the presented profiles are only two of the possible ones for a fixed trajectory shape - we can state that optimization must be included in the trajectory study, also because the Taguchi method does not give the optimal trajectory, because we discretely vary the trajectory parameters. Is it possible, to change one (or more) of the parameters a few percent to win the last bit of payload? As one can easily understand, the answer to this question must be YES. Optimization gives a continuous variation of the relevant parameters, finally arriving at the optimal solution. Besides, we already stated that in order to compare the performance of different configurations we must at least work with optimal trajectories for each configuration. Another advantage of using an optimization method is, that in principle only one simulation is required for each configuration we want to study, whereas with the Taguchi method we need 36 simulations per configuration. Moreover, in the latter case we have to redefine the segments of the trajectory, whereas in the case of optimization the trajectory is shaped by the optimization process.

**Remark**

This Section appears in almost the same form in the report on WP-151 (Mooij et al., 1993).

3.1. Introduction.

The flight-simulation model, which has to be developed in parallel with the space-plane related conceptual design modules, consists of a Trajectory Optimizer (TO) and a Flight-Dynamics Software (FDS). A typical mission, which has to be simulated by the flight-simulation model, is shown in Fig. 3.1. In this Figure, a mission of the Sänger future space transportation system is depicted, including the Horus second-stage separation.

![Flight-simulation model diagram](image)

**Fig. 3.1 - Sänger ascent trajectory with Horus stage separation (Koelle *et al.*, 1990)**

The TO must be able to generate an optimized trajectory, which will serve as input for the FDS. The FDS must in the end give a proof of the steerability and controllability of the space plane. To minimize the development time, we want to make use of existing software as much as possible. On the other hand, duplicating development can be performed to gain knowledge and understanding.

At this moment, we have defined two possible paths, which can be followed. These paths are indicated in Fig. 3.2. The alternative paths arise from the side of the TO. The Advanced Launcher Trajectory Optimization Software (ALTOS) is a software package, developed under ESTEC contract, which is currently not yet available for the SPP Team. However, it is likely that in the near future the software can be used and, at least parts of it, further developed.
S/W #1

PATH A

ALTOS

PATH B

ASCENT

EXTENSION

OPTIMIZER

S/W #2

START V2.0

EXTENSIONS

- PROPULSION
- VAR. MASS PROPERTIES
- GUIDANCE & CONTROL
- EXTENSION UI
- INPUT REF. TRAJECTORY
- VERTICAL LAUNCH
- ORBITAL INTERFACE
- AEROTHERM

3 DOF OPTIMIZED

REF. TRAJECTORY

START V3.0

PROOF OF STEERABILITY AND CONTROLLABILITY

The development of the Flight Simulation Software for the Aerospace Plane.
ALTOS is a sophisticated TO, for which the user has to be quite experienced to fully use all its capabilities and to come to good solutions. To get insight in the optimization problem and to be able to come up with proposals for further development, it is necessary that we are well posted in the subject. Therefore, it was decided to continue developing the ascent trajectory software ASCENT (already available at DUT/FAE). This will also give us the opportunity to have an independent program to compute an optimized ascent trajectory, so that results can be compared.

The current version of ASCENT can only compute (non-optimized) ascent trajectories; at this moment, a study has started to extend ASCENT with optimization routines. As a guideline, we can concentrate ourselves on the Program to Optimize Simulated Trajectories (POST), developed by the US industry for NASA. Although this program cannot be commercially obtained, the program documentation is readily available (Brauer et al., 1974, 1975a, 1975b and 1977).

Another software package, which is currently available at DUT/FAE, is the Simulation Tool for Atmospheric Re-entry Trajectories (START). START Version 2.1 is a full 6 dof re-entry-simulation software, which can serve as a basis to simulate ascent trajectories. Of course, a number of extensions is necessary before guided and controlled ascent trajectories can be simulated. These extensions are mainly related to the implementation of variable mass properties, propulsion, and guidance and control capabilities (Fig. 3.2).

In the next Section, we will discuss the TO. As has been indicated above, two alternative paths can be followed, using two different TOs (ALTOS and ASCENT). Therefore, both optimizers are discussed. An overview of the capabilities is given, as well as possible future extensions.

Section 3.3 describes the 6-dof FDS START, which is going to be used to get an indication of the steerability and controllability of the space plane. An overview of the development of the program, the status of the current version and related extensions to fulfil the objectives of ALTAP is given. The Section continues with some remarks on its implementation as flight-simulator software. Furthermore, the current status of START and the future work will be briefly discussed.

3.2. Trajectory Optimizer.

3.2.1. ALTOS.

The development of ALTOS was part of an ESTEC contract (No. 8046/88/NL/MAC), carried out by MBB (Ottobrunn) and DLR (Oberpfaffenhofen). The development objectives of ALTOS can be summarized as (Deutsche Aerospace, 1992 and Well et al., 1992):

- the development of a comprehensive trajectory optimization tool, for both conventional launchers with vertical take-off and advanced winged launchers with horizontal take-off, which includes
- flexible modelling capabilities,
- state-of-the-art optimization methods, and
- an efficient User Interface (UI).
The main functionality of the UI consists of (Well et al., 1992) i) input file definitions (alphanumeric and graphic) and conversion of these files into a form needed by the optimization and simulation programs, templates of all files need to be available, ii) file management and process control, iii) graphical simulation and optimization result analysis. The basic architecture of ALTOS is shown in Fig. 3.3.

![Diagram](image)

Fig. 3.3 - The ALTOS User Interface and the program task separation (Deutsche Aerospace, 1992).

The trajectory model is based on a point-mass representation of the vehicle, meaning 3-dof equations of motion. These equations must be suitable for both vertical and horizontal take-off, which resulted in the following state variables: $r$ (distance to the Centre of Mass (CoM) of the central body), $\tau$ (planetocentric longitude) and $\delta$ (planetocentric latitude) for the position; cartesian components w.r.t. the local horizontal plane for the velocity; the mass $m$, to take the decreasing fuel mass into account, concludes the set of state variables. The trajectory itself can be represented by a number of phases (see also Fig. 3.4). A motivation for this is the occurrence of discontinuities in models and controls. Examples are discontinuous dynamics (jettisoning of stages, switching of propulsion systems), switch between different kinds of control (pitch control transition from time dependent to constant, vehicle orientation control changed from load factors to aerodynamic angles) and reaching the target orbit.

The environmental model represents the Earth as an oblate spheroid. The gravity field is modelled as an axially symmetric field, up to the second zonal harmonic ($J_2$). An altitude dependent density and sonic velocity (table input) form the atmospheric model.
minimize fuel consumption of upper stage and lower stage for given initial mass

Fig. 3.4 - The branched optimization problem for a TSTO space plane; the subdivision of the trajectory into phases (Deutsche Aerospace, 1992).

Apart from gravitational forces, also aerodynamic and propulsion forces are modelled. The aerodynamic forces are represented by lift and drag as a function of the angle of attack $\alpha$ and the Mach number $M$, for both axisymmetric and non-axisymmetric vehicles. The propulsion forces can come from either rocket or air-breathing engines. Both the thrust and the fuel mass flow are represented by functions of (at the most) four independent parameters: $h$ (height), $M$, $\alpha$ and an engine-control parameter. For interpolation and extrapolation, however, a preprocessing software (PROPIT) is needed, which represents the engine characteristics as (complex) functions of a large number of coefficients.

Beside the environmental and vehicle (mass, aerodynamic and propulsion properties) models, also the trajectory with its path and boundary constraints, and several types of control functions can be modelled. The path constraints can be chosen from a list of 24 types, e.g., loads acting on the vehicle (dynamic pressure, heat flux, etc.), operational constraints for engines and the flight corridor. A list of 40 types is available for the boundary constraints. Typical examples are the target orbit parameters, the impact area of lower stages, fairing jettisoning and the load on the ozone layer. The control functions can be related to the vehicle orientation (Euler angles for conventional launchers, angle of attack/bank angle or vertical/horizontal load factor for winged vehicles) or the engine operation (power setting/equivalence ratio, throttling). Typical cost functions are based on the maximum payload or the minimum initial fuel mass.

Two optimizers have been implemented in ALTOS. The first method (TROPIC) is based on direct collocation. In this method, the control functions and states are parametrized. The corres-
ponding (implicit) integration technique is of an approximation nature. The second method is called PROMIS, and is based on direct multiple shooting. In this case, the equations of motion are explicitly integrated. The Non-Linear Programming (NLP) solver to find the optimal solution is based on Sequential Least Squares Programming (SLSQP). Up to a maximum of 500 parameters can be optimized with the implemented methods. It is beyond the scope of this report to focus deeper on the optimization problem. More details can be found in, for instance, Gottfried et al. (1973), Vinh (1981) and Yiyuan Zhao (1991).

ALTOS was presented at the 12th IFAC Symposium on Aerospace Control (Well et al., 1992 and Lecohier et al., 1992) for the first time. Lecohier presented the modelling, simulation and optimization of conventional launchers (Ariane 5). Buhl (1992) showed, at the 4th National Aerospace Plane Conference, the results of optimizing the branched trajectory of a TSTO space plane. As an example, both test cases will be briefly discussed below.

The objective of the study of Lecohier et al. (1992) was to determine the optimal payload mass that can be launched from Kourou into a target orbit of 70x340 km, 28.5° inclination. Next to an extensive modelling of the launcher, i.e., aerodynamic, mass and propulsion-system data, also the control of the ascent trajectory had to be provided. A total of 5 phases was defined: vertical lift-off, a steering manoeuvre, a constant-pitch phase, a gravity turn or zero incidence flight and finally the jettisoning of the 2 P230 solid boosters. In the fifth phase pitch and yaw had to be optimized. The problem described here, was first optimized using the direct collocation method (TROPIC). The optimized solution was then refined using the PROMIS optimizer. This way of solving the optimization problem was found to be convenient, because TROPIC accepts more crude initial guesses than PROMIS.

The ALTOS software proved to be a powerful tool when applied to conventional launcher problems with numerous complex constraints. Besides, it is useful in case of trajectory validation, of auxiliary calculations and of a priori or a posteriori control strategy evaluation. While using ALTOS, the user is helped by the software thanks to a powerful associated graphical user interface where all the important problem features can be displayed, and thanks to instructive error messages or help menus for better understanding of conflicting situations.

Buhl (1992) describes the TSTO ascent to a LEO transfer orbit, with perigee and apogee altitudes of 70 and 450 km, and an orbit inclination of 28.5°. Launch takes place from a European launch site, i.e., Istres (southern France). The mission comprises 3 branches, i.e., the combined (air-breathing) ascent of the upper and lower stage to the separation point, the (rocket-powered) ascent of the upper stage to the target orbit and the (air-breathing) descent of the lower stage to the landing site. The chosen optimization criterion is the maximization of the payload transported to the target orbit by the upper stage, treating the vehicle data (e.g., structural and fuel masses) as fixed. The branches are optimized separately, as well as simultaneously.

The outcome of the study was, that ALTOS proved to be capable of handling the problem of branched trajectory optimization, comprising a rather detailed modelling of the air-breathing engines. Compared to the suboptimal trajectories based on predefined flight segments like climb and cruise phases, the optimal trajectory revealed some similarities like the occurrence of a pull-up manoeuvre before stage separation. An evident dissimilarity was the absence of any
constant parameter phase. The number of parameters, which can be handled by the NLP solver was fully used, meaning that for a smoother representation of state parameters and controls, an enhanced NLP solver is required.

**Future extensions**

The development of ALTOS is not yet at an end. The extensions, which were mentioned at the final presentation of the current version at ESTEC, are (Deutsche Aerospace, 1992):

- The number of available optimization parameters (= 500) has been fully exploited. To incorporate, for instance, a narrower spacing of the control function grid, an enhanced NLP solver, which can handle more parameters (up to 5000), is necessary.
- The User Input could be 'streamlined', and use could be made of graphical Input/Output instead of textual.
- The equations of motion could be extended to a so-called 'enhanced' 3-dof mode, which means that for the point mass an attitude is defined without including the full 6-dof equations. However, the latter equations could be added for normal flight simulation.
- Apart from ascent-trajectory optimization, also the re-entry phase could be computed and optimized.
- Extensions towards vehicle design optimization.

### 3.2.2. ASCENT.

**Current status (Version 1.0)**

The development of ASCENT Version 1.0 started in March 1992 as a thesis study at the Faculty of Aerospace Engineering, and lasted 9 months. The outcome of the study was a program, with which 3-dof ascent trajectories of both SSTO and TSTO can be simulated (Korswagen 1993a and 1993b). To do this, the trajectory can be divided into a number of flight segments, like take-off, climb, cruising, and the rocket ascent. The state variables on which ASCENT is based are: \( r, \tau \) and \( \delta \) for the position; \( V \) (relative velocity w.r.t. the central body), \( \gamma \) (flight-path angle, i.e., the angle between the velocity vector and the local horizontal plane) and \( \chi \) (heading, i.e., the angle between the projection of the velocity in the local horizontal plane and the local north) for the velocity; \( m \) concludes the set of state variables.

Although the vehicle is represented as a mass point, the attitude is implicitly included in the form of the angle of attack \( \alpha \). This attitude can change during the flight as a result of the used guidance law (pitch motion only). No out-of-plane manoeuvres, e.g., banking, are possible. In case of a TSTO space plane, staging is taken into account as an instantaneous change of mass, aerodynamic and propulsion properties. No interference between the first- and second-stage vehicle is considered.

The layout and UI of ASCENT are based on those of START (see the next Section). For a good
understanding of the capabilities of ASCENT, a short description (based on Korswagen (1993b)) will be given below.

ASCENT can, roughly spoken, be divided into five modules (see Fig. 3.5). The main module is the UI and is called the Main Menu. This module is the main driver of the program from which all activities can be initiated. It also controls the incoming and outgoing data flow.

![Diagram of ASCENT Version 1.0](image)

Fig. 3.5 - The schematic layout of ASCENT Version 1.0.

The input data consist of all data, which are necessary to execute a trajectory simulation. They can be read from a pre-defined data file, and can be modified using a line editor (the second module). The third module is the actual core of the program and is called the simulation module. Using the current values of the input data, this module generates the so-called coded output data (a file, to which all output variables are written with a specified filing rate). These coded output data can then be converted to a formatted output using the fourth module, the conversion module, or can be stored for later use. Formatted, in this case, means that the output data are converted to a specified format such that the result can be read by an external plotting program (ASCENT has no graphical output capabilities). The last module is the initialization module, which resets all variables to their default values.
The menu-oriented editor enables the user to change all values of the input data (apart from the aerodynamic and propulsion data). The input data can be divided into four categories: vehicle data, environmental data, mission data and simulation data.

The first data set consists of the vehicle related parameters, such as the mass and reference geometry, aerodynamic coefficients and parameters concerning the propulsion system, for both the first- and second-stage vehicle. Mass and geometry data include the initial mass (at take-off), the reference area and the reference nose radius. Both aerodynamic and propulsion data can be changed by reading existing data files. These files must meet specific requirements, with respect to the format and sequence of the data. It is (not yet) possible to edit these data from within ASCENT.

The second data set deals with the environment, i.e., the atmosphere and the gravity field. The atmosphere can be modelled in two ways, by selecting either the exponential atmosphere or the US Standard Atmosphere 1976. The gravity field is based on the central field representation, but it is possible to include the $J_2$ term.

The third data set consists of mission related data. These data include the initial state of the vehicle concerning position, velocity, attitude and throttle setting ($\delta_T$). Furthermore, the defined trajectory, the operational constraints, and the engine criteria are contained in this set.

When defining the trajectory, it is possible to select one control law for each flight segment from five available ones. The boundary condition, defining the transition from one flight segment to the next one must be specified. One can select a boundary criterion from a list with flight conditions (flight time ($t$), height ($h$), $V$, Mach number ($M$), dynamic pressure ($q_{dyn}$) and $\gamma$). Besides, it is also possible to specify for each flight segment whether thrust vectoring will be used. The operational constraints are divided into trajectory and steering constraints. The latter limit the values and the rate of change of the steering variables. The trajectory constraints limit the trajectory flight conditions. So far, only $q_{dyn}$ and $n_a$ (axial acceleration) have been implemented as trajectory constraints. The engine criteria specify the flight condition at which an engine will be switched on or off. The criteria can be chosen from the same list as the boundary conditions, except for $\gamma$. Besides, the flight-segment number can be chosen.

The third module is the simulation module, containing a loop, which is continuously executed, until a specified stop criterion is met. During the execution of the loop, the boundary, engine and stop criteria are checked. Meeting a criterion will either initiate a new flight segment, change the engine status or terminate the simulation run. In the latter case, control is returned to the UI. When no stop criterion is met, the program continues to generate the steering variables, i.e., $\alpha$, $\delta_T$ and $\varepsilon_T$ (the thrust vector angle). $\alpha$ is generated with the use of the assigned control law for the current flight segment. Thrust vectoring implies here, that $\varepsilon_T$ is set equal to $\alpha$. $\delta_T$ is an outcome of checking the trajectory constraints. After generating the steering variables, the equations of motion are integrated, resulting in a new state vector for the vehicle. Using these new values, the loop starts all over again with the criteria check.

The fourth module, the conversion module, converts the coded output to formatted output. The coded output can be any saved output-data file. The formatted output consists of 19 files, in
which related variables, like lift and drag, are stored together. Nota bene: the current version of the conversion module is based on input requirements (and limitations) of the plotting program Slidewrite 3.0. Other plotting programs may need different formats.

The last module is the initialization module. Activating this module from the Main Menu resets all variables to their default value. For more information on these default settings, one is referred to the User Manual of ASCENT Version 1.0 (Korswagen, 1993b).

**Improvement of ASCENT Version 1.0**

At this moment, two extensions have been added to ASCENT Version 1.0. These relate to the cruise flight of a space plane and the circulation manoeuvre. It is now possible to define a cruise-flight segment (flight with a constant speed, with a flight-path angle of $\gamma = 0^\circ$). The circulation manoeuvre has been introduced to arrive at the exact final conditions, without having the problems of flying with $\gamma = 0^\circ$ and covering the last few meters (which takes a long time and a disproportional amount of fuel). This problem results from the simplicity of the implemented guidance law and has been solved as follows. When the space plane arrives at a certain distance from the final altitude (depending on the value of $\gamma$), the magnitude and direction of the velocity are impulsively adjusted to the current altitude, under the condition that the velocity is the circular velocity for that altitude and $\gamma = 0^\circ$. The fuel, that is needed for this manoeuvre, is then subtracted from the available fuel mass and serves as a penalty.

**Future developments**

A next step in the development of ASCENT is extending the software with an optimizer, to compute optimized ascent trajectories. This is part of a second thesis study, which is currently being performed. The optimization routines will be obtained from the IMSL library, in a similar manner as has been described by Blaauw (1992). These optimization routines will be based on the collocation method. Final goal of the implementation will be the generation of an optimized ascent trajectory in such a way, that this trajectory can serve as input for a flight dynamics software with guidance and control (G&C) capabilities (START Version 3.0). Besides, the model should be able to interact with the DUT/FAE conceptual design model.

Examples of trajectory optimization are described by numerous authors:

- General: the rocket trajectory optimization problem can be found in Lawden (1991), Jänisch et al. (1990), Kumar et al. (1987) and Well et al. (1982); Strohmaier et al. (1990), Jänisch et al. (1990 and 1991), Lemanie (1992) and ACRI (1992a-b) show optimal re-entry trajectories; descriptions of the non-linear programming and collocation technique in trajectory optimization can be found in Betts et al. (1984), Bauer et al. (1984), McLain et al. (1990) and Nguyen (1990); Venugopal et al. (1991), finally, discuss a generalized algorithm based on energy management techniques and calculus of variations for determining optimal trajectories of hypersonic vehicles.
- SSTO: Noton et al. (1989), Martel (1990), Nguyen (1991), Wagner et al. (1991) and Lu
(1991) describe the optimal ascent of HOTOL, STS-2000 and general SSTO configurations.

- TSTO: Bulirsch et al. (1991 and 1992) and Sachs et al. (1990) treat the optimal trajectories of the Sänger space plane; Christophe (1992) and Nguyen (1991) describe a general configuration; second-stage trajectories are found in Staufenbiel (1990).

The output of the study will include

- implementation of the optimization method in ASCENT,
- evaluation of the new software, preferably a comparison with the output of another program (ALTOS), but at least a comparison with a number of mission analyses found in literature,
- an optimization analysis of a TSTO aerospace plane, taking several optimizing parameters into account (e.g., propellant mass or payload mass), and
- an output file with an optimized reference trajectory.

3.3. Flight-dynamics software.

The flight-dynamics software, to be used in parallel with the trajectory-optimization software, is based on the re-entry software START. The development of START began as a part of a thesis study in January 1991. This study has been done at ESTEC, covering a period of 10 months. The result was START Version 1.0, which will be briefly discussed in the next Section. START Version 2.0 was the outcome of a seven months part-time activity for the Huygens Project Team at ESTEC, and started in January 1992; a description will be given in Section 3.3.2. The version of the software, which is currently available, is Version 2.1. It is discussed in Section 3.3.4. To fit within the ALTAP, several extensions are necessary. These extensions, ultimately leading to START Version 3.0, will be introduced in the Section 3.3.5. In the last Section, finally, some remarks will be given on possible use of the START software w.r.t. a flight simulator.

3.3.1. START Version 1.0.

The first version of START can be defined to be a 6-dof open-loop re-entry trajectory simulation tool. Open-loop means here, that there are no guidance and control capabilities. The equations of motion, which can be divided into equations of translational motion (position and velocity) and equations of rotational motion (attitude and angular motion), are based on the following state variables: $r$, $\tau$ and $\delta$ for the position; $V$, $\gamma$ and $\chi$ for the velocity; $p$, $q$ and $r$ (the rotational rate of the body w.r.t. to the inertial planetocentric frame, i.e., the roll, pitch and yaw rate) for the angular motion; and the aerodynamic angles $\alpha$ (angle of attack), $\beta$ (angle of sideslip) and $\sigma$ (bank angle) for the attitude of the re-entry vehicle w.r.t. the airflow.

As is indicated in Fig. 3.6, the actual flight dynamics code, the core of the program, is embedded in a UI. It provides the user with a friendly tool to edit all the input data necessary for the trajectory analysis and to start the simulation itself. In other words, all possible actions can only be activated by means of this UI.
Fig. 3.6 - Schematic layout of START Version 1.0.
The input data can be divided into four blocks, in Fig. 3.6 indicated with trapezoids. The first block is related with the re-entry vehicle. A vehicle can be described as a number of mass elements, each with its own mass, CoM and inertia tensor. This way of entering the vehicle enables a user to 'build' a re-entry vehicle on basis of fundamental geometrical shapes with readily available inertia tensors. The global inertia tensor will be computed during the simulation. Besides, this concept of mass elements can also be used for configuration changes by just deleting one (or more) of the mass elements.

A major part of the vehicle data consists of the aerodynamic database. Each of the (six) force and moment coefficients can be written as a Taylor series, as a function of a number of independent variables, for instance:

\[ CF = CF_0 + CF_{X_1}X_1 + CF_{X_2}X_2 + \ldots + CF_{X_1,X_2}X_1X_2 + \ldots \]

The variables \( X_i \) are called derivation variables (e.g., \( \alpha, \beta \), one of the three angular rates); the order of derivation can be 0, 1 or 2. Besides, each of the coefficients \( CF_0 \) can be entered as a table (max. 10x10), as a function of 0, 1 or 2 table variables (e.g., \( \alpha, \beta, q_{dyn}, M \)). This way of entering aerodynamic data is very flexible. And, by storing the data in a separate file, the aerodynamic data can be updated after each configuration change, by simply reading in an aerodynamic database file. The aerodynamic forces and moments consist of the drag, side and lift forces \( D, S, \) and \( L \), and the rolling, pitching and yawing moments \( L, M \) and \( N \), respectively.

As a parachute model, a two-dimensional, 3-dof model has been implemented (the third state variable, in addition to the position \( (x,y) \), is the angle between the parachute body axis and the local vertical). The model may also include reefing (i.e., step-wise opening) of the parachute.

The second block deals with the environment. In the first place, a central body can be selected. This can either be the Earth, the Moon, Mars or Titan. Depending on the selection of the central body is the choice of the gravity model (central field plus harmonics up to \( J_2 \)) and that of the atmospheric model. For the Earth, there are three available models: US Standard atmosphere 1962 (US62), US76 and a simple tabulated model based on US76 (up to 90 km) and CIRA86 (above 90 km). The Martian atmosphere is also tabulated and based on Viking-1 data. For the Titan atmosphere, the tabulated nominal Lellouch-Hunten model of October 1987 is used.

The mission block enables the user to define the mission of the re-entry vehicle. Data correspond to the initial state of the vehicle, parachute deployment, reefing and release and (other) configuration changes.

The fourth block, simulation, contains data for executing the actual simulation, i.e., the choice of the integrator (fourth-order fixed-step Runge-Kutta, seventh-order variable-step Runge-Kutta-Fehlberg, variable-order variable-step Adams-Bashforth, Adams-Moulton), the maximum integration step size, the output filing rate for both the quicklook data (information for user during the simulation) and the plot data, the tolerances of the state variables (used by the variable step size integrators) and the stop criteria (e.g., \( t, h, M \)).

Input data can be stored in a data file and retrieved when necessary.
The actual simulation can be performed in two modes, namely 3 and 6 dof. In the first mode, the re-entry vehicle is considered to be a mass point and only translations can be analyzed (little CPU-time is used). In the second mode, the vehicle is thought of as a rigid body and both translations and rotations can be examined.

At each time step, the equations of motion are integrated to give position and speed (3 dof) and attitude and angular velocities (6 dof). Mission events can be executed whenever a predefined flight condition occurs. Computations proceed until a stop criterium is met. In that case, control is returned to the UI.

The development of START Version 1.0 has been extensively described by Mooij (1991a and 1991c). The program was used to do a first mission analysis of the ESA Huygens Probe (Mooij, 1991b); it proved to be very useful.

3.3.2. START Version 2.0.

In 1992, a follow-on activity was defined to do an extended entry and descent analysis of the ESA Huygens Probe (Mooij, 1992). To meet with the adjusted mission scenario and based on the experiences of the first mission analysis (Mooij, 1991b), it became evident that START had to be improved and extended. Summarized, the improvements and extensions are the following:

1) General
   • Speeding up of S/W
   • Batch mode
   • Additional integration methods (Adams-Bashforth)

2) Entry model
   • Quaternions

3) Vehicle model
   • Spin vanes

4) Environment
   • Wind model
     - Adjustment of the equations of motion
     - Extension of UI
     - Steady state wind (including the Flasar model for Titan)
     - Horizontal wind gust
   • Atmosphere model
     - Minimum and maximum Lellouch/Hunten 1987
     - Dynamic viscosity computation

5) 3D (6 dof) parachute model

The above extensions will now be discussed in more detail. A complete description of all the
newly developed models can be found in (Mooij, 1992).

General

The first version of START appeared to be very slow, compared with other re-entry simulation packages. Measurements taken for decreasing the CPU-time usage were a general reprogramming of the software and an improvement of the integration algorithm. As a result, the performance of START has been significantly improved.

In addition, the UI has been adjusted at several places to increase the ease of use.

As may have become clear from previous discussions, START is a menu-oriented program. This implies that to start a simulation, several commands have to be given before entering the corresponding Simulation Menu. Besides, the names of the two output files have to be specified. To include a simulation in a batch job is therefore not very convenient. However, a full mission simulation creates a heavy computational burden for the computer system. If one is not a single user, this can become a substantial problem, so operating in batch mode is preferable. For this reason, a so-called batch mode has been added to the program. By specifying a batch-input file with the names of the input and output files, and the simulation mode (3 or 6 dof), the UI is overruled and the simulation is executed with minimum output to the screen. The batch mode is also very convenient when a complete sensitivity with multiple runs has to be executed. In that case, the user can suffice with generating the input and batch files for each of the runs, and combine all the executions of START in a batch job.

Within the framework of an on-going study on the numerical aspects of entry flight simulation, three integration methods have been added to the already existing ones. These integrators are the second-, third- and fourth-order fixed-step Adams-Bashforth methods.

Finally, supporting software has been developed to keep all common blocks, in which all input data and a major part of the auxiliary variables are stored, up to date when the START software is extended or changed.

Entry model

In START Version 1.0, the attitude of the re-entry vehicle w.r.t. the on-coming flow was defined by the angle of attack, the angle of sideslip and the bank angle. However, the corresponding equations of motion are very complex and intuitively one might think that integrating these

---

3 Because of the large number of subroutines in which common blocks are used, it has become rather complex to change one of the common blocks throughout the program package. To prevent any errors, introduced by possible incomplete updates of the common blocks, a method has been developed to do this automatically. One file has been created comprising all the common blocks. Besides, a computer program has been made that scans the START source code and inserts the common-block information from this file at the appropriate place. In this way, common blocks have to be changed at one place only, thus eliminating any errors related to incompatible common-block information.
equations creates a heavy computational load. An alternative for the description of the attitude is using quaternions as variables. The corresponding equations of motion, including all related transformations between quaternions and \( \alpha, \beta \) and \( \sigma \), have been implemented in START as an alternative, so that results based on the two different models can now be checked against each other. It appeared, that the differences in results and CPU-time usage were small, the latter mainly the result of the large number of frame transformations involved for obtaining interpretable results.

**Vehicle**

To guarantee a nominal spin rate, the current design of the Huygens descent module is equipped with so-called spin vanes. These spin vanes cause an additional external rolling moment. A simplified model has been developed to get an indication of the spin-rate behaviour. The basic idea of the spin-vane model is as follows.

Each vane is considered to be a rectangular flat plate. The lift and drag contribution for one vane is computed, based on flat-plate theory. These forces result in a rolling moment that drives the descent module. The contribution of all the vanes is taken into account by multiplying the computed rolling moment by the number of vanes. This implies, that the flow direction for each vane is assumed to be identical, so, for instance, no transverse winds can be studied.

Limitations of the developed model are the following:

- flow disturbances due to interactions between the descent-module wall and the vanes are not considered,
- partial visibility of a vane due to shielding of the descent module and a deviation in flow direction is not taken into account,
- the flow direction for each vane is different, unless we consider a vertical descent with no wind. For realistic computations, wind should be taken into account and therefore, the forces acting on each vane should be computed separately.

**Environment**

The extension of the environment can be divided into two parts. The first part is related to the implementation of wind, and the second part to the extension of the atmosphere models.

The implementation of wind contained two aspects. In the first place, the equations of motion and computation of aerodynamic forces had to be modified to deal with the effect of wind. This derivation of the wind equations is extensively described in (Moolj, TO BE PUBLISHED). In the second place, the UI of START had to be adjusted to enter steady-state wind and horizontal
wind gusts\(^4\).

The steady-state wind can be defined as having two components, either a zonal and a meridional component, or a modulus and direction component. The components can be entered in tabular form, being a function of two independent variables at the most (i.e., the atmospheric pressure, the height and the latitude). It is also possible to define a zonal component only, being a (simple) function of the planetary rotation. In addition, the so-called Flasar wind model has been implemented for Titan. This is an engineering model, defining a zonal steady-state wind as a function of altitude (atmospheric pressure) and latitude.

A total of 10 horizontal wind gusts can be specified. Main parameters are the initial altitude, the thickness, the maximum velocity and the direction of each gust. Two shapes have been predefined.

The extension of the atmosphere models mainly restricts to Titan. The nominal Lellouch-Hunten atmosphere model, which has been implemented in START, is based on tabulated data of atmospheric density and temperature as a function of altitude. To interpolate in the tables for a given altitude, cubic spline interpolation is used. This interpolation method requires, apart from the function values, also the corresponding second derivatives. Press et al. (1989) give an algorithm to compute these second derivatives. This algorithm has been used to develop the tabulated models for the so-called minimum and maximum atmosphere. The minimum, nominal and maximum atmosphere define the complete Lellouch-Hunten model. Besides, an additional computer program has been developed, which reads the function values and corresponding second derivatives from data files, and generates FORTRAN source code containing the atmosphere tables. To conclude, the atmosphere models are Titan related but the algorithm to generate tabulated atmospheres is commonly applicable for any atmosphere.

Apart from the above extensions, also some small adjustments have been made for the speed-of-sound and the dynamic-viscosity computation for each of the three Titan atmospheres.

**Parachute model**

In the (2D, 3 dof) parachute model initially implemented in START, parachute and payload were represented by point-masses, connected by a rigid bar. Only the 2D motion over a flat surface (in a fixed plane) and the pendulum type of motion could be studied. For the Huygens mission, the spin-rate profile is an important mission parameter, so the parachute model must be capable of simulating 3D rotations. Besides, when a parachute descent with strong winds is studied, resulting in a large trajectory drift, the 2D-plane cannot be considered to be fixed in space any more. Therefore, 3D-motion must be simulated. A 6 dof parachute model is capable of studying the effects mentioned above.

---

\(^4\) For the mission analysis of Huygens, it was sufficient to restrict the wind model to a steady state wind and horizontal wind gusts. However, the layout of the User Interface and the simulation algorithm is already such that future extensions of the wind model, e.g. vertical wind gusts, wind shear and turbulence can easily be implemented.
The following characteristics underlie the model. The parachute/payload system is considered to be one single body. The equations of motion describe the translation of the global Centre of Mass (CoM) of the system and the rotation around this CoM. The equations of translational motion are based on cartesian coordinates, to overcome the singularity for spherical coordinates when a vertical motion is studied. To describe the attitude of the system, quaternions are used. The effect of wind is taken into account.

Both the parachute and the re-entry vehicle are thought to be rigid bodies (no mass points), connected by a rigid bar. This rigid-bar connection prevents the two bodies to rotate w.r.t. each other. However, the two bodies can spin w.r.t. each other, along this hypothetical bar, because a (non-ideal) swivel has been implemented in the connection between parachute and re-entry vehicle. Non-ideal means here, that the swivel causes a friction moment that counteracts the spinning motion.

The air in and under the parachute is taken into account as added mass. The shape of the canopy is defined to be a hemisphere.

To minimize the CPU-time usage, the parachute/payload system is considered to be rotational symmetric. In that case, the inverse of the inertia tensor does not have to be computed every time step, since the Euler equations for a rotational symmetric body are directly available in an analytical form. Nota bene: the inertia properties of the system change continuously because of the added mass, which is dependent on the atmospheric density and therefore on height.

### 3.3.3. START Version 2.1.

Within the scope of the transfer of START to ESTEC, START had to be extended with a capability of simulating a (n impulsive) deorbit-burn manoeuvre. To demonstrate the updated version of START, a sensitivity analysis had to be performed of the deorbit-burn manoeuvre of the Assured Crew Return Vehicle (ACRV), a so-called rescue or return vehicle for the International Space Station 'Freedom' (Mooij, 1993b).

The extensions can be divided into three categories:

- initial conditions; an alternative way of entering the initial conditions, and the transformation between the related parameters has been developed,
- deorbit-burn manoeuvre; both an impulsive and a thrusted deorbit-burn model have been developed,
- user-interface extensions; the UI had to be extended to handle the new capabilities.

They will be briefly discussed below.

*Initial conditions*

In orbital mechanics, the position and velocity of a satellite or a spacecraft orbiting the Earth,
is usually given by either orbital or cartesian components. Both ways of defining the state of the
vehicle are confined to inertial space, which means that position and velocity are expressed with
respect to the inertial planetocentric frame. The equations of motion, which are implemented
in START, define the state of the vehicle with respect to the rotating planetocentric frame;
besides, the position and velocity are expressed with spherical parameters. The related transfor-
mations from cartesian and orbital to spherical parameters have been derived and implemented.

Deorbit-burn model

Two models for the deorbit-burn manoeuvre have been derived, i.e., an impulsive and a
thrusted one.

- The impulsive $\Delta V$-manoeuvre. The idea behind the impulsive $\Delta V$-manoeuvre is, that the
  (inertial) $\Delta V$-vector is vectorially added to the initial velocity of the spacecraft. $\Delta V$ is
given as a modulus and two direction angles, relative to the inertial velocity of the
vehicle.

- The $\Delta V$ manoeuvre with finite thrust. Also in this case, the inertial $\Delta V$-vector is given
as a modulus and two (relative) direction angles. The $\Delta V$ is now achieved by a thrusted
manoeuvre. The direction of the thrust vector $T$ is computed on basis of the (fixed)
orientation in inertial space. So when the spacecraft is orbiting the central body, the
thrust direction is constantly adjusted, because the orientation of the body frame (in
which the thrust is defined) is continuously changing w.r.t. the inertial frame. As a stop
criterion for thrusting, the achieved (total) $\Delta V$ in thrust direction is taken. This implies,
that we assume a drag-free deorbit-burn manoeuvre, i.e., an exo-atmospheric
manoeuvre.

User-Interface extensions

The UI extensions are related to two of the four input-data blocks, i.e., the vehicle and the
mission block. Part of the vehicle data is the propulsion system; for the deorbit-burn engine the
constant specific impulse and thrust in vacuum can be edited. W.r.t. the mission data, the entry
of the initial state has been extended with the possibility to edit cartesian position and velocity,
and to enter the classical orbital elements. Finally, propulsion operations have been added to
the menu structure: the deorbit-burn manoeuvre can be selected, and a choice can be made
out of three models, i.e., the impulsive $\Delta V$, and two finite-thrust manoeuvres ($m_f$ computed
according to Tsiolkovsky’s Equation or $m_f$ specified by the user). Due to the large number of
extensions since START Version 1.0, a new User Manual has been written (Mooij, 1993a).

---

5 The proposed, simple guidance law is just one possibility out of many. The goal of the study was not to
develop efficient guidance algorithms, so that is why we used a very simple scheme. An even simpler guidance
law would have been a constant thrust direction, based on the initial orientation of the $\Delta V$-vector.
3.3.4. START Version 3.0.

The previous versions of START are only suitable for doing entry and descent analyses. Only the uncontrolled motion can be studied, meaning that there are no guidance and control capabilities. To use the software for conventional and advanced launchers, several substantial extensions are necessary. A typical mission of the Sänger system as to be simulated by START is given in Fig. 3.1; in Fig. 3.7 the Horus mission operations are shown.

![Mission Diagram](image)

**Mission A**

**Mission B**

![Trajectory Profile](image)

**Fig. 3.7** - The Horus mission operations (top) and corresponding trajectory profile (Koelle et al., 1990).

We can conclude from this Figure, that a complete mission of Sänger includes more than the air-breathing ascent of the two stages, the separation of Horus, the return flight of the first stage and the rocket-powered ascent of the second stage to orbit. Mission aspects of interest are, for
example, the orbital phase (transfer orbits, phasing orbits, circulation manoeuvre, the deorbit burn) and the re-entry phase.

The extensions, needed to perform a mission analysis as described above, are listed below (see also Fig. 3.2):

- propulsion, including reaction control
- variable mass properties
- guidance and control
- input of a reference trajectory
- vertical launch
- orbital interface
- aeroheating
- extension of the UI

Each of the above items will now be discussed in more detail.

**Propulsion, including reaction control**

The reason for implementing propulsion (forces and moments) in the equations of motion, is twofold. In the first place, looking at re-entry only, until now the re-entry vehicle is going down to the planetary surface in free fall, i.e., uncontrolled. When we want to exert influence on the landing place, an Attitude Control System (ACS) is essential. This system generates torques, which change the attitude of the vehicle and therefore the direction (and possibly also the magnitude) of the aerodynamic force. As a result, the re-entry vehicle will deviate from its free-fall trajectory and both the crossrange and downrange are influenced by this.

The ACS can be aerodynamics- or propulsion-based, and it is usually a combination of both (it may even be necessary to include CoM-control for efficient control during the ascent flight). At higher altitudes, where the atmospheric density is practically zero, there are no aerodynamic forces and moments of importance. In that case, a propulsion system is mandatory. Besides, if we look at an Apollo-type of re-entry vehicle, there are no aerodynamic control surfaces, and the ACS is based on a hydrazine propulsion system.

Restricting to attitude control, only the equations of rotational motion are affected. The Euler equations, giving the relation between the angular accelerations and the external moments, have to be adjusted in the sense that propulsion moments are taken into account. The propulsion forces are usually small compared with the aerodynamic and gravitational forces acting on the main body, and do not influence the trajectory significantly.

In the second place, concentrating at the ascent of any type of launcher, the driving force for gaining altitude and velocity is of propulsive origin. Furthermore, Thrust-Vector Control (TVC)

---

6 We restrict to propulsion torques here. For the implementation of a complete guidance and control system, also aerodynamic control torques have to be considered. This will come up for discussion later in this Section.
of the main propulsion system can be used to exert influence on the ascent trajectory (part of the guidance system of the vehicle). Besides, TVC can be used to change the attitude of the launcher, thus being a part of the ACS.

We already discussed that in case of an ACS, the equations of rotational motion have to be adjusted. However, for simulating ascent trajectories also the equations of translational motion have to be extended, notably with propulsion forces. A full derivation of the (six-degrees-of-freedom) equations of motion, including propulsion forces and moments, has already been made (Mooij, TO BE PUBLISHED). These equations allow for ascent simulation, as well as for TVC and attitude control. Nota bene: within the frame work of the deorbit-burn analysis, as mentioned in the previous Section, the equations of translational motion were extended with thrust forces. However, variable propulsion properties (thrust and specific impulse), have not been implemented yet.

Once propulsion forces and moments have been implemented, we can include these in the ACS, next to the aerodynamic control surfaces, and study different control strategies. In this way, a(n) (sub-)optimal control scheme can be defined. In this light, one can think of doing a kind of sensitivity analysis using different values for maximum deflection and thrust-vector angles and study the influence on flight time, used propellant mass, resulting trim drag, etc. The Taguchi method, as introduced in Section 2.3, could very well be used.

Besides, it should be possible that a combination of LH2 and/or LCH4 as fuel is used. These two fuels have different densities, so we might save on tank volume if we can get a similar thrust with less fuel consumption. This means that the overall size of the space plane can be smaller, thus decreasing the drag area. W.r.t. a combination of two fuels, we can think of either sequential burning, i.e., the (same) engine switches from LH2 to LCH4, or parallel burning. Parallel burning can either be two engines using different fuels (compare the Space Shuttle: the main engines are fed from the external tank with liquid propellant, while the rocket boosters use solid propellant), or one engine using dual fuels. Martin (1985) discusses the effects of tripropellant engines on Earth-to-orbit vehicles. For SSTO vehicles, the decrease of dry mass resulting from using these kind of engines is significant (up to 19% for the studied configuration). For TSTO vehicles, the effect is smaller, but can still amount some 9%. These results mean, that our flight-simulation tool must have the capability to study the use of tripropellant engines as well.

---

7 Tripropellant engines are, in fact, dual-fuel engines. The third propellant is in that case the oxidizer. Tripropellant engines use, for instance, hydrocarbon fuel (LCH4) with oxygen (LO2) for most of the thrust produced. Hydrogen (LH2) is used as an auxiliary fluid, because it has excellent characteristics as a coolant and as a drive gas to provide pump power. After providing pump power, the hydrogen can also provide some thrust (Martin, 1985).
Variable mass properties

The implementation of variable mass properties is related to having a propulsion system. Although the fuel consumption of an ACS does not significantly influence the total mass of the vehicle, this cannot be said when the propulsion system has to provide the accelerating force during the ascent to orbit. A conventional launcher carries all its fuel in on-board tanks. A typical launch of a 7000 kg payload into a 7°-inclination LEO by Ariane 44L consumes 428,000 kg of fuel, which is 90% of the total mass (Wilson, 1991). An advanced launcher, such as the German Sänger (TSTO), has an air-breathing first stage and a rocket-propelled second stage. In the air-breathing phase, the fuel consumption is relatively low (Table 3.1). However, to reach Low Earth Orbit the second stage is separated a Mach number of $M = 6.5$, corresponding with an altitude of about 28 km (Staufenbiel, 1990). The second stage has also all its fuel on-board, so the decrease in mass upon reaching the final orbit is considerable.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description of mass</th>
<th>Mass $10^3$ kg</th>
<th>Mass %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Sänger</td>
<td>Total launch mass</td>
<td>366.0</td>
<td>100.0</td>
</tr>
<tr>
<td>First stage</td>
<td>Total mass</td>
<td>254.0</td>
<td>100.0</td>
</tr>
<tr>
<td>(EHTV)</td>
<td>Nominal net mass</td>
<td>156.0</td>
<td>61.4</td>
</tr>
<tr>
<td></td>
<td>Maximum propellant mass (LH₂)</td>
<td>98.0</td>
<td>38.6</td>
</tr>
<tr>
<td>Second stage</td>
<td>Total mass</td>
<td>112.0</td>
<td>100.0</td>
</tr>
<tr>
<td>HORUS-M</td>
<td>Nominal net mass</td>
<td>28.1</td>
<td>25.1</td>
</tr>
<tr>
<td></td>
<td>Maximum propellant mass (LOX/LH₂)</td>
<td>80.9</td>
<td>72.2</td>
</tr>
<tr>
<td></td>
<td>Payload (28°-Space Station orbit)</td>
<td>3.0</td>
<td>2.7</td>
</tr>
<tr>
<td>Second stage</td>
<td>Total mass</td>
<td>112.0</td>
<td>100.0</td>
</tr>
<tr>
<td>HORUS-C</td>
<td>Nominal net mass</td>
<td>24.9</td>
<td>22.2</td>
</tr>
<tr>
<td></td>
<td>Maximum propellant mass (LOX/LH₂)</td>
<td>79.4</td>
<td>70.9</td>
</tr>
<tr>
<td></td>
<td>Payload (LEO, 200-km orbit)</td>
<td>7.7</td>
<td>6.9</td>
</tr>
</tbody>
</table>

Table 3.1 - Preliminary Sänger mass summary (based on Koelle et al., 1990)

The change in fuel mass is too large to be ignored. Therefore, this change of mass must be incorporated in START. However, there is more to it than just a change in total mass. Also the location of the CoM is influenced, as well as the inertia properties. The total mass can directly be computed from the fuel mass flow. The location of the CoM and the moments and products of inertia, on the other hand, are directly dependent on the location of the fuel tanks, and the outgoing mass flow per tank. In principle, solving this problem can be very complicated and may

---

8 In principle, we should also include CoM control (as part of the ACS). Since this variation of the mass properties is of different origin, it will not be discussed here.
also not be suitable for our purposes.

A trajectory simulation is computationally intensive, and adding computations of the location of the CoM, the inertia tensor and its inverse (needed in the Euler equations, when we consider an asymmetrical body) create an additional burden and must be avoided. One way of solving this problem is, to compute the mass properties off-line, i.e., separated from the trajectory simulation. In this case, a mass and geometry model of the launcher must be assumed, so that for a discrete number of fuel masses the location of the CoM and the inertia tensor can be calculated. The results are then stored in tabular form (as a function of fuel mass, or, as an alternative, the total mass) and form an integral part of the input data. During the trajectory simulation, the actual mass properties can be determined using, for instance, linear interpolation.

The fill ratio of fuel tanks is usually about 90%. Besides, operating the engines means that fuel is being used, so we have to deal with partially-filled fuel tanks. In that case, the attitude of the launcher influences the orientation of the fuel surface, because the fuel level will always be perpendicular to the resulting acceleration. As an (extreme) example, we consider a single, partially filled, axisymmetrical fuel tank with a vertical and a horizontal attitude (Fig. 3.8). Going from one attitude to the other, we see that the CoM has moved to a different location and one can imagine that also the inertia tensor has changed, especially if we realize that the total fuel mass is high. Besides, at first the tank was symmetrical about the Υ-axis, which is not the case for the horizontal attitude. This results in products of inertia unequal to zero. This means that the attitude has to be added as an independent parameter in some form.

A model, which can compute the mass properties of launchers, may have the following properties. The user must be able to select the shape of the tanks, based on a number of predefined shapes, e.g., a sphere, a cylinder, or a cylinder with hemispheres at the top and bottom. Depending on the selected shape, some reference dimensions must be given (radius, height, etc.). Furthermore, the number of tanks and their respective locations w.r.t. a (body-fixed) reference frame must be specified. For a low level of sophistication, we can assume that all tanks are emptying at the same rate. This means, that for each total fuel mass, the fill ratio of the tanks is known. If we now define the boundary values of the attitude angles, i.e., the orientation of the reference frame w.r.t. an initial orientation, then the location of the CoM and the elements of the inertia tensor can be computed for a discrete number of attitudes, taking the total mass (and also the fill ratio) and the mass properties of the standard shapes into account.
Fig. 3.8 - The location of the CoM is depending on the orientation of the fuel tank w.r.t. the resulting acceleration. Also the elements of the inertia tensor are influenced by an attitude difference.

Guidance and control

Once an optimal trajectory for a given design of a space plane is known, we want to make sure whether the space plane can be guided along that trajectory in the presence of changing atmospheric conditions. During the optimization process, particular rates of change of attitude variables have been assumed, naturally based on control system specifications. Flying the actual trajectory will increase insight in the controllability of the space plane and whether a change of, for instance, the angle of attack of 0.4 °/s is feasible. Of course, the stability and control must be guaranteed over the entire trajectory.

A proper propulsion-system performance depends, amongst others, on the angle of attack. This shows a need for an accurate angle-of-attack control. Analyses to evaluate the effects of gusts, wind shear and turbulence aloft on control system effectiveness are therefore required, to see whether the system can cope with this kind of disturbances. Fig. 3.9 indicates that a more advanced performance sensitive control system may have to be considered.

Mclver et al. (1990) report a NASA study on a conical hypersonic vehicle. Stability and control analyses for that configuration have shown that at higher Mach numbers, aerodynamic control effectiveness is sharply reduced. In addition, the aerodynamic centre moved forward of the centre of gravity creating a pitching moment, which when offset by the elevons, introduces significant trim drag. Hence, centre of gravity management and thrust vectoring concepts have to
be considered. Also Powell et al. (1990) arrive at similar conclusions. They state that the use of active centre-of-gravity control or thrust vectoring should decrease the amount of propellant required.

![Gusts, Shear, Turbulence](image)

Fig. 3.9 - Atmospheric effects on control system performance (Mclver et al., 1990).

The conclusion, which can be drawn from the above discussion, is that the control system for hypersonic vehicles like a space plane will have to accommodate a wide range of functions and will be very complex because of the high level of system integration (Fig. 3.10). The system will be responsible for flight, thermal, propulsion and structural controls. The control laws will be complex, interactive, and be able to provide restructuring in case of subsystem failures (Mclver et al., 1990).

Basically, the development of the guidance and control system exists of two parts. The first part deals with the guidance of the space plane, i.e., the controlled motion of the CoM. To be more precise, guidance can be used to compute a new trajectory if deviations occur (so-called re-optimization) or to restore the original path after deviations from this nominal path (perturbation guidance). To gain insight, an inventory of applicable guidance strategies should be made. Within the framework of the SPP, Marée (1992) has made such an inventory. Besides, the application of classical control theory on space-plane guidance was studied. A number of additional treatments on this subject were found in literature: Tylee et al. (1988) and Hardtla et al. (1987) give an overview of the guidance requirements for future launch vehicles; Cochran et al. (1991) give analytical solutions to the guidance problem; McEneaney (1991) presents three guidance algorithms for aeroassisted orbital transfer and aerocapture, using Loh's term approximations; Corban et al. (1989a-b and 1991) describe fast, near-optimal guidance schemes, and Schultz et al. (1991) present a computationally efficient, real-time trajectory optimization and guidance approach for hypersonic vehicles; the guidance of conventional launchers is discussed by White (1992) and Martin et al. (197); Wagner et al. (1990), finally, discuss the guidance design of the NASP for vehicle autonomy.
The guidance system provides the necessary control variables (e.g., the angle of attack, the bank angle, the throttle setting, the axial or normal load factor, the location of the CoM). These variables are input for the second part, the control of the space plane, i.e., the motion around the CoM. The control system must provide, for instance, the deflections of control surfaces so that a particular angle of attack can be reached. We found a number of guidance-and-control-system designs in literature, as will be summarized below. These systems were applied to re-entry\(^9\) only, but they can serve as a basis from which further development is possible. It must be noted that the design of an optimal controller is not one of the goals of the study. The software must be extended with possible elements of a control system, i.e., aerodynamic surfaces, TVC and possibly CoM motion. Besides, an algorithm must be developed of how the system will respond to deviations of the nominal control variables. Input for the model will have to be generated in close cooperation with the Section Stability and Control of DUT/FAE.

MBB (1988a-e) describes the principle re-entry problems of winged re-entry vehicles, with emphasis on guidance and control strategies. A demonstration of related modelling and computer programs is included. The reference vehicle, which was used for the simulations, is the winged upper stage of the Sänger system, the Horus 2. Serrano-Martínez et al. (1987) present a Hermes hypervelocity re-entry trajectory study, including the computation of optimal trajectories and the guidance and control along these trajectories. Additional information on these topics can be found in Legenne et al. (1992). Serrano-Martínez et al. (1989) also discuss the

---

\(^9\) We must keep in mind that the flight simulation of space planes is not restricted to ascent flights only. The saying 'What goes up, must come down' is perfectly true for both a first and a second stage. After staging, the first stage will go back to its launch base (or another landing site, when this appears to be more convenient). The second stage continues its ascent to orbit, but after completion of the mission it deorbits and re-enters. Also this descent flight is completely guided and controlled. For a SSTO vehicle, a similar reasoning is of course true.
guidance and navigation of aerocapture capsules, applied to the ESA Comet-Nucleus-Sample-Return mission. ACRI/LAN (1992a-b), finally, describe the development of simulation software for the guidance of moderate lift and drag re-entry vehicles, hereby reviewing several guidance and control strategies.

To conclude the overview of the literature, we want to refer to Bryson et al. (1975), who give general background information on applied optimal control, Chowdhry et al. (1991), who describe optimal rigid body motion, Kokotovic et al. (1986) and Hodges et al. (1991). More specific details on space-plane dynamics and control can be found in McGraw et al. (1990), Schmidt et al. (1991), Raney et al. (1991), Hattis et al. (1992), Kugelmann et al. (1991) and Schmidt (1992).

The implementation of a guidance and control system in START is not completed with guidance and control laws alone. Also the geometry and aerodynamic properties of control surfaces must be considered. TVC and CoM motion should be studied as well. This includes the dynamic response time and operational intervals.

Input of a reference trajectory

Once an optimized trajectory for a launcher is known, it is interesting to see whether this trajectory can actually be flown if we are dealing with perturbations, like wind gusts and uncertainties in the atmosphere. A complete analysis, with varying perturbations, finally leads to a proof of steerability and controllability of the launcher, and whether it is more efficient to re-optimize the remaining part of the trajectory instead of trying to get back to the nominal trajectory.

For this reason it is necessary that START provides a means to input a reference trajectory. This trajectory can be, for instance, a table with values for the velocity as a function of altitude but also an energy function of time (ACRI/LAN, 1992a and 1992b). Besides, tables with the nominal (i.e., optimal) control variables (e.g., the bank angle, the angle of attack and/or the throttle setting) as a function of time may be needed. As an example, we refer to Lemarie (1992), who describes the Hermes nominal re-entry trajectories, Corban et al. (1991) and Hattis et al. (1989).

Jackson et al. (1991) describe a way of using dynamic interpolation, i.e., to simplify the specification of a reference trajectory in terms of a set of intercept points and continuity constraints. As an advantage of the discretely specified path they state that the system controls may be segmented between intercept points, allowing the application of piecewise-continuous polynomials to define the segment control. Further study is necessary, of course in combination with the controllability of the space plane, to find out whether this way of storing the trajectory is applicable for our purpose.

Vertical launch

The equations of translational motion in START Version 2.1 are based on spherical coordi-
nates. As a consequence, there are two singularities in the equations. The first one is of minor importance, and deals with a latitude position of $\delta = \pm 90^\circ$ (either the North or South Pole). Since it is not likely that a space plane is going to fly over the Poles, this singularity can be discarded for the time being. If, in any case, a conventional launcher is bound to go into a Polar orbit, special attention must be paid to this case.

The second singularity is of greater importance, not only for conventional launchers but also for an advanced launcher as the Delta Clipper. This singularity is related to moving vertically (along the gravity force), or, in numerical terms, with a flight-path angle $\gamma = \pm 90^\circ$. Since conventional launchers$^{10}$ in principle lift off vertically, one can understand the problem. In both Version 1.0 and Version 2.0, this singularity has been solved in a simplified way (Mooij, 1991a), which cannot be used for vertical launches. This is due to the fact that this solution implies that once the vehicle is moving vertically, it continues to do so, because there are no out-of-plane forces possible to change the attitude.

To solve this problem, one can approach it in two ways. Either one improves the current solution of overcoming the singularity, or an alternative set of equations of motion must be derived. Concentrating on the second solution, one can think of using cartesian position and velocity variables. A full derivation of the equations has been given by Mooij (TO BE PUBLISHED), and the cartesian model has been used (i.e., implemented, evaluated and executed) in the 6-dof parachute model, as described in the previous Section. Parachute descent is most of the time vertically downwards, so the model has shown its usefulness$^{11}$. Again, a complete description of the parachute model, including force modelling and the influence of wind, can be found in Mooij (1992).

It is also possible to stick to spherical position variables, but to change the velocity components (in which the singularity is present). In that case, we arrive at using velocity components in radial, longitudinal and latitudinal direction (i.e., the velocity components w.r.t. the local horizontal plane). These variables are better interpretable, and the equations are easier to use in combination with guidance and control. However, it is necessary that the attitude of the vehicle w.r.t. the local horizontal plane is known. A similar kind of modelling has been used for ALTOS (Deutsche Aerospace, 1992).

**Orbital interface**

A complete mission of a space plane consists of an ascent flight to orbit, the actual mission in

---

$^{10}$ This report deals, in principle, with advanced (future) launchers, which take off horizontally. However, since no space plane has ever flown, there are no flight data available for validation of the developed models. This does not apply to conventional launchers. For that reason it is possible to validate developed software on basis of these conventional launchers. Since a lot of the modelling, needed for space planes, can directly be used for conventional rockets, it has been decided to include the possibility of vertical launch in START. The User Interface shall be adapted such, that there is a clear distinction between the two launch categories, although internally there will hardly be a difference in treatment of the launchers.

$^{11}$ The presence of the parachute model enables us to simulate the parachute descent of, for instance, rocket boosters as part of a conventional-launcher mission.
orbit (e.g., the delivery of payload or research), and finally, after deorbiting, re-entry and the
descent flight back to the surface of the Earth. The latter part of the mission, i.e., re-entry and
descent, can already be simulated with the current version of START. As has been mentioned
before, the ascent flight will be modelled as part of one of the essential extensions. That leaves
the orbital phase.

For a complete mission simulation it would be convenient to have this phase included, although
it creates some problems. It would be possible to include a simple (analytical) model, which
gives, based on some initial conditions (usually the final conditions of the ascent) the final
conditions of the orbital phase (based on in-orbit mission duration and orbital information). At
that time, the de-orbit burn can be simulated (either numerical or analytical), resulting in the
entry conditions at the end of this phase. From there, a normal (numerical) re-entry simulation
can be performed. A disadvantage of an analytical method may be that the influence of the
atmosphere and the gravitational field\textsuperscript{12} cannot be included accurately enough. Here, it is
good to refer to the work of Regan (1984) and Battin (1987), who give a lot of analytical
formulations.

A full (numerical) simulation can take quite some CPU time, especially since the space plane
can be in orbit for more than one revolution around the Earth. The simulations become very
long (with correspondingly large data files, unless the output filing rate is adjusted accordingly),
and special attention has to be paid to the numerical integration method (a possible propagation
of numerical errors). Debatin et al. (1986) describe a fast numerical integration of interplanetary
orbits. This method should at least be compared with, for instance, the classical Runge-Kutta-
Fehlberg method, which is implemented in START. Also Battin (1987) gives a detailed overview
of numerical integration methods in the field of astrodynamics.

As we discussed in the previous Section, a short study has been conducted for ESTEC w.r.t.
the deorbit-burn manoeuvre of a re-entry capsule. However, the current implementation could
be extended and/or refined. A list with possible extensions can be found in Mooij (1993b).

\textit{Aeroheating}

The heat loads during the ascent, and especially during the re-entry, are severe. This means
that when the vehicle is flown by an autopilot, information about the heat loads is of vital
importance. Combined with the maximum allowable wall temperature and heating rate, the
guidance and control system must be able to fly such a trajectory that these limits are not
exceeded. The maximum wall temperature and heating rate act as so-called constraints on the
trajectory to be flown.

For the implementation in START, we will restrict to a simplified model. This model must give
us at least an overestimate of the actual values, computed by complex CFD codes. Examples
of thermodynamical models can be found in, for instance, Hankey (1988) and Serrano-Martínez

\textsuperscript{12} Apart from the attraction of the Earth, it may also be necessary to include the perturbing effect of third
bodies, like the Moon and the Sun.
et al. (1987 and 1989).

Extension of the User Interface

It is evident, that the UI is heavily affected by all the extensions, which are mentioned above. It would be beyond the scope of this report to consider each of the extensions in great detail. We will restrict ourselves to a few remarks.

For simulating propelled flights, we need to be able to define (and edit) propulsion systems. This modelling exists of the number of engines, their respective locations, the thrust and specific impulse as a function of, for instance, the Mach number, altitude, the angle of attack and the fuel equivalence ratio. Besides, during an ascent flight to orbit, several engines are going to be used, i.e., turbo-, RAM- or SCRAM- jets and rocket engines. Switching conditions must be added, to allow for a certain mission profile. It must be possible to enter (and edit) the propulsion parameters in tabular form, as a function of a number of independent variables. However, it may be more convenient to enter the parameters as some kind of polynomial or spline function. The same remarks are applicable for the (variable) mass properties.

Related to the guidance and control system are, amongst others, the input of control surfaces, consisting of geometrical and aerodynamic data. Besides, when there is a number of guidance and/or control laws available for different flight segments, it must be possible edit the choice of these laws. This can be generalized to edit possibilities w.r.t. the mission scenario. Also steering and trajectory constraints fall into this category.

It may be obvious that the complete input-data set will be quite large, which gives a reason to divide the data into different modules, or databases. In START Version 2.1, it is already possible to load and save parts of the input data, i.e.

- an aerodynamic database, in which a complete aerodynamic configuration is stored (three force and three moment coefficients),
- a wind database, in which the total user-defined wind model is stored, and
- a parachute database, defining the number of parachute systems, mass and geometry, and aerodynamic properties.

For START Version 3.0, the number of databases can be extended with

- a propulsion system database, in which a number of different engines with corresponding data is stored,
- a configuration database with geometry and mass properties; it may even be necessary to split this database into two separate ones,
- a control database, with information about the control surfaces,
- a mission database, with complete mission scenarios

The subdivision of the input data in different modules, justifies the existence of one overall database, from which blocks of data can be retrieved to 'compose' an input set.
3.3.5. START Version Flight Simulator?

This Section has been added for the sake of information only. A possible implementation as Flight-Simulator software will not be part of the current study.

It can be disputed that the final version of START (3.0) is suitable to be implemented on the flight simulator BARESIM of the Section Stability and Control (Faculty of Aerospace Engineering, Delft University of Technology). It is obvious that the major part of the UI is superfluous w.r.t. implementation on the simulator. Besides, it is unlikely that a vertical launch will ever be simulated. On the other hand, all the developed models for aerodynamics, propulsion, flight dynamics, guidance and control, etc. can without more ado be used as a basis for a possible recoding of the software, and some parts can without doubt directly be transferred. Also the developed modular layout is applicable. This means, that once a fully tested, evaluated and (hopefully) validated version of START including all documentation is available, the effort to come to flight-simulator software is not that substantial any more.

As an example, we have included Fig. 3.11. In this Figure, the schematic layout of a space-plane flight-simulation model is depicted. Three main modules are visible, i.e., from left to right the space-plane input data, the flight-dynamics software and the flight-simulator related software and hardware. When we focus on the second block, we see that the layout can be exactly the same as the one of START Version 3.0. The core module is the driver of the software: all internal data transport will be coordinated by this module. Contribution to the forces and moments module come from the propulsion (thrust), aerodynamics (lift, drag, etc.), environment (gravity field), description-of-space-plane (mass and geometry) modules. State-vector computation takes place in the module equations-of-motion, with input of the forces-and-moments module, of course. The guidance-and-control module forms the basis of a steerable and controllable space plane; the aeroheating module, finally, gives information about heat loads. For an autopilot, this information can be treated as a constraint while flying a nominal trajectory.
3.3.6. Current status of START.

As by this moment (early November 1993), some work has already been done towards START Version 3.0. Basically, the work is related to the implementation of ASCENT in START. This
has been decided, because it is the shortest way to arrive at a 6-dof tool, capable of simulating ascent trajectories of space planes. Besides, ASCENT has been developed as a descendant of START, which makes implementing the related modules of ASCENT in START relatively easy. The status is, that the data structure of START has been extended, so that the extra features of ASCENT can be inputted and edited. The related data blocks are dealing with the propulsion system and the mission definition, and include a full extension of the UI. Since the propulsion data could not be edited in ASCENT, a completely new part of the UI has been developed\(^{13}\). The next step will be the implementation of the guidance module of ASCENT.

Furthermore, the UI has been adjusted in such a way, that the mass properties, the propulsion systems, and the mission definition can be saved as separate databases. Nota bene: these databases are for the use by START only. They do not have a relation with the database, as mentioned in Section 2.2.3. Besides, it is now possible to write the input data in both binary and ASCII form. The latter was until now the only possibility, but due to the possible huge amount of input data, we had to look at a more efficient way of storing the data.

Finally, a beginning has been made, with the development of a variable-mass properties model in START. The Euler equations of rotational motion have been fully written out, so that the inverse inertia tensor does not have to be computed any more.

### 3.3.7. Future development.

For the future development, a number of steps is foreseen. The major ones will be given below. Since START is originally a re-entry tool, the first phase of the future development will be related to finalizing the re-entry capabilities (i.e., guidance and control) before we will start working on the ascent trajectory. The steps of the work are now as follows:

1) Implementation of aerodynamic control. As a reference vehicle, the HORUS 2B has been selected, because a complete study is available on this vehicle (MBB, 1988a-e). START will be evaluated on the basis of the results, which have been presented there.

2) Implementation of reaction control. Two studies will serve as a basis: the MBB study of HORUS (1988a-e) and the aerocapture of ROSETTA, a capsule which is during the re-entry controlled by reaction-control jets only (Serrano-Martín et al., 1989).

3) Implementation of hybrid control (combination of the two previous control possibilities). The basis is again the MBB study; however, two studies of the re-entry of Hermes (Serrano-Martín et al. 1987) en Gockel (1993) have been found to provide additional information.

\(^{13}\) It is possible to enter 15 jet engines and 15 rocket engines. For each of the engines, one can enter tables for the thrust and the specific impulse, as a function of maximal three independent variables (Mach number, dynamic pressure, height and equivalence ratio). Besides, the initial thrust-vector angles and the location of the thrust centre can be entered. Furthermore, it is possible to copy table from one engine to another, and it is possible to link engines, so that only one table will be used to compute the performance parameters of a group of engines.
4) Modelling of the ascent trajectory, with as a first step the further implementation of ASCENT in START. With respect to the guidance and control in this phase, the previous implementations will serve as a basis. As extensions, TVC and TMC will be implemented. Active CoM will be studied as an alternative and possibly implemented as well. The implementation includes the development of the data structure and UI. As the reference vehicle for this phase of the development, the Winged Cone Configuration (Shaughnessy, 1990) has been selected.

5) An equation will be derived which describes the moment equilibrium about the pitch axis. As a start, we will consider as input the aerodynamic forces and pitching moment, the location of the CoM, the aerodynamic reference centre and the thrust centre, and the magnitude of the thrust vector. The direction of the thrust vector in order to guarantee equilibrium can be the output. A careful study has to be conducted to see the interference with other elements of the simulation model, for instance the guidance module: if we change the direction of the thrust, how will it influence the required angle of attack? How will the change in angle of attack influence the moment equilibrium? An iteration process is foreseen. As an alternative, it has to be possible to compute the required location of the CoM, with the moments and forces as an input.

6) Adjustment of the guidance module, as implemented in ASCENT. The trajectory constraints have to be extended with at least the normal acceleration. It must be studied, whether there are more to be added. As a control variable of the engine, the equivalence ratio will be used, maybe in combination with the throttle setting.

7) The inclusion of trim drag in the related coefficients. It has to be studied what will be the most efficient way. The forces computation has to be adjusted accordingly; an iterative procedure is foreseen.

8) A review will be made of the available thermodynamical models for space vehicles, and based on the outcome one should be selected and implemented. The use of a tabular model will be considered as well. The output of this model (trajectory constraint) will be used as input for the guidance system.

9) Since ALTOS can be obtained, it is not necessary that an optimizer has to be linked with ASCENT within the framework of this study. However, time will be invested in getting acquainted with the use of ALTOS. This also includes a literature study on optimization techniques. Linked with generating an optimal trajectory including the control history, is the input of this trajectory in START, so that it can serve as a basis for the guidance system. A data structure and related UI will be developed.

10) The development of a simulation scheme. As a basis, we will use Section 2.2.2. In principle, we want to study the topics, which were introduced in that Section.
References.

1) ACRI/LAN;
   Guidance and control for moderate lift/drag reentry - Executive summary;
   ESA Contract no. 9359/91/NL/JG (SC);
   ACRI, Valbonne, 1992a.

2) ACRI/LAN;
   Guidance and control for moderate lift/drag reentry - Final report;
   ESA Contract no. 9359/91/NL/JG (SC);
   ACRI, Valbonne, 1992b.

3) AEOLUS Team;

4) Alberico, J.F.;
   The development of an interactive computer tool for synthesis and optimization of
   hypersonic air-breathing vehicles;
   AIAA-92-5076;
   From: AIAA Fourth International Aerospace Planes Conference (Orlando, FL, 1-4 Dec.

5) Arend, H.;
   Costs and benefits of future heavy space freighters;

6) Battin, R.H.;
   An introduction to the mathematics and methods of astrodynamics;
   AIAA Education Series;

7) Battrick, B. (editor);
   Aerothermodynamics for Space Vehicles;
   Proceedings of the First European Symposium, Noordwijk, 28-30 May 1991;
   ESA SP-318;

8) Bauer, T., Betts, J.T., Hallman, W.P., Huffman, W.P. and Zondervan, K.P.;
   Solving the optimal control problem using a nonlinear programming technique part 2:
   Optimal Shuttle ascent trajectories;
   AIAA-84-2038;

9) Betts, J.T., Bauer, T., Huffman, W.P. and Zondervan, K.P.;
   Solving the optimal control problem using a nonlinear programming technique part 1:
   General formulation;
AIAA-84-2037;

10) Blaauw, F.L.;
Four-dimensional trajectory optimization and guidance techniques for terminal area traffic management;

11) Bowcutt, K.G.;
Hypersonic aircraft optimization including aerodynamic, propulsion, and trim effects;
AIAA-92-5055;

12) Branscome, D.R. and Reese, T.G.;
NASA launch vehicles: the next 20 Years;

Program to Optimize Simulated Trajectories (POST);
Volume 1: Formulation Manual (NASA CR-132689);
NASA, 1975a.

Program to Optimize Simulated Trajectories (POST);
Volume 3: Programmer's Manual (NASA CR-132691);
NASA, 1975b.

Capabilities and applications of the Program to Optimize Simulated Trajectories (POST);
NASA CR-2770;

Six-degree-of-freedom Program to Optimize Simulated Trajectories (6D POST);
Volume 1: Formulation Manual (NASA CR-132741);

17) Bryson Jr., A.E. and Ho, Y.-C.;
Applied optimal control;

18) Buhl, W., Ebert, K. and Herbst, H.;
Optimal ascent trajectories for advanced launch vehicles;
AIAA-92-5008;
From: AIAA Fourth International Aerospace Planes Conference (Orlando, FL, 1-4 Dec.
19) Bulirsch, R. and Chudej, K.;
Ascent optimization of an air-breathing space vehicle;
AIAA-91-2656-CP;

20) Bulirsch, R. and Chudej, K.;
Staging and ascent optimization of a dual-stage space transporter;
Zeitschrift für Flugwissenschaften und Weltraumforschung, Vol. 16, No. 3, pp. 143-151,

21) Chaput, A.J.;
Preliminary sizing methodology for hypersonic vehicles;

22) Chowdhry, R.S., Cliff, E.M. and Lutze, F.H.;
Optimal rigid-body motions;

23) Christophe, B.;
Two-stage air-breathing launcher trajectory optimization including first-stage return;
AIAA-92-5007;
From: AIAA Fourth International Aerospace Planes Conference (Orlando, FL, 1-4 Dec.

24) Cochran, J.E., No, T.S. and Thaxton, D.G.;
Analytical solutions to a guidance problem;

25) Conchie, P.J.;
A horizontal take-off and landing satellite launcher or aerospace plane (HOTOL);

26) Corban, J.E., Calise, A.J., and Flandro, G.A.;
A real time guidance algorithm for aerospace plane optimal ascent to low Earth orbit;

Optimal guidance and propulsion control for transatmospheric vehicles;

28) Corban, J.E., Calise, A.J. and Flandro, G.A.;
Rapid near-optimal aerospace plane trajectory generation and guidance;

29) Cruz, C.I. and Wilhite, A.W.;
Prediction of high-speed aerodynamic characteristics using the Aerodynamic Preliminary Analysis System (APAS);
AIAA-89-2173;

30) Darwin, C.R. Austin, G., Varnado, L. and Eudy, G.;
A view toward future launch vehicles: a civil perspective;

31) Debatin, F., Tiligner, A. and Hechler, F.;
Fast numerical integration of interplanetary orbits;
ESA SP-255;

32) Deutsche Aerospace;
ALTOS - Advanced Launcher Trajectory Optimization Software;

33) Dorrington G.E.;
Optimum Directions for Air-breathing Launch Vehicle Design;
AIAA-87-1818;
AIAA 23rd Joint Propulsion conference (San Diego, June 29-July 2, 1987).

34) Feustel-Büechl, Isakeit, D. and Pfeffer, H.;
Wings for European Spaceflight - The Future of Space Transportation;

35) Furniss, S.G. and Walters, I.M.;
Ascent and descent optimizations of an air-breathing launch vehicle;
AGARD-CP-489;
From: Space Vehicle Flight Mechanics Conference (paper 22);

36) Gockel, W.;
Angular control of a reentry vehicle in hypersonic flight regime;
AIAA-93-5090;
From: AIAA/DGLR Fifth International Aerospace Planes and Hypersonics technologies Conference, 30 November - 3 December 1993, Munich, Germany.

37) Gord, P.R., Langan, K.J. and Stringer, M.E.;
Advanced launch vehicle configurations and performance trades;
AGARD-CP-489;
From: Space Vehicle Flight Mechanics Conference (paper 5);
38) Gottfried, B.S. and Weisman, J.;
Introduction to optimization theory;

39) Grallert, H.K.H. and Berry, W.;
Progress in ESA winged launcher configuration studies;
IAF-91-199;
From: 42nd Congress of the International Astronautical Federation (Montreal, Canada; October 5-11, 1991).

40) Guyenne, T.D. and Hunt, J.J. (editors);
Progress in space transportation;
ESA SP-293;

41) Hankey, W.L.;
Re-entry aerodynamics;

42) Hardtla, J.W., Pichler, M.J. and Bradt, J.E.;
Guidance requirements for future launch vehicles;
AIAA 87-2462.

43) Hattis, P.D.;
Optimal air-breathing launch vehicle design;

44) Hattis, P.D. and Malchow, H.;
Evaluation of some significant issues affecting trajectory and control management for air-breathing hypersonic vehicles;
AIAA-92-5011;

45) Hattis, P.D., Malchow, H., Shaughnessy, J. and Chowdhry, R.;
Integrated trajectory and control analysis for generic hypersonic vehicles;
AIAA-91-5052;
From: AIAA Third International Aerospace Planes Conference (Orlando, FL, 3-5 dec. 1991).

Optimal trajectory generation and design trades for hypersonic vehicles;
47) Heppenheimer, T.A.;
The National Aerospace Plane;

48) Hodges, D.H. and Bless, R.R.;
Weak Hamilton finite element method for optimal control problems;

49) Jackson, J.W. and Crouch, P.E.;
Dynamic interpolation and application to flight control;

50) Jänsch, C. and Markl, A.;
Trajectory optimization and guidance for a Hermes type re-entry vehicle;
AIAA-91-2659-CP.

51) Jänsch, C., Schnepper, K. and Well, K.H.;
Ascent and descent trajectory optimization of Ariane V/Hermes;
AGARD-CP-489;
From: Space Vehicle Flight Mechanics Conference (paper 4);

52) Koelle, D.E.;
Cost-optimized launch vehicle design;
AIAA paper 92-1281.

Sänger II, an advanced launcher system for Europe;

54) Koelle, D.E. and Kuczera, H.;
Sänger Space Transportation System - Progress Report 1990;
Paper No. IAF-90-175;
From: 41st IAF Congress, Dresden, October 1990.

55) Kokotovic, P.V., Khalil, H.K. and O'Reilly, J.;
Singular perturbation methods in control: analysis and design;

56) Korswagen, J.H.;
ASCENT trajectory simulation for aerospace planes: development of a simulation program;
Report SSR-2;
Delft University of Technology, Faculty of Aerospace Engineering, 1993a.

57) Korswagen, J.H.;
ASCENT user manual - version 1.0;  
Report SSR-4;  
Delft University of Technology, Faculty of Aerospace Engineering, 1993b.

58) Kugelmann, B. and Pesch, H.J.;  
Real-time computation of feedback controls with applications in aerospace engineering:  
controllability analysis of a TSTO vehicle;  
AIAA-91-2658-CP.

Singular optimal atmospheric rocket trajectories;  
AIAA 87-2513.

60) Lawden, D.F.;  
Rocket trajectory optimization: 1950-1963;  

61) Lecohier, G. and Mehlem, K.;  
A new trajectory optimisation tool (ALTOS) applied to conventional launchers;  
From: 12th IFAC Symposium on Aerospace Control (Ottobrunn, Germany, September 7-11, 1992).

62) Legenne, J., Broca, R. and Alby F.;  
Hermes phasing GNC functions;  
IAF 92-0043;  

63) Lemarie, D.;  
Hermes nominal re-entry trajectories;  
Paper No. IAF-92-0018;  

64) Lu, P.;  
Trajectory optimization and guidance for a hypersonic vehicle;  
AIAA-91-5068;  
From: AIAA Third International Aerospace Planes Conference (Orlando, FL, 3-5 dec. 1991).

65) Marée, A.G.M.;  
Guidance system design for spaceplanes; a design based on gain scheduling;  
Report SSR-1;  
Delft University of Technology, Faculty of Aerospace Engineering, 1992.

66) Marée, A.G.M.;  
Inlet performance and aerothermodynamic loads, using Intake-1, -2, -3;
AEOLUS Document code AE-DUT-TN-9301 (WP-2233);
Report LR-730;
Delft University of Technology, Faculty of Aerospace Engineering, 1993.

67) Marée, A.G.M., Mooij, E. and Zandbergen, B.T.C.;
Study on a typical space plane and its trajectory: the generation of a reference trajectory
(PRELIMINARY TITLE WP-212);
Delft University of Technology, Faculty of Aerospace Engineering, 1993, TO BE PUBLISHED.

68) Martel, F.;
Global optimization of air-breathing launch vehicle trajectories;
AGARD-CP-489;
From: Space Vehicle Flight Mechanics Conference (paper 24);

69) Martin, J.A;
Effects of tripropellant engines on Earth-to-orbit vehicles;

Saturn V guidance, navigation, and targeting.

71) MBB Space Communication and Propulsion Systems Division;
Study on re-entry guidance and control. Final report - Volume 1: Executive summary.
ESA report reference: ESA CR (P) 2652;
MBB, Munich, 1988a.

72) MBB Space Communication and Propulsion Systems Division;
Study on re-entry guidance and control. Final report - Volume 2: Ref. vehicle definition
and orbital constraints. ESA report reference: ESA CR (P) 2652;
MBB, Munich, 1988b.

73) MBB Space Communication and Propulsion Systems Division;
Study on re-entry guidance and control. Final report - Volume 3: Re-entry trajectory
simulation. ESA report reference: ESA CR (P) 2652;
MBB, Munich, 1988c.

74) MBB Space Communication and Propulsion Systems Division;
Study on re-entry guidance and control. Final report - Volume 4: FCS and guidance
concepts. ESA report reference: ESA CR (P) 2652;
MBB, Munich, 1988d.

75) MBB Space Communication and Propulsion Systems Division;
Study on re-entry guidance and control. Final report - Volume 5: Simulation software.
ESA report reference: ESA CR (P) 2652;
MBB, Munich, 1988e.
76) McEneaney, W.M.;
Optimal aeroassisted guidance using Loh's term approximations;

77) McGraw, G.A. and Mak, P.H.;
Adaptive control for advanced launch vehicles;
AAS 90-052;

National Aero-Space Plane - flight mechanics;
AGARD-CP-489;
From: Space Vehicle Flight Mechanics Conference (paper 20);

79) McLain, M.G. and Nguyen, H.N.;
An assessment of computational efficiency for a nonlinear programming and collocation technique in trajectory optimization;
AIAA-90-3416-CP.

80) Mooij, E.;
Development of START, a six degrees of freedom Simulation Tool for Atmospheric Re-entry Trajectories;
ESTEC Working Paper EWP 1633;
ESTEC, Noordwijk, 1991a.

81) Mooij, E.;
ESA Huygens probe entry trajectory analysis;
ESTEC Working Paper EWP 1634;
ESTEC, Noordwijk, 1991b.

82) Mooij, E.;
START User Manual, Version 1.0;
Final thesis Delft University of Technology, faculty of Aerospace Engineering;
Delft, 1991c.

83) Mooij, E.;
ESA Huygens probe entry and descent analysis;
ESTEC Working Paper EWP 1679;

84) Mooij, E.;
START User Manual; Version 2.1;
Report LR-726;
Delft University of Technology, Faculty of Aerospace Engineering, 1993a.
85) Mooij, E.; The influence of the deorbit-burn manoeuvre on the footprint of a re-entry capsule; Report LR-727; Delft University of Technology, Faculty of Aerospace Engineering, 1993b.

86) Mooij, E., Marée A.G.M., and Zandbergen, B.T.C.; Flight simulation for aerospace planes. Tool development plan; AEOLUS Document code AE-DUT-TN-9302 (WP-151); Report LR-733; Delft University of Technology, Faculty of Aerospace Engineering, TO BE PUBLISHED.

87) Mooij, E.; The motion of a vehicle in a planetary atmosphere; Delft University of Technology, Faculty of Aerospace Engineering, TO BE PUBLISHED.

88) Nguyen, H.N.; Optimal ascent trajectories of the horizontal take-off single-stage and two-stage-to-orbit launchers; AIAA-91-2740-CP.

89) Nguyen, H.N.; Optimal solutions to flight mechanics problems using a nonlinear programming and collocation technique; AIAA-90-3415-CP.


91) Powell, R.W., Naftel, J.C. and Cruz, C.I.; Flight control issues of the next generation space transportation launch vehicles; AGARD-CP-489; From: Space Vehicle Flight Mechanics Conference (paper 3); Neuilly sur Seine, France, 1990.


94) Raney, D. and Lallman, F.; Control concept for maneuvering in hypersonic flight; AIAA-91-5055;
From: AIAA Third International Aerospace Planes Conference (Orlando, FL, 3-5 December 1991).

95) Regan, F.J.;
Re-entry vehicle dynamics;
AIAA Education Series;

96) Sachs, G., Bayer, R. and Drexler, J.;
Optimal trajectories for Sänger-type vehicles;
AGARD-CP-489;
From: Space Vehicle Flight Mechanics Conference (paper 21);

97) Schmidt, D.K., Mamich, H. and Chavez, F.;
Dynamics and control of hypersonic vehicles - The integration challenge of the 1990's;
AIAA paper 91-5057;

98) Schmidt, D.K.;
Problem in control design for hypersonic vehicles;
From: 12th IFAC Symposium on Aerospace Control (Ottobrunn, Germany, September 7-11, 1992).

99) Schoettle, U.M.;
Performance Analysis of Rocket-Ramjet Propelled SSTO Vehicles;

100) Schoettle, U.M. and Hillesheimer, M.;
Performance optimization of an air-breathing launch vehicle by a sequential trajectory optimization and vehicle design scheme;
AIAA 91-2655;

Trajectory optimization for hypersonic aircraft guidance;

Study on aerocapture guidance and navigation;
ESA contract no. 7260/87/D/MD;
Final report GMVSA 2049/89;
103) Serrano-Martinez, J.B. and Parra, J.A.;
Hermes hypervelocity re-entry trajectory study;
ESA contract no. 6932/86/F/CG;
Final report GMVSA 212/87;

104) Shaughnessy, J.D., Zane Pinckney, S., McMinn, J.D., Cruz, C.I. and Kelley, M.L.;
Hypersonic vehicle simulation model: Winged-Cone configuration;
NASA Technical Memorandum 102610;
Langley Research Center, Hampton, Virginia, November 1990.

Conceptual design of a fully reusable manned launch system;

Application of Taguchi methods to propulsion system optimization for SSTO vehicles;

Comparison of single-stage and two-stage air-breathing launch vehicles;

108) Staufenbiel, R.W.;
Second-stage trajectories of air-breathing space planes;

109) Staufenbiel, R.W.;
Transport efficiency of spaceplanes with air-breathing phases;

110) Strohmaier, P., Kiefer, A., Burkhardt, D. and Horn, K.;
Re-entry trajectory optimization and control;
AGARD-CP-489;
From: Space Vehicle Flight Mechanics Conference (paper 7);

111) Taguchi, G.;
System of experimental design. Engineering methods to optimize quality and minimize costs (Volume 1);

112) Townend, L.H.;
Research and design for hypersonic aircraft;
113) Tylee, J.L. and Hardtla, J.W.;
Adaptive guidance for future launch vehicles;
AIAA 88-4124-CP.

114) Venugopal, N., Grandhi, R.V., Hankey, W.L. and Belcher, P.J.;
Automated trajectory synthesis for hypersonic vehicles using energy management and
variational calculus techniques;

115) Vinh, N.X.;
Optimal trajectories in atmospheric flight;

116) Wagner, E., Li, I., Nguyen, D. and Nguyen, P.;
NASP guidance design for vehicle autonomy;
AIAA-90-5245;
1990).

STS 2000: a reference air-breathing SSTO;
AIAA-91-5013;
From: AIAA 3rd International Aerospace Planes Conference (Orlando, Florida; December
3-5, 1991).

118) Well, K.H. and Ebert, K.;
Trajectory optimization techniques and software implementation;
From: 12th IFAC Symposium on Aerospace Control (Ottobrunn, Germany, September 7-
11, 1992).

Rocket ascent trajectory optimization via recursive quadratic programming;

120) White, J.E.;
Guidance and targeting for the strategic target system;

121) Wilson, A.W. (ed.);
Interavia space directory; 1991-92;

122) Yiyuan Zhao;
A classification of trajectory optimization algorithms;
AIAA-91-2724-CP.
123) Zandbergen, B.T.C.;
Spaceplane flight simulation; project definition;
memorandum M-661;
This page has been intentionally left blank
Appendix A: Work Packages AEOLUS Project

January 1994

Ir. E. Mooij
WORK PACKAGE DESCRIPTION

Project: AEOLUS  Phase: 1993  W.P. Ref.: AEOLUS-0151

W.P. Title: Dynamic flight simulation tool  Sheet: 1 of 1
Major Constituent: Technology Development
Contractor: TUD-L&R
Start event: Kick-off  Issue: 1
End event:  Issue Date: 25-9-92
W.P. Manager: 

1. **Objective:**
   Development of dynamic flight simulation and trajectory optimization tool

2. **Input:**
   1. Document containing requirements for the development (or acquisition) and verification of spaceplane analysis tools (output of WP 211)
   2. Available existing software
   3. Relevant literature

3. **Tasks:**
   1. Review of requirement specifications
   2. Inventory of existing tools
   3. Review of existing tools
   4. Definition of tool development program
   5. Reporting

4. **Output:**
   1. Document containing a review of the requirement specifications (input to WP 211)
   2. Document containing the results of tasks 4 and 5 (input to WP 211)

5. **1993: all**
WORK PACKAGE DESCRIPTION

Project: AEOLUS                          Phase: 1993                          W.P.Ref.: AEOLUS-0212

W.P. Title:                           Spaceplane analysis                          Sheet: 1 of 1
Major Constituent:                   Spaceplane
Contractor:                          TUD-L&R
Start event:                          Kick-off
End event:                           
W.P Manager:                         

1. Objective:
WP 2121: Review of spaceplane concepts
WP 2122: Definition of flight conditions along the trajectory

2. Input:
WP 2121:
1 Document containing the requirements for the baseline spaceplane concept(s) (output of WP 211)
2 Relevant literature

WP 2122:
1 Document containing the requirements for the reference trajectories (output of WP 211)
2 Relevant literature

3. Tasks:
WP 2121:
1 Review of requirement specifications
2 Investigation into typical spaceplane concepts
3 Reporting

WP 2122:
1 Review of requirement specifications
2 Investigation into typical spaceplane trajectories, trajectory constraints, controls and control laws
3 Reporting

4. Output:
WP 2121
1 Document containing a review of the requirement specifications (input to WP 211)
2 Document containing the results of task 2 (input to WP 211)

WP 2122
1 Document containing a review of the requirement specifications (input to WP 211)
2 Document containing an overview of typical spaceplane trajectories, trajectory constraints, controls and control laws (input to WP 211)

5. 1993: all