Development of a Post-Optimality Analysis Algorithm for Optimal Control Problems

Von der Fakultät Luft- und Raumfahrttechnik und Geodäsie
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieur (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von

Sven Oliver Erb

aus Waldshut-Tiengen

Hauptberichter:	Prof. Klaus Hinrich Well, Ph.D.
Mitberichter:	Prof. Dr. rer. nat. Ernst W. Messerschmid

Tag der mündlichen Prüfung: 12. Januar 2007

Institut für Flugmechanik und Flugregelung
der Universität Stuttgart

2007

Acknowledgements

The foundation for this work was laid in the early summer of 2002 when Prof. Klaus Well encouraged me in a short, but decisive conversation to deepen my knowledge in optimization methods and their utilization in aerospace engineering. My activities as a research scientist in the following years at the Institute of Flight Mechanics and Control (IFR) at the University of Stuttgart, then, led to this dissertation.
In numerous discussions Prof. Klaus Well sharpened my understanding of scientific research with an engineering twist. My deepest gratitude goes to him not only for the first encouragement, but also for his ongoing support and confidence in me.

I also want to thank Prof. Ernst Messerschmid, who took to time to review my dissertation and who gave me a number of very helpful remarks.

My colleagues at the IFR and TTI have been a constant source of inspiration and motivation. I want to specifically thank Andreas Wiegand and Sven Weikert for the input they gave for the demonstration cases and their ADA knowledge-base that I exploited frequently.

The active collaboration with several people from the Institute of Space Systems (IRS) at the University of Stuttgart was certainly fruitful in the sense that it brought me closer to applied space technology. I recollect each single discussion at the IRS as a valuable contribution.

For scientific guidance and a friendly word whenever we were in tough I would like to thank John Betts from The Boeing Company. He has significantly influenced my personal development. And so has Seattle.

Meiner Familie möchte ich für die Unterstützung über all die Jahre danken. Ihr habt mir einige Steine aus dem Weg geräumt und den Rücken frei gehalten. Das weiss ich.

Det är inte enkelt at bara tilsammans med en rymd ingenjör. Tack för at du har lyssnad på mig altid. Jag älskar dig!

Munich, April 2007 Sven O. Erb

Contents

Acknowledgements **i**

Table of Contents **ii**

Nomenclature **vi**

Zusammenfassung **ix**

Summary **xi**

1 Introduction **1**

1.1 The Optimal Control Problem 2

1.1.1 Mathematical Description 2

1.1.2 Understanding an Optimal Solution 5

1.2 Sensitivity as a Means of Analysis 6

1.3 The Question of Solution Analysis in Optimization 9

1.4 Previous Works on Optimal Solution Analysis 9

1.5 Design of a Post-Optimality Analysis Algorithm 11

2 Sensitivity Analysis of Optimal Control Applications **13**

2.1 The Parameterized Sensitivity Equations 13

2.1.1 Nonlinear Programming Problem 14

2.1.2 Derivation of First Order Sensitivity Equations 16

2.1.3 Second Order Sensitivity Coefficients 18

2.2 Exemplary Analysis of an Optimal Solution 20

2.2.1 Simplified Earth Mars Transfer 20
2.2.2 Interpretation of the Solution 21
2.3 Scaling Considerations 23
2.4 Direct Control Optimization 26
2.4.1 Shooting Methods 27
2.4.2 Collocation Methods 28

3 Extension for Post-Optimality Analysis 29
3.1 Modified Zermelo Problem 29
3.1.1 Problem Description 30
3.1.2 Impairment of the Conventional Analysis 31
3.2 Identification of Active Inequality Constraints 32
3.3 Prediction of Active Set Changes 34
3.3.1 Estimation of Perturbed Lagrange Multipliers 34
3.3.2 Confidence Region 36
3.3.3 Completion of Active Set Data 36
3.3.4 Practical Example 37

4 Integration with an Existing Optimization Software 43
4.1 Transcription of the Optimal Control Problem 44
4.1.1 Multiple Shooting Scheme 45
4.1.2 Collocation Scheme 45
4.2 The Hessian Matrix 47
4.2.1 Finite Difference Approximations 47
4.2.2 Efficient Computing 49
4.3 Inclusion of Boundary Conditions 50
4.4 Lagrange Multiplier Accuracy 52

5 Example 1: Hopper Optimal Re-Entry Analysis 55
5.1 Description of Model and Mission 56
5.2 Parameter Perturbations in the Model 58
5.3 The Optimized Nominal Trajectory 59
5.4 Results of the Post-Optimality Analysis 62

6 Example 2: Ariane 5 Post-Optimality Analysis 75
6.1 Description of Model and Mission 75
6.2 Post-Optimal GTO-Launch Analysis 79
6.2.1 Sensitivity with Respect to Component Properties 80
6.2.2 Effects of Path Constraint Variations 82
6.2.3 Sensitivity with Respect to the Deployment Orbit 85
6.3 Dual-Payload Performance Sensitivity 86
6.3.1 Sensitivity with Respect to the Component Properties 89
6.3.2 Sensitivity with Respect to the Deployment Orbits 91
6.3.3 Coast-Arc Sensitivity Matters 92

7 Conclusions 95

References 99

Appendices 107

A The Earth Mars Transfer Parameterization 107
A.1 List of Optimization Parameters 107
A.2 List of Optimization Constraints 109

B Collocation Methods 111
B.1 Trapezoidal Rule 111
B.2 Classical Runge-Kutta Method 111
B.3 Hermite-Simpson Method 112

C HOPPER Results 113

D Ariane 5 - Dual-Payload Launch 117

Curriculum Vitae 119

Nomenclature

Acronyms

AU	Astronomical Unit
CAMTOS	Collocation And Multiple shooting Trajectory Optimization Software
DOF	Degrees Of Freedom
GESOP	Graphical Environment for Simulation and Optimization
ISP	Impuls, Specific
KKT	Karush-Kuhn-Tucker
MDO	Multi Disciplinary Optimization
NLP	Nonlinear Programming
OCP	Optimal Control Problem
POA	Post-Optimality Analysis
PSA	Parameterized Sensitivity Analysis
SA	Sensitivity Analysis
SQP	Sequential Quadratic Programming
SSA	System Sensitivity Analysis
SSTO	Single Stage To Orbit
TAEM	Terminal Area Energy Management
UL	Unit Length

Greek Symbols - Scalars

γ	Flight-path angle, coefficient
δ	Geocentric declination
ζ	Coefficient
Θ	Heading angle, coefficient
κ	Coefficient
λ	Geocentric longitude
μ_g	Gravitational constant
ρ	Barrier gradient
τ	Coefficient

Φ	Angular position
Φ	Mayer term
χ	Flight-path azimuth

Greek Symbols - Vectors and Matrices

ϵ	Perturbation vector
λ	Lagrange multipliers
λ	Adjoint vector
μ	Lagrange multipliers
ν	Adjoint vector
$\mathbf{\Psi}$	Boundary constraint vector

Latin Symbols - Scalars

c	Constraint
d	Distance
h	Altitude
H	Hamiltonian
J	Cost functional
k	Coefficient
L	Integrand, Lagrange term
$\mathcal{L}$	Lagrange function
m	Mass
Ma	Mach number
n	Load factor
q	Dynamic pressure
$\dot{Q}$	Heatflux
r	Radius
s	Shift
$\mathcal{S}$	Sensitivity coefficient
t	Time, independent variable
T	Thrust
u	Velocity
v	Velocity
V	Relative velocity

Latin Symbols - Vectors and Matrices

$\mathcal{C}$	Set of constraints
$\mathbf{g}$	Inequality path constraint vector
$\mathcal{G}$	Set of inequality constraints

$\mathbf{h}$	Equality path constraint vector
$\mathcal{H}$	Set of equality constraints
$\mathbf{p}$	Design parameter vector
$\mathbf{u}$	Control vector
$\mathbf{x}$	State vector

Mathematical Symbols

Δ	Difference between two values
$\dot{X}$	First order time derivative of X
∇	Nabla operator
∂	Partial derivative

Subscripts/Superscripts

$*$	Optimum, nominal set
$\bar{X}$	Updated vector X
$\tilde{X}$	Subset of original vector X
0	Initial value
ϵ	Optimum, perturbed set
ϵ	Derivative with respect to ϵ
a	Active set
f	Final value
lb	Lower bound
$p1$	First order prediction
$p2$	Second order prediction
r	Radial
s	Scaled quantity
th	Tangential
ub	Upper bound
w	Working set
x	Derivative with respect to x

Zusammenfassung

Die Beschreibung von technischen Fragestellungen in Form von Optimalsteuerungsproblemen wird zunehmend zu einem populären Werkzeug. Dabei zeigt sich, dass die Aufbereitung und Auswertung der gefundenen optimalen Lösung eine Schlüsselkompetenz darstellt. Die vorgelegte Arbeit setzt hier an. Mittels post-optimaler Analyse wird eine zuvor ermittelte optimale Lösung so weiterverarbeitet, dass der Anwender auf effizientem Wege zusätzliche Daten über Optimalitätsverhalten, Einflussfaktoren und, in weiteren Schritten, über benachbarte Lösungsräume erhält.

Der entwicklte Algorithmus nutzt Verfahren aus der Nichtlinearen Optimierung zur Transkription des Optimalsteuerungsproblems. Dieser Vorgang ist insofern generisch gehalten, als verschiedene Parametrisierungsschemata der Kollokation, wie auch der Mehrzielverfahren implementiert sind. Eine darauf aufbauende parametrisierte Sensitivitätsanalyse dient als Grundlage für die eingehende post-optimale Analyse. Für die Untersuchungen erster und zweiter Ordnung wird zur Effizienzsteigerung und Erhöhung der Transparenz, soweit möglich, auf bereits vorhandene Basisdaten zurückgegriffen. Nicht verfügbare Daten, wie etwa Hesse-Matrix, werden neu berechnet.

Der entwickelte Algorithmus identifiziert Sensitivitäten der Kostenfunktion innerhalb der bestehenden Problembeschreibung und erlaubt darüber hinaus Aussagen zur Beeinflussung des Lösungsraumes durch die Variation nicht optimierbarer Parameter bzw. Gleichungs- oder auch Ungleichungsbeschränkungen.
Letztere Fähigkeit schließt die Prädiktion des Einflusses von finiten Variationen auf die Zusammensetzung des Set der aktiven Beschränkungen ein. Daraus wird dann unter Einschluß von Optimalitätsaspekten und Angabe über den Vertrauensbereich, die Lösung benachbarter Entwurfsräume vorhergesagt.
Damit ist die post-optimale Analyse nicht nur zur besseren Bewertung bereits ermittelter Lösungen geeignet, sondern bietet auch eine effiziente Alternative zur konsekutiven Optimierung, variierter Problemstellungen.

Um den Nutzen der post-optimalen Analyse für umfassende, moderne Anwendungen zu demonstrieren, wird der neue Algorithmus auf zwei typische Probleme aus der Raumfahrt angewendet. Es handelt sich um die optimale Wiedereintrittsbahn des wiederverwendbaren Transportfahrzeuges HOPPER, sowie um optimale Aufstiegsbahnen einer Ariane 5 bei unterschiedlichen Nutzlast-Konfigurationen.
Dabei werden Robustheit und Güte der bisherigen Entwürfe bewertet und das Verbesserungspotential quantifiziert. Außerdem wird demonstriert, welchen zusätzlichen Nutzen der neu entwickelte Algorithmus zur post-optimalen Analyse im Bereich von Optimalsteurungsproblemen hat.

Summary

The description of technical problems in the form of optimal control problems is becoming increasingly popular. And the assessment and evaluation of the obtained optimal solutions is developing into a key competence. This dissertation departs from such an optimal solution. By means of post-optimality analysis an earlier obtained result is processed in order to allow a user to efficiently produce information about optimality criteria, the main factors that influence the solution and, in consecutive steps also about neighboring solution spaces.

The developed algorithm exploits methods from the domain of nonlinear optimization for the transcription of the optimal control problem. This procedure is made generic in the sense that different parameterization schemes are incorporated, ranging from collocation to multiple shooting. The basis of consecutive post-optimality analysis is a parameterized sensitivity analysis. First and second order evaluation takes efficiently advantage of data that has already been computed during the process of the prior optimization. Data that is not readily available, like the Hesse matrix, is newly computed.

The developed algorithm identifies sensitivities of the cost function within the existing problem description, but also permits to investigate the influence of variations in non-optimizable parameters, equality or inequality constraints on the solution space. The latter includes the prediction of changes in the active set of constraints due to finite variations. Under full consideration of optimality conditions and the trust radius, the solution of neighboring problems is predicted. Thus, post-optimality analysis is not only suitable for evaluation of already computed optimal solutions. It also provides an efficient alternative to consecutive optimization of varied problem descriptions.

In order to demonstrate the benefits of post-optimality analysis for modern comprehensive problems, the algorithm is applied to two typical aerospace problems. These are an optimal entry of the reusable launch vehicle HOPPER, and ascent trajectories of Ariane 5 with several payload configurations. Robustness and cost quality of the current design is evaluated and potential for improvement quantified. Further, the added value is demonstrated that the newly developed algorithm for post-optimality analysis provides in the area of optimal control problems.

Chapter 1

Introduction

> *He who has knowledge does not predict. He who predicts does not have knowledge.*
>
> *Chinese teaching*

This statement was supposedly made by ancient Chinese philosopher Lao-Tse more than two thousand years ago. It is a popular phrase, variously quoted whenever people are encouraged to question the professionalism and qualification of self-promoted leaders and specialists. It essentially distinguishes between knowledge and prediction and, thus, suggests that there is an either-or. A proper understanding of cause and effect guarantees knowledge while a lack of this understanding triggers prediction. The connotation of prediction is undoubtedly negative, since knowledge is generally considered to be a positive quality.

Today, in research and engineering it becomes more and more important to combine knowledge and prediction in order to improve model understanding and accelerate product development.

Post-optimality analysis (POA) is seen as such a hybrid. It is a technique to study the behavior of an earlier obtained problem solution and helps to gather information about its sensitivity. At the same time, it also suggests an interpretation of the state space around the solution providing stability information. In other words, it serves to predict the solution of perturbed problems without need for recomputation.

Following, a definition is given for optimal control problems to familiarize the reader with this particular class of problems, which is in the focus of our efforts

to develop an expertise in post-optimality analysis. The role and importance of Sensitivity Analysis (SA) is briefly addressed in sec. 1.2 together with a synopsis of common methods. Then, the basic aspects of post-optimality analysis and prior contributions by other authors are given.
Considerations on the requirements for post-optimality analysis and applicable methods conclude the introduction and define the strategy.

1.1 The Optimal Control Problem

Problem formulations of dynamic systems with optimizable parameters and controls have a long lasting tradition in aerospace engineering. This class of Optimal Control Problems (OCP), has found its entry into flight maneuver optimization [42], launcher and return vehicle trajectory optimization [35], [2], [16], [63], satellite transfer [23], [18] and interplanetary travel [24]. The community of practitioners has also spread into other branches, like the automotive sector [20] and medicine [26], and has created a growing interest in solution and analysis methods.
Model complexity and the need for efficiency have raised the interest in methods to compute optimal solutions. They are frequently the only chance to bring about improvements towards enhanced performance.

To fully immerse into the topic, the proper mathematical formulation of optimal control problems is given in the following section. Afterwards, the value of their solution is assessed with regard to practical usefulness.

1.1.1 Mathematical Description

The optimal control problem is detailed in a large number of publications [15], [31], [8]. In order to provide a concise nomenclature and to allow fundamental understanding of the later chapters, a basic description is given here.
The control problem for which an optimal solution shall be computed, is as follows: minimize the cost functional

$$min\ J(x,u,p,t) = \Phi(x_f,p,t_f) + \int_{t_0}^{t_f} L(x,u,p,t)dt. \tag{1.1.1}$$

The objective function is stated in Bolza format with Φ being the Mayer term and a Lagrange term with the integrand L. The vector $\mathbf{u} = \mathbf{u}(t)$ represents the time-variant control vector and t represents the independent variable. The state vector of the system is given as $\mathbf{x} = \mathbf{x}(t)$ and has the dynamics

$$\dot{\mathbf{x}} = f(x(t),u(t),p,t) \tag{1.1.2}$$

It is often convenient to assign time-invariant parameters $\mathbf{p}$, which describe certain system properties. Commonly, they are design parameters or related qualities of the model, and mission, respectively. Their value is optimizable, but does not change over time.

Additional conditions for the system are stated as path constraints. These can be defined as equality constraints

$$\mathbf{h}(x(t), u(t), p, t) = 0 \tag{1.1.3}$$

or as inequality constraints

$$\mathbf{g}(x(t), u(t), p, t) \geq 0. \tag{1.1.4}$$

The same holds for boundary constraints which can be equality

$$\mathbf{\Psi}_t = \mathbf{\Psi}_t\left(x(t), u(t), p, t\right) = 0 \tag{1.1.5}$$

or inequality constraints

$$\mathbf{\Psi}_t = \mathbf{\Psi}_t\left(x(t), u(t), p, t\right) \geq 0 \tag{1.1.6}$$

under the condition that either $t = t_0$ or $t = t_f$.

The classical way to solve such an optimal control problem are indirect methods. They are based on the calculus of variations.

The first formulation of first order necessary conditions for optimal control problems was published by Euler and Lagrange in 1744. The Euler–Lagrange equations compose the fundament for the solution of this kind of mathematical problem. In the time since, scientists have refined the formulations and have extended their use. The introduction of the Hamilton function in 1834/35, for instance, was a major contribution to improve the analytic structure of the conditions.

The first order necessary conditions for a problem with no path constraints, with terminal equality constraints and mixed initial constraints can be found in eqs. 1.1.7.

$$
\begin{aligned}
&\text{Hamiltonian:} && H = L + \lambda^T f \\
&\text{Dynamics:} && \dot{\mathbf{x}} = f(x,u,t) = \tfrac{\partial H}{\partial \lambda} \\
&\text{Adjoint differential equations:} && \dot{\lambda} = -\tfrac{\partial L}{\partial \mathbf{x}} - \left(\tfrac{\partial f}{\partial \mathbf{x}}\right)^T \lambda = -\tfrac{\partial H}{\partial \mathbf{x}} \\
&\text{Optimality condition:} && 0 = \tfrac{\partial H}{\partial \mathbf{u}} = \tfrac{\partial L}{\partial \mathbf{u}} + \left(\tfrac{\partial f}{\partial \mathbf{u}}\right)^T \lambda \\
&\text{Initial conditions:} && \mathbf{x}(t_o) \text{ given or } \lambda(t_0) = 0 \\
&\text{Terminal constraints:} && \boldsymbol{\Psi}_f = \boldsymbol{\Psi}_f(x(t_f), t_f) = 0 \\
&\text{Transversality conditions:} && \lambda_f = \left[\tfrac{\partial \Phi}{\partial \mathbf{x}} + \left(\tfrac{\partial \Psi_f}{\partial \mathbf{x}}\right)^T \nu\right]_{t=t_f} \\
&\text{Transvers. for optimizable } t_f\text{:} && \Omega = \left[\tfrac{\partial \Phi}{\partial t} + \nu^T \tfrac{\partial \Psi_f}{\partial t} + H\right]_{t=t_f} = 0
\end{aligned}
\tag{1.1.7}
$$

The Euler–Lagrange equations are only of first order and do not formulate sufficient conditions. Therefore extensive research has been undertaken to complete and extend them. The Legendre–Clebsch condition, demanding

$$
\frac{\partial^2 H}{\partial \mathbf{u}^2} \geq 0 \tag{1.1.8}
$$

is an example for a necessary condition of second order.

It was the Russian mathematician Pontryagin in 1954, who extended the optimal control theory to cases with constrained variables. This was in so far an important contribution as it enabled the optimization of problems with path constraints, which are a common element in engineering problems. A detailed discussion of additional conditions can be found in [15].

Indirect methods have the potential to provide closed solutions for OCPs. They work with exact analytical terms and solve the problem via an intermediate elimination of the control and later back-calculation of the optimal control history. This technique has also been the name giver for the class of indirect methods.
The very attractive features of indirect methods are paid for by enormous mathematical overhead. Application of indirect methods requires a deep understanding

of the mathematical problem structure and extensive knowledge about its solution space. The structure of the equations describing the optimality conditions is very complex. And even small changes in the problem outline can trigger major modifications of, for instance, the Hamiltonian function.
The same holds for the adjoint, or costate, variables [11]. They do not have physical meaning which makes their estimation not at all intuitive and overly time consuming. However, an accurate estimate is essential, since the mathematical problem description is very sensitive to changes in the costate values. And convergence to an optimal solution requires a qualitatively good initial guess.

An alternative to indirect methods has appeared with the emergence of digital computers in the mid of the 20th century -the so called direct methods [46], [39], [9]. As the name suggests, they are straightforward techniques to calculate the optimal solution. The concept is to parameterize the solution space and to solve a constrained nonlinear programming problem. The algorithm ensures compliance with all constraints while in an iterative process closing in on the optimal solution.
The direct methods are numerical and, thus, not as elegant and analytically exact as indirect methods. But they exhibit a series of advantages, which make them the first choice more and more often in practical applications. It is the flexibility, adaptability and robustness that gives these methods a larger convergence radius and makes them well appropriate for users, who do not have an in-depth knowledge of optimization theory.

1.1.2 Understanding an Optimal Solution

It is one thing to describe an optimal control problem and obtain a solution. It is another to fully understand and exploit the found solution. There are numerous aspects that determine the usefulness of the results that an optimization algorithm delivers.

First of all, there is the question about the optimality of the solution with respect to a certain criterion. Is the solution strictly optimal, or is the cost function gradient only gently inclined?
Essentially, optimization algorithms terminate providing an optimal design point with a certain performance number, but lack a comprehensive survey of the design point sensitivity.
Unfortunately, the exact solution is purely theoretical in the eyes of practitioners, since the problem formulation generally describes a simplified model of reality. Hence, a transfer of the results is only reasonable when the behavior of the model is known and understood. Otherwise, reduction of main features of the problem result in erroneous solutions, from which faulty conclusions are drawn.

This motivates the assessment of perturbations in auxiliary design parameters, which could compensate for model short-comings. These parameters typically describe model properties that are held constant during the optimization, even though they are naturally not constrained to a particular value.
This leads to the identification of critical components, the evaluation of uncertainty sources and the question, what effect certain parameters have on the optimal solution.

The matter can be summarized in the term *Sensitivity*. The key to enhanced problem understanding is knowledge about the sensitivity of the optimum solution.

1.2 Sensitivity as a Means of Analysis

Sensitivity analysis has become a main competence for the modeling and analysis of complex systems in general. Blackwell [12] has congregated its meaning in the following definition:

> **Sensitivity analysis is defined as the study how variations in input parameters of a computational model cause variations in output. [...] A measure of this sensitivity is termed the sensitivity coefficient and is (mathematically) defined as a partial derivative of the output variable with respect to the parameter of interest.**

This gives us the mathematical expression

$$\mathcal{S}_{ij} = \frac{\partial y_i}{\partial x_j}. \tag{1.2.9}$$

for the sensitivity coefficient $\mathcal{S}_{ij}$. The scalar variable y_i represents an output of the investigated system. It commonly characterizes the objective and contains key performance properties of the dynamic system.
The scalar variable x_j constitutes an input to the system. Most often it is a design parameter, a model parameter, or a parameter defining a condition.

Another expression which is commonly used for $\mathcal{S}_{ij}$ is sensitivity derivative. It has been introduced by Sobieski [65], [58], [64].

The dependencies illustrated by the sensitivity coefficients provide useful information about the behavior and character of the problem under investigation. Hence, the coefficients are valuable for analysis and can be processed for various tasks. Bose

et al. [13] performs sensitivity analysis to identify uncertainty risk and categorizes as follows:

1. Structural Uncertainty
   All mathematical models work with assumptions and simplifications to represent physical phenomena. These pose the risk of not being realistic.

2. Parametric Uncertainty
   This type of uncertainty arises for uncertainties in the model parameter estimates. This happens frequently for parameters of dynamic systems which can not be measured explicitly, but have to be guessed.

3. Stochastic Uncertainty
   Natural fluctuations can cause this kind of uncertainty. A common example are atmospheric anomalies with stochastic behavior.

There is a multitude of scientific applications which go under the title of sensitivity analysis and document the very heterogeneous perception of its usefulness. Empirical methodologies can be found in environmental model analysis [50], labor market evaluation [62], or in chemistry [69]. It can be used to compute safety or probability margins [27] by widening the parameter range or provide gradient information for an optimization algorithm [55].

The rising interest in sensitivity computation in recent years has led to the development of a number of different methods. In general the selection of the most suitable method is dominated by the structure of the model and its accessibility. Therefore model properties and computational interests can be taken to broadly classify the various analysis methods.

If the model is completely unknown and the model equations are not accessible or if dependencies shall be scanned for a wide variational range, then sampling methods promise to show best performance. The relationship between input and output parameters in the state space is established empirically via model runs at sets of sampling points [51].
In this context, simulation campaigns are a widely used means for uncertainty and sensitivity analysis [48]. Telaar [68] has worked with them to identify sensitivities of a reentry vehicle. And Bose et al. [13] applied the concept to compute sensitivities in the thermochemical model for a Titan atmospheric entry.
The selection scheme for the sampling sets distinguishes the various sampling methods from each other and defines the computational expenses. This ranges from Monte Carlo-like approaches with almost random distribution, to Latin Hypercube

Sampling [66] with equal probability segments. Also Response Surface methods are in use. They process the obtained sampling information towards the definition of a secondary model with reduced parameter number [72]. The surrogate model is convenient for trend analysis and allows rapid, but also inaccurate sensitivity derivation.

Other methods are directly focused on producing derivatives. There are the analytic methods, which rely on the accessibility and differentiability of the model equations. One such technique compiles the Forward Sensitivity Equations. Another is the Reverse Adjoint Equations method. As the name suggests, it analyzes the origin of an anomaly by means of reverse signal flow [36] and, thus, allows an identification of perturbation sources. In [55] it is shown, how this method can be used to provide sensitivity data for aero-structural optimization.

All these analytic methods provide exact gradients, but require an enormous mathematical overhead. Particularly in engineering applications it is frequently the case that spreadsheets and switching functions are part of the model and make analytical techniques a prohibitively expensive or even impossible task.
Then, numerical methods become a convenient alternative. Finite differences are generally the prime choice for numerical differentiation. But Martins et al. [56] has also successfully tested the method of Complex Steps for sensitivity analysis purposes. The accuracy is superior. However, complex variation of the model requires a comprehensive complex algebra environment.
In any case, the sensitivity coefficients are computed for a certain reference point and therefore local.

Another method is Parameterized Sensitivity Analysis (PSA). The fundamental idea behind this is to reduce the model size by parameterization, while retaining its complexity. It is often possible to make reasonable assumptions for a model and establish parametric relationships as a substitute to function dependencies, for instance, through use of polynomial fitting. The infinite number of state propagators is approximated with a limited number of parameters, which subsequently constitute the parameter space of the new problem description.
The computation of sensitivities is turned into the computation of parameter dependencies. And the chain of such dependencies determines the impact of an input variation on a specific output. Each parameter which correlates to the input of interest potentially stimulates the output. Fiacco [28] and Sobieski [65] have provided valuable contributions for the advancement of PSA.

1.3 The Question of Solution Analysis in Optimization

Having surveyed a number of different methods, the question arises of how sensitivity analysis in practice can help to better understand solutions to optimal control problems, fulfilling the demands postulated in sec. 1.1.2.

Sensitivity analysis is often employed to examine natural system behavior. Especially, sampling methods are a popular tool to study open loop systems. But in post-optimality analysis the task is to assess optimal behavior of constrained dynamic systems. Therefore it is a prerequisite that the analysis algorithm complies with optimality conditions.
This specifically addresses the sensitivity of the optimal solution to infinitesimal perturbations in all kinds of design parameters. The type of output enables researchers and engineers to qualify the nominal design with respect to its actual application.

The spreading use of optimization methods also substantiates other needs. In design optimization and system development the focus shifts from infinitesimal perturbations to finite variations. This leads to an assessment of the design space in the vicinity of the optimality point for properties, which are genuinely not part of the optimizable parameters.
Such an analysis requires not only sensitivity information about the nominal solution. But it needs a measure for the reliability of the data in the broader neighborhood.

1.4 Previous Works on Optimal Solution Analysis

As said before, the growing interest in optimization techniques and the raised awareness for the benefits of solution analysis have led to a number of scientific contributions and to the question of how and for what exactly to conduct post-optimaliy analysis.
One of the foundations has been laid down by Fiacco in his "Introduction to Sensitivity and Stability Analysis in Nonlinear Programming" [28], with which he tries to foster the theoretical unification and practical implementation of such methods. He is well aware of the importance of post-optimality analysis in general and concentrates on the analysis of NLP programs. In his view NLP can only then become a widely used method, if practitioners get the necessary algorithms at hand to evaluate the computed optimality.
Bose et al. [13] follows a simple approach. He investigates the uncertainty sources in a thermochemical model for a Titan atmospheric entry by running a plain simulation campaign for the constrained trajectory with both local and global interests.

From the results the input-output correlations are identified and uncertainties in the model ranked with respect to their influence. Bose is aware of the deficiencies of this method. He describes it as somehow subjective and limited to the nominal trajectory. The found ranking and the "*conclusions drawn here may not be valid if mission specifications differ in terms of entry velocity, vehicle size, etc. Each candidate mission must, therefore, be analyzed individually*" [13]. However, Bose gives a good reasoning for his efforts, which also helps us to understand the charm of post-optimality analysis. He uses the sensitivities to evaluate uncertainty confidence limits and to identify target areas, where further research effort would grant maximum payoff.
The preparatory work by Fiacco, gathering the scattered contributions to sensitivity analysis, has been taken up by practicioners and has led to various implementations. Sobieski [64], for instance, calls it System Sensitivity Analysis. From the early 80's till the end of the 90's he advances the technique. First focusing on structural optimization, his interest successively moves towards aerodynamic applications and he utilizes the method in MDO.
Olds picks this up and does a conceptual design of a dual-fuel SSTO rocket [58] based on sensitivity coefficients. His algorithm requires extensive reoptimization and falls short of exploiting the available optimality information. Therefore in his remarks he promotes the idea of proper post-optimality analysis in the future, which takes advantage of secondary output of the optimizer.

An extensive investigation of parameterized sensitivity analysis for launch vehicle sizing and trajectory design has been conducted by Hallmann [44], [45] with contributions by Beltracchi [6]. Hallman's publications on the analysis of launcher cases has provided fundamental insights into the mechanisms of NLP algorithm interconnection with PSA. He presents first and second order sensitivity analysis for design parameters of the nonlinear programming problem and identifies trends for parameter deviations.
As an advanced practical example for the capability of the developed software Hallman presents a booster/upper stage optimum trajectory design. The case demonstrates the good performance of the algorithm, but also witnesses the limitations of the concept. With a total of 17 variables the problem is small and documents the care with which the NLP problem needs to be set up in order to allow for sensitivity analysis. Besides, the data interpretation is very conservative and does not provide confidence margins.

In recent years the focus of the scientific community has been on the utilization of simple methods, customized for specific needs [68] and the theoretical deepening of the insight in sensitivity analysis [17]. The interpretation of the sensitivity data has remained basic and stayed limited to neighboring designs with similar properties.

1.5 Design of a Post-Optimality Analysis Algorithm

The development of a post-optimality analysis algorithm is motivated by the aspiration to give practitioners an additional means at hand to study, interpret and, ultimately, to better exploit the before computed solution of an optimal control problem.

The intention is to establish a highly efficient analysis procedure, which is generic and, hence, suitable for a broad variety of applications. Under consideration of the optimizer output, it shall first provide processable optimality information about the computed solution.
Further, it is the ambition to use the algorithm to identify decisive or critical parameters. This represents a reverse variation assessment to determine the variation's cause.

Forward perturbation analysis constitutes a fundamental capability. It targets at evaluating the influence of non-optimizable parameters on predefined model outputs and extends optimality information to parameters, which have not been part of the initial control problem description.

Optimal control problems in general engineering tend to be set up inefficiently with inferior condition and restricted access to the dynamic equations. This needs to be considered for the development of a heuristic post-optimality analysis algorithm in order to permit an economic evaluation even of large problem descriptions.

An added value arises when the analysis information can also be used to interpret the broader vicinity around the optimum design point. This means the assessment of larger variations. Among a series of benefits does this feature permit to derive optimal performance of neighboring problems.

The strategy to develop a post-optimality analysis algorithm that meets the postulated requirements has to conform with optimality conditions even for perturbed problems. Therefore the link between optimal control theory and sensitivity gradients of first and second order [28] under inclusion of optimality conditions and constraints is established in chapter 2. The concept is to utilize parameterized sensitivity analysis for the purpose of computing variational data. In chapter 3 the algorithm is extended to update the constraint set and compute confidence radii for the extrapolation with sensitivity gradients.
In chapter 4 the developed concept for post-optimality analysis is mated with the optimizer CAMTOS [31]. The primary aspects of algorithm interconnection are described.

In order to demonstrate and validate performance and added value of the method, two applications are included. The analysis of an optimal HOPPER reentry trajectory resembles the first case in chapter 5. An assessment of the launcher and optimal trajectory design of an Ariane 5 in the light of sensitivity analysis is described in chapter 6.
Conclusions and an outlook on further potential use of the method ends the thesis and gives a perspective for consecutive research activities.

Chapter 2

Sensitivity Analysis of Optimal Control Applications

A brief introduction to sensitivity analysis was given in the first chapter and the general usefulness illustrated. The various methods can serve a wide range of purposes. In any way, they help to correlate input and output variations and provide statistical information.
It has also been shown, how this data can help to better understand various mathematical problems. In this thesis, emphasis is put on the assessment of solutions to optimal control problems and their design space, for instance, by evaluating the stability and sensitivity of the variable set.
The necessary connection between sensitivity analysis and optimal control theory comprises the content of this chapter. At first, a short discussion explains the selection of PSA for sensitivity computations. The name of the method itself already reveals one of its key properties. It only works for parameterized problems. Hence, the continuous optimal control problem needs to be transcribed into a parameterized problem, namely a Nonlinear Programming NLP problem as described in sec. 2.1. Following, first and second order sensitivity equations are derived.
The method is more thoroughly studied and applied to a simplified satellite transfer in sec. 2.2. Considerations on problem scaling in sec. 2.3 and on transcription schemes (see sec. 2.4) conclude the chapter.

2.1 The Parameterized Sensitivity Equations

The urge for optimization expertise in today's engineering environment is more and more dominated by the benefits of ever growing computing power. On one hand, it enables the practitioner to design more realistic models for use in simulation and optimization. But on the other hand, it is seductive to compensate poor problem assessment with brute force.

The rapid development of NLP algorithms [8], [39], [29], [70] and computer hardware allows the optimization of larger and more complex problems. The inclusion of nonlinear constraints into the nonlinear programs is a routine exercise today. And the successful handling of more than 100 000 degrees of freedom is a proven quality [10] which underlines the capabilities of the modern algorithms. Their reliability and adaptability motivates the spreading use in research and development.

Along with the growing complexity of the analyzed models, it becomes prohibitively expensive to apply analytical methods to examine model behavior. This is even truer for optimization problems in aerospace engineering, where encapsulated modules do not even permit access to the governing equations.

It was Fiacco, who gave a boost to parameterized sensitivity analysis when he published his book on sensitivity and stability analysis [28]. Gathering the scattered knowledge and consolidating the contributions in a survey-like manner, he made them accessible to a broad audience. And the growth of the NLP community makes them even more attractive today. Thats why we further on concentrate on PSA for sensitivity computation and why former authors [28], [44], [65] vigorously promoted the method.

2.1.1 Nonlinear Programming Problem

The first step is to describe the NLP program. The subsequent definition for nonlinear programming problems follows common nomenclature as presented in [8], [29], [28]. The cost function

$$f = f(x, \epsilon) \tag{2.1.1}$$

shall be minimized with respect to the parameter vector $\mathbf{x} \in \Re^n$. This vector basically comprises all parameters which are optimizable. We also include the vector $\epsilon \in \Re^p$ in the problem description. It contains a set of non-optimizable parameters and will later be required for sensitivity computations.
The solution is subject to the equality constraints

$$\mathbf{h}(x, \epsilon) = 0, \text{ with } \mathbf{h} \in \Re^{m_e} \tag{2.1.2}$$

and the inequality constraints

$$\mathbf{g}(x, \epsilon) \geq 0, \text{ with } \mathbf{g} \in \Re^{(m-m_e)}. \tag{2.1.3}$$

It is desirable to develop necessary and sufficient optimality conditions. This is frequently accomplished via a transformation of the constrained into an unconstrained problem [30]. One such transformation works using the Lagrange function

$$\mathcal{L}(x, \lambda, \mu, \epsilon) = f(x, \epsilon) - \lambda^T \cdot \mathbf{g}(x, \epsilon) + \mu^T \cdot \mathbf{h}(x, \epsilon) \tag{2.1.4}$$

with the Lagrange multipliers $\lambda \in \Re^{(m-m_e)}$ being associated to the inequality and $\mu \in \Re^{m_e}$ to the equality constraints.

The stationary point $\mathbf{x}^*$ is a local minimum, if there exist λ^* and μ^* such that the first order Karush-Kuhn-Tucker KKT conditions hold:

$$\begin{aligned}
\mathbf{g}(x^*, 0) &\geq 0 && (2.1.5)\\
\mathbf{h}(x^*, 0) &= 0 && (2.1.6)\\
\lambda_i^* g_i(x^*, 0) &= 0, \quad i \in E^{m-m_e} && (2.1.7)\\
\lambda_i^* &\geq 0, \quad i \in E^{m-m_e} && (2.1.8)\\
\mathcal{L}_x(x^*, \lambda^*, \mu^*, 0) &= f_x(x^*, 0) - \left(\mathbf{g}_x(x^*, 0)\right)^T \lambda^* + \left(\mathbf{h}_x(x^*, 0)\right)^T \mu^* && (2.1.9)\\
&= 0
\end{aligned}$$

In practice, the optimal solution does not always satisfy these conditions [29]. Hence, NLP algorithms usually either terminate when a KKT point

$$(\mathbf{y}^*)^T = \left[(\mathbf{x}^*)^T \, (\lambda^*)^T (\mu^*)^{\mathbf{T}}\right] \tag{2.1.10}$$

is reached or no further progress appears to be possible.
This is important to remember. As soon as progress falls below a certain tolerance level -a threshold- the optimization is stopped. The obtained solution is presumably close to the true optimum. However, this depends on the tolerance level and the convergence behavior of the problem.

Whereas the cost function usually shows good convergence and is close to the actual optimum, this does not necessarily hold for all elements of the KKT point vector. Particularly, the Lagrange multipliers tend to be very sensitive in the neighborhood of the optimal solution and, hence, are often inaccurate [57]. The influence of these deviations on the quality of our sensitivity analysis will be addressed in sec. 4.4.

2.1.2 Derivation of First Order Sensitivity Equations

The definition of the NLP problem is taken to derive sensitivity equations. The aim is to find relationships between variations in the sensitivity parameters ϵ and variations in the NLP parameters $\mathbf{x}$. Two premises hold for these sensitivity equations. They shall not work with assumptions and reductions to avoid unnecessary inaccuracies. And the equations shall provide straight forward relationships which can easily be computed. Fiacco [28] describes a differentiation method to obtain these equations. We follow his suggestion and differentiate eq. 2.1.9 with respect to the parameter vector ϵ:

$$\frac{\partial^2 f}{\partial \mathbf{x} \partial \epsilon} + \frac{\partial^2 f}{\partial \mathbf{x^2}} \frac{\partial \mathbf{x}}{\partial \epsilon} - \sum_{i=m_e+1}^{m} \left[\lambda_i \left[\frac{\partial^2 g_i}{\partial \mathbf{x} \partial \epsilon} + \frac{\partial^2 g_i}{\partial \mathbf{x^2}} \frac{\partial \mathbf{x}}{\partial \epsilon} \right] \right] \tag{2.1.11}$$

$$- \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right]^T \frac{\partial \lambda}{\partial \epsilon} + \sum_{j=1}^{m_e} \left[\mu_j \left[\frac{\partial^2 h_j}{\partial \mathbf{x} \partial \epsilon} + \frac{\partial^2 h_j}{\partial \mathbf{x^2}} \frac{\partial \mathbf{x}}{\partial \epsilon} \right] \right] + \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right]^T \frac{\partial \mu}{\partial \epsilon} = 0.$$

We proceed in the same manner with eq. 2.1.7

$$\lambda_i \left[\frac{\partial g_i}{\partial \epsilon} + \frac{\partial g_i}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \epsilon} \right] + g_i \frac{\partial \lambda_i}{\partial \epsilon} = 0 \tag{2.1.12}$$

with $i \in E^{m-m_e}$ and eq. 2.1.6

$$\frac{\partial \mathbf{h}}{\partial \epsilon} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \epsilon} = 0. \tag{2.1.13}$$

The vector ϵ comprises the non-optimizable sensitivity parameters. They are in the center of attention of the sensitivity analysis, since it is their influence on the optimal solution, which is supposed to be examined.
The first equation of the system can be reshaped as follows:

$$\frac{\partial^2 f}{\partial \mathbf{x^2}} \frac{\partial \mathbf{x}}{\partial \epsilon} - \sum_{i=m_e+1}^{m} \left[\lambda_i \frac{\partial^2 g_i}{\partial \mathbf{x^2}} \frac{\partial \mathbf{x}}{\partial \epsilon} \right] + \sum_{j=1}^{m_e} \left[\mu_j \frac{\partial^2 h_j}{\partial \mathbf{x^2}} \frac{\partial \mathbf{x}}{\partial \epsilon} \right] + \frac{\partial^2 f}{\partial \mathbf{x} \partial \epsilon} \tag{2.1.14}$$

$$- \sum_{i=m_e+1}^{m} \left[\frac{\partial^2 g_i}{\partial \mathbf{x} \partial \epsilon} \right] + \sum_{j=1}^{m_e} \left[\mu_j \frac{\partial^2 h_j}{\partial \mathbf{x} \partial \epsilon} \right] - \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right]^T \frac{\partial \lambda}{\partial \epsilon} + \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right]^T \frac{\partial \mu}{\partial \epsilon} = 0$$

which is equal to

$$\nabla_{xx} \mathcal{L} \cdot \frac{\partial \mathbf{x}}{\partial \epsilon} + \nabla_{x\epsilon} \mathcal{L} - \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right]^T \frac{\partial \lambda}{\partial \epsilon} + \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right]^T \frac{\partial \mu}{\partial \epsilon} = 0 \tag{2.1.15}$$

It can be reformulated in matrix form

$$\begin{bmatrix} \nabla_{xx}\mathcal{L} & -\left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right]^T & \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]^T \\ \lambda_{\mathbf{1}}\left(\frac{\partial \mathbf{g_1}}{\partial \mathbf{x}}\right)^{\mathbf{T}} \; \ddots & & 0 \\ \vdots & diag(g) & \vdots \\ \lambda_{\mathbf{q}}\left(\frac{\partial \mathbf{g_q}}{\partial \mathbf{x}}\right)^{\mathbf{T}} & \ddots \; 0 & \\ \frac{\partial \mathbf{h}}{\partial \mathbf{x}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \epsilon} \\ \left(\frac{\partial \lambda_1}{\partial \epsilon}\right)^T \\ \vdots \\ \left(\frac{\partial \lambda_q}{\partial \epsilon}\right)^T \\ \frac{\partial \mu}{\partial \epsilon} \end{bmatrix} = \begin{bmatrix} -\nabla_{x\epsilon}\mathcal{L} \\ -\lambda_1\left(\frac{\partial g_1}{\partial \epsilon}\right)^T \\ \vdots \\ -\lambda_q\left(\frac{\partial g_q}{\partial \epsilon}\right)^T \\ -\frac{\partial \mathbf{h}}{\partial \epsilon} \end{bmatrix} \tag{2.1.16}$$

with $q = m - m_e$. Or in short form

$$\mathbf{M} \cdot \nabla_\epsilon \mathbf{y} = \mathbf{N}, \quad \textbf{with } \mathbf{y^T} = \left[\mathbf{x^T} \; \lambda^{\mathbf{T}} \; \mu^{\mathbf{T}}\right] \tag{2.1.17}$$

If matrix $\mathbf{M}$ is invertible, then the linear system can be solved. A solution for the partial derivative of $\mathbf{x}$ with respect to ϵ is then available for further analysis. For instance, it can be used to perform a first order assessment of the parameter values of a perturbed problem. This means, after having obtained an optimal solution for an NLP program, this solution along with the sensitivity coefficients can be taken to predict the NLP parameter values of a derived problem with perturbations in the elements of vector ϵ.

This statement is not entirely correct and reliable in its generalization. Curtailing conditions will be described in detail in chapter 3. Until then, sensitivity information it taken as it is and used to compute first order propagations. Therefore, we perform a Taylor expansion for the cost function and the NLP parameters. The first order system

$$\mathbf{x}^{p1}(\epsilon + \Delta\epsilon) = \mathbf{x}^*(\epsilon) + \frac{\partial \mathbf{x}}{\partial \epsilon}\Delta\epsilon \tag{2.1.18}$$

$$f^{p1}(x(\epsilon + \Delta\epsilon), \epsilon + \Delta\epsilon) = f^*(x^*, \epsilon) + \left(\frac{\partial f}{\partial \epsilon}\right)^T \Delta\epsilon + \left(\frac{\partial f}{\partial \mathbf{x}}\right)^T \frac{\partial \mathbf{x}}{\partial \epsilon}\Delta\epsilon \tag{2.1.19}$$

delivers first order predictions for the optimal cost of the perturbed problem f^{p1}.

2.1.3 Second Order Sensitivity Coefficients

With sensitivity gradients at hand it is possible to approximate the optimal behavior of the dynamics with linear terms. Even though linearization techniques are common practice and popular in many fields, the shortcomings are obvious, since variable dependencies are in general not linear. This limits the range for which the linearization represents a reliable approximation.
Second order sensitivity coefficients promise to further improve expressiveness of a post-optimality analysis by giving indications about the degree of non-linearity in a parameter relationship and its extrapolatory trend.

Consider the Lagrange function eq. 2.1.4 and differentiate with respect to ϵ

$$\frac{\partial \mathcal{L}}{\partial \epsilon} = \frac{\partial f}{\partial \epsilon} + \left[\frac{\partial \mathbf{x}}{\partial \epsilon}\right]^T \frac{\partial f}{\partial \mathbf{x}} - \left[\frac{\partial \mathbf{g}}{\partial \epsilon} + \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \epsilon}\right]^T \lambda + \left[\frac{\partial \mathbf{h}}{\partial \epsilon} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \epsilon}\right]^T \mu \quad (2.1.20)$$

It is derived a second time with respect to ϵ. The result is

$$\begin{aligned}\frac{d^2\mathcal{L}}{d\epsilon^2} = {} & \frac{\partial^2 f}{\partial \epsilon^{\mathbf{2}}} + \frac{\partial^2 f}{\partial \epsilon \partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \epsilon} + \left[\frac{\partial^2 f}{\partial \mathbf{x}\partial \epsilon} + \frac{\partial^2 f}{\partial \mathbf{x}^{\mathbf{2}}}\frac{\partial \mathbf{x}}{\partial \epsilon}\right]^T \frac{\partial \mathbf{x}}{\partial \epsilon} + \left[\frac{\partial f}{\partial \mathbf{x}}\right]^T \left[\frac{\partial^2 \mathbf{x}}{\partial \epsilon^{\mathbf{2}}}\right] \\ & - \sum_{j=m_e+1}^{m} \left[\lambda_j \left[\frac{\partial^2 \mathbf{g}_j}{\partial \epsilon^{\mathbf{2}}} + \frac{\partial^2 \mathbf{g}_j}{\partial \epsilon \partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \epsilon} + \left[\frac{\partial^2 \mathbf{g}_j}{\partial \mathbf{x}\partial \epsilon} + \frac{\partial^2 \mathbf{g}_j}{\partial \mathbf{x}^{\mathbf{2}}}\frac{\partial \mathbf{x}}{\partial \epsilon}\right]^T \frac{\partial \mathbf{x}}{\partial \epsilon}\right.\right. \\ & \left.\left. + \left[\frac{\partial \mathbf{g}_j}{\partial \mathbf{x}}\right]^T \frac{\partial^2 \mathbf{x}}{\partial \epsilon^{\mathbf{2}}}\right]\right] + \sum_{i=1}^{m_e} \left[\mu_i \left[\frac{\partial^2 \mathbf{h}_i}{\partial \epsilon^{\mathbf{2}}} + \frac{\partial^2 \mathbf{h}_i}{\partial \epsilon \partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \epsilon}\right.\right. \\ & \left.\left. + \left[\frac{\partial^2 \mathbf{h}_i}{\partial \mathbf{x}\partial \epsilon} + \frac{\partial^2 \mathbf{h}_i}{\partial \mathbf{x}^{\mathbf{2}}}\frac{\partial \mathbf{x}}{\partial \epsilon}\right]^T \frac{\partial \mathbf{x}}{\partial \epsilon} + \left[\frac{\partial \mathbf{h}_i}{\partial \mathbf{x}}\right]^T \frac{\partial^2 \mathbf{x}}{\partial \epsilon^{\mathbf{2}}}\right]\right].\end{aligned} \quad (2.1.21)$$

Which can be reformulated as

$$\begin{aligned}\frac{d^2\mathcal{L}}{d\epsilon^2} = \frac{d^2 f}{d\epsilon^2} = {} & \begin{bmatrix} \left(\frac{\partial \mathbf{x}}{\partial \epsilon}\right)^T & I_p \end{bmatrix} \begin{bmatrix} \frac{\partial^2 L}{\partial \mathbf{x}^{\mathbf{2}}} & \frac{\partial^2 L}{\partial \mathbf{x}\partial \epsilon} \\ \frac{\partial^2 L}{\partial \epsilon \partial \mathbf{x}} & \frac{\partial^2 L}{\partial \epsilon^{\mathbf{2}}} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \epsilon} \\ I_p \end{bmatrix} \\ & + \left[\frac{\partial f}{\partial \mathbf{x}}\right]^T \frac{\partial^2 \mathbf{x}}{\partial \epsilon^{\mathbf{2}}} - \sum_{j=m_e+1}^{m} \left[\lambda_j \left[\frac{\partial g_j}{\partial \mathbf{x}}\right]^T \frac{\partial^2 \mathbf{x}}{\partial \epsilon^{\mathbf{2}}}\right]\end{aligned} \quad (2.1.22)$$

$$+\sum_{i=1}^{m_e}\left[\mu_i\left[\frac{\partial h_i}{\partial \mathbf{x}}\right]^T\frac{\partial^2\mathbf{x}}{\partial\epsilon^{\mathbf{2}}}\right]$$

with I_p being the p-dimensional identity matrix, corresponding to the dimension of ϵ.

We take a closer look at the last three terms of the above equation and see that they can be rearranged as follows:

$$\left[\frac{\partial f}{\partial \mathbf{x}}-\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^T\lambda+\left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T\mu\right]^T\frac{\partial^2\mathbf{x}}{\partial\epsilon^{\mathbf{2}}}=\left[\frac{\partial\mathcal{L}}{\partial\mathbf{x}}\right]^T\cdot\frac{\partial^2\mathbf{x}}{\partial\epsilon^{\mathbf{2}}} \tag{2.1.23}$$

Since the optimality condition requires $\frac{\partial\mathcal{L}}{\partial\mathbf{x}}$ to be zero, the corresponding terms can be eliminated from eq. 2.1.22. Which leads to

$$\frac{d^2\mathcal{L}}{d\epsilon^2}=\frac{d^2f}{d\epsilon^2}=\begin{bmatrix}\left(\frac{\partial\mathbf{x}}{\partial\epsilon}\right)^T & I_p\end{bmatrix}\begin{bmatrix}\frac{\partial^2L}{\partial\mathbf{x}^{\mathbf{2}}} & \frac{\partial^2L}{\partial\partial\mathbf{x}\epsilon}\\ \frac{\partial^2L}{\partial\partial\epsilon\mathbf{x}} & \frac{\partial^2L}{\partial\epsilon^{\mathbf{2}}}\end{bmatrix}\begin{bmatrix}\frac{\partial\mathbf{x}}{\partial\epsilon}\\ I_p\end{bmatrix} \tag{2.1.24}$$

Thus, second order information about the objective function sensitivity becomes available at the expense of providing Hessian matrix information.
The differentiation of the Lagrange function as performed above takes advantage of the fact that the terms $\left[\frac{\partial\mu}{\partial\epsilon}\right]^T\cdot\mathbf{h}$ are zero for all equality constraints $\mathbf{h}=\mathbf{0}$ at the KKT point. Further, the terms $\left[\frac{\partial\lambda}{\partial\epsilon}\right]^T\cdot\mathbf{g}$ are zero either because of the constraint being active, or, because $\lambda_i=0$ does not depend on any ϵ_j and, thus, $\frac{\partial\lambda_i}{\partial\epsilon}=\mathbf{0}$ at the KKT point.

2.2 Exemplary Analysis of an Optimal Solution

2.2.1 Simplified Earth Mars Transfer

After having deducted the fundamental sensitivity equations it is time to get some hands-on experience by applying the knowledge to a small demonstration case. The exemplary optimization problem that we take a look at is a satellite transfer from earth to mars. It is a simplified mission in a heliocentric system where the satellite departs from a circular orbit with a semimajor axis of exactly one astronomical unit AU. The target orbit is circular as well with a semimajor axis of $a_f = 1.523396\ AU$, identical to the semimajor axis of the mars orbit. Inclination changes will not be considered.
Divided into three legs the mission consists of a first arc with continuous thrust, a coast arc and then a third leg with continuous thrust again. The corresponding equations of motion for the four states radius r, radial velocity v_r, tangential velocity v_{th} and angular position Φ are

$$\dot{r} = v_r \tag{2.2.25}$$

$$\dot{v}_r = \frac{v_{th}^2}{r} - \frac{\mu_g}{r^2} + \frac{T}{m}\sin\Theta \tag{2.2.26}$$

$$\dot{v}_{th} = -\frac{v_r \cdot v_{th}}{r} + \frac{T}{m}\cos\Theta \tag{2.2.27}$$

$$\dot{\Phi} = \frac{v_{th}}{r} \tag{2.2.28}$$

The symbol μ_g represents the gravitational constant and T is the thrust magnitude, which is zero during the coast arc. The thrust is steered via the angular control Θ. It defines the thrust direction and is discretized with 5 nodes each in phase 1 and 3. Mass is a function of the thruster's mass-flow

$$m = m_0 - \dot{m}t \tag{2.2.29}$$

The objective is to minimize the duration of the first and third leg, thus, minimizing propellant consumption.
A total of 26 parameters and 16 equality constraints are sufficient to describe the parameterized problem. Their complete description can be found in app. A.
The NLP program is minimized with the sequential quadratic optimizer SLSQP [53]. Figs. 2.2.1 and 2.2.2 show the spacecraft trajectory of the initial guess and the solution in the inertial orbit plane along with the paths of mars and earth at mission time. As can be seen, phasing conditions with the two planets apply for the spacecraft. The optimum cost function value is $f^* = 95.812\ days$.

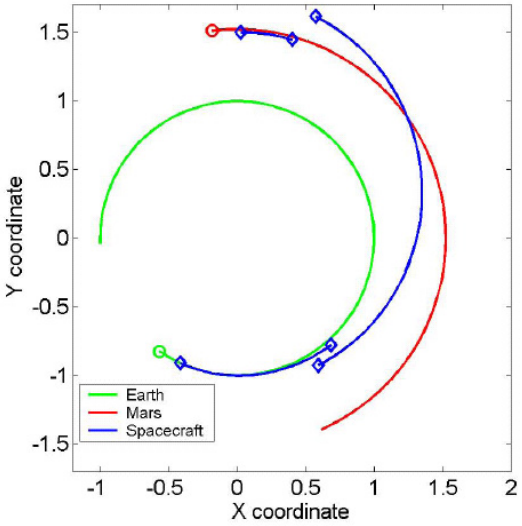


Fig. 2.2.1: Initial guess

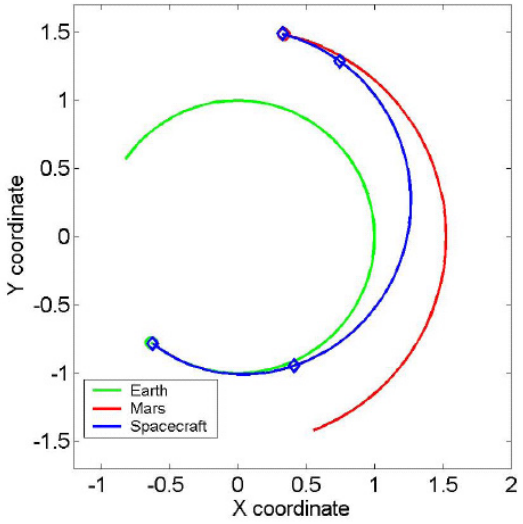


Fig. 2.2.2: Optimal solution

SLSQP provides the optimum set of parameters, the Jacobian matrix of the constraints and the cost function gradient for a consecutive sensitivity analysis as described in sec. 2.1.2 and 2.1.3. Complexity is reduced by the fact that all constraints are equalities.

2.2.2 Interpretation of the Solution

In order to assess the impact of certain perturbations and input changes, it is necessary to first define the perturbation vector ϵ. The composition for the present demo case can be found in tab. 2.2.1. The goal here is to identify the impact that these non-optimizable parameters have on the overall performance, characterized by the duration of the propelled mission legs.

The first order prediction f^{p1} of the cost function value for perturbations of 5.0 % and 10.0 %, respectively, in the elements of ϵ results in the numbers listed in tab. 2.2.2.

Coherence of the predictions for changes in the initial spacecraft mass ϵ_1 and the thrust level ϵ_2 is exceptionally good, compared to the corresponding true optima f^{ϵ}. The error is between 0.0 % and 5.8 % of the total variation. It does not come as a surprise, because of the approximately linear relationship between mass and thrust on one side and the total Δv that the spacecraft needs for the orbit manipulation on the other side.

ϵ_1: Initial spacecraft mass	4500 kg
ϵ_2: Thrust	6.0 N
ϵ_3: Initial radius	1.0 AU
ϵ_4: Final radius	1.523396 AU
ϵ_5: Final tangential velocity	0.81020 [ ]

Tab. 2.2.1: Composition of the perturbation vector

	ϵ_1		ϵ_2		ϵ_3		ϵ_4		ϵ_5	
$\Delta\epsilon_i$	5%	10%	5%	10%	5%	10%	5%	10%	5%	10%
f^*	95.812									
f^{p1}	100.61	105.40	92.73	89.65	79.67	63.53	108.9	121.9	109.1	122.4
f^{ϵ}	100.61	105.40	92.82	89.99	78.45	58.60	107.4	116.5	108.0	118.1

Tab. 2.2.2: Prognosis of the optimal cost function values - first order

For ϵ_3 through ϵ_5 the quality of the predictions is acceptable, but shows a disproportionately increasing error for larger perturbations. Taking a look at the equations of motion the reason becomes obvious. Perturbations in the parameters ϵ_3 through ϵ_5 affect the radius and velocity, which both have a complex influence on the propagation. This strongly motivates a second order analysis, which is exemplified for ϵ_4 according to

$$f^{p2}(x(\epsilon_4+\Delta\epsilon),\epsilon_4+\Delta\epsilon) = \quad f^*(x^*,\epsilon_4) + \frac{\partial f}{\partial \epsilon_4}\Delta\epsilon + \left(\frac{\partial f}{\partial \mathbf{X}}\right)^T \frac{\partial \mathbf{X}}{\partial \epsilon_4}\Delta\epsilon + \frac{1}{2}\frac{d^2 f}{d\epsilon_4^2}\Delta\epsilon \quad (2.2.30)$$

The lack of inequality constraints conveniently reduces the effort for the computations. The results for second order prediction can be found in fig. 2.2.3. They support the earlier assumption.

The use of second order terms significantly improves the prediction of the perturbed costs. The error for a variation of 10 % in ϵ_4 is reduced by about 75%. Further numbers can be found in tab. 2.2.3.
However, the computations that led to the reported numbers were based on a simplification. An explanation is required. The described demo case has a control in the form of a time-variant thrust direction. This control is parameterized. Even though the resulting parameters are optimizable, they are not unbounded. Which

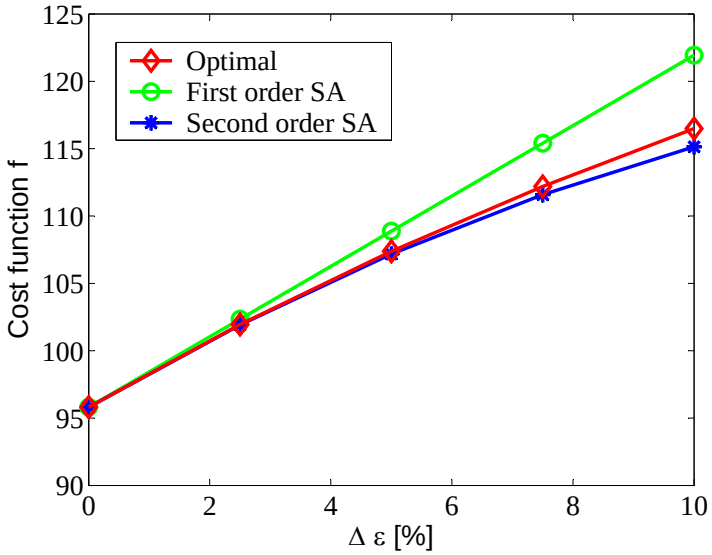


Fig. 2.2.3: First and second order approximation for perturbations in ϵ_4

means, a variation inside the model can cause control parameters to saturate which ultimately changes the set of active constraints. However, the algorithm mentioned in the previous section works with a certain predefined set of active constraints. A method to cope with this obstacle is to take the control parameters and fix them at their optimal values. The sensitivity analysis is performed as if the control parameters were non-optimizable. This kind of model reduction can certainly not be generally applied. But for the purpose of this demo case, where the goal is to indicate the usefulness of second order analysis, we accept it and refer to sec. 3 for an extensive discussion of active set manipulations.

2.3 Scaling Considerations

The fact that the whole concept behind NLP programming is gradient-based makes it mandatory to apply special care when computing gradients. This holds for the elements of the Jacobian, the cost function gradient and others. Only with precise knowledge about the derivatives is it possible to compute correct search directions and, in the end, obtain converged solutions, which obey the imposed optimization tolerances [8].

This is to the same extend true for post-optimality analysis. An accurate assessment relies on exact gradients and optimality of the provided solution.

	ϵ_3		ϵ_4		ϵ_5	
$\Delta\epsilon_i$	5%	10%	5%	10%	5%	10%
f^*	95.812					
f^{p1}	79.67	63.53	108.88	121.94	109.09	122.37
f^{p2}	77.95	56.65	107.18	115.14	108.10	118.39
f^{ϵ}	78.45	58.60	107.38	116.49	107.96	118.12

Tab. 2.2.3: Prognosis of the optimal cost function values - second order

The demanding task of computing accurate data is afflicted by numerous effects. In sec. 4.2.1 emphasis will be put on the selection and implementation of numerical finite difference algorithms.
Another source of inaccuracies emerges from round-off errors which are caused by general matrix operations. In essence, the critical factor, constraining the exactness of algebraic operations is the computer hardware, that works with a limited number of significant figure. With a regular PC, for instance, it is a necessity to avoid matrix calculations that require more than 15 significant figures to display deviations. Because otherwise the outcome is dominated by round-off errors and canceling effects [59].

The answer to this challenge is adequate conditioning of vectors and matrices. In NLP-based optimization, scaling techniques for parameters and constraints are commonly applied to enhance algorithm performance [8], [47], [31]. They can enormously improve the quality of the computations. Or, the other way around, if the quantities describing the optimization problem are poorly scaled and several orders of magnitude lie between them, the problem might be so degenerate that it's not even possible to solve it.

The benefits of scaling have also been documented for parameterized sensitivity analysis, tackling comparable challenges [7], [44]. For obvious reasons, we also incorporate a scaling strategy. It is a straightforward technique to scale the parameters, constraints and costs describing the transcribed optimization problem. The intention is to bring all parameters and constraints down to the range between -1 and 1. This motivates the following scaling for the different quantities:

$$x_s \quad = \quad Kx + \theta, \text{ with } K = diag(\kappa_1, ...\kappa_n), \theta = (\theta_1, ..., \theta_n)^T \qquad (2.3.31)$$

$$c_s = Tc, \text{ with } T = diag(\tau_1, ..., \tau_m) \tag{2.3.32}$$

$$f_s = \nu f \tag{2.3.33}$$

The selection of the scale weight for the perturbation ϵ with

$$\epsilon_s = \eta\epsilon \tag{2.3.34}$$

should take the structural properties of the perturbation parameter into consideration. For instance, if it is associated to a certain state parameter, then it is preferable to apply identical weights for scaling of the state and scaling of ϵ.

To better understand the internal processing of the scaled quantities inside the algorithm it is useful to take a look at the most common transformations of the various vectors and matrices required for PSA.

$$[\partial x]_s = K\partial x \tag{2.3.35}$$

$$[\partial c]_s = T\partial c \tag{2.3.36}$$

$$\left[\frac{\partial c}{\partial x}\right]_s = T\frac{\partial c}{\partial x}K^{-1} \tag{2.3.37}$$

$$\left[\frac{\partial f}{\partial x}\right]_s = \nu K^{-1}\frac{\partial f}{\partial x} \tag{2.3.38}$$

$$\left[\frac{\partial c}{\partial \epsilon}\right]_s = \eta^{-1}T\frac{\partial c}{\partial \epsilon} \tag{2.3.39}$$

$$\left[\frac{\partial f}{\partial \epsilon}\right]_s = \eta^{-1}\nu\frac{\partial f}{\partial \epsilon} \tag{2.3.40}$$

The calculation of the Lagrange multipliers inside the algorithm happens with respect to the scaled problem. Hence, the multipliers are automatically scaled. In order to obtain unscaled values one needs to scale them back. To derive the rule, we take a look at

$$min||G_s^T\lambda_s - \nabla f_s||, \quad with\ G_s = \left[\frac{\partial c}{\partial x}\right]_s \tag{2.3.41}$$

The unscaled Lagrange multipliers for the linear least squares problem then are

$$\lambda = \nu^{-1} T \lambda_s. \tag{2.3.42}$$

In addition, we take the opportunity here to mention several other transformations of matrices and vectors for reference purposes. These are

$$\left[\frac{\partial^2 \mathcal{L}}{\partial x^2}\right]_s = \nu K^{-1} \frac{\partial^2 \mathcal{L}}{\partial x^2} K^{-1} \tag{2.3.43}$$

$$\left[\frac{\partial^2 \mathcal{L}}{\partial x \partial \epsilon}\right]_s = \nu \eta^{-1} K^{-1} \frac{\partial^2 \mathcal{L}}{\partial x \partial \epsilon} \tag{2.3.44}$$

$$\left[\frac{\partial^2 \mathcal{L}}{\partial \epsilon^2}\right]_s = \nu \eta^{-1} \frac{\partial^2 \mathcal{L}}{\partial \epsilon^2} \eta^{-1} \tag{2.3.45}$$

2.4 Direct Control Optimization

The parameterized sensitivity analysis concept presented earlier in this chapter describes, as the name suggests, a method which is tailored for parameterized problems. And the definition of nonlinear programming problems composes the fundament for this method.
However, the ambition is to perform sensitivity analysis for continuous optimal control problems. An important link is still missing in the analysis strategy.

In sec. 1.1 two ways are listed, how optimal control problems can be solved. One is indirect optimization. The other is direct transcription. It is an NLP based solution method and, thus, establishes the missing link between optimal control theory and parameterized sensitivity analysis.
The design of more and more capable NLP solvers is attracting an increasing number of practitioners (see sec. 2.1.1). This has fostered the development of direct transcription methods to make the benefits of NLP programming available for the solution of continuous optimization problems with dynamics.
Even though it was not explicitly discussed, the earth-mars transfer gave a practical example for transcription. In a nutshell the term describes just the procedure that was necessary to fit the problem into an NLP frame.
The literature mentions two different methods to reformulate the mathematical description of an OCP. These are the shooting method and the collocation method [5], [31]. Both approximate the control history through a grid of nodes and an interpolation scheme (see fig. 2.4.4). Based on the specific requirements, the interpolation can be either piecewise constant, linear or even polynomial.
The whole concept ultimately relies on the conversion of the infinite number of

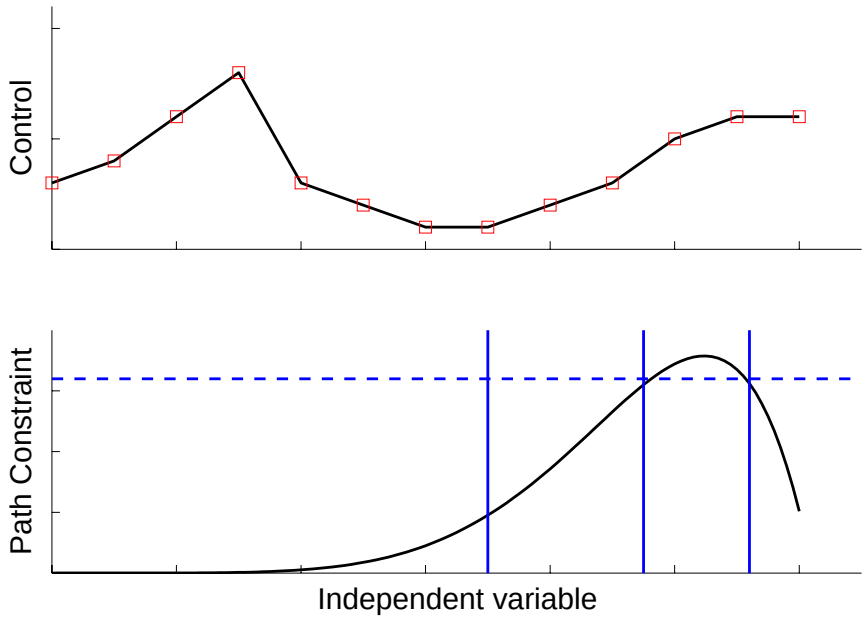


Fig. 2.4.4: Linear control approximation and path constraint evaluation grid

parameters of the continuous problem into a finite number of parameters and constraints. This has implications on possible path constraints. To suit the NLP program they are converted to point constraints which are evaluated at predefined nodes along the time history as shown in fig. 2.4.4.

2.4.1 Shooting Methods

Besides all the features that the two transcription methods have in common, there is also one, which distinguishes them. It's how they ensure compliance of the NLP program with the dynamics of the genuine OCP.
The shooting method exploits the exact equations of motion. Set up as an initial value problem, the initial state is defined as an optimizable parameter and then integrated along the independent variable (see fig 2.4.5). Adherence of the complete set of parameters with the integrated result is then forced via inclusion of defect constraints.

This method has the advantage of being very exact, since it uses the original equations. Generally, the integration is numerical. On the other hand, practitioners often wince at the time consuming integration and the stiffness of the technique.

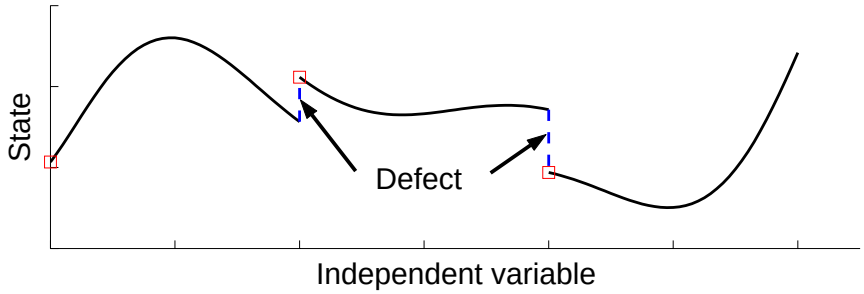


Fig. 2.4.5: Shooting scheme for state

2.4.2 Collocation Methods

In contrast to the first method, collocation does not only disretize the control history, but also the state history (see fig. 2.4.6). The set of schemes for parameterization ranges from trapezoidal and Euler approximations to Hermite-Simpson or Gaus-Lobatto polynomials [4], [49]. They represent different levels of sophistication and accuracy. It is the user's responsibility to identify the appropriate level for a particular application. Defect equations for a selection of collocation schemes can be found in app. B.

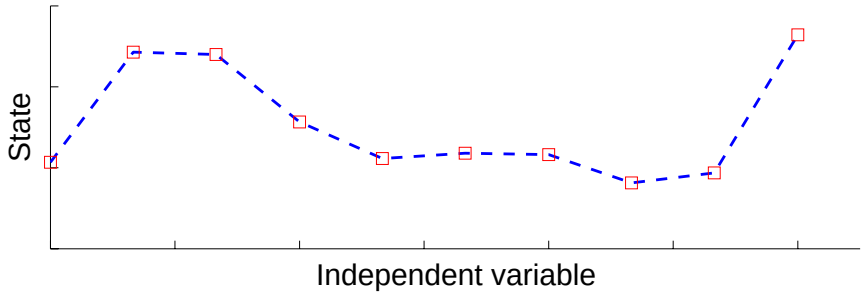


Fig. 2.4.6: Collocation scheme for state

Choosing the number and position of collocation nodes is a second important task of the user and requires careful assessment [75]. For a poor choice, the deviation between differential equation and approximate polynomial gets overly large. Then, the NLP program is not representative of the genuine OCP anymore. On the other hand, a very large number of optimization parameters unnecessarily degrades the performance of the solver. Here, automatic mesh refinement algorithms, as suggested by Betts [8] can serve as remedy.

Chapter 3

Extension for Post-Optimality Analysis

The concept of first and second order sensitivity calculation, which has been developed in the previous chapter, composes the core of our analysis method. However, so far, it falls short of the capabilities expected from a proper post-optimality analysis algorithm.
The example of sec. 3.1 illustrates the impairment of the algorithm, lacking information about the reliability of the computed gradient information in the broader vicinity of the optimality point. In principle, an enhanced evaluation technique is needed that examines the composition and steadiness of the employed active set of constraints and that derives profound data, which can be used to tag confidence properties to the sensitivity gradients.
This issue is tackled in sec. 3.2 to sec. 3.3, where a stepsize control-like technique is developed in order to provide improved optimality information in a non-iterative fashion.

3.1 Modified Zermelo Problem

The so called *Zermelo Problem* [15] is a popular example of an optimal control problem. It is about finding a minimum-time transfer trajectory for a boat in a flow field between two fixed points. The analytic solution is easily obtained with the help of the Euler-Lagrange equations for optimal control problems.
In the classical configuration the problem formulation does not contain any inequality constraints. For demonstration purposes we extend the description to contain such inequalities.

3.1.1 Problem Description

The mathematical description is plain and corresponds to the idea of having a simple example. Given the two dimensional dynamics

$$\dot{x} = V\cos\Theta + u \tag{3.1.1}$$

$$\dot{y} = V\sin\Theta + v \tag{3.1.2}$$

with (x, y) being the rectangular coordinates of the boat and $u(x, y)$ and $v(x, y)$ being the position dependent velocity components of the flow field in the direction of the x and y axis. The independent variable is the mission time t.
An engine is propelling the boat with a constant relative velocity V. The heading angle Θ can be controlled. Thus, it is continuously changeable.
Boundary constraints $\mathbf{\Psi}$ are imposed on the initial states as well as the final states:

$$\Psi_1 : x(t_0) = x_0 \tag{3.1.3}$$

$$\Psi_2 : y(t_0) = y_0 \tag{3.1.4}$$

$$\Psi_3 : x(t_f) = x_f \tag{3.1.5}$$

$$\Psi_4 : y(t_f) = y_f \tag{3.1.6}$$

A path constraint is added to the conventional problem:

$$g_1 : (x(t)\tan\rho + s) - y(t) \geq 0 \tag{3.1.7}$$

The linear function defines a barrier that the boat is not permitted to violate. The parameter s is the shift and ρ the gradient of this barrier.

In the earth-mars transfer example in sec. 2.2 the whole problem was configured in a way that only equality constraints were present. Because of very loose bounds on the state parameters, saturation of the states was practically impossible. In the Zermelo case, the bounds are chosen to be much tighter.

With a shooting scheme the problem is transcribed into an NLP program and then solved with an SQP method. The optimal solution of a nominal configuration is depicted in fig. 3.1.1. It shows the optimal trajectories of the unrestricted mission and for inclusion of the path constraint and tight state bounds.
The meaning of the term nominal configuration will be explained in sec. 3.3.4. Until then, the focus will entirely be on the general implications posed by the inclusion of inequality constraints.

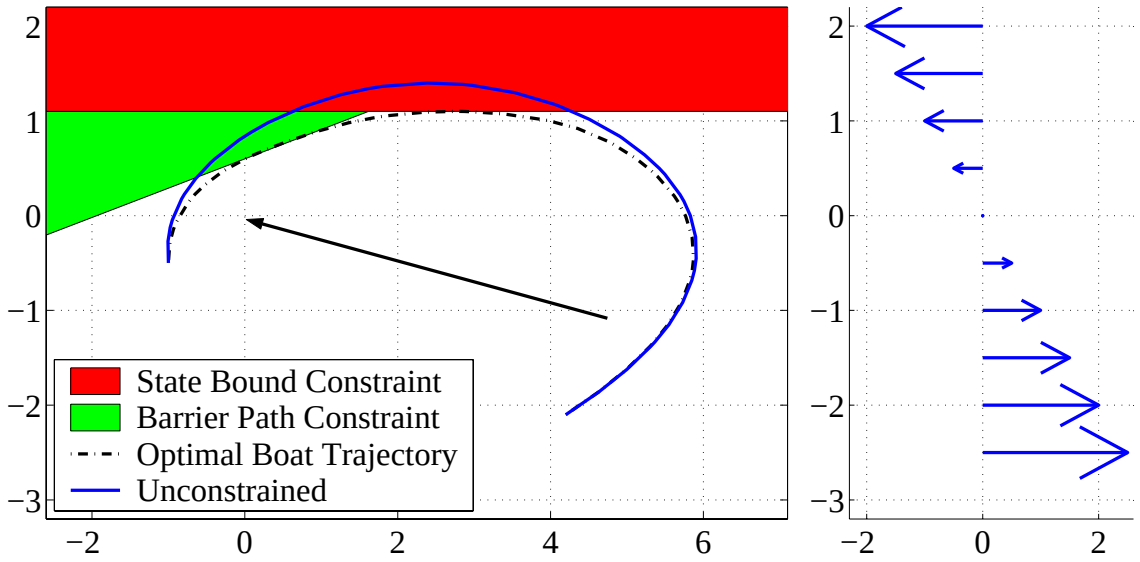


Fig. 3.1.1: Optimal transfer trajectory with partly active path constraints; flow field

3.1.2 Impairment of the Conventional Analysis

Sequential quadratic programming, like other NLP solver techniques, is iterative. The problem is solved in a series of steps. First, a search direction is computed and then the size of the next step adapted to produce optimum progress [43], [8], [39]. This process is generally gradient-based. The Jacobian matrix is evaluated and depending on the specific method also the Hessian matrix. Therefore, it is important to precisely know the active set of constraints $\mathcal{C}_a$, which, to some extend, defines the matrix elements.

This assignment is trivial for problems with solely equality constraints. In the earth-mars transfer case no inequalities were explicitly imposed. And the fact that the lower and upper bounds on the parameters were loosely defined with

$$(x_i)_{lb} \ll x_i \ll (x_i)_{ub}, i \in E^m \tag{3.1.8}$$

guaranteed an unchanged active set throughout the entire optimization.
But this does not hold for the modified Zermelo problem. The inclusion of the inequality constraint described by g_1 adds uncertainty to the composition of the active set. It depends on the actual parameter set and the location of the evaluation points, whether the corresponding point constraints are enforced or not. Implicitly, this has also major influence on the feasibility of the mission.

NLP solvers have sophisticated, embedded strategies for updating the active set

during the iterative solution of the problem [57], [52]. They ensure feasibility of the final trajectory by evaluating all kinds of inequality constraints, be it path constraints, boundary constraints or parameter limits and including or excluding them from the active set depending on the momentary status.

The correctness of the active set of constraints is just as important for parameterized sensitivity analysis and post-optimality assessment as it is for optimization algorithms. This need arises from the ambition to predict the behavior of the optimal solution not only in the direct neighborhood around the nominal solution, but also in the broader vicinity for finite variations of non-negligible magnitude.
In order to enable a qualified statement about the reliability of the POA results, it is mandatory to know about the steadiness of the active set composition. This information shall be computed with minimum effort. A routine is required, which either updates the active set in a very time efficient way - an optimizer-like iterative procedure is not aspired. Or it shall make a statement about the parameter space for which the provided sensitivities are applicable with a certain level of confidence.

3.2 Identification of Active Inequality Constraints

We understand how important it is to work with the correct and complete active set of constraints $\mathcal{C}_a^*$. But how exactly is it composed? Fig. 3.2.2 shows a generic overview of constraint types commonly used in transcriptions. These are the kind of constraints, which are explicitly defined and handed over to an NLP solver. So they are readily available, including the corresponding parameter derivatives in the Jacobian matrix. The symbol $\mathcal{C}$ is assigned to the entirety of these constraints.

The standard set $\mathcal{C}^*$ of the optimal solution contains equality constraints $\mathcal{H}^*$ and inequality constraints $\mathcal{G}^*$ respectively. In order to obtain the concise active subset, the active components of $\mathcal{G}^*$ are extracted

$$\mathcal{G}_w^* = \{c(x^*, 0) \in \mathcal{G}^* \mid c(x^*, 0) = 0\} \tag{3.2.9}$$

and merged with $\mathcal{H}^*$ to form the working set:

$$\mathcal{C}_w^* = \{c(x^*, 0) \in \mathcal{C}^* \mid (c(x^*, 0) \in \mathcal{H}^*) \vee (c(x^*, 0) \in \mathcal{G}_w^*)\} \tag{3.2.10}$$

The active set is not complete yet. Besides the explicitly formulated constraints there are also implicit conditions. They emerge from the boundedness of the optimizable parameters and are linear in their representation:

$$\mathbf{c_{lb}} \quad : \quad x_i - (x_i)_{lb} \geq 0, i \in E^m \tag{3.2.11}$$

$$\mathbf{c_{ub}} \quad : \quad (x_i)_{ub} - x_i \geq 0, i \in E^m \tag{3.2.12}$$

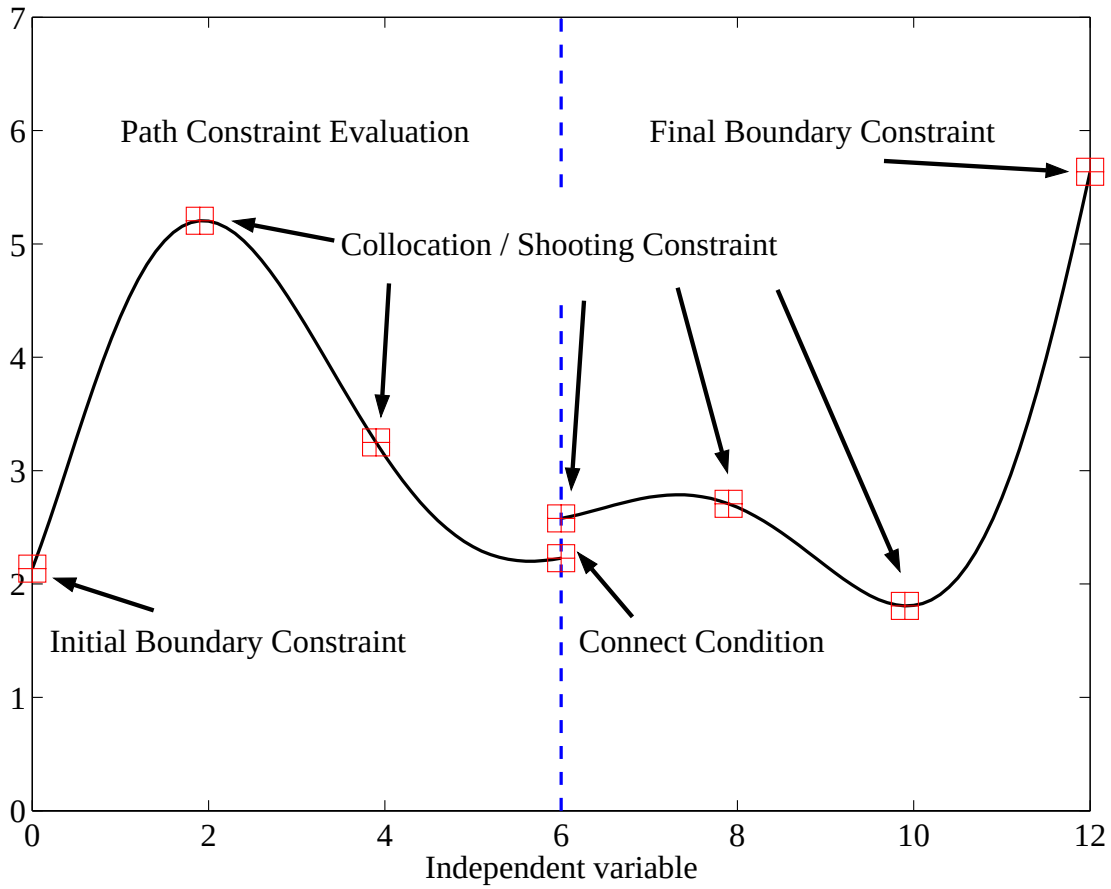


Fig. 3.2.2: Survey of constraint types commonly considered in transcriptions

Linearity implies that the first derivative is constant and that higher derivatives are zero. NLP solvers, therefore, often treat linear constraints separately [38]. A more thorough study of related implications for the Jacobian and Hessian matrix computations will be given in sec. 4.3. For the time being we will concentrate on completing the active set.
Unlike the parameter bounds in the earth-mars transfer, tight limits can force inclusion of bound constraints $\mathcal{B}^*$ in the active set of constraints for the optimal solution to the nonlinear program. This completes $\mathcal{C}_a^*$ with

$$\mathcal{C}_a^* = \{\mathcal{C}_w^* \cup \mathcal{B}^*\} \tag{3.2.13}$$

The prior explanations on the composition of the active set sound trivial. Nonetheless, they are necessary to understand the updating and consecutive handling of the active set in the context of post-optimality analysis.

3.3 Prediction of Active Set Changes

In general, the optimal solution of a problem is reached at the end of an iterative process. Step by step the solution is refined until the optimality conditions are finally met. An inherent part of this process is the routinely update of the active set of constraints. It is necessary to ensure feasibility and optimality of the terminal solution while permitting maximum flexibility. In the end, also the composition of the active set converges towards a steady state.
Conventional sensitivity analysis follows a different concept. It is a single step calculation which exploits the information available about the active set. Hence, the sensitivity data is only valid as a linear approximation in the neighborhood around the KKT point, where the active set remains unchanged.
This is sufficient to identify tendencies of, for instance, the costs for infinitesimal perturbations in the design parameters and to analyze problems with a very small number of considered constraints [44], [60]. But the performance quickly degrades when predicting the optimal behavior for finite variations in a more complex problem.
It is advised to estimate the changes in $\mathcal{C}_a^*$ that can be expected for a defined perturbation. The fundamental thought behind this is to provide an enhanced prediction for the perturbed solution based on the information that is readily available. Since the Lagrange multipliers establish the status of inequality constraints, they are the key to the prediction of changes in the active set and will further on be investigated more closely.

Before submerging into the subspaces of active and inactive constraints, it is useful to look at fig. 3.2.2 once more. Because there is one distinct implication connected to the discretization which has not been mentioned yet: the post-optimality analysis can't consider any conditions which are not part of the transcribed problem description. This means for path constraints, for instance, that activation of a path constraint is out of the question if no evaluation node is defined in the region of interest. Practitioners should bear this in mind, when evaluating PSA performance.

3.3.1 Estimation of Perturbed Lagrange Multipliers

In order to assess the adequacy of the information about the active set for consecutive use in post-optimality analysis we start with the optimality condition for the Lagrangian function as formulated in eq. 2.1.9

$$\mathcal{L}_x(x^*, \lambda^*, \mu^*, 0) \;=\; f_x(x^*, 0) - \left(\mathbf{g}_x(x^*, 0)\right)^T \lambda^* + \left(\mathbf{h}_x(x^*, 0)\right)^T \mu^* = 0$$

It holds that

$$\lambda_i \left[\left(\frac{\partial g_i}{\partial \epsilon} \right)^T + \left(\frac{\partial g_i}{\partial \mathbf{x}} \right)^T \frac{\partial \mathbf{x}}{\partial \epsilon} \right] \Delta\epsilon = 0, \quad i \in E^{m-m_e} \tag{3.3.14}$$

and

$$\left[\left(\frac{\partial h_j}{\partial \epsilon}\right)^T + \left(\frac{\partial h_j}{\partial \mathbf{x}}\right)^T \frac{\partial \mathbf{x}}{\partial \epsilon}\right] \Delta\epsilon = 0, \quad j \in E^{m_e} \tag{3.3.15}$$

Hence, it follows from

$$\frac{df}{d\epsilon} = \frac{\partial f}{\partial \epsilon} + \left(\frac{\partial \mathbf{x}}{\partial \epsilon}\right)^T \frac{\partial f}{\partial \mathbf{x}} \tag{3.3.16}$$

that

$$\frac{df^*}{d\epsilon} = \frac{\partial f}{\partial \epsilon} - \left(\frac{\partial \mathbf{g}}{\partial \epsilon}\right)^T \lambda^* + \left(\frac{\partial \mathbf{h}}{\partial \epsilon}\right)^T \mu^* \tag{3.3.17}$$

The gradient of the objective function at the KKT point $\mathbf{y}^*$ is a linear combination of the derivatives of the problem constraints. The Lagrange multipliers are the coefficients to these derivatives. For inactive constraints these coefficients are zero.

Local convergence of SQP methods is applicable under the assumption that the active set does not change. This assumption is valid in the vicinity of the KKT point where the Hessian matrix is positive definite [22]. In this neighborhood, the KKT point can also be formulated as $\tilde{y}^* = \left[(x^*)^T \ (\tilde{\mu}^*)^T\right]$ with $\tilde{\mu}^*$ representing all Lagrange multipliers of $\mathcal{C}_a^*$.
An SQP step assuming optimal variation can then be formulated as

$$\tilde{y}^p \quad = \quad \tilde{y}^* + \Delta\tilde{y}. \tag{3.3.18}$$

The vector $\Delta\tilde{y}$ is computed from the first order Taylor expansions

$$\Delta\tilde{y} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \epsilon} \\ \frac{\partial \tilde{\mu}}{\partial \epsilon} \end{bmatrix} \Delta\epsilon. \tag{3.3.19}$$

As shown in sec. 2.1.2, the sensitivity of the Lagrange multipliers of the active constraints with respect to variations in ϵ can be calculated from

$$\nabla_{xx}\mathcal{L}\frac{\partial \mathbf{x}}{\partial \epsilon} + \left[\frac{\partial c_a}{\partial \mathbf{x}}\right]^T \frac{\partial \tilde{\mu}}{\partial \epsilon} = -\nabla_{x\epsilon}\mathcal{L} \tag{3.3.20}$$

This is the basis for the prediction of the Lagrange multipliers as part of the post-optimality analysis for variations in ϵ.

The calculation of the Lagrange multiplier gradients requires the Hessian matrix. It is, hence, a task for second order analysis. Nonetheless, the additional effort seems justified and will variously be addressed in following chapters.

3.3.2 Confidence Region

The Lagrange multipliers of inactive constraints are zero. Hence, in a coarse approximation, the subset

$$\tilde{\mu}^p \;=\; \tilde{\mu}^* + \frac{\partial \tilde{\mu}}{\partial \epsilon}\Delta\epsilon \tag{3.3.21}$$

of eq. 3.3.18 can serve as an indicator for the tendency of the inequality constraints to remain active or to turn inactive.

In an SQP iteration, the length $\Delta\epsilon$ would set the size of the step. We will use the formula in reciprocal fashion to identify a so called confidence region. For a single element perturbation, the perturbed Lagrange multipliers are set to zero

$$\tilde{\mu}^p \stackrel{!}{=} 0 \tag{3.3.22}$$

and the equation is reformulated as

$$0 = \tilde{\mu}^* + \left[\frac{\partial \tilde{\mu}}{\partial \epsilon}\right] \cdot \Delta\epsilon_c \tag{3.3.23}$$

It is solved for the column vector $\Delta\epsilon_c$. The elements with the smallest magnitude then define the confidence radius for constraint deactivation for as long as they belong to inequality constraints:

$$(\Delta\epsilon)_{lb} = max\{\Delta\epsilon_c \mid ((\Delta\epsilon_i)_c \leq 0) \cap (c_i \in \mathcal{G}^*)\} \tag{3.3.24}$$

$$(\Delta\epsilon)_{ub} = min\{\Delta\epsilon_c \mid ((\Delta\epsilon_i)_c \geq 0) \cap (c_i \in \mathcal{G}^*)\} \tag{3.3.25}$$

The remaining elements of the KKT point vector, namely the optimizable parameters, are extrapolated in the same fashion. This provides an insight into the potential saturation of parameter bounds and, thus, additional references in the form of confidence radii for the update of the active set of constraints.
Usage of the confidence radii requires cautious handling. The linear extrapolation is far from being exact. Especially the Lagrange Multipliers of active constraints are very sensitive and might hold superordinate nonlinear properties.

3.3.3 Completion of Active Set Data

Once the composition of the active set of constraints is determined, the regarding vectors, matrices and derived quantities need to be computed or updated.
The elementary approach is to search the pre-computed data from the optimization for information linked to the components of $\mathcal{C}_a^*$. This includes columns of the Jacobian matrix and constraint values.

Since completeness and coherence is a prerequisite for expedient post-optimality analysis, all information that is not provided by the optimization algorithm needs to be computed separately. The extend of this task depends on the capabilities and accessibility of the particular solver for the optimization problem. Routinely, these solvers return information for the generalized set of explicit constraints $\mathcal{G}$ and $\mathcal{H}$, including Lagrange multipliers and scale weights. Several algorithms also provide cost gradients and constraint derivatives. However, it is not common that an accurate Hessian matrix is returned.
In chapter 4 data completion is demonstrate for a specific algorithm, namely the optimizer CAMTOS.

The idea of using first order results of the Lagrange multiplier prediction to update the active set composition for enhanced second order analysis sounds appealing. However, a closer look reveals that this poses more problems than benefits.
A comprehensive update of Jacobian and Hessian matrix requires techniques that are usually used in iterative optimization algorithms. They are either efficient and coarse, or extensive and accurate. Both concepts do not satisfy the demands of post-optimality analysis. Further contributions to this discussion can be found in chapter 4.
Overall, it can be said that the concept is not worth serious consideration.

3.3.4 Practical Example

The likeliness of impairments of the method, developed in chapter 2, when applied to the modified Zermelo problem has motivated the prediction of perturbed Lagrange multipliers. It is reasonable to use the Zermelo example now for an assessment of the capabilities of the newly established features. Hence, the post-optimality analysis is extended. The confidence radii for different problem configurations are computed and exploited towards an identification of the perturbed active set composition. It provides predictive information along with a reliability judgement based on the confidence radii. The results are cross-checked with reoptimized solutions of the perturbed cases.

Some preparatory work needs to be conducted to qualify the model of the modified Zermelo problem from sec. 3.1 for use in a post-optimality analysis. Since the POA concentrates on the influence of non-optimizable parameters, it is necessary to declare the quantities of interest as elements of such a parameter set. The list of parameters together with their nominal values is given in tab. 3.3.1.

At first, a possible variation in the shift s of the barrier is assumed. A sensitivity analysis is run for $\epsilon_s = 0.6$. The results can be found in fig. 3.3.3.
The confidence radius is computed to be $(\Delta\epsilon_s)_c = 0.204$. This value turns out to

Name	Reference	Description	value
ϵ_v	V	Velocity base value	1.00
ϵ_ρ	ρ	Gradient of the barrier [rad]	0.30
ϵ_s	s	Shift of the barrier [UL]	0.60
ϵ_{x0}	x_0	Initial value of state x [UL]	4.20
ϵ_{yf}	y_0	Initial value of state y [UL]	-2.10
ϵ_{xf}	x_f	Final value of state x [UL]	-1.00
ϵ_{yf}	y_f	Final value of state y [UL]	-0.50

Tab. 3.3.1: Non-optimizable parameters available for the POA

specifically limit the trust in the status of the inequality constraint that is related to the point evaluation of the path constraint. Interpretation suggests, that for a variation in s in the range of 0.2, the trajectory detaches from the barrier, thus, changing the set of active constraints. First and second order SA use the active set from the optimization. Accordingly, this effect should become tangible in the accuracy of the sensitivity data.
Results can be seen in the given references for $\Delta s = 0.2$ and $\Delta s = 0.4$, respectively. Whereas the predicted cost function $f^{p2}(\Delta\epsilon_s = 0.2)$ is within 1.5% accurate compared to the optimal cost, the error increases enormously when going beyond the confidence radius. For $\Delta\epsilon_s = 0.4$ the deviation in the cost prediction increases to more than 20%, even though the variation in the input only doubles.
Since the cost function prediction is directly affected by $\frac{dx}{d\epsilon}$ we can implicitly expect enhanced predictions for state variations. For one representative point is this visualized in the trajectory plots of fig. 3.3.3. An extrapolation $(x, y)^p$ for $\Delta\epsilon_2$ beyond the confidence radius keeps constraints enforced which otherwise would not be active. Even for this simple example, the data then quickly becomes overly erroneous.

The first demo configuration neglected saturated state bounds. This is corrected in the next configuration by setting the upper bound of y to $y_{ub} = 1.1$. Once more, the confidence radius proves to be of good use. The new boundary constraint is identified to turn inactive for about $\Delta\epsilon_{x0} = -0.98$. This is verified empirically (see table in fig. 3.3.4). The first order cost prediction delivers good results. For second order analysis, the quality degrades slightly to an accuracy of about 5 %. The reason has to be attributed to the overall large variations, based on a small number of parameters and the evaluation point placement.

The proper placement of evaluation points can become an enormously important task in parameterized sensitivity analysis. This is addressed in another example, where variations in the initial state y_0 are investigated.

s	y_{ub}	f^*
0.6	∞	7.1655

	Case ①	Case ②
$\Delta\epsilon_s$	0.2	0.4
$(\Delta\epsilon_s)_c$	0.2040	0.2040
$f^{p1}(\Delta\epsilon_s)$	7.0900	7.0146
$\left\lvert\frac{f^{p1}-f^{\epsilon}}{f^{\epsilon}-f^*}\right\rvert$	0.5253	1.7092
$f^{p2}(\Delta\epsilon_s)$	7.1167	7.1213
$\left\lvert\frac{f^{p2}-f^{\epsilon}}{f^{\epsilon}-f^*}\right\rvert$	0.0141	0.2065
$f^{\epsilon}(\Delta\epsilon_s)$	7.116	7.1098

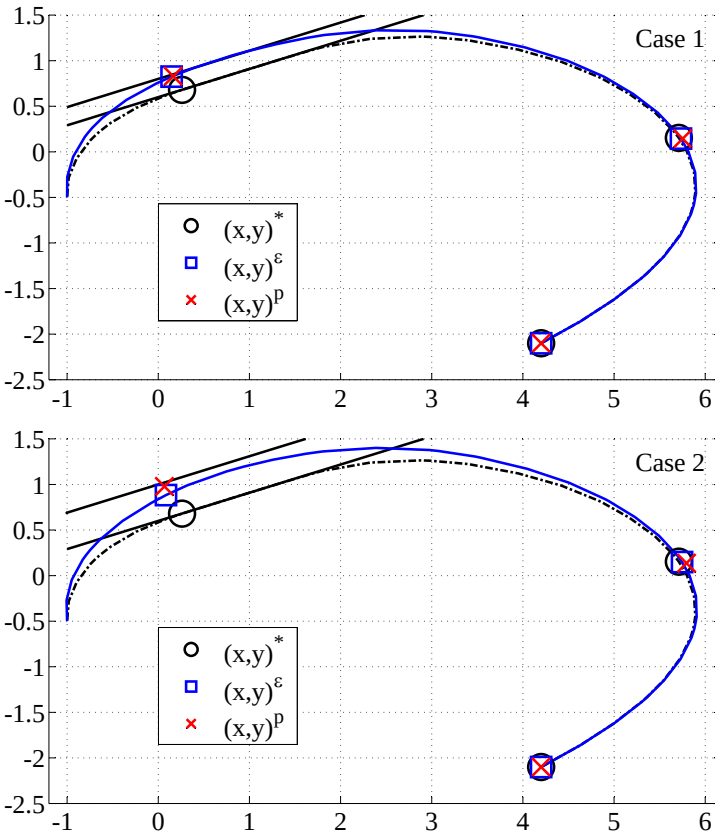


Fig. 3.3.3: Cases ①/②: The table shows predicted cost function values (p1 is order one, p2 is order two) for two different finite variations of ϵ, including reoptimized value (index ϵ) and error. The graphs show the nominal, predicted and reoptimized boat trajectory including discretization nodes.

Shooting methods for transcription are popular and attractive. They do not require a large number of grid nodes to stabilize the approximation. This reduces the NLP problem size but also poses the following risks in connection with PSA. Take the analysis results for a variation of $\Delta\epsilon_{y0} = -0.7$, for instance. The upper graph in fig. 3.3.5 shows the trajectories that were optimized with a grid of four state nodes. The third node saturates at the state's upper limit. The confidence radius of $(\Delta\epsilon_{y0})_c = -0.8136$ of this particular inequality suggests, that it comes close to a change of status for $\Delta\epsilon_{y0} = -0.7$. This prediction is correct but falls short. The enlargement of the trajectory via $\Delta\epsilon_{y0}$ moves the grid node to the right, away from the center of interest. The lack of evaluation nodes, then, tolerates the violation of the upper state bound, causing an infeasible trajectory. This can be overcome by inclusion of an additional grid point during the NLP optimization, as can be seen in the lower graph of fig. 3.3.5. The results of the post-processing of the op-

x_0	y_{ub}	f^*
4.2	1.1	7.1825
$\Delta\epsilon_{x0}$		-1.05
$(\Delta\epsilon_{x0})_c$		-0.9802
$f^{p1}(\Delta\epsilon_{x0})$		6.6796
$\left\lvert\frac{f^{p1}-f^{\epsilon}}{f^{\epsilon}-f^*}\right\rvert$		0.0030
$f^{p2}(\Delta\epsilon_{x0})$		6.7036
$\left\lvert\frac{f^{p2}-f^{\epsilon}}{f^{\epsilon}-f^*}\right\rvert$		0.05056
$f^{\epsilon}(\Delta\epsilon_{x0})$		6.6781

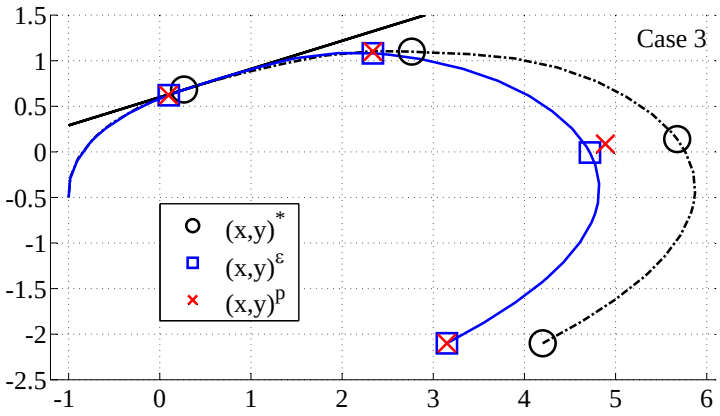


Fig. 3.3.4: Case ③: Parameters, processed results and trajectory. For reference see fig. 3.3.3

timal solution delivers the following results now. The confidence radius is reduced to $(\Delta\epsilon_{y0})_c = -0.2595$, because of the upper bound y_{ub} at the fourth state node. Ultimately, this node secures feasibility of the perturbed solution. Looking at the predicted cost function value, one might not consider the improvement from 1.27% to 0.58% to be noteworthy. Nonetheless, the enhanced feasibility information is.

The calculation of confidence radii, based on the extrapolation of the Lagrange multipliers and parameters has shown to be of good use for the post-optimality prediction of variational influences. Their interpretation increases reliability and therefore also accuracy of the prognoses.

y_0	y_{ub}	f^*
-2.1	1.1	7.1789

	Case ④	Case ⑤
$\Delta\epsilon_{y0}$	-0.7	-0.7
$(\Delta\epsilon_{y0})_c$	-0.8136	-0.2595
$f^{p1}(\Delta\epsilon_{y0})$	8.5391	8.5409
$\left\lvert\frac{f^{p1}-f^{\epsilon}}{f^{\epsilon}-f^{*}}\right\rvert$	0.0749	0.0807
$f^{p2}(\Delta\epsilon_{y0})$	8.6678	8.6691
$\left\lvert\frac{f^{p2}-f^{\epsilon}}{f^{\epsilon}-f^{*}}\right\rvert$	0.0127	0.0058
$f^{\epsilon}(\Delta\epsilon_{y0})$	8.6492	8.6605

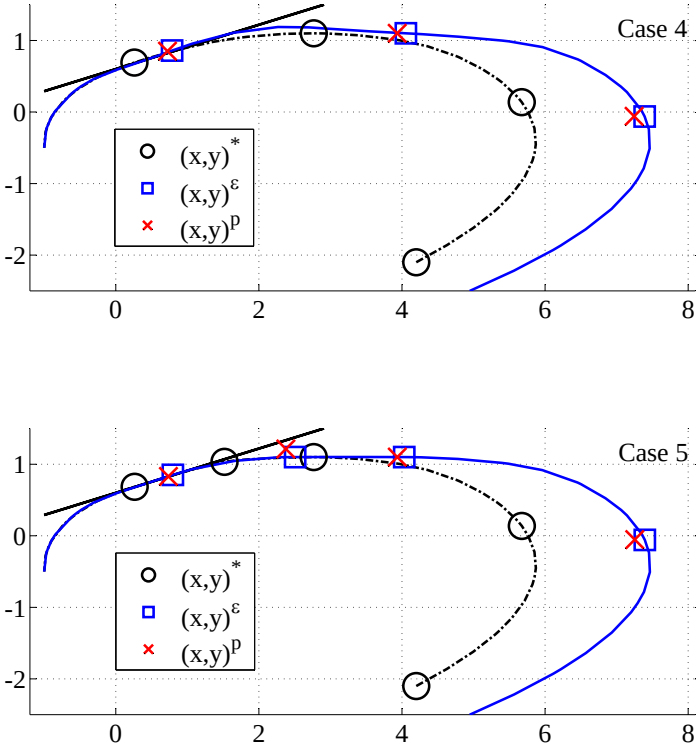


Fig. 3.3.5: Case ④/⑤: Parameters, processed results and trajectory. Better discretization in case ⑤ allows more exact prediction of active set, leading to more accurate post-optimality analysis results.

Chapter 4

Integration with an Existing Optimization Software

The intention behind this thesis is, to develop an efficient, POA-based tool for post-processing of optimal control problems. Therefore, the POA algorithm needs to be integrated with an optimizer that provides respective optimal solutions in a convenient structure. A practitioner, who wants to analyze an OCP shall be enabled to optimize it and run a POA immediately afterwards, taking full advantage of the optimizer output.
It has been decided to integrate the POA algorithm with the software CAMTOS [31]. The acronym stands for *Collocation And Multiple Shooting Trajectory Optimization Software*. It is a hybrid method allowing the user to select from a set of collocation and shooting schemes to transcribe the optimal control problem. The software has been developed at the Institute of Flight Mechanics and Control IFR, University of Stuttgart, Germany [33], [34], [35]. CAMTOS is embedded in the GESOP package, a so called *Graphical Environment for Simulation and OPtimization* [4], which manages the interfacing with the NLP solver, integrators and does the data handling.

The underlying concept behind CAMTOS is motivated by the same interests as the POA efforts, namely to offer simple-to-use optimization expertise to practitioners, who are no optimal control experts, but specialists in a certain field of application. This makes CAMTOS most suitable for the aimed purpose.

In a number of steps the POA algorithm needs to be integrated and adapted to the parameter and constraint structure of the optimizer. Further data retrieval from the NLP solver via the optimizer (see fig. 4.0.1) is considered to be a key competence to ensure efficient computing and increase practical usefulness of the integrated tool. Therefore, the next section deals with the elementary details of the transcription procedure and how POA fits in. Then, in sec. 4.2.1 the computation of the Hessian matrix is discussed and evaluated with respect to accuracy matters.

Practical aspects of the boundary constraint update and their inclusion in the Jacobian matrix and other relevant vectors are contained in sec. 4.3.
An assessment of the accuracy of the Lagrange multipliers and implications for the quality of the parameterized sensitivity analysis in sec. 4.4 completes the chapter on practicality and implementation of the POA algorithm.

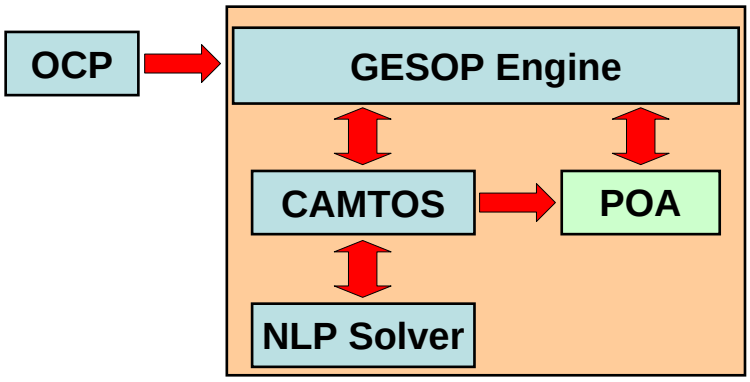


Fig. 4.0.1: Structure of and data flow between the different modules of GESOP

4.1 Transcription of the Optimal Control Problem

The transcription of the optimal control problem into an NLP program is a critical task. It entails consequences for the optimization process as well as for consecutive processing of the results. A first categorization of transcription methods into collocation and shooting schemes has been given in sec. 2.4, where direct solution approaches are described. To be able to effectively interface the post-optimality analysis with the optimizer, it is essential to understand the parameterization of the original problem.
CAMTOS offers various schemes, which have a series of features in common. For instance, the handling of discontinuities, which commonly appear in practical use. They are treated as connect conditions of multi-phase setups. In order to allow enhanced flexibility and avoid the risk of overlapping ranges of the independent variable of the genuine problem, the runtime variable is phasewise normalized. This constitutes a scaling of the independent variable and is beneficial for the optimizer accuracy.

4.1.1 Multiple Shooting Scheme

The direct shooting method in CAMTOS works with Runge-Kutta integrators of order 2/3 and 4/5, respectively. The parameter set is developed according to model specifics and continuity conditions for one shooting interval after the other. In tab. 4.1.1 the sparsity pattern of the Jacobian matrix is given for one interval block, also illustrating the applicable parameter and constraint types. The block-diagonal structure does not come naturally, but is achieved by duplicating the optimizable real parameters and phase times at each shooting point. Despite its advantageous structure for efficient optimization, there are also drawbacks caused by the multitude of similar parameters. Sensitivity analysis will compute multiple independent derivatives with respect to just one real parameter. The matter is addressed when dealing with practical examples in chapters 5 and 6.
A more detailed discussion of the shooting scheme in CAMTOS can be found in [31].

Parameter / Constraint	x_0	p_0	u_0	t_{00}	t_{f0}	x_1	p_1	u_1	t_{01}	t_{f1}
Initial Boundary	x	x	x	x	x					
Parameter Constraint		x		x	x					
Path Constraint	x	x	x	x	x					
State Connect	x	x	x	x	x	x				
Parameter Connect		x					x			
Control Connect			x					x		
t_0 Connect				x					x	
t_f Connect					x					x
Path Constraint						x	x	x	x	x
Final Boundary						x	x	x	x	x

Tab. 4.1.1: Sparsity pattern of the Jacobian Matrix in a multiple shooting phase with one node according to [31]

4.1.2 Collocation Scheme

Collocation schemes are popular since they do not require time-consuming variable integration and show appreciable robustness of the nonlinear program. On the other

hand, they are, in fact, approximations. Which means, they are not per se accurate. Therefore, CAMTOS offers several schemes that the user can choose from to refine the accuracy of an optimized trajectory. Whereas the controls are always modeled as piecewise constant or piecewise linear, the states can be approximated as Hermite-Simpson polynomials, Runge-Kutta polynomials or with a trapezoidal rule (see app. B). The defect conditions are added to the set of constraints.

The sparsity pattern of the collocation is coarser organized than the one of the shooting scheme (see tab. 4.1.2). The enormous increase in parameters and constraints due to duplication is judged to be more critical than the possible savings traded by the slimmer block structure of the Jacobian matrix. Therefore, the collocation scheme abandons the idea of parameter duplication.

Parameter / Constraint	x_1	u_1	x_2	u_2	x_3	u_3	x_4	u_4	p_1	t_0	t_f
Initial Boundary	x	x							x	x	x
Parameter Constraint									x	x	x
Path Constraint	x	x	x	x					x	x	x
Defect 12	x	x	x	x					x	x	x
Control Continuity		x		x							
Path Constraint			x	x	x	x			x	x	x
Defect 23			x	x	x	x			x	x	x
Control Continuity				x		x					
Path Constraint					x	x	x	x	x	x	x
Defect 34					x	x	x	x	x	x	x
Control Continuity						x		x			
Final Boundary							x	x	x	x	x

Tab. 4.1.2: Sparsity pattern of the Jacobian Matrix in a collocation phase which consists of three intervals (Ref. [31])

There is another peculiarity. The implementation works with a dual set of control parameters at each grid point. Defining a left-hand and a right-hand control enlarges the convergence radius of the optimizer. During the optimization the NLP solver permits defects which eventually have to be reduced to zero.

This poses an obstacle for sensitivity analysis, since again multiple parameters describe just the same. This will be addressed in later chapters.

4.2 The Hessian Matrix

We have learned that the calculation of higher order sensitivity information requires the Hessian matrix of the Lagrangian function. In the following section we will discuss the computation of the Hessian using finite differences. First the Hessian matrix itself is introduced. Then, the finite difference approximation is derived and accuracy limitations are discussed.

The Hessian of the Lagrangian is the second derivative of the Lagrangian function with respect to the system parameters. Thus, after dual differentiation of eq. 2.1.4 we obtain the Hessian

$$\frac{\partial^2 \mathcal{L}}{\partial x^2} = \frac{\partial^2 f}{\partial x^2} - \sum_{k=m_e+1}^{m} \lambda_{\mathbf{k}} \frac{\partial^{\mathbf{2}} \mathbf{g_k}}{\partial \mathbf{x^2}} + \sum_{\mathbf{l=1}}^{\mathbf{m_e}} \mu_{\mathbf{l}} \frac{\partial^{\mathbf{2}} \mathbf{h_l}}{\partial \mathbf{x^2}}. \tag{4.2.1}$$

It is helpful to take a look at certain characteristics of the Hessian matrix. First of all, and most obviously, the Hessian is symmetric. For a more thorough discussion, see [40], p.320.
Another practical property is the sparsity structure of the Hessian alluded in eq. 4.2.1. Every element $\frac{\partial^2 \mathcal{L}}{\partial x_i \partial x_j}$ is composed of the sum of the partial derivatives of all constraints with respect to x_i and x_j. For certain combinations of parameters the Hessian element can't be non-zero when following the sparsity pattern of the transcription methods. The correlation defines a particular super-block diagonal structure of the Hessian matrix for every individual NLP program.
For single-phase problems in connection with a collocation scheme, the block is so large, that the matrix has to be understood as dense. But for multi-phase setups the sparsity can becomes truly beneficial for efficient calculation.

4.2.1 Finite Difference Approximations

The expenses when differentiating the cost function and all constraints with respect to all parameters are enormous. Most often an analytical differentiation is not even possible. Therefore, it is common practice to approximate the derivatives with finite differences. We distinguish between forward/backward differences which only require one additional function value computation and central differences, where two additional function value computations are necessary. The benefit of the central difference method is the higher order accuracy.

Following, we develop the equations for the finite difference approximation of second order derivatives and apply them to the Hessian matrix.
Given the function

$$f = f(x).$$

The forward difference approximation for the first derivative with respect to x_i is

$$\frac{\partial f(x)}{\partial x_i} = \frac{f(x_i + \Delta x) - f(x_i)}{(x_i + \Delta x) - x_i} + \mathcal{O}(\Delta x) \tag{4.2.2}$$

for a Δx of appropriately small size.
The central difference approximation for the first derivative with respect to x_i is

$$\frac{\partial f(x)}{\partial x_i} = \frac{f(x_i + \Delta x) - f(x_i - \Delta x)}{(x_i + \Delta x) - (x_i - \Delta x)} + \mathcal{O}(\Delta x^2) \tag{4.2.3}$$

We use the method of forward differences a second time to obtain the second derivative approximation

$$\frac{\partial^2 f}{\partial x_i \partial x_j} \approx \frac{\left[\frac{f(x_i+\Delta x, x_j+\Delta x)}{\Delta x} - \frac{f(x_i+\Delta x, x_j)}{\Delta x}\right] - \left[\frac{f(x_i, x_j+\Delta x)}{\Delta x} - \frac{f(x_i, x_j)}{\Delta x}\right]}{\Delta x} \tag{4.2.4}$$

In the same fashion we obtain with central differences

$$\begin{aligned} \frac{\partial^2 f}{\partial x_i \partial x_j} &\approx \frac{\left[\frac{f(x_i+\Delta x, x_j+\Delta x)}{2\Delta x} - \frac{f(x_i-\Delta x, x_j+\Delta x)}{2\Delta x}\right] - \left[\frac{f(x_i+\Delta x, x_j-\Delta x)}{2\Delta x} - \frac{f(x_i-\Delta x, x_j-\Delta x)}{2\Delta x}\right]}{2\Delta x} \\ &\approx \frac{f(x_i+\Delta x, x_j+\Delta x) - f(x_i-\Delta x, x_j+\Delta x) - f(x_i+\Delta x, x_j-\Delta x) + f(x_i-\Delta x, x_j-\Delta x)}{4\Delta x^2}. \end{aligned} \tag{4.2.5}$$

For i=j this simplifies to

$$\frac{\partial^2 f}{\partial x_i^2} \approx = \frac{f(x_i + 2\Delta x) - 2f(x_i) + f(x_i - 2\Delta x)}{4\Delta x^2} \tag{4.2.6}$$

For the elements of the Hessian matrix in eq. 4.2.1 it follows

$$\begin{aligned} \frac{\partial^2 \mathcal{L}}{\partial x_i \partial x_j} &\approx \frac{f(x_i+\Delta x, x_j+\Delta x) - f(x_i-\Delta x, x_j+\Delta x) - f(x_i+\Delta x, x_j-\Delta x) + f(x_i-\Delta x, x_j-\Delta x)}{4\Delta x^2} \\ &- \sum_{k=m_e+1}^{m} \left[\lambda_k \frac{g_k(x_i+\Delta x, x_j+\Delta x) - g_k(x_i-\Delta x, x_j+\Delta x) - g_k(x_i+\Delta x, x_j-\Delta x) + g_k(x_i-\Delta x, x_j-\Delta x)}{4\Delta x^2}\right] \\ &+ \sum_{l=1}^{m_e} \left[\mu_l \frac{h_l(x_i+\Delta x, x_j+\Delta x) - h_l(x_i-\Delta x, x_j+\Delta x) - h_l(x_i+\Delta x, x_j-\Delta x) + h_l(x_i-\Delta x, x_j-\Delta x)}{4\Delta x^2}\right] + \mathcal{O}(\Delta x) \end{aligned} \tag{4.2.7}$$

The expression in eq. 4.2.7 is the fundament for the computation of the Hessian matrix. However, the indicated order of accuracy is not of practical use. It establishes only a theoretically best possible value. Because, when implemented in a computer code, numerical effects dominate the accuracy of the result and further degrade it. It is, then, time to take a look at efficient and effective computing.

4.2.2 Efficient Computing

Two considerations drive the effort to be invested in the implementation of the Hessian matrix calculation. First of all, there is the need to compute most exact and reliable information. Since the Hessian matrix information is foreseen for consecutive second order sensitivity analysis, it is essential to obtain a Hessian matrix which is not degenerate, but provides second order gradient information of certain accuracy. This is even more crucial for finite difference methods with decaying exactness.

The second aspect is efficiency in the computation of the elements of the Hessian matrix. With an increasing dimension of the nonlinear program, meaning an increasing number of parameters as well as constraints, the number of operations required for a straightforward computation of the finite differences grows exponentially.
Various methods appear to be suitable to support efficient computation of the Hessian elements. First of all, it is advised to exploit the super-block sparsity of the Hessian by evaluating the correlations between x_i, x_j and the constraints. If these a-priori prove certain elements to be zero, then costly computation can be avoided. The same hold for components, where the Lagrange multiplier is zero.
Other economization strategies suggest to take advantage of system parameter decoupling and to calculate multiple elements at the same time. One such strategy is described by Curtis, Powell and Reid [19]. This can significantly reduce the number of required function calls, but always depends on the structure of the particular application.

The numerical calculation of the Hessian matrix elements is prone to several error sources. The use of finite differences triggers truncation and roundoff errors. The magnitude of the applied variation Δx is crucial to the numerical accuracy of the approximation. For overly large values the approximation will suffer from enormous truncation errors. On the other hand, if the magnitude of the variation is very small, then this will cause the round-off errors to dominate the accuracy [40].

A recommendation for appropriate variations in a second order system that best balances the contrary interests, is

$$\Delta x \sim \epsilon_f^{1/3} \Delta x_c \tag{4.2.8}$$

given in [59].
In this context ϵ_f is the machine precision of the computer processor. The variation Δx_c is either the parameter value if larger than one, or one itself.

4.3 Inclusion of Boundary Conditions

The constraint output that is made available by the optimization algorithm after successful termination, is incomplete in so far, as the data is limited to the constraint subset $\mathcal{C}_w^*$. For parameterized sensitivity analysis the subset and derived quantities, like the Jacobian matrix, need to be expanded to cover the full set of active constraints $\mathcal{C}_a^*$. The consecutive instructions detail the required measures.

Not included are the state bound constraints, which limit the allowed range of system parameters. They are inequalities and always defined in pairs of two constraints for each parameter $\mathbf{x}$, one for the lower bound $\mathbf{x}_{lb}$ and one for the upper bound $\mathbf{x}_{ub}$. If not applicable, a bound can be set to an infinite value.

According to 3.2.11 and 3.2.12 their formulation is

$$\mathbf{c_{lb}} \quad : \quad x_i - (x_i)_{lb} \geq 0, i \in E^n$$

$$\mathbf{c_{ub}} \quad : \quad (x_i)_{ub} - x_i \geq 0, i \in E^n$$

General constraint as well as parameter values of the NLP program are scaled already. Because of this and the linearity of the boundary conditions, it is sufficient to utilize the related factors for scaling of the new boundary constraints. The idea of separate scaling can therefore be abandoned.
This results in

$$[(\mathbf{c})_{\mathbf{lb}}]_s : \quad [x_i]_s - [(x_i)_{lb}]_s \quad \geq 0, i \in E^n \tag{4.3.9}$$

$$[(\mathbf{c})_{\mathbf{ub}}]_s : \quad [(x_i)_{ub}]_s - [x_i]_s \quad \geq 0, i \in E^n. \tag{4.3.10}$$

All state bound constraints, which are zero at the optimal solution are extracted to form $\mathcal{B}^*$ and are added to the set of active working constraints $\mathcal{C}_w^*$, which is then called $\mathcal{C}_a^*$. We derive this set with respect to the system states and obtain the Jacobian matrix. Due to practical considerations we differentiate the equations of the state bound constraints analytically. The gradient for lower bound constraints is

$$\frac{\partial c_{j\ lb}}{\partial x_i} = +1 \text{ for } i = j;\ i, j \in E^n \tag{4.3.11}$$

and for upper bound constraints

$$\frac{\partial c_{j\ ub}}{\partial x_i} = -1 \text{ for } i = j;\ i, j \in E^n. \tag{4.3.12}$$

In case of $i \neq j$ there is no correlation and the gradients are zero:

$$\frac{\partial c_{j\ ub}}{\partial x_i} = \frac{\partial c_{j\ lb}}{\partial x_i} = 0 \text{ for } i \neq j. \tag{4.3.13}$$

Column-by-column the correct gradients can be appended to the Jacobian matrix now.

In the same manner the state bound constraints are included in the calculation of the Hessian matrix. Therefore, we analytically differentiate the constraints and get

$$\frac{\partial^2 c_{i\ b}}{\partial x^2} = \frac{\partial^2 c_{i\ b}}{\partial x \partial \epsilon} = \frac{\partial^2 c_{i\ b}}{\partial \epsilon^2} = 0;\ c_{i\ b} \in \mathcal{B}^*. \tag{4.3.14}$$

Hence, the state bound constraints can be neglected in the calculation of the Hessian matrix.

Next comes the gradient $\frac{\partial c_b}{\partial \epsilon}$ with respect to the perturbation parameter. The assertion

$$\frac{\partial c_b}{\partial \epsilon} = 0. \tag{4.3.15}$$

is true for practical applications, since state bound sensitivity would generally not be examined for singular evaluation nodes, but by use of a path constraint.

With these measures the optimizer output is qualified for use in post-optimality analysis.

The same procedure can theoretically be employed to update vectors and matrices for second order sensitivity analysis, based on first order analysis predictions about the active set composition of the perturbed solution. However, one should bear in mind that this also requires the update of the parameter vector. The parameters of the updated nominal vector $\bar{\mathbf{x}}^*$ have to fulfill the newly included boundary conditions

$$c_b : \ = [\bar{x}_i^*]_s - [(x_i)_b]_s \ = 0, i \in E^n. \tag{4.3.16}$$

Therefore, the procedure should not be utilized for complex problems, since the update is incomplete and, thus, increases the risk of erroneous sensitivity coefficients. The full update of Jacobian and Hessian matrix as well as the perturbation related gradients for nonlinear constraints requires substantial computation and would ultimately lead to an optimizer-like iterative method, but not to a slim POA algorithm.

4.4 Lagrange Multiplier Accuracy

We recollect the description of the NLP program in sec. 2.1.1. The fundamental equations explain the importance of Lagrange multipliers, which are part of the KKT vector and, thus, define the optimization problem. Securing their accuracy is both a critical and also sometimes tedious task, in the sense that only for a truly converged optimum the Lagrange multipliers are unique [22].

Fortunately, the relevance of Lagrange multiplier accuracy is somewhat different when the quality of PSA results is concerned. Hallman [44] relies on them for each and every analysis. But the derivation in sec. 2.1.2, specifically, eq. 2.1.13, shows that the sensitivity gradients $\frac{\partial \mathbf{x}}{\partial \epsilon}$ of the NLP parameters can be computed without using the Lagrange multipliers. The same holds for first order cost function sensitivity $\frac{\partial f}{\partial \epsilon}$.
Lagrange multipliers become relevant when computing the complete KKT point sensitivity and when performing second order analyses. They form the core of the confidence radius estimation and are a key element in the computation of the Hessian matrix.

In any case, it is advisable to consult the first order optimality conditions to clarify the status of an obtained solution, for instance, by computing $(\tilde{\mu}^*)^T \cdot c_a^*$ as a low level optimality criterion. For a converged solution its product is supposed to equal zero. Deviation from zero indicates suboptimality and relativizes POA results.

It is difficult to exactly solve the NLP program and accurately invert the required matrices in order to calculate the Lagrange multipliers:

$$\left(\frac{\partial c_a}{\partial x}\right)^T \tilde{\mu}^* = -\frac{\partial f}{\partial x}. \tag{4.4.17}$$

Among other methods suggested in [37], [7], singular value decomposition is the prime choice in this context.
For a decomposition of the active set Jacobian matrix

$$\frac{\partial c_a}{\partial x} = U \Sigma V^T \tag{4.4.18}$$

with U, V being orthonormal matrices, the pseudo-inverse can be used to compute the Lagrange multipliers as

$$\tilde{\mu} = -V \bar{\Sigma}^{-1} U^T \frac{\partial f}{\partial x} \tag{4.4.19}$$

where $\bar{\Sigma}$ is a diagonal matrix, which contains the singular values of the Jacobian $s = diag(\Sigma)$ down to a certain threshold.

Equation 4.4.17 represents an over-determined linear system. Generally, Σ is not singular and the system can immediately be solved. Occasional degeneracy in the Jacobian, however, can make it necessary to zero small singular values [59]. This is related to possible linear dependency of constraints, which can not be ruled out for optimal control problems in engineering. Parameter duplication by the transcription method is also one possible source for such dependencies.

Proper scaling is just about as important for the Lagrange multipliers as it is for any other variable or constraint. Degradation of the accuracy has been reported in detail in [6], [44]. Practical use shows that the methodology addressed in sec. 2.3 provides sufficient relieve in order to make the effects secondary with respect to multiplier accuracy.

Chapter 5

Example 1: Hopper Optimal Re-Entry Analysis

It is a continuing effort in the spacecraft community to develop concepts for more efficient, more economic and more reliable launcher vehicles. One of them is known under the name HOPPER [21]. It is a reusable launcher concept with a suborbital vehicle that deploys an upper stage at an altitude of about 130 km to 140 km into an intermediate orbit. Thanks to its winged configuration, it afterwards lands horizontally on a runway, while the upper stage delivers the payload into the target orbit. With Kourou, French-Guiana being the launch site, the foreseen landing site for missions with payload bound for equatorial orbits is on Ascension Island in the Atlantic Ocean.

The optimization of HOPPER trajectory segments [67], of end-to-end trajectories [73], possibly with multiple branches [31], [74], has been and still is a topic of

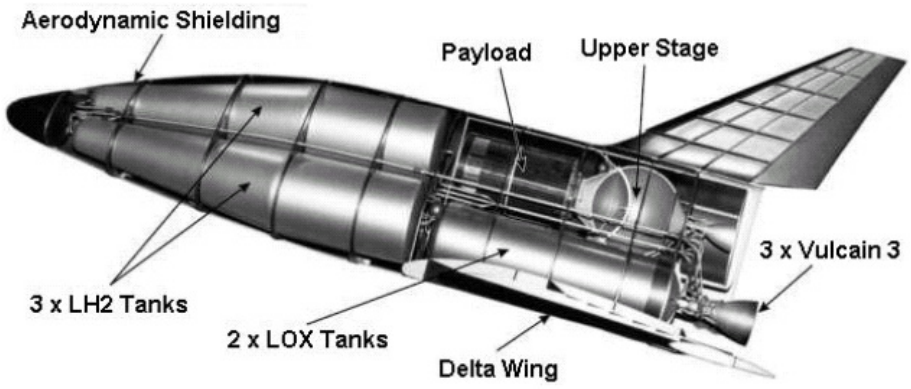


Fig. 5.0.1: The HOPPER reusable launch vehicle

active research. Nominal conditions are assumed when computing such reference trajectories. The outcome is then processed in order to develop suitable controllers. Still today, it is common practice to establish the controller performance by running a series of Monte Carlo simulations, introducing uncertainties in the model that had not been considered during the optimization [68].

The assessment of controller stability is necessary because of inaccuracies in the model equations and uncertainty of model parameters. For instance, in real life the position of a vehicle can only be determined within certain tolerances. The gravitational model might as well be inaccurate. And most important of all, the atmosphere information and the actual aerodynamic properties of the lifting body might vary from results of computational fluid dynamics analysis.

Following, we will take a look at the re-entry leg of the HOPPER mission and execute a parameterized sensitivity analysis to come up with a first estimate on the impact of uncertainties in the model.

5.1 Description of Model and Mission

The trajectory segment under consideration covers the re-entry leg of the mission. For the optimization it is modeled as a 3-DOF system with the state vector

$$\mathbf{x} = [r, \lambda, \delta, v, \gamma, \chi]^T$$

representing the radius, geocentric longitude and declination, relative velocity and flight-path angle and azimuth. The re-renty is unpropelled. Therefore, the mass of the vehicle is considered constant throughout the flight. The equations of motion are given as subsets of kinematic equations

$$\dot{r} = v \cdot \sin\gamma \tag{5.1.1}$$

$$\dot{\lambda} = \frac{v \cdot \cos\gamma \cdot \sin\chi}{r \cdot \cos\delta} \tag{5.1.2}$$

$$\dot{\delta} = \frac{v \cdot \cos\gamma \cdot \cos\chi}{r} \tag{5.1.3}$$

and of force equations

$$\dot{v} = a_{h,x}\cos\gamma\cos\chi + a_{h,y}\cos\gamma\sin\chi - a_{h,z}\sin\gamma \tag{5.1.4}$$

$$\dot{\gamma} = -\frac{(a_{h,x}\cos\chi + a_{h,y}\sin\chi)\sin\gamma - a_{h,z}\cos\gamma}{v} \tag{5.1.5}$$

$$\dot{\chi} = \frac{-a_{h,x}\sin\chi + a_{h,y}\cos\chi}{v\cos\gamma} \tag{5.1.6}$$

which are coupled.

The modeling of gravitational acceleration is limited to J_2. This measure has been taken for the sake of the intended exemplary use of the re-entry case. The atmosphere model is US Standard 62.

The vehicle is controlled via the angle of attack α and the bank angle μ. Both are not explicitly mentioned in the equations of motion, but are part of the formulation of the acceleration vector $\mathbf{a_h} = [\mathbf{a_{h,x}}, \mathbf{a_{h,y}}, \mathbf{a_{h,z}}]^{\mathbf{T}}$. It sums up the gravitational and aerodynamic forces and is given in the horizontal frame [61].

In general, optimization of re-entry trajectories concentrates on maximizing the cross- or downrange. However, since the end conditions are given by the location of Ascension Island and the specifications of the terminal entry interface [67], the final state is firmly constrained.
In return, there need to be other degrees of freedom. Therefore, the starting conditions are not fixed to the culmination point of the suborbital trajectory. But the initial longitude λ_0 becomes the objective to be minimized during the optimization, thus, maximizing the flight arc. Further initial state conditions can be found in tab. 5.1.1. They conform with typical HOPPER culmination conditions for a geostationary orbit transport.
The total mass of HOPPER at re-entry is $m_0 = 63,335\ kg$.

Altitude h_0	: 140 km
Longitude λ_0	: *optimizable*
Declination δ_0	: 3.2°
Flight-path velocity v_0	: 4960.2 m/s
Flight-path angle γ_0	: 0.0°
Flight-path azimuth χ_0	: 100.0°

Tab. 5.1.1: Initial conditions

The phase structure applied to this trajectory optimization problem is depicted in tab. 5.1.2. The division into five phases mates constraint necessities and aerodynamic coefficient-steadiness precaution for the sake of an efficient gridding. More details on the aerodynamic database can be found in [74] and [25]. Reference on the

	Time [sec]	Control Limits	Path constraints	Boundary constraints
1	0-190	$\alpha = 40°$ $-80° \leq \mu \leq 80°$	$q \leq 40\ kPa$ $\dot{Q} \leq 450\ kW/m^2$	$[r_0, \delta_0, v_0, \gamma_0, \chi_0]$ $n = 4.5\ g$
2	190-544	$0° \leq \alpha \leq 40°$ $-80° \leq \mu \leq 80°$	$q \leq 40\ kPa$ $\dot{Q} \leq 450\ kW/m^2$ $n \leq 4.5\ g$	Ma = 8.00
3	544-825	$0° \leq \alpha \leq 40°$ $-80° \leq \mu \leq 80°$		Ma = 3.92
4	825-939	$0° \leq \alpha \leq 40°$ $-80° \leq \mu \leq 80°$		Ma = 2.31
5	939-1049	$0° \leq \alpha \leq 40°$ $-80° \leq \mu \leq 80°$	$q \leq 40\ kPa$	$h_f = 15\ km$ $\lambda_f = -14°23'$ $\delta_f = -7°58'$ Ma = 1.50

Tab. 5.1.2: Phase structure of HOPPER re-entry optimization

phase separation can also be found in [67].

There are a number of structural, thermal and thermo-dynamic restrictions which commonly need to be preserved. In the given case these are limits on the allowed load factor n, the heatflux $\dot{Q}$ and the dynamic pressure q. The actual restrictions are assigned in tab. 5.1.2.

5.2 Parameter Perturbations in the Model

The given mathematical equations do not accurately describe the true vehicle dynamics, but represent an approximate model. From a controllability perspective, the degree of inaccuracy along with safety margins determines, whether the optimized solution is only theoretically feasible or can be flown in practice. Therefore, it is crucial to assess the effects of inaccuracies and parameter uncertainties on the nominal trajectory.
The design parameters and model properties that are prone to perturbations can be separated into three groups according to their character. There are uncertainties related to the vehicle itself. Respective design parameters are among others, system masses and aerodynamic coefficients. Since the re-entry flight is unpowered, the propulsion specifications are not an issue.

Then, there are navigational perturbations. In our case they are considered in the determination of the initial state vector elements. Perturbations can arise as conventional inaccuracies, which, according to [41], are usually no larger than 3 m/s for the velocity, and 800 m for the position. Or they can be included to study trade-off potential for system staging. For the latter, larger variations of the design parameters can be expected.
Environmental inaccuracies compose a third group of uncertainties. These can, for instance, be atmospheric or gravitational anomalies. In general, the earth model is well established and therefore highly accurate. Nonetheless, because of simplifications in the optimization model, the assessment of environmental inaccuracies seems reasonable given the controllability of the vehicle.
A set of design parameters for post-optimality analysis can be found in tab. 5.2.3.

Symbol	**Description**	**Reference value**
ϵ_r	Initial altitude [km]	140
ϵ_δ	Initial declination [°]	3.2
ϵ_v	Initial velocity $[\frac{m}{s}]$	4960.2
ϵ_χ	Initial flight-path azimuth [°]	100.0
ϵ_{cd}	Aerodynamic drag deviation factor [ ]	1.0
ϵ_{cl}	Aerodynamic lift deviation factor [ ]	1.0
ϵ_{J_2}	Zonal harmonics J_2 [ ]	0.0010826
ϵ_m	Vehicle mass [kg]	63335.0
ϵ_{hfl}	Heatflux limit $[\frac{kW}{m^2}]$	450.0

Tab. 5.2.3: Parameter selection for POA

5.3 The Optimized Nominal Trajectory

Using the CAMTOS algorithm, the optimization of the HOPPER re-entry trajectory can not be carried out in a straightforward fashion, but needs to be developed iteratively. The problem is first solved neglecting path constraints. They are incorporated in a consecutive step.
The final nonlinear program consists of a total of 315 parameters and 225 constraints of which 195 are equalities. The applied scheme represents a shooting method.

The optimal trajectory is shown in fig. 5.3.2. The cost function is minimized to a value of $f^* \equiv \lambda_0^* = -36.139°$, which falls short of the value computed for the cul-

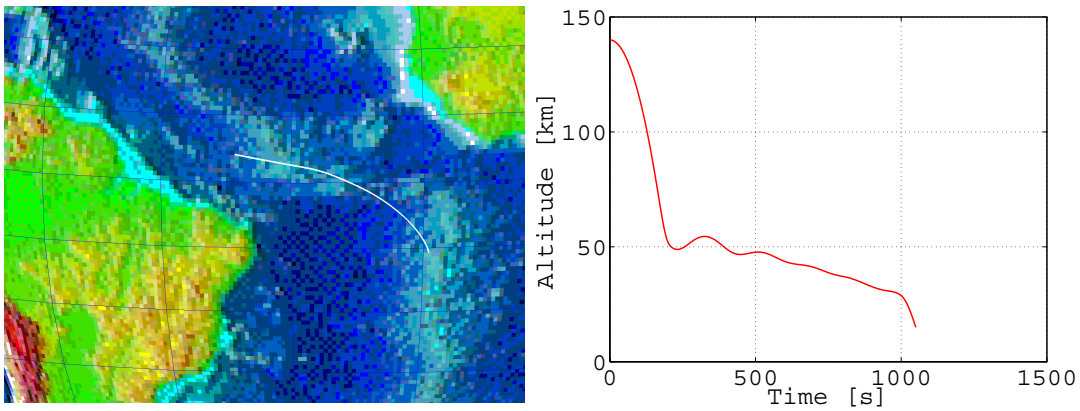


Fig. 5.3.2: The HOPPER optimal re-entry trajectory

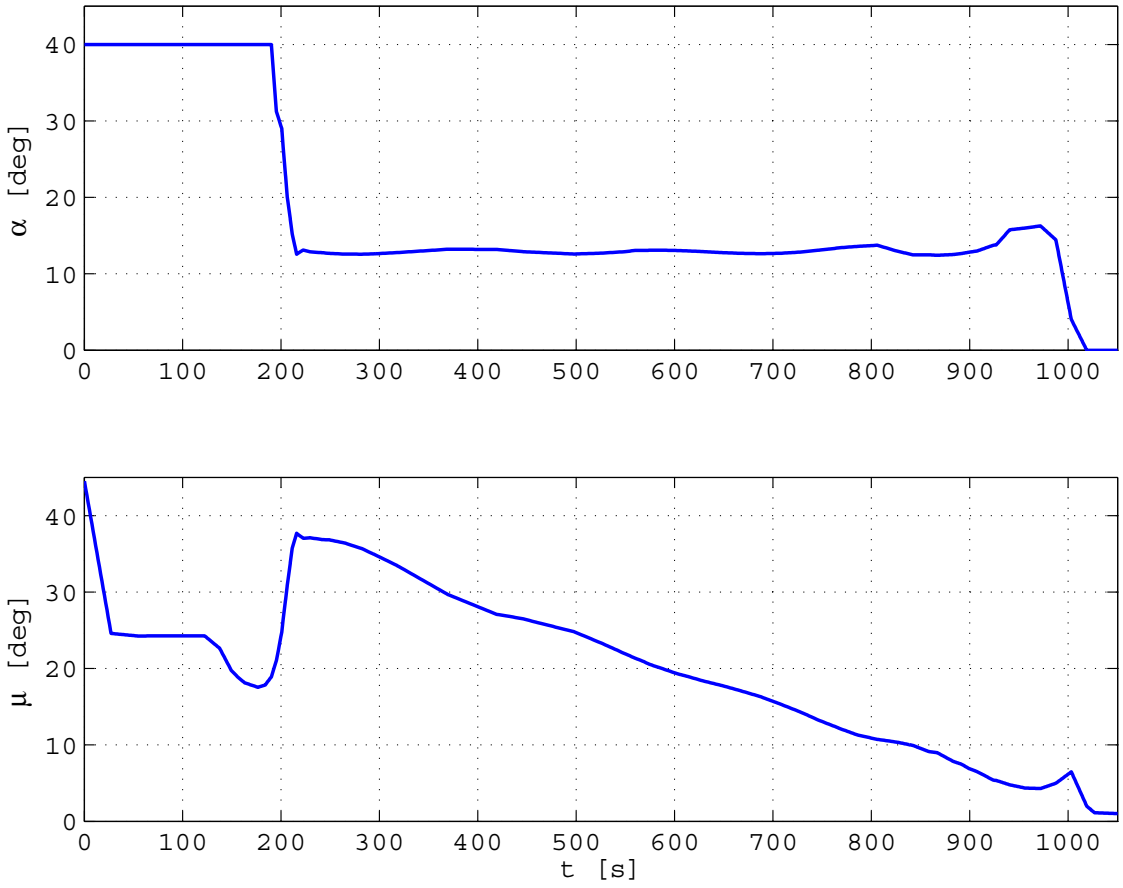


Fig. 5.3.3: Control history for the optimal re-entry

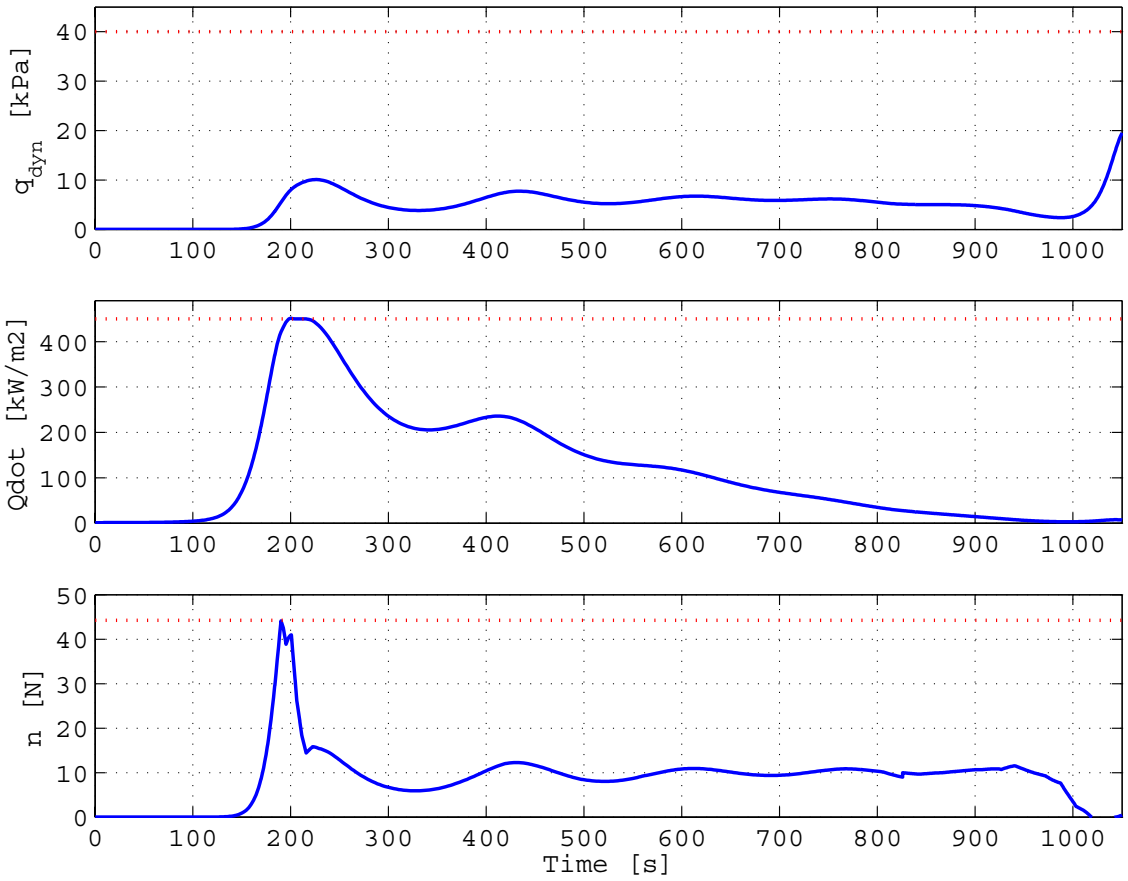


Fig. 5.3.4: Heatflux, dynamic pressure and load factor graphs

mination point of the branched trajectory in [31]. Minor deviations are attributed to updates in the model and mission requirements, including tighter restrictions. It shows that the current HOPPER configuration requires basically almost maximum downrange from the stage separation point in order to reach the TAEM (Terminal Area Energy Management) interface, Ascension Island respectively.

The altitude profile in fig. 5.3.2 shows a number of receding skips. They are not unusual for HOPPER re-entry trajectories. Damping of these skips is coupled with tighter limits on γ_{ub} and the heatflux limit. It results in more agile profiles of the controls. As has been described in [67], the magnitude of the control variations versus the definition of the skips can be influenced by a Lagrange term in the cost

function. In an intermediate optimization step the term

$$L = \int \left[k_\alpha(t) \cdot \left(\frac{d\alpha}{dt} \right)^2 + k_\mu(t) \cdot \left(\frac{d\mu}{dt} \right)^2 \right] dt \tag{5.3.7}$$

is included in order to serve just this purpose. Appropriate values for the coefficients k_α and k_μ depend on the degree of damping and usually range between 10^{-4} and 10^{-1}. For the presented trajectory they had a magnitude of about 10^{-3}.
The optimal control profiles are depicted in fig. 5.3.3.

The load factor n is the first path constraint to saturate (see fig. 5.3.4). The peak is steep and limited to the phase bound 1-2. Immediately afterwards, the heatflux reaches 450 kW/m^2 which activates the constraint. It remains active for about 20 seconds. Then follows the peak of the dynamic pressure. But due to the heatflux restriction, the dynamic pressure stays well below critical levels.
The tight limit on the heatflux turns out to be a major hurdle during the optimization process. It appears to dampen the skipping, but also to significantly shorten the mission arc.

5.4 Results of the Post-Optimality Analysis

As a first step a post-optimality analysis of the nominal solution is conducted with respect to variations in the design parameter ϵ_δ, which defines the initial geocentric declination. For reference, an increment of $\Delta\epsilon_\delta = +0.24°$ is assumed (see tab. 5.4.4), which correspondents to a deviation of approximately $+25$ km. According to second order PSA, this shortens the starting longitude by about one degree to a value of

$$\lambda_0^{p2} = f^{p2}(\Delta\epsilon_\delta = 0.24) = -35.219°. \tag{5.4.8}$$

The confidence radius $(\epsilon_\delta)_{c+} = +0.74°$ is considerably larger than the foreseen variation. In the other direction, for negative perturbations in ϵ_δ, the analysis delivers results with a confidence radius of $(\epsilon_\delta)_{c-} = -0.75°$.
In both cases, a foreseen perturbation of $\Delta\epsilon_\delta = \mp 0.24$ is much smaller than the confidence radius. This suggests that the active set of constraints remains unchanged for the perturbed trajectories.
Keeping in mind the coarseness of the method, the predictions should produce reasonable, but by no means exact results. Comparison of the predicted f^{p2} with the reoptimized f^ϵ supports this assumption. The normalized errors σ are 20.1% and -17.5%, respectively.

Par.	ϵ_c	$\Delta\epsilon$	$\Delta\epsilon$ [%]	f^* [°]	f^{p1} [°]	f^{p2} [°]	f^{ϵ} [°]	σ [%]
ϵ_δ	-0.74883	-0.24	-7.5	-36.139	-37.048	-37.039	-36.905	-17.5
ϵ_δ	0.74161	+0.24	+7.5	-36.139	-35.229	-35.219	-34.988	20.1

$\sigma = \frac{f^{\epsilon} - f^{p2}}{f^{\epsilon} - f^*}$

Tab. 5.4.4: Convergence radii and cost function prediction for design parameter perturbations

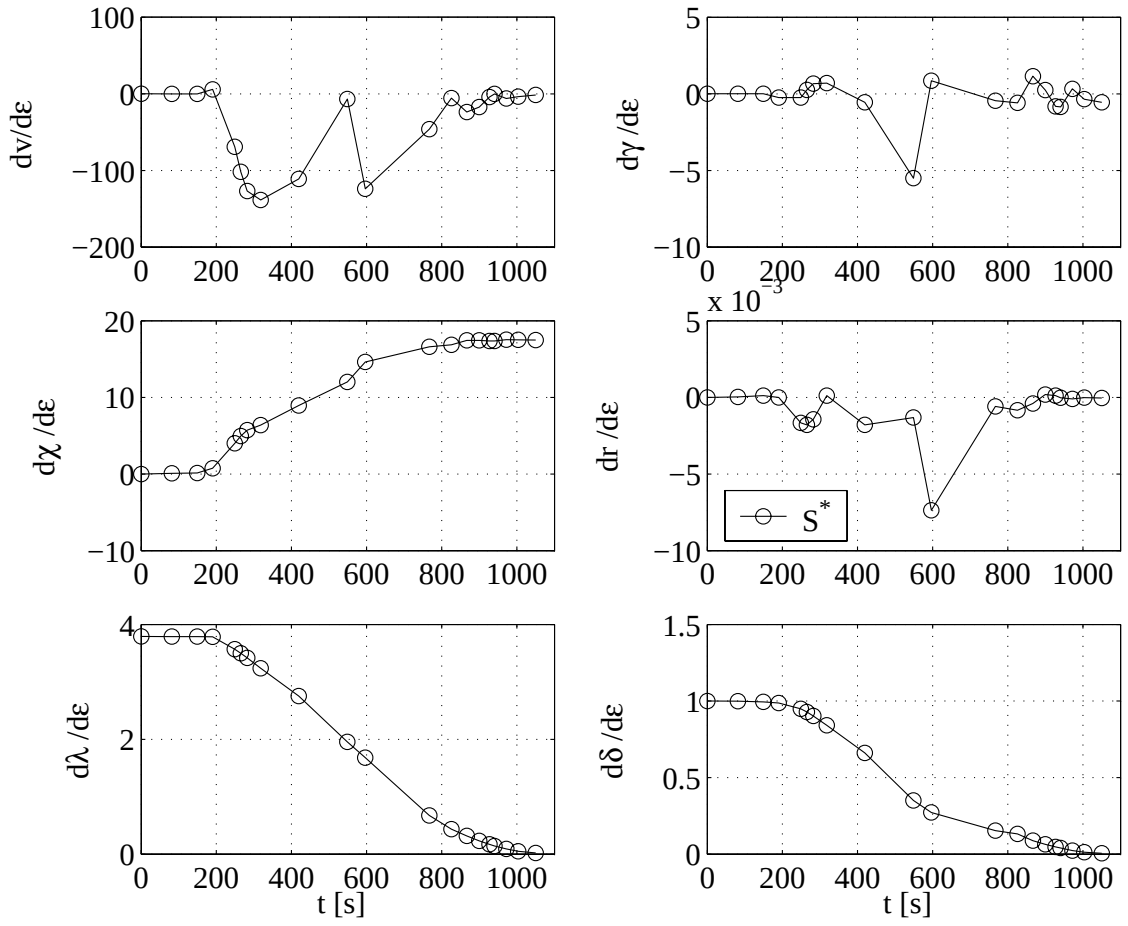


Fig. 5.4.5: State sensitivities S^* for variations in ϵ_δ

In order to further analyze the PSA results, we take a closer look at the individual state sensitivities for $\Delta\epsilon_\delta$. Figure 5.4.5 gives the sensitivities $\mathcal{S}^*$ for the nominal case. They are plotted against the true mission time.
The graph of $\frac{d\chi}{d\epsilon}$ indicates no influence of the perturbation on to early flight-path angle. This had to be expected, since the initial flight-path angle χ_0 is constrained. Conversely, the graph for the longitude shows high sensitivity for the early parameters. It shrinks down to zero towards the end of the mission in compliance with the terminal constraint. This coincides with the expected behavior.

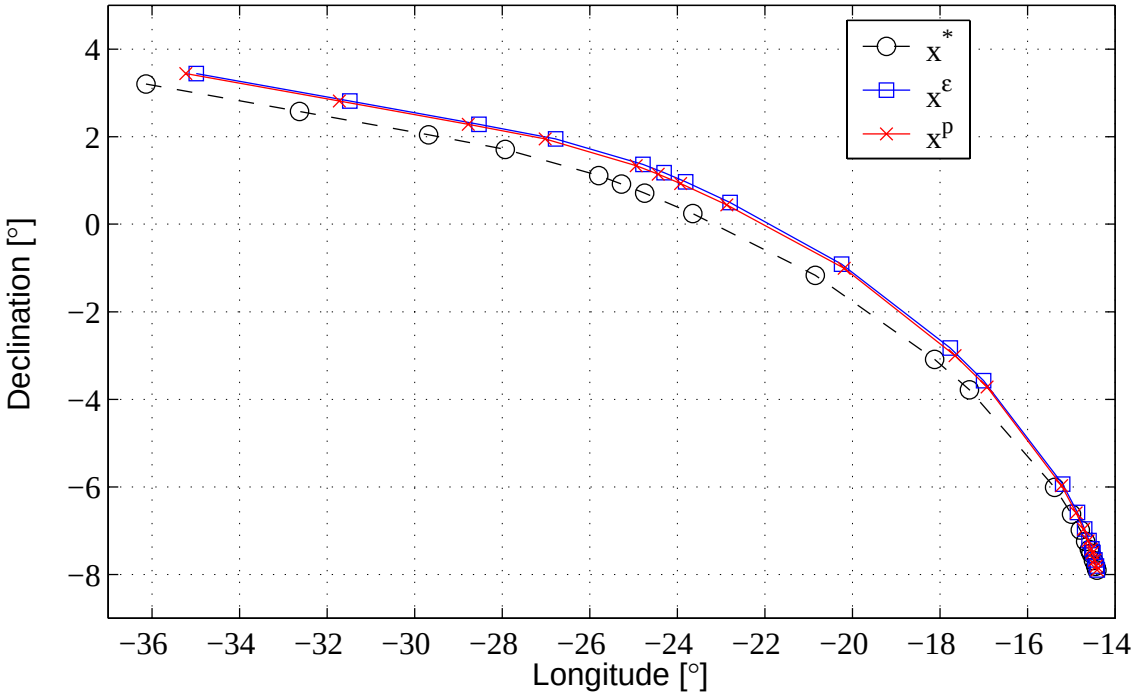


Fig. 5.4.6: Ground track for $\Delta\epsilon_\delta = +0.24$

The sensitivities $\frac{d\lambda}{d\epsilon}$, $\frac{d\delta}{d\epsilon}$ and $\frac{d\chi}{d\epsilon}$ exhibit strictly monotonic development, suggesting a very smooth bending of the trajectory with increasing perturbation.
For further assessment, it is helpful to take a look at the ground track of the re-entry for $\Delta\epsilon_\delta = 0.24°$ in fig. 5.4.6. The predicted trajectory as such almost accurately matches the reoptimized trajectory. The locations of the grid points differ slightly more. In the light of the good overall performance the displayed mild inconsistency seems secondary. The underlying effect, however, is attributed to a cause that the user needs to bear in mind, when interpreting the sensitivity results for the continuous control problem.

The explanation is as follows. The mission has been set up with a normalized time grid. The actual time in seconds for each grid point is a function of the fixed normalized mission time t_n and the parameters that contain the true phase duration (see tab. 5.4.5). Hence, the complete time grid can shift, when perturbation triggers a variation in the time related optimizable NLP parameters.
This does not yet explain, why the trajectory prediction delivers satisfactory re-

	Δt^*	Δt^ϵ	Δt^p	$\Delta t^\epsilon - \Delta t^p$
Phase 1	190.1	190.2	190.2	0.0
Phase 2	358.7	372.7	265.7	7.0
Phase 3	277.0	276.6	285.6	-9.0
Phase 4	114.7	114.9	115.5	-0.6
Phase 5	109.7	109.9	110.0	-0.1
Sum	1050.3	1064.3	1067.0	-2.7

Tab. 5.4.5: Phase durations in seconds with reference $\Delta\epsilon_\delta = +0.24°$

sults, even though the time grid develops differently. The reason is to be found in the ultimately deviating search directions of the PSA algorithm and the NLP solver. Parameterized sensitivity analysis is programmed to produce a feasible solution and move best possible towards an optimal result. But it is not an iterative NLP algorithm. Only for trivial problems could a single step algorithm converge to the optimal solution.
The different deformation of the time grid in phases two and three is also the reason for the dual dip of $\frac{dv}{d\epsilon}$ in fig. 5.4.5, which at first glance gives the false impression that the variations in the velocity would be larger than they actually are.

Conclusively, it can be said that a comprehensive comparison of predicted and re-optimized histories for the verification of the algorithm performance is only possible in the framework of the optimal control problem description under full consideration of time grid shifts and deformation. The plots of state and control profiles against mission time can be found in fig. 5.4.7 and fig. 5.4.8.
A representation with normalized time might only be sufficient for the assessment of pure parameter sensitivity.

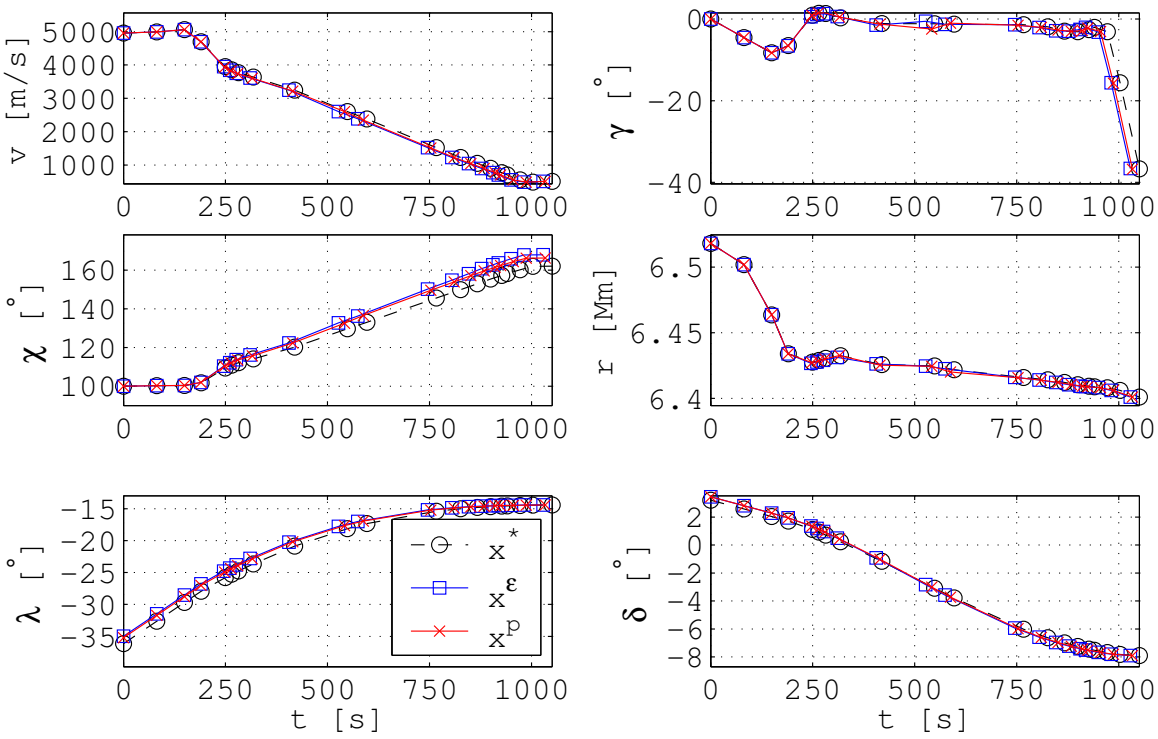


Fig. 5.4.7: State history with reference $\Delta\epsilon_\delta = +0.24°$

The question arises, how accurate the post-optimality analysis is. The deviations between reoptimized control and state histories and the nominal, as well as the predicted histories are computed (see fig. 5.4.9 and fig. 5.4.10). It shows that the sensitivity analysis performs fairly well for states λ, δ and χ. Also for the velocity v the error stays in satisfactory margins.
This does, however, not hold for the remaining two states. Overall, the errors in the predicted radius and flight-path angle profiles are in the order of the variations. Temporally, the region can be narrowed down to the range from about 200 to 700 seconds. This is the mission segment in which the skips occur.
The inferior prediction of the two states in the vertical plane goes back to the control that governs their propagation -the angle of attack. As can be seen in fig. 5.4.10, the estimated angle of attack α^p is inaccurate. In fact, it appears not to be useful at all. The errors $\tau_{\alpha,1}$ have the same magnitude like the variations $\tau_{\alpha,2}$. In contrast, the bank angle profile shows variations $\tau_{\mu,2}$ of up to 4.5° that are reasonably well predicted.

Several of the prior insights have been based on knowledge about the exact solution. It has been made available for reference purposes. But the practical idea

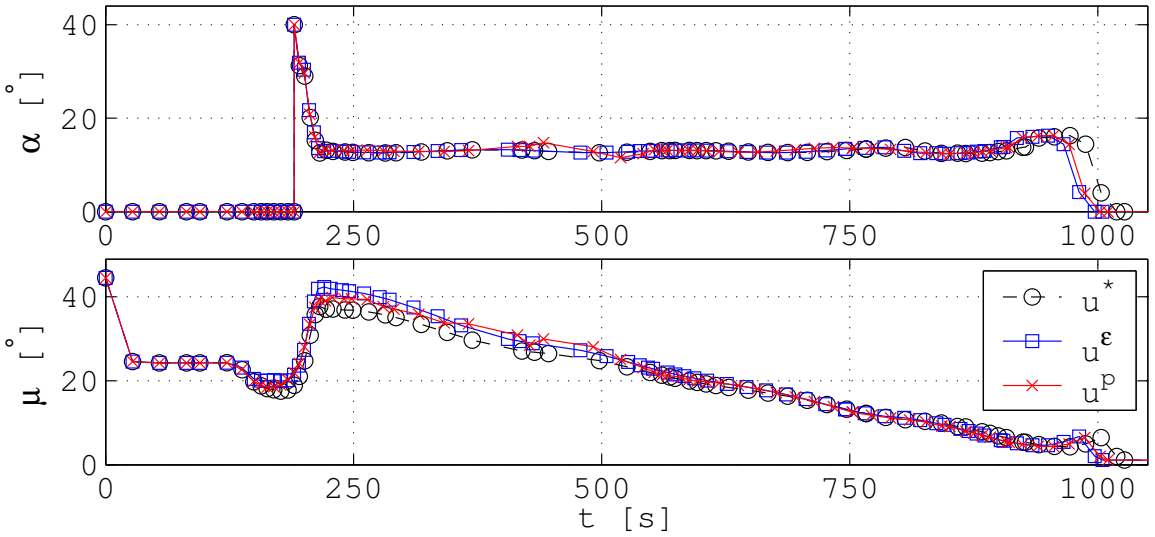


Fig. 5.4.8: Control history with reference $\Delta\epsilon_\delta = +0.24°$

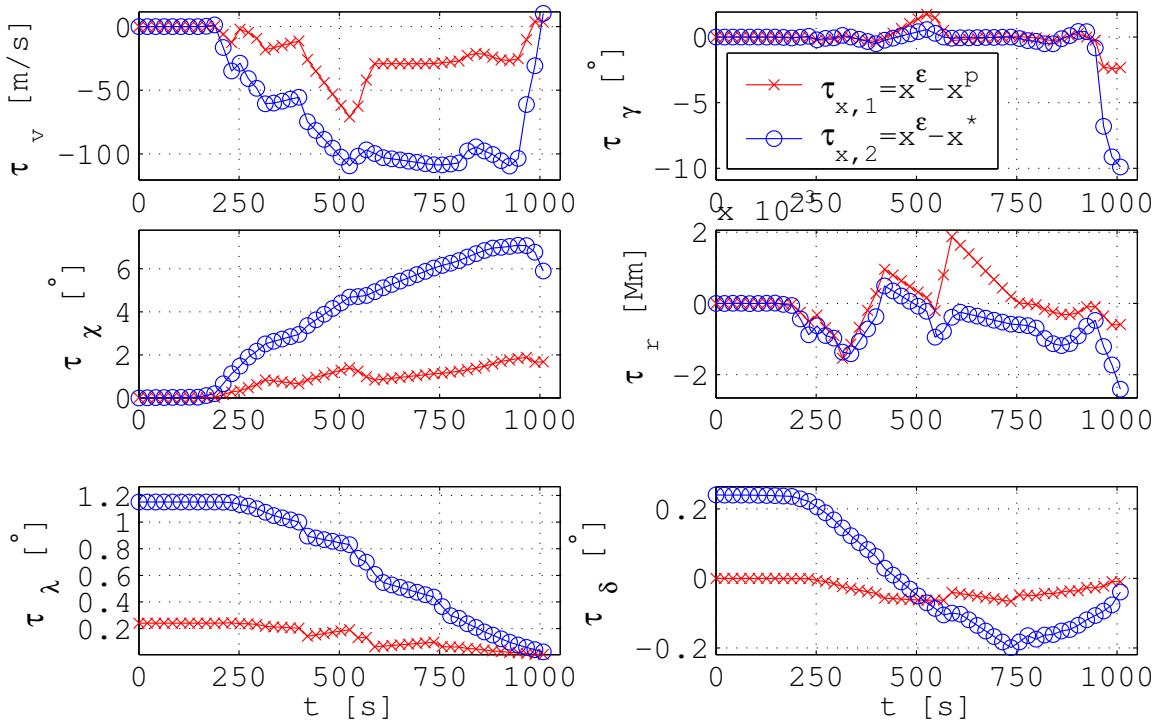


Fig. 5.4.9: State history deviations for $\Delta\epsilon_\delta = +0.24°$

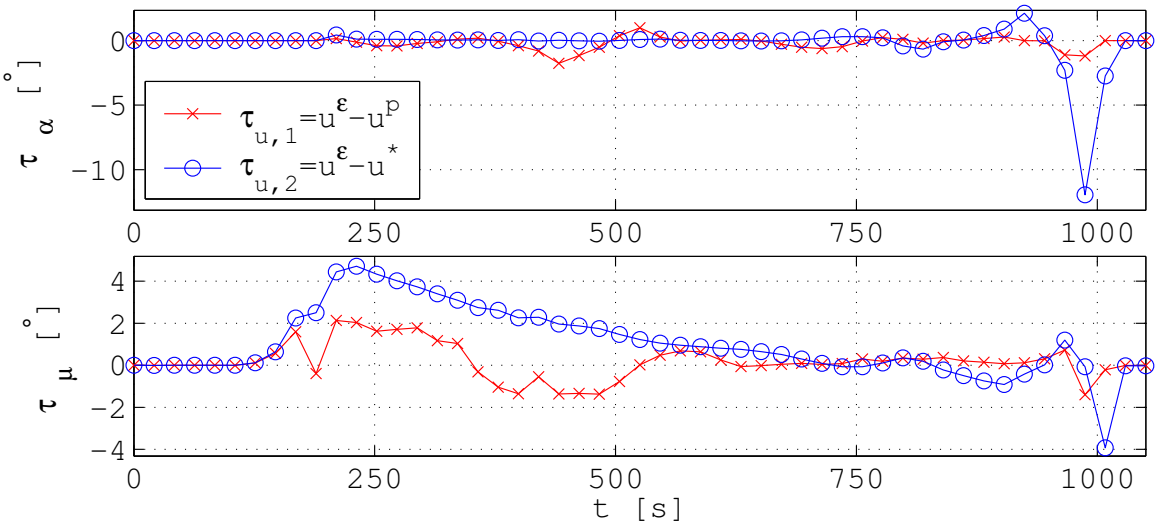


Fig. 5.4.10: Control history deviations for $\Delta\epsilon_\delta = +0.24°$

behind post-optimality analysis is to perform a rapid, approximate assessment of an optimal solution. So the question is, what information can be extracted from the sensitivity data itself. Thereto we turn to the sensitivities of the control parameters in fig. 5.4.11.

The values S^*_μ are all positive. The effect of a variation in ϵ_δ on the bank angle

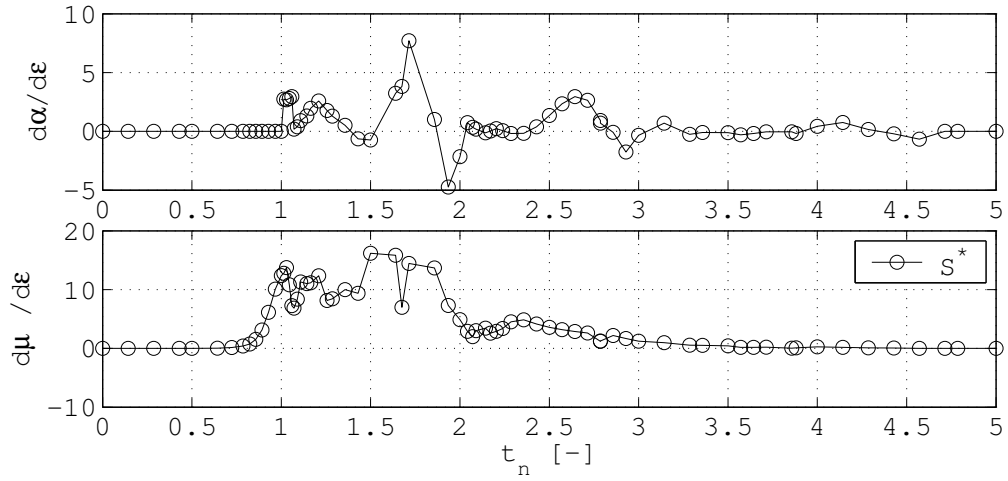


Fig. 5.4.11: Control sensitivities with reference $\Delta\epsilon_\delta = +0.24$

profile is a uniform increase or decrease, causing the trajectory to bend more or less, which is consistent with plausibility considerations.
The highest values are found in phase 2, where they are evenly distributed and form a plateau with increased sensitivity. It has to be expected that, particularly during this phase, the bank angle changes for variations in the initial declination δ_0. Hence, from an optimization standpoint it is advisable to discretize the optimal control problem with a tighter grid in this region.

It also holds for the α-control that the largest sensitivities are computed in phase 2. However, the profile does not feature an even plateau, but shows sharp and narrow peaks with quickly changing signs. The graph visualizes the tendency to give the later skips of the trajectory a more defined shape. Unfortunately, this tendency is not beneficial. The increased drag even thwarts the cost function value. The conflict arises from the linear character of the PSA method, which is not able to accurately trade lift versus drag in a nonlinear system.
The sharp peaks denote the flight regime, in which the trajectory is most sensitive to α-variations.

So far, the analysis has concentrated on a single design parameter, namely ϵ_δ. In a next step we take a look at sets of parameters to rank their influence. The plots in fig. 5.4.12 describe the sensitivity of the optimum cost function to changes in the initial state vector of the optimal control problem. The interpretation is straightforward. A higher initial altitude of the vehicle, for instance, results in a slightly higher initial longitude, moving the starting point of the re-entry trajectory eastward. This is negative for the mission. But in total the influence of perturbations in ϵ_r is small. Changing the orientation of HOPPER by decreasing ϵ_χ has a negative effect on the downrange. It shows, however, that this can easily be compensated by slightly increasing the velocity or decreasing the declination, which both have a more significant influence. Overall, this should be viewed in the context of upper stage performance benefits.
The total sensitivities with respect to initial state vector elements are

$$\frac{d\lambda_0}{d\epsilon_r} = 0.0413\ [^\circ/km]$$

$$\frac{d\lambda_0}{d\epsilon_\delta} = 3.79\ [\ ]$$

$$\frac{d\lambda_0}{d\epsilon_v} = -0.0289\ [^\circ s/m]$$

$$\frac{d\lambda_0}{d\epsilon_\chi} = -1.28\ [\ ]$$

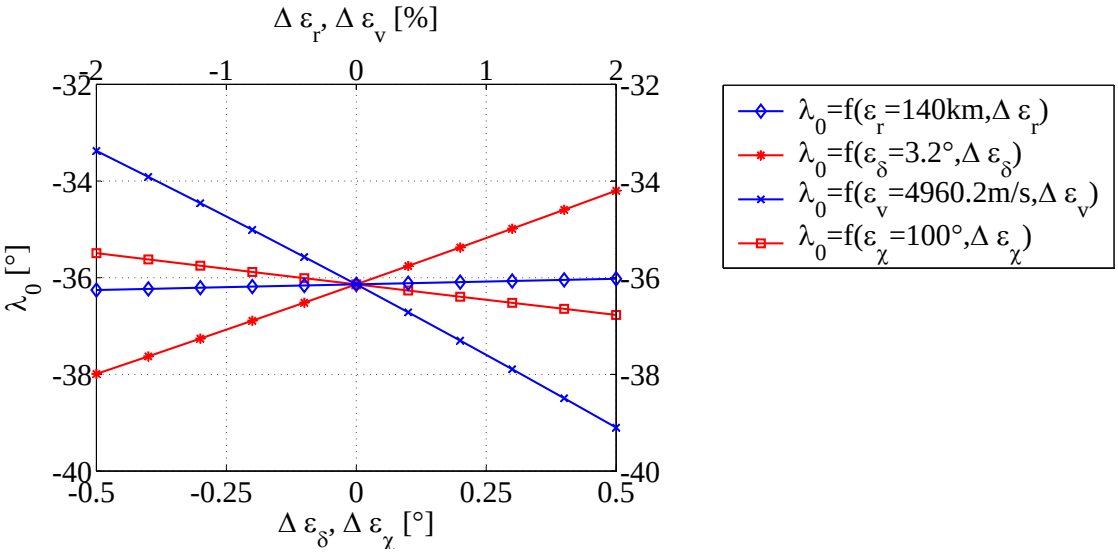


Fig. 5.4.12: Cost function value versus initial state variation

They also provide answers to the question of navigational uncertainties. For an error of 0.05° in the initial declination the landing site Ascension Island, or better the terminal entry interface, can only be reached if a loss of 0.19° can otherwise be compensated.

The plot in fig. 5.4.13 shows the cost function versus variations of the aerodynamic drag and lift coefficient. The lift/drag ratio of the HOPPER vehicle is in the order of 2 during supersonic and hypersonic flight for an angle of attack above 10° [68], as is the case with the presented nominal trajectory. The sensitivity gradients can be utilized to approximate the vehicle's performance for deviating ratios. Thus, an increase of L/D by one percent trades a gain of about 0.9 degrees in the longitude.

With variations of about 0.027° the cost function is very insensitive to common uncertainties of no more than 180 kg in the vehicle mass, which can be taken from fig. 5.4.14. The graph also shows the sensitivity to variations in the heatflux limit. It predicts a gain of about one degree in longitude when raising the limit of $\dot{Q}$ by five percent to 472.5 kW/m^2. This reveals significant potential for improvement in mission performance. Other than changes of the culmination point state vector, as caused by $\Delta\epsilon_\delta$ or $\Delta\epsilon_v$ does this not have an impact on the upper stage performance. Hence, it is worthwhile to take a closer look at the reliability of the prediction for $\Delta\epsilon_{hfl}$. The computed confidence radius is $(\epsilon_{hfl})_{c+} = +58.7\ kW/m^2$. The cost extrapolation with $\Delta\epsilon_{hfl} = +22.5\ kW/m^2$ therefore appears to be well within the reliability limits.

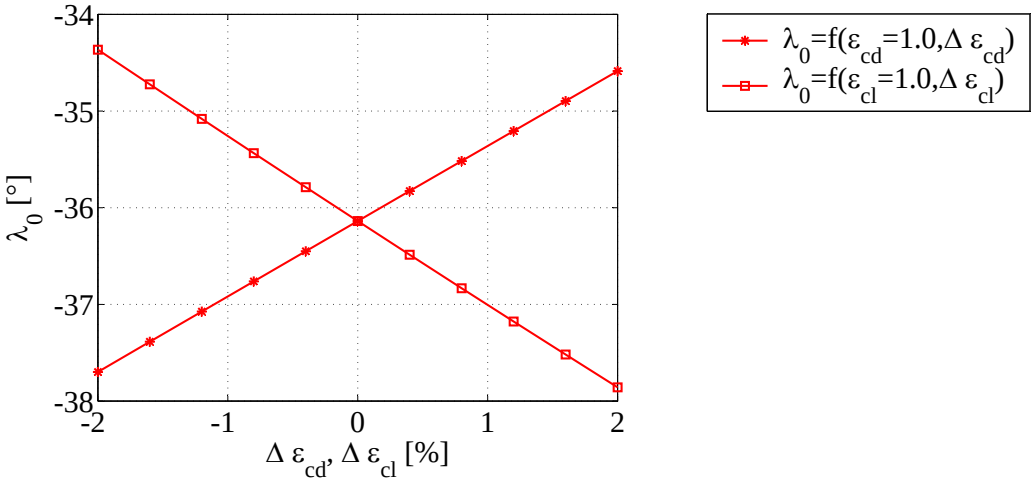


Fig. 5.4.13: Sensitivity with respect to uncertainties in the aerodynamic coefficients

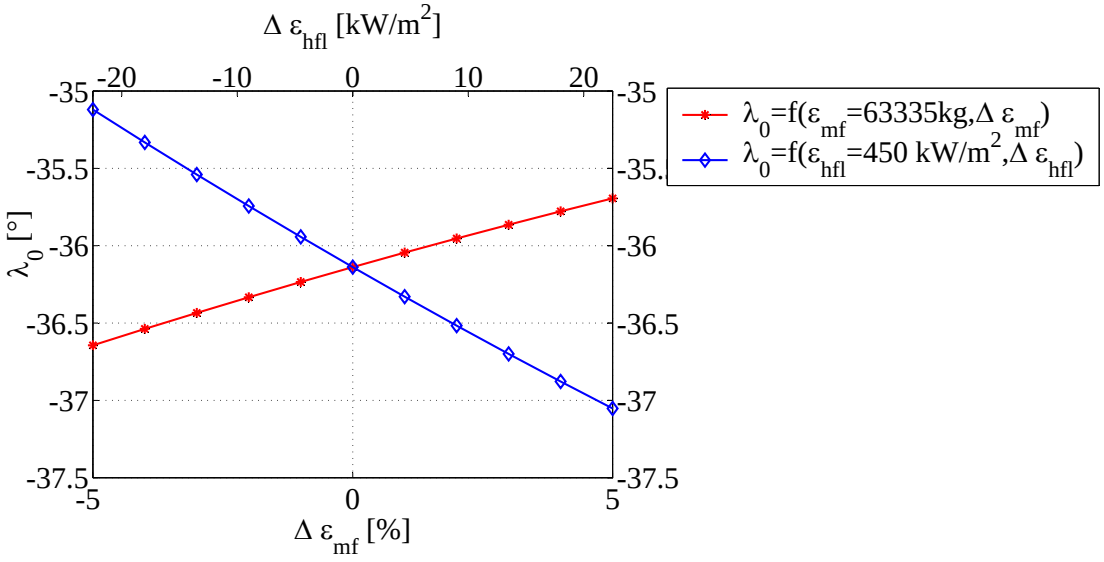


Fig. 5.4.14: Sensitivity with respect to vehicle mass and heatflux limit

A later performed reoptimization of the perturbed trajectory serves as reference. The minimum longitude is $\lambda_0^\epsilon = -36.869°$. The normalized error in the sensitivity-based extrapolation of the optimum cost function is 25.1%.
This error behavior is verified with a collocation based transcription of the optimal

control problem with a significantly larger number of parameters and constraints. The result is essentially the same. Which on one hand means, the collocation based method produces the same quality of result, even though the NLP description is much different, with a completely different set of Lagrange multipliers. On the other hand it underlines the limitations of post-optimality analysis. The degree of nonlinear dependencies and the finiteness of the number of grid nodes limits the performance.

Taking mission planning and engineering perspectives into account, it is not only interesting to perform sensitivity gradient mining, where trade-offs are computed for deviating designs in the vicinity of the nominal configuration, like done earlier. But it is also of value to assess the development of, for instance, the heatflux in a broader range by applying POA, since it has been shown already that it is a key limiting factor for mission performance.

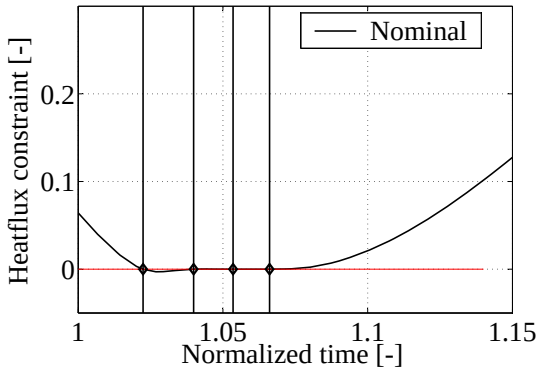
The heatflux constraint at four active evaluation nodes in phase 2 -depicted in fig. 5.4.15 together with the Lagrange multipliers and constraint values- along with a number of bound constraints for saturated α and μ parameters in mission phase 5 compose the set of active inequality constraints of the nominal solution. The set plays a major role in post-optimality analysis when estimating the confidence radius of the computed sensitivity gradients.

For the various investigated design parameters ϵ the smallest radius in the confidence region (see tab. C.0.2) is always caused by one out of three reasons:

- Potential deactivation of heatflux point constraint
- Potential saturation of flight-path angle in phase 2
- Potential punctual saturation of α

An analysis of perturbations in the aerodynamic drag coefficient via ϵ_{cd} produces results with $(\epsilon_{cd})_{c+} = 0.044$. Here, heatflux constraints are identified as the most critical inequalities. We perform two predictions, one for a variation of $\Delta\epsilon_{cd} = 0.02$, equal to a two percent growth of the drag coefficient, which is well below the confidence bound. The other for a variation of $\Delta\epsilon_{cd} = 0.04$. Based on the post-optimality analysis, the first prediction should be trustworthy, the latter one not.

Since POA is based on linear extrapolation of the state parameters without intermediate update of the active set, the predicted heatflux histories of the two cases (see fig. 5.4.16) do not differ significantly. The reoptimized graphs, however, document a dramatic change, as was indicated by the confidence radius. The larger of the two variations causes the heatflux at the evaluation nodes to detach from the limit. The peak moves up the normalized time and by-passes the discretization



Constraint value	Lagrange multiplier
$5.45676 \cdot 10^{-8}$	$4.86278 \cdot 10^{-2}$
$-4.33408 \cdot 10^{-7}$	$9.07964 \cdot 10^{-2}$
$4.03456 \cdot 10^{-7}$	$9.20511 \cdot 10^{-2}$
$1.27493 \cdot 10^{-6}$	$1.05107 \cdot 10^{-1}$

Fig. 5.4.15: Active section of the heatflux constraint grid

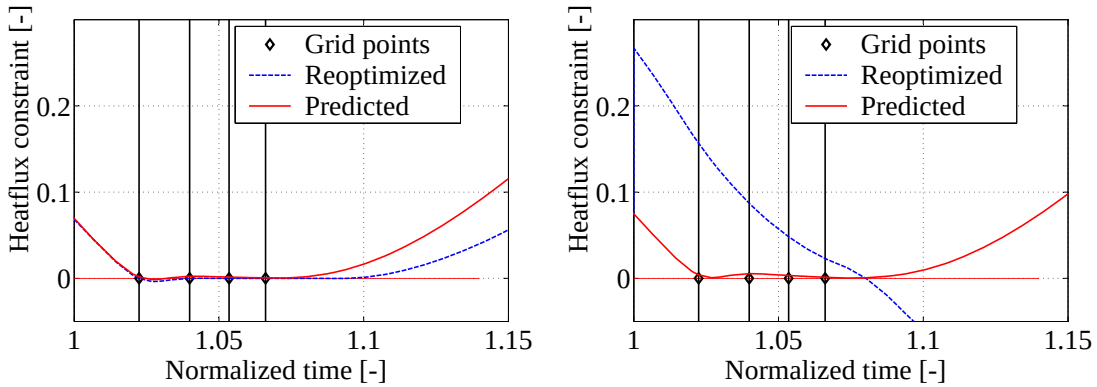


Fig. 5.4.16: Heatflux constraint, $\Delta\epsilon_{cd} = 0.02$ and $\Delta\epsilon_{cd} = 0.04$

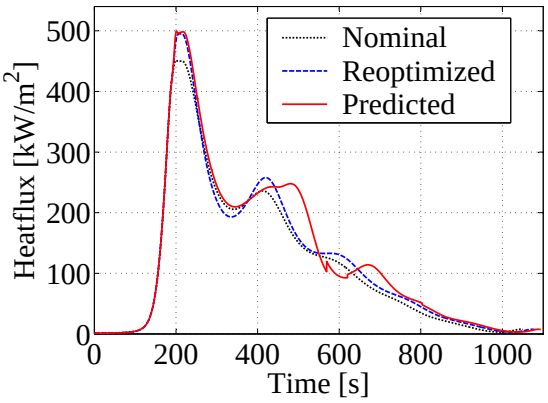


Fig. 5.4.17: Time history of heatflux, $\Delta\epsilon_{hfl} = 45\ \frac{kW}{m^2}$

nodes of the constraint. The result is an infeasible trajectory.
Parameterized analysis is not able to accurately forecast such effects, when the genuine evaluation grid placement does not cover respective areas. But the confidence radii give overall good indication on the operational range of the PSA and constraint/parameter bounds that are eminent.
For the smaller variation $\Delta\epsilon_{cd} = 0.02$ the prediction error in the cost function is 32 %.

There is more to be said about heatflux prediction and confidence radius interpretation. We turn to the impact of variations in the actual heatflux limit and take a look at the graph of heatflux versus mission time in fig. 5.4.17.
The maximum allowed $\dot{Q}$ is raised by 10 %, $\Delta\epsilon_{hfl} = 45\ \frac{kW}{m^2}$ respectively. The POA algorithm takes full advantage of the new margin and adapts the parameters of the NLP problem in such way that a later simulation of the heatflux shows good alignment of the peak with the imposed limit at 495 $\frac{kW}{m^2}$. Once again, heatflux is not a discretized state, but a function of the states. Hence, the good alignment does not come naturally, but is attributed to the POA performance.

The later segment of the graph shows what has already been discussion earlier in this section about the sensitivity of α. The model tends to further define the unconstrained skipping maneuvers because of inexact data about α-sensitivity. Once more it documents the sensitiveness and reduced stability of the angle of attack.

Chapter 6

Example 2: Ariane 5 Post-Optimality Analysis

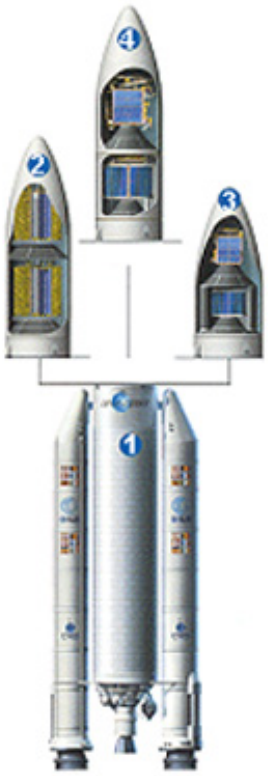
Demand for ever better optimized launcher trajectories has been driving the development of optimal control algorithms for decades. Therefore it is only consequent to take a closer look at the benefits of post-optimality analysis concerning launch vehicles. Interest is put on optimality assessment and the provision of sensitivity data for planning and evaluation of derived missions and launcher concepts.

The reference launcher vehicle to be used is the Ariane 5 rocket, currently Europe's prime satellite transporter. Since its first successful flight in 1997 [1] it has undergone several modifications and enhancements. The Ariane 5 *Versatile*, in particular, is the configuration of choice for this study. It consists of the main engine H 173, two boosters P 241 and a new Aestus engine in the upper stage, which is restartable.

More details on model and missions are given in sec. 6.1. Then, an optimal direct launch into GTO is analyzed in sec. 6.2 followed by a dual payload mission in sec. 6.3, when a first payload is delivered into a LEO and a second payload into GTO.

6.1 Description of Model and Mission

Other than in the case of the HOPPER re-entry there is a mass-flow due to propellant consumption. It is considered in the modeling of the dynamics. Besides the mass, the model states represent 3-DOF dynamics.
The optimizable controls are defined as aerodynamic angles, being the angle of attack α and the sideslip-angle β.
The employed gravity model includes harmonics up to J_2.



1. Cryogenic main stage, Solid boosters
2. EPS L 10 upper stage
3. EPS L 9 upper stage
4. ESC-A upper stage

Fig. 6.0.1: Ariane 5 version overview, ©Arianespace

The shape of the ascent trajectory is mainly defined by the stage engines and their corresponding tanks and structural masses. The H 173 engine with a burntime of 650 seconds propels the lower stage. For the first 130 seconds of the mission it is supported by two P 241 boosters.
The upper stage L 10 with a new Aestus engine can contain up to 10,000 kg of propellant with a maximum burntime of 1100 seconds. Further data on propulsion and components can be found in tab. 6.1.1.

Particularly mass-flow and I_{SP} of the boosters and, hence, their thrust output, is not linear or even constant, but follows a certain profile. Therefore the value of 274 seconds for the I_{SP} only represents an average vacuum value. In the optimization a profile is considered which is an adaptation of the P 230 profile presented in [2].

Starting from the Kourou launch pad the rocket does a vertical take-off. An explicit pitch over maneuver is not required, since γ_0 establishes a well defined flight-path angle right from the beginning. A gravity turn follows, where α is fixed

H 173	Structural mass [kg]	12,700
	Propellant mass [kg]	173,300
	I_{sp} [s]	434.0
P 241	Structural mass [kg]	38,200
	Propellant mass [kg]	248,130
	I_{sp} [s], (reference)	274.0
L 10	Structural mass [kg]	2,750
	Propellant mass [kg]	10,000
	I_{sp} [s]	324.0

Tab. 6.1.1: Component specifications according to [71]

to zero till the boosters reach shut-off and are jettisoned. From then on, the controls are freely optimizable.
In order to compile an acceptable initial guess to start the optimization a set of control laws has been used. Further reference on the background and the exact equations are available in [3].

Two path constraints are imposed. The one related to the maximum tolerable dynamic pressure is especially important during the early ascent, while the heatflux limit rather becomes critical in the later part of the trajectory, depending on the mission outline (see fig. 6.1.4). They are set to $\dot{Q} \leq 60\ \frac{kW}{m^2}$ and $q_{dyn} \leq 40\ kPa$.
Table 6.1.2 exemplifies the phase structure for an ascent mission to a GTO. The boundary and path constraints along with the rocket components assigned to the various phases, are given therein.

Overall, two different scenarios are investigated. The first scenario is a direct launch into GTO with permanent thrust. The intention is to deliver a maximum payload into a 200km x 35,800km orbit with 7° inclination.
The L 10 upper stage has been designed for dual payload missions. Hence, in a second scenario it is taken full advantage of its restart capabilities. A first payload of 3,500 kg is delivered into a 500km x 550km near circular orbit. It follows a coast arc which is beneficial for this kind of dual payload mission, as has been demonstrated in [32]. Then, the engine is restarted in order to bring the second payload into a 200km x 35,800km orbit with 7° inclination. Once again the objective is to maximize the mass of the payload to be deployed into GTO.

	Time [sec]	Active Propulsion	Path constraints	Boundary constraints
1	0-18	$2 \cdot$ P 241 $1 \cdot$ H 173	$\chi = 90°$	$h_0 = 5\ m$ $\lambda_0 = -52.7686°$ $\delta_0 = 5.2434°$ $v_0 = 5\ \frac{m}{s}$ $\gamma_0 = 89.6°$ $\chi_0 = 90°$
2	18-t_{pf}	$2 \cdot$ P 241 $1 \cdot$ H 173	$\dot{Q} \leq 60\ kW/m^2$ $q \leq 40\ kPa$ $\alpha = 0°$	
3	t_{pf}-t_{hf}	$1 \cdot$ H 173	$\dot{Q} \leq 60\ kW/m^2$	
4	t_{hf}-1561	$1 \cdot$ L 10	$\dot{Q} \leq 60\ kW/m^2$	$h_{peri} = 200\ km$ $h_{apo} = 35,800\ km$ $i = 7.0°$

Tab. 6.1.2: Phase structure of Ariane 5 launch optimization to GTO

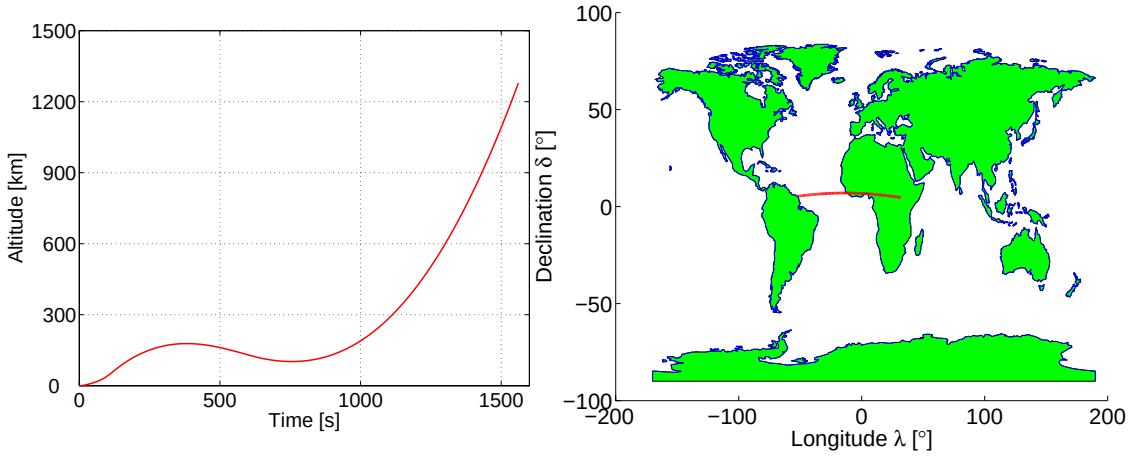


Fig. 6.1.2: Groundtrack and altitude profile of optimal GTO transfer

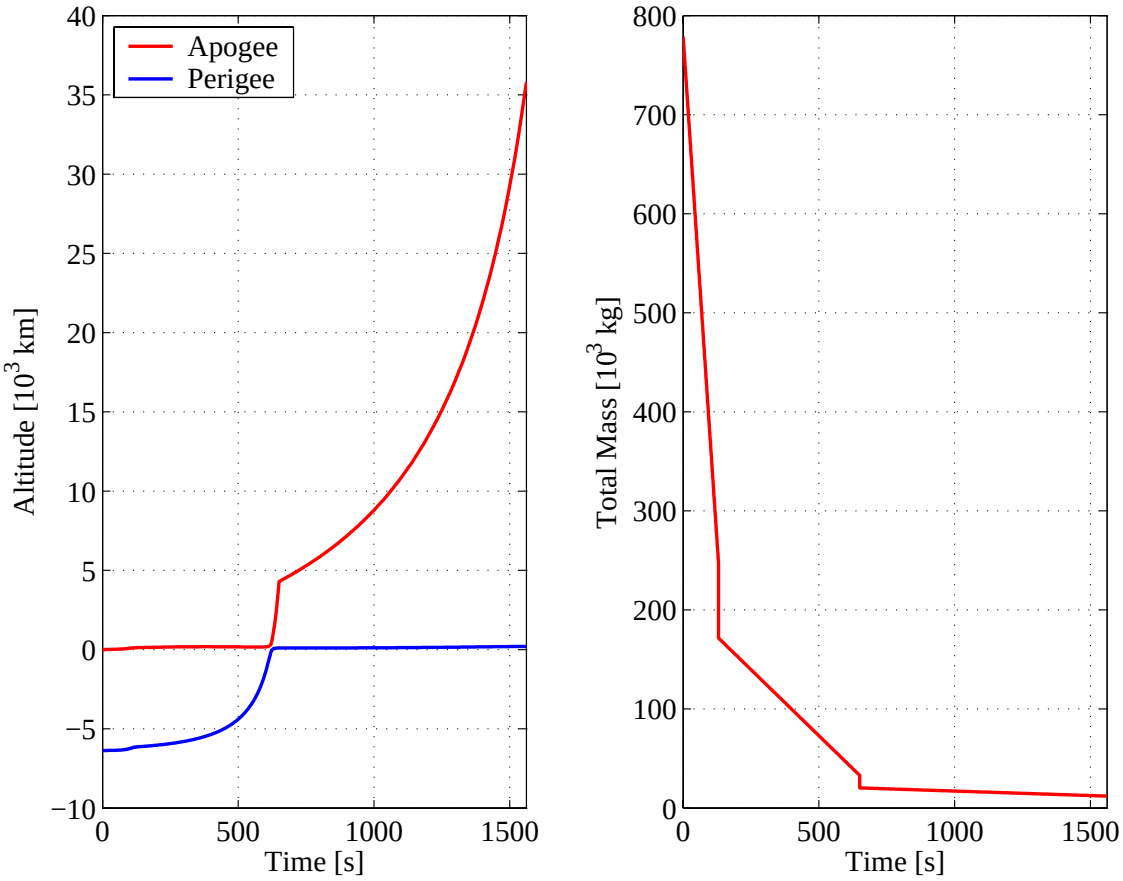


Fig. 6.1.3: Direct GTO transfer: Perigee, apogee and total mass

6.2 Post-Optimal GTO-Launch Analysis

The ground track and altitude profile of the optimal direct GTO launch are shown in fig. 6.1.2. After leaving the earth atmosphere, the launcher sags with a decreasing flight-path angle in order to reach orbital speed, also raising the perigee altitude (see fig. 6.1.3). At shut-down of the Aestus engine the upper stage reaches the final orbit with a mass of 12,038 kg of which 2,750 kg are structural and 9,288 kg represent the maximized payload.

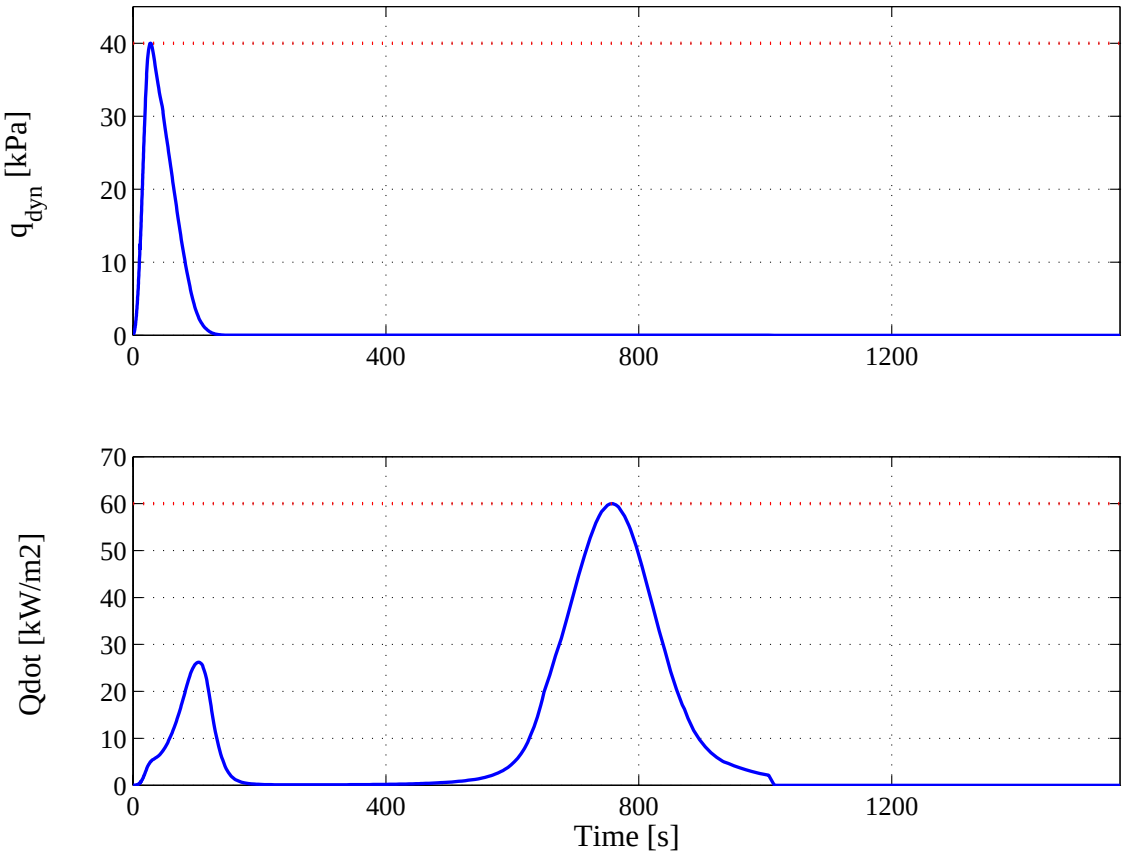


Fig. 6.1.4: Direct GTO transfer: Heatflux and dynamic pressure

6.2.1 Sensitivity with Respect to Component Properties

During the optimization the characteristics of the launcher stages and the mission parameters (see tab. 6.3.3) are fixed to their nominal value and therefore not optimizable. Since Ariane 5 *Versatile* has a configuration which has been customized for payload transport into GTO, it should be expected that the found design constitutes optimal staging configuration. Evaluation of the sensitivities of the optimal cost with respect to fuel masses of the three propellant components partly confirms the assumption. Both a lower and a higher propellant mass ϵ_{P173} (subscript P stands for propellant) of the main engine do not increase the payload, but instead results in a significant and strongly nonlinear decrease of the optimal payload, according to post-optimality analysis (see fig. 6.2.5).

The assumption of optimal staging design is not entirely correct for the booster dimensions. According to sensitivity analysis, an 8 % higher propellant mass per

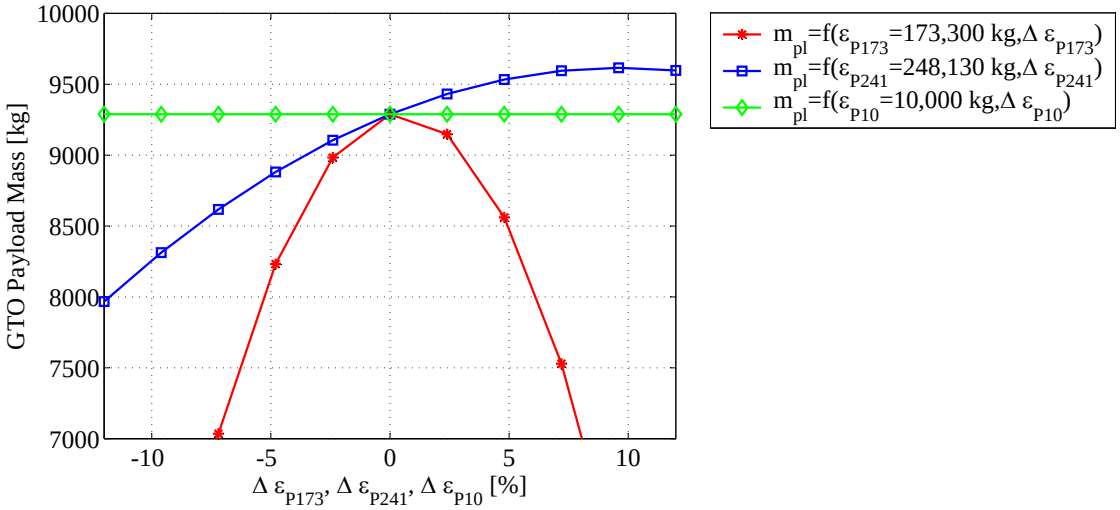


Fig. 6.2.5: Direct GTO transfer: Propellant sensitivity

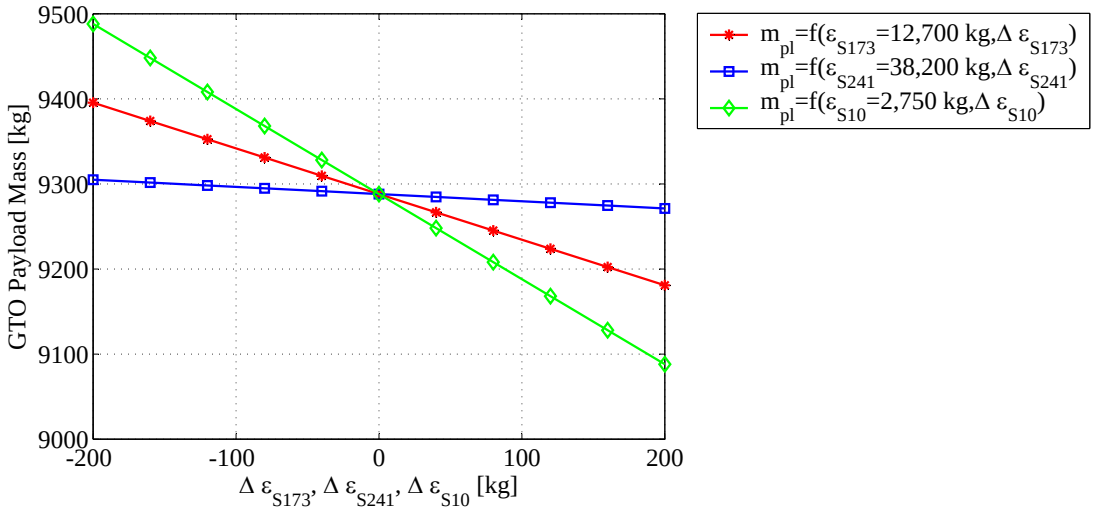


Fig. 6.2.6: Direct GTO transfer: Structural mass sensitivity

booster should raise the payload by roughly 300 kg. This statement, however, is afflicted with major uncertainties. Thus, it can not be expected that the utilized profile of the specific impulse is precise. Further, the model does not accommodate for design correlations. In real life a higher tank volume for an already optimized booster would most likely require a larger cross section with negative effects upon the aerodynamic drag.
Post-optimality analysis displays also another reason why the statement about better effectiveness is conditional. Because the design location at $\Delta\epsilon_{P241} = 8\%$ is well outside the confidence radius.

When examining how sensitive the payload is to variations in the upper stage propellant mass, it turns out ϵ_{P10} does not have any impact at all. This is surprising at first glance, because one would expect that changing the fuel mass should at least have some sort of effect on the maximum payload.
The way the optimal control problem is modeled, it is not required that the maximum allowed propellant mass of the L 10 stage is entirely used. And, in fact, revision of the trajectory data shows that the tank is only filled to 83 %. We will get back to this issue later on when discussing the imposed mission constraints. In any way, it can be noted that an upper stage filling ratio of one is not optimal for the aspired mission.

The findings of PSA give profound quantitative insight into the payload gains/losses related to the structural masses of the three key launcher components. Logically, a higher structural mass ϵ_{S10} of L 10 maps one-to-one to losses in the payload, as can be seen in fig. 6.2.6 and tab. 6.3.3. Per additional kilogram structure of H 173 the payload is reduced by -537 grams; for the boosters it is -84 grams.

The effects of the ISP on the optimum cost function value can be ranked as follows. The highest sensitivity of $86.2\frac{kg}{s}$ is computed for the main engine ISP ϵ_{I173}, while it is $62.9\frac{kg}{s}$ for the reference booster ISP ϵ_{I241} and $22.3\frac{kg}{s}$ for the Aestus ϵ_{I10} (see fig. 6.2.7).

6.2.2 Effects of Path Constraint Variations

The graphs in fig. 6.2.8 show second order analysis results for percentage variations of the heatflux limit, the dynamic pressure limit and the reference area F, which is required for the computation of the aerodynamic forces.
We learn that variations in ϵ_{hfl} can have a positive but very limited impact. It turns out, for an allowed $\dot{Q} \leq 80\frac{kW}{m^2}$, being a third larger than the nominal value,

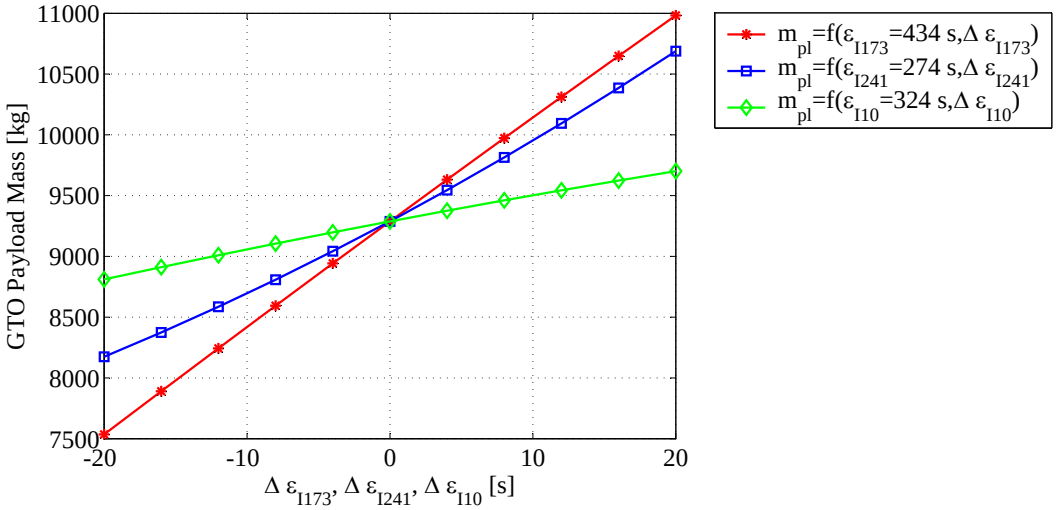


Fig. 6.2.7: Direct GTO transfer: ISP sensitivity

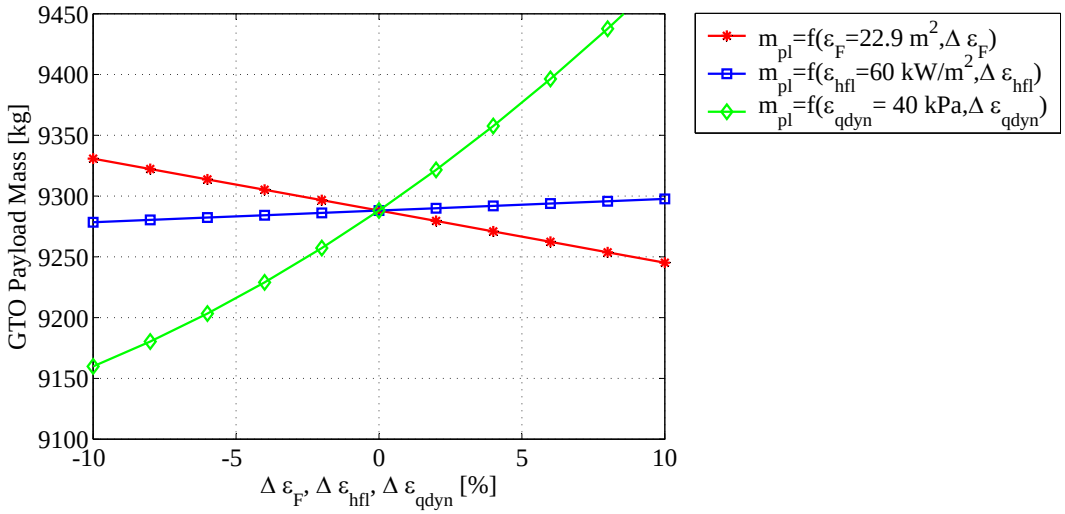


Fig. 6.2.8: Direct GTO transfer: Path constraint sensitivity

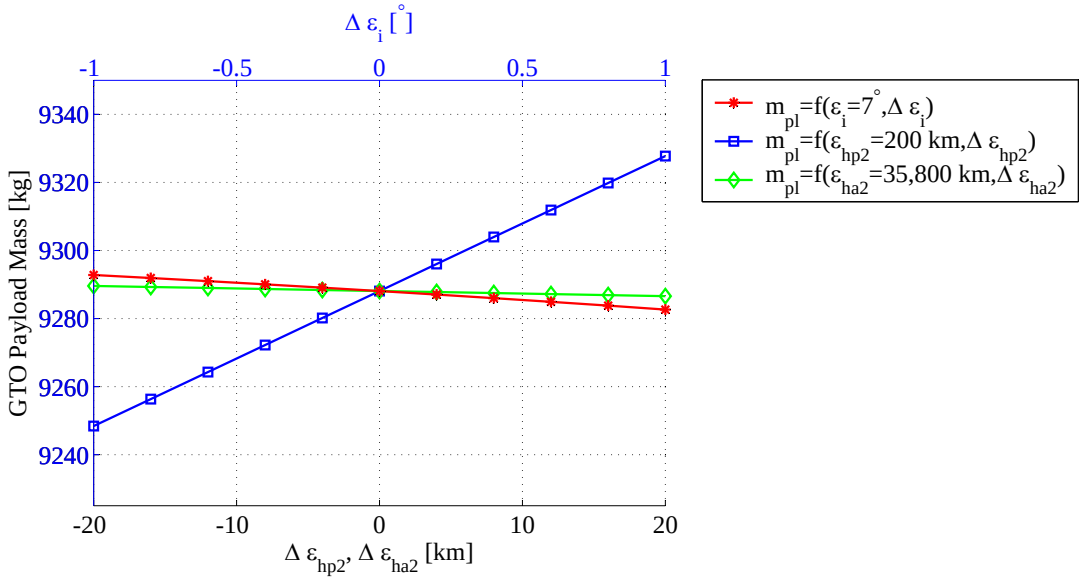


Fig. 6.2.9: Direct GTO transfer: Target orbit sensitivity

the PSA predicts a gain in payload of only 32 kg while staying within the confidence radius. Implicitly, it also means that the heatflux constraint will remain active even for much higher limits.
It is different for the dynamic pressure. The cost function at the Karush-Kuhn-Tucker point exhibits a sensitivity of 40.2 $\frac{kg}{kPa}$. The confidence radius with an order of 10^{-4} is very small, suggesting that an extrapolation of the costs will not be accurate, because of an erroneous active set of constraints. Since the parameters, defining the radius is related to α in phase 1, we do not bother for the moment and instead take a closer look at the dynamic pressure path constraint. Extrapolation of the Lagrange multipliers delivers a confidence radius of $(\Delta\epsilon_{qdyn})_c = 0.28$ predicting that the dynamic pressure will detach from its allowed limit if the limit is raised to a value of 40.28 kPa.
In order to validate the capabilities of the post-optimality algorithm the mission is reoptmized without restrictions for the dynamic pressure. The maximum observed is $[q_{dyn}]_{max} = 40.22\ kPa$. POA has indeed produced quite accurate information about the deactivation of the path constraint.

The third graph in fig. 6.2.8 depicts how the cost function value is affected by a variation of the reference area F. This is of particular interest in the context of design optimization [54]. We have seen earlier that additional propellant in the boosters could increase payload. A larger reference area in turn reduces payload.

Neglecting structural implications cost-benefit assessment looks as follows: a five percent increase in the volume ($[UL]^3$) booster propellant superposed with the plane ($[UL]^2$)enlargement of F results in

$$\begin{aligned} m_{pl}^{p} &= m_{pl}^{*} + \left[\frac{dm_{pl}}{d\epsilon_F} \quad \frac{dm_{pl}}{d\epsilon_{P241}}\right] \cdot \xi + \frac{1}{2}\xi^T \cdot \begin{bmatrix} \frac{d^2 m_{pl}}{d\epsilon_F^2} & 0 \\ 0 & \frac{d^2 m_{pl}}{d\epsilon_{P241}^2} \end{bmatrix} \cdot \xi \qquad (6.2.1) \\ &= 9701.8\ kg \end{aligned}$$

with

$$\xi = \begin{bmatrix} \Delta\epsilon_F \\ \Delta\epsilon_{P241} \end{bmatrix} = \begin{bmatrix} \epsilon_F \cdot (1.05^{2/3} - 1) \\ \epsilon_{P241} \cdot (1.05 - 1) \end{bmatrix}. \qquad (6.2.2)$$

In words, sensitivity analysis identifies potential for design improvements assuming the simplified modeling of launcher and mission.

6.2.3 Sensitivity with Respect to the Deployment Orbit

The answer is still pending to the question, why the optimal solution does not take full advantage of the allowed upper stage propellant mass. Therefore we turn to inclination, apogee altitude and perigee altitude of the final orbit. These are stated as equality constraints of the optimal control problem. Post-optimality analysis is used to determine what effect the three orbit elements have on the optimum payload.

As can be seen in fig. 6.2.9, the payload does increase when the perigee altitude is raised. A $\Delta\epsilon_{hp2} = 10\ km$ results in a gain of $\Delta m_{pl} = 19.8\ kg$. Since GTOs are only used as transfer orbits to a geostationary orbit, the consequence can be drawn to raise the transfer orbit perigee, providing the satellite with a higher orbital energy while at the same time enabling higher payload. It is paid for by consumption of additional propellant in the upper stage. From the sensitivity data related to the final mission time it can be extrapolated that the maximum allowed burntime will be reached for a perigee altitude of 380 km.
Since this is outside the confidence radius the finding has to be handled with caution. Comparison with reoptimized data shows that the burntime maximum is actually reached for $\epsilon_{hp2} = 342\ km$.
A short note on the optimality. As reoptimization also shows, the true maximum payload is reached for $\epsilon_{hp2} = 386\ km$.

The payload mass is very insensitive with respect to the apogee altitude and also with respect to final inclination. This facilitates the utilization of super geostationary transfer orbits following the concept of bielliptic transfers [14].

As a last remark on the overall analysis of the direct GTO transfer it is worth noting that most sensitivities are primarily linear. Hence, for studies of the design sensitivity and perturbed design performance in the close vicinity of the KKT point, it appears sufficient to limit the efforts to first order analysis.

6.3 Dual-Payload Performance Sensitivity

In a second ascent scenario the Ariane 5 *Versatile* is sent to deliver two payloads. The first payload with a mass of $3,500\ kg$ is deployed into a $500\ km\ x\ 550\ km$ orbit with $7°$ inclination. Following a coast arc the upper stage engine is restarted and transports the second payload into GTO. Its mass is the objective to be maximized.

The new mission outline changes the character of the launcher trajectory in various ways. First of all it becomes much longer. With a coast arc duration of 1137 seconds and maximum burntime of the L 10, the mission is extended to 2886 seconds, or about 48 minutes.

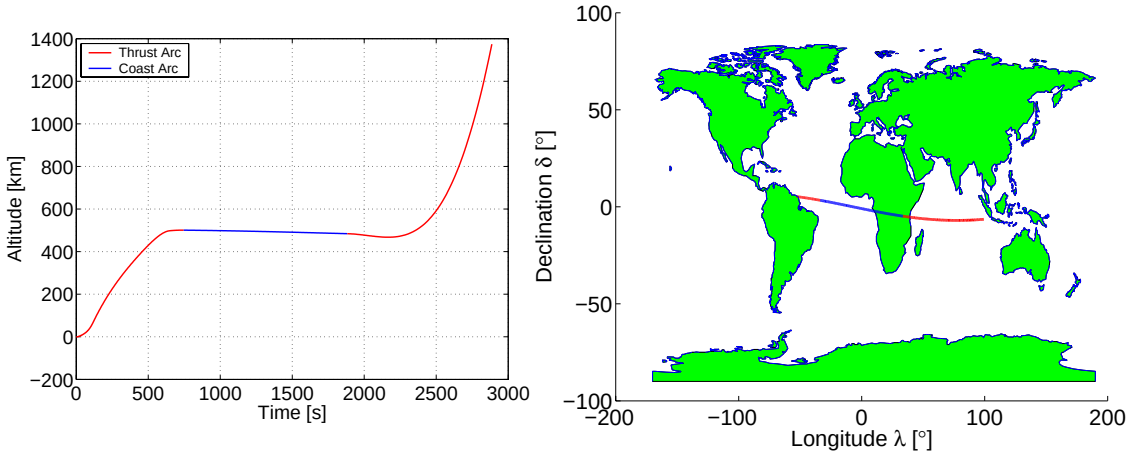


Fig. 6.3.10: Groundtrack and altitude profile of optimal dual payload transfer

The first stage ascends to higher altitude than in the direct GTO case. It is an effect of the $500\ km$ perigee constraint. The following first burn of the upper stage is short. During the coast the vehicle sags slightly (see fig. 6.3.10). The reason is to

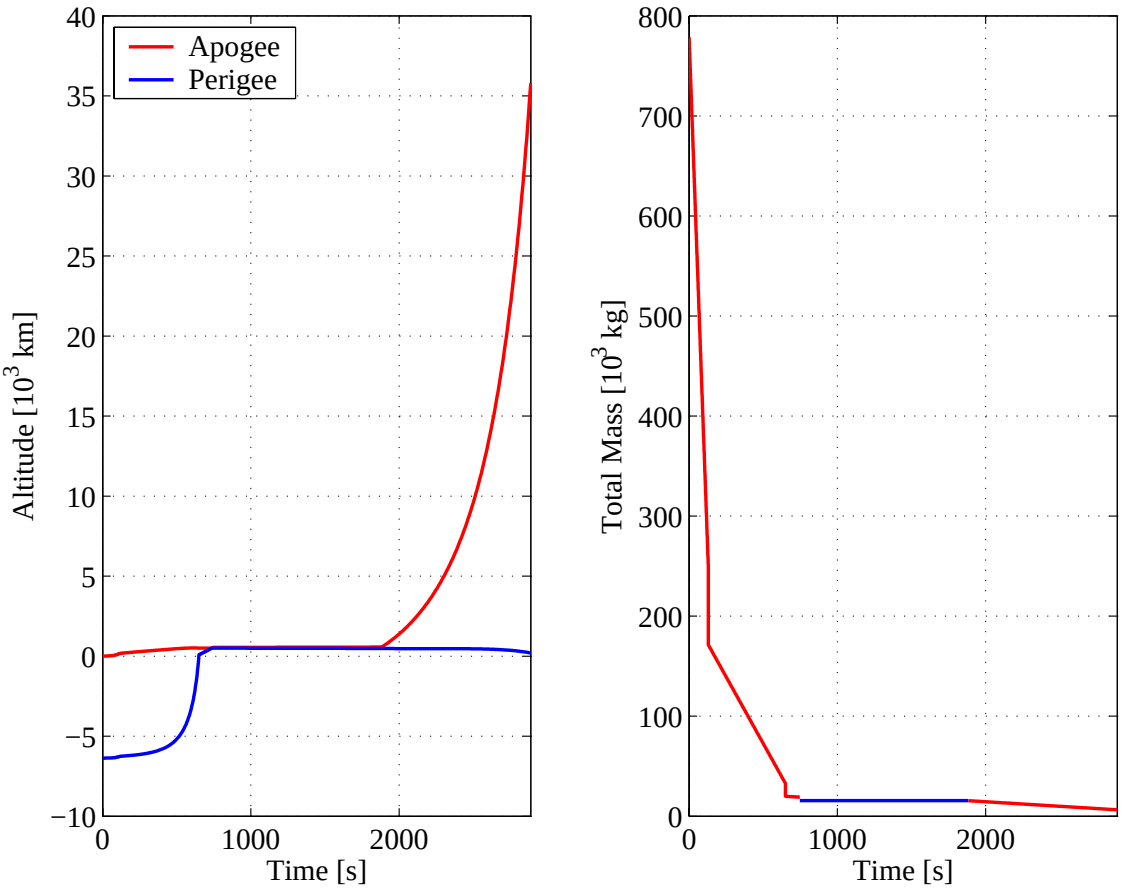


Fig. 6.3.11: Dual payload transfer: Perigee, apogee and total mass

be found in earth oblateness effects.
The final orbit is reached with an optimized payload mass of $3,695.5\ kg$ (see fig. 6.3.11).

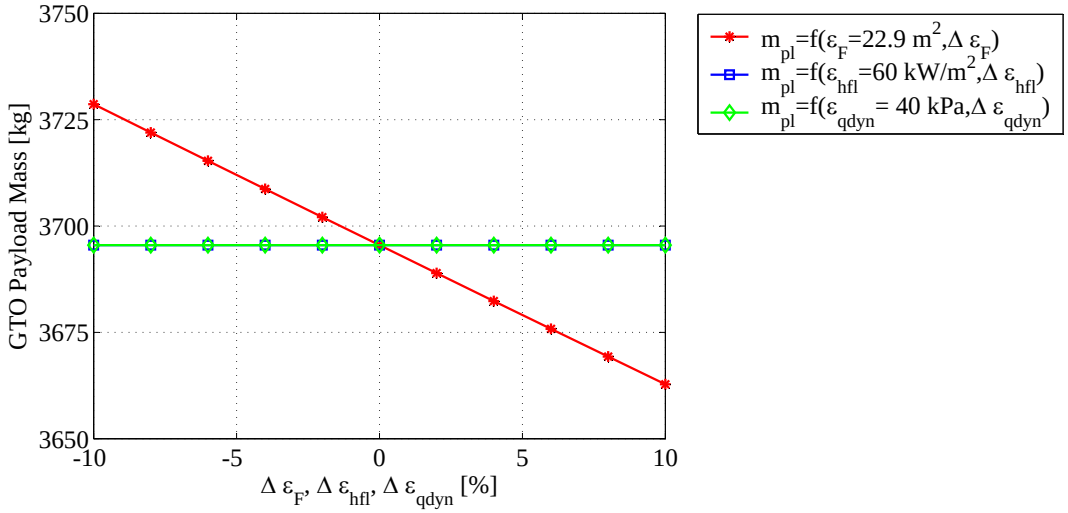


Fig. 6.3.12: Dual payload transfer: Path constraint sensitivity

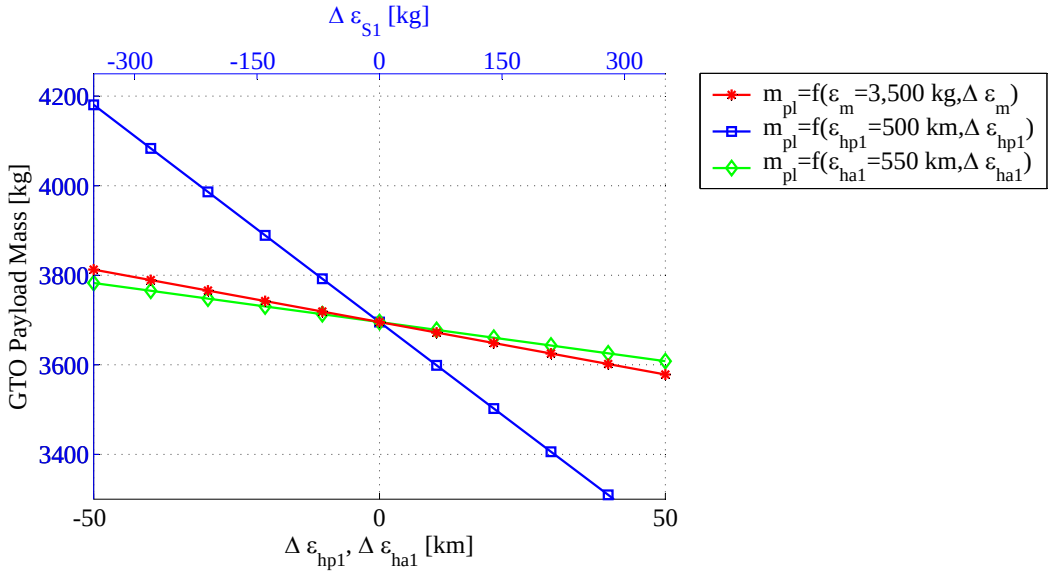


Fig. 6.3.13: Dual payload transfer: Low earth orbit sensitivity

6.3.1 Sensitivity with Respect to the Component Properties

A quick look at the path constraint sensitivity (see fig. 6.3.12) reveals that the new trajectory with the steeper altitude profile is not affected by variations of ϵ_{hfl} or ϵ_{qdyn}. Because both constraints are at no time active.

In order to allow a rapid assessment of the deviations between the sensitivities of the direct GTO transfer and of the dual payload transfer, the first order sensitivity coefficients are listed in tab. 6.3.3. For ϵ_F it shows that the effect on the payload shrinks from $-18.69\ \frac{kg}{m^2}$ to $-14.38\ \frac{kg}{m^2}$ as a consequence of the shorter time spent in the atmosphere. Besides, the effect of ϵ_F is only secondary.

The same shrinking can be observed for the ISP related sensitivity coefficients. The new trajectory is much more robust. So, for instance, a loss of one second in the ISP of the boosters is expected to result in a loss of no more than 52 kg of payload to GTO.

Other than in the first scenario, this time the optimizer fully exploits the fuel allowance of the upper stage. This is shown by the non-zero number for $\frac{dm_{pl}}{d\epsilon_{P10}} = 0.134\ \frac{kg}{kg}$. The respective plot for propellant sensitivities can be found in app. D.0.2.

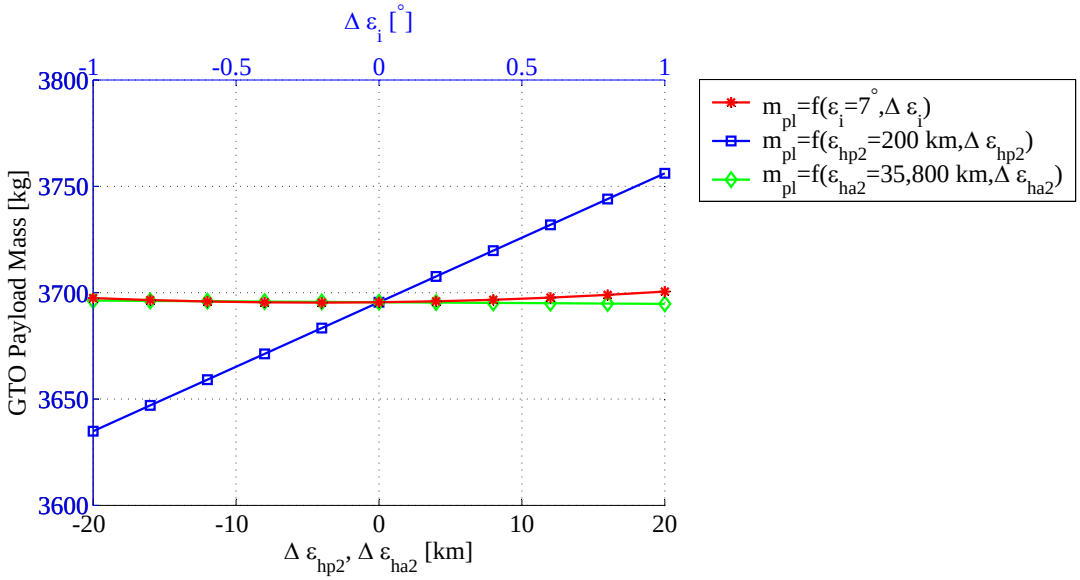


Fig. 6.3.14: Dual payload transfer: Target orbit sensitivity

Name	Sensitivity Parameter			Value	Sensitivities $\frac{dm_{pl}}{d\epsilon}$ in kg per unit	
	Description	Component	Unit		Direct GTO	Dual payload
ϵ_{S173}	Structural mass	H 173	[kg]	12,700	-0.537	-0.316
ϵ_{S241}		P 241		38,200	-0.0844	-0.068
ϵ_{S10}		L 10		2,7500	-1.000	-1.000
ϵ_{P173}	Propellant mass	H 173	[kg]	173,300	0.0198	-3.025e-4
ϵ_{P241}		P 241		248,130	0.0274	0.0223
ϵ_{P10}		L 10		10,000	0.000	0.134
ϵ_{I173}	ISP	H 173	[s]	434.0	86.19	51.86
ϵ_{I241}		P 241		274.0	62.86	59.13
ϵ_{I10}		L 10		324.0	22.27	21.15
ϵ_F	Reference area	Constr.	$[m^2]$	22.9	-18.69	-14.38
ϵ_{hfl}	Heatflux limit	Constr.	$[\frac{kW}{m^2}]$	60.0	1.600	0.00
ϵ_{qdyn}	Dynamic pressure limit	Constr.	[kPa]	40.0	40.21	0.00
ϵ_m	Total mass	Payload 1	[kg]	3,500	0.000	-0.335
ϵ_{hp1}	Perigee altitude	Payload 1	[km]	500.0	0.000	-9.672
ϵ_{ha1}	Apogee altitude	Payload 1	[km]	550.0	0.000	-1.7466
ϵ_i	Inclination	Payload 2	[°]	7.0	-5.082	1.516
ϵ_{hp2}	Perigee altitude	Payload 2	[km]	200.0	1.984	3.032
ϵ_{ha2}	Apogee altitude	Payload 2	[km]	35,800.0	-0.075	-0.0389

Tab. 6.3.3: Description of sensitivity parameters and first order sensitivities for direct GTO and for dual payload transfer; values are given for payload mass m_{pl} in kg

6.3.2 Sensitivity with Respect to the Deployment Orbits

A series of mission constraints define the two deployment orbits. In fig. 6.3.13 second order sensitivity information is given for ϵ_m, ϵ_{ha1} and ϵ_{hp1}. The graphs show that it is indeed costly to deploy payload into the predefined LEO. The ratio of the LEO versus the GTO payload is

$$\frac{dm_{pl}}{d\epsilon_m} = -0.335 \tag{6.3.3}$$

In words, for every additional kilogram of LEO-bound payload the GTO payload is reduced by 0.335 kg. The analysis also shows a confidence radius of $(\Delta\epsilon_m)_c = 934\ kg$. Hence, the prediction is assumed to be reliable for GTO payload losses of up to 313 kg.
Both enforced apogee and enforced perigee altitude further lower the optimal objective value. The perigee altitude ϵ_{hp1} in particular demonstrates a major impact on the GTO payload, reducing it by 9.67 kg for every additional kilometer. Lowering the perigee to 300 km is predicted to produce a gain in payload of 1934 kg.
Even though the variation is very large, post-optimality analysis approves the result to be trustworthy. Comparison with the true optimum indicates a mismatch of 22.5 %, which, under the given circumstances can be taken as qualitatively good.

The sensitivity of the cost function with respect to GTO orbit constraints exhibits the same tendencies like in the direct GTO transfer, as can be seen in fig. 6.3.14. Variations in the perigee altitude ϵ_{hp2} appear to have more critical consequences now. The sensitivity coefficient is more than 50 % larger. But even with a value of $\frac{dm_{pl}}{d\epsilon_{hp2}} = 3.032$ it still has only a third of the effect that a variation in the perigee altitude of the LEO orbit has.

After review, it can be stated that the cost gradient information given for the various perturbation parameters is qualitatively and quantitatively reliable within the computed confidence radii.

The provided sensitivities are of good use to verify optimality of the costs with respect to certain design parameters, like the main stage propellant mass. They can also serve to identify potential for performance enhancement and may suggest changes in the rocket design or mission outline.
So far, this study has solely concentrated on the optimal cost function value. Its analysis is certainly of major interest. But there are also other aspects of the optimal trajectory that can be investigated through post-optimality analysis. For instance, the relevance of the coast-arc.

Phase	t_f^* [s]	$\frac{dt_f}{d\epsilon_{hp1}}$ $[\frac{s}{km}]$	$\frac{dt_f}{d\epsilon_{ha1}}$ $[\frac{s}{km}]$	$\frac{dt_f}{d\epsilon_{hp2}}$ $[\frac{s}{km}]$	$\frac{dt_f}{d\epsilon_{ha2}}$ $[\frac{s}{km}]$
1: Lift-off	21.1	-1.7e-4	-4.6e-5	4.4e-5	-2.5e-7
2: With Boosters	130.0	0.0	0.0	0.0	0.0
3: Main engine	650.0	0.0	0.0	0.0	0.0
4: First burn L 10	746.1	7.2e-3	2.6e-3	-3.2e-3	-6.7e-5
5: Coast-arc	1882.7	7.4e-1	4.2e-2	7.3e-2	-9.7e-4
6: Second burn L 10	2886.6	7.4e-1	4.0e-2	7.6e-2	-9.0e-4

Tab. 6.3.4: Dual payload transfer: Sensitivity coefficients for phase end times

6.3.3 Coast-Arc Sensitivity Matters

The optimal coast-arc is 1137 seconds long. It has been introduced in order to further increase payload. The questions arise, to what extend the coast-arc is beneficial and how its shape changes when design parameters variate.

The sensitivities of the phase times with respect to the orbit parameters are listed in tab. 6.3.4. According to these numbers, the length of the coast-arc

$$d_{ca} \;=\; t_{f5} + \frac{dt_{f5}}{d\epsilon} \cdot \Delta\epsilon - t_{f4} - \frac{dt_{f4}}{d\epsilon} \cdot \Delta\epsilon \tag{6.3.4}$$

does not change considerably even for significant variations in the perigee or apogee constraints and it is almost exclusively dependent on the final time parameter. Further, raising the LEO apogee by 10 km would shorten the coast-arc by about 0.4 seconds. This value is very small and motivates a closer examination of the coast-arc final time.
A sensitivity analysis is conducted for perturbations in t_{f5}. The computed sensitivity is

$$\frac{dm_{pl}}{dt_{f5}} \;=\; 0.0078\ \frac{kg}{s}. \tag{6.3.5}$$

Compared to other factors, the impact of the coast-arc duration turns out to be practically negligible. This is advantageous for predicting optimal payload. But it also means that the shape of the coast-arc is very insensitive. Hence, the prediction of the trajectory shape might suffer. Figures 6.3.15 and 6.3.16 exemplify this dilemma for a variation of 10 % in ϵ_{ha1}. The predicted trajectory is feasible and complies with the imposed orbital constraints. And the error in the first order cost

function approximation $m_{pl}^{p1}(\Delta\epsilon_{ha1} = 55\ km) = 3599.4\ kg$ is only 19.3 % compared to the absolute variation. The shape of the trajectory, however, looks much different. This duality is related to the poorly predicted coast-arc duration with its low sensitivity.

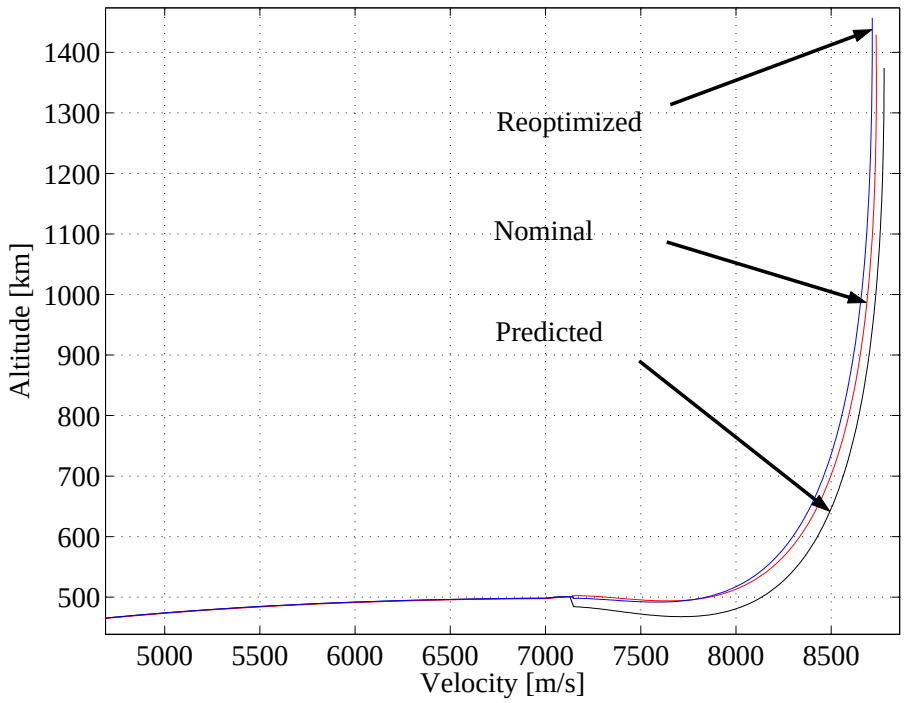


Fig. 6.3.15: Dual payload transfer: State prediction

The shown illustration is only one example. But it depicts a general matter. Analysis results are best when sensitivities are reasonable. A linearization of the optimum payload sensitivity about the broader vicinity around the KKT point produces valuable results. But the practitioner should once more bear in mind that the search direction of the post-optimality algorithm is in general not identical with the heading towards the truly optimal solution of the perturbed optimal control problem.

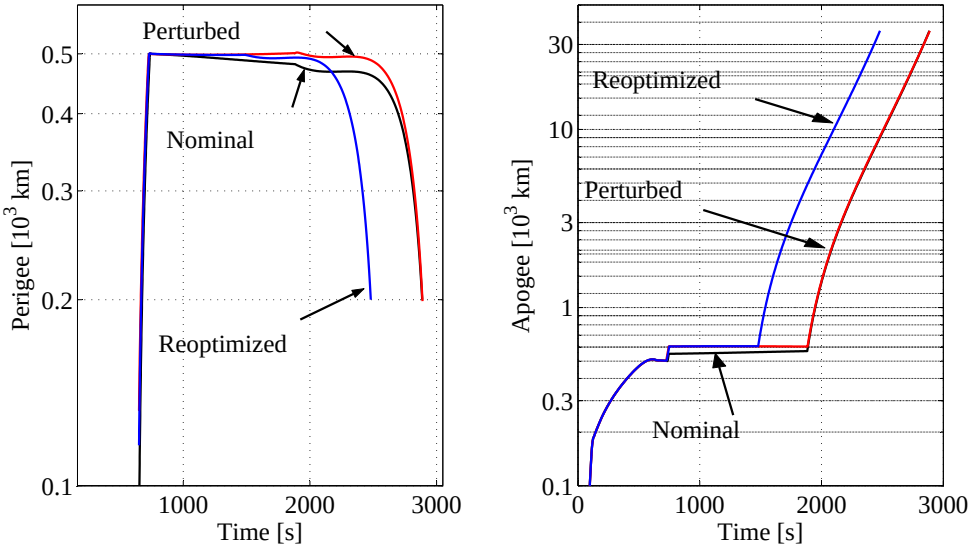


Fig. 6.3.16: Dual payload transfer: Comparison of predicted and perturbed trajectory

Chapter 7

Conclusions

In the framework of this thesis a post-processing algorithm for the analysis of optimal solutions has been developed and implemented. The work has been motivated by the ambition to overall enhance optimization procedures in practice. Because the research invested in the development of ever better optimization algorithms, has not made it any easier for practitioners to comprehend the informative content of an obtained optimal solution.

This thesis focuses on post-processing solutions of optimal control problems, since these constitute a class of optimization problems, which is of particular interest in aeronautics and astronautics. Nonetheless, as it turned out, the achievements are generic and can also be used for other classes of problems and fields of application.

The taken approach is to format the solution of the optimal control problem in a fashion that makes it easily and efficiently accessible to sensitivity analysis. This is accomplished by direct transcription of the OCP into an NLP program. Several methods are available to perform this task. The selection is adapted from the optimizer CAMTOS and makes different collocation schemes as well as multiple shooting schemes available. This measure is taken to make not only the solution itself, but also the solution structure available for analysis.

The core activities of the post-optimality analysis of the transcribed solution commence with a parameterized sensitivity analysis of first order. Securing an efficient analysis on this level, the computational input is, as far as possible, limited to data that is already available from prior optimization. In case of incompleteness of the active set information, the developed algorithm automatically completes the matrix data related to the active set of constraints including the Lagrange multipliers.

The result are the sensitivities $\mathcal{S}$ of individual program parameters with respect

to pre-identified design or mission parameters ϵ and the sensitivity gradient of the cost function with respect to ϵ. The method is numerical, which means, it does not depend on the analytical accessibility of the dynamic equations. This proves to be useful in practical use. However, it also means that the output is local in so far as it gives gradients at the KKT point. These gradients are very accurate, since the sensitivity analysis is set up in a fashion so that it does not rely on the exactness of the Lagrange multipliers.
These multipliers are needed for second order analysis, where the sensitivities of the costate variables of the KKT vector are computed, as are second order terms of the cost function gradients with respect to ϵ.
The second order analysis requires the Hessian matrix of the Lagrange function. Numerical methods have been implemented to perform its calculation.

The second order sensitivity analysis, as introduced in this thesis, represents a capacious module for the assessment of neighboring designs in the vicinity around the optimal solution of the nominal case.
Firstly, it allows an estimate of higher order effects on the cost function development for finite perturbations. Secondly, and even more critical, by extrapolating the sensitivity gradients of the Lagrange multipliers the post-processing does a single step prediction of the active set composition. This provides confidence radii for the applicability of the sensitivity gradients, when the optimal behavior of deviating designs is estimated.

The post-optimality analysis algorithm has been applied to two comprehensive problems in the field of trajectory optimization, namely a re-entry flight of the winged HOPPER vehicle and Ariane 5 single- and dual-payload launch missions.

With the HOPPER re-entry flight various purposes have been demonstrated which sensitivity analysis can be used for. The set of studied perturbation parameters contained initial states, and vehicle as well as model properties.
The sensitivity gradients help to rank the parameters according to their impact and depict the individual responsibility for the overall performance of the system. This has been shown in particular for uncertainties in the trajectory's initial states.
Superposition of sensitivity gradients related to the aerodynamic coefficients gives a quantitative measure for possible controller design margins: an increase of one percent in the ratio of lift versus drag trades with a gain of about 0.9 degrees in longitude.
Another result of the sensitivity gradient mining is the potential gain of one degree in longitude, when raising the heatflux limit by 5 percent to 472.5 $\frac{kW}{m^2}$.
The results are directly derived from a single optimal solution without reoptimization.

The consequent utilization of sensitivity coefficients for prediction of derived missions or designs has been cross-checked with reoptimized solutions. Comparison shows good compliance and justifies the confidence put in POA results. This holds particularly for states of the lateral motion and to a lesser extend also for the states of the longitudinal motion. The results have been verified with different transcription methods, which underlines the generic capacity.

The quality of the control sensitivities permits trend interpretation and an assessment of perturbation induced control excitation levels. However, for the angle of attack more tangible results can not be obtained. The reason is to be found in the unstable behavior of the linearized sensitivity model, which tends to further define the unconstrained skipping maneuvers along the trajectory. The effect disqualifies the individual α-sensitivities for prediction purposes, but depicts, where the trajectory is most sensitive to variations in α.

Especially, the cost function trade-offs can quickly be computed by the post-optimality analysis algorithm and immediately show the user, how the design space is shaped in the vicinity around the optimal solution. The results can be used to identify design potential or stability margins.
The computed confidence radii give overall good indication of the operational range of the PSA and constraint/parameter bounds that are eminent.

For the post-optimality analysis of Ariane 5 optimal trajectories two different launch missions have been studied -a direct launch of payload into GTO and a dual-payload mission to LEO and onwards to GTO.
The results confirm the optimal configuration of the launcher's main stage and boosters, and give profound qualitative and quantitative insight into the payload gains/losses related to structural or propellant masses and I_{SP} of the three stages.

The perigee constraint for direct GTO insertion degrades the performance. As PSA shows, raising the perigee by 10 km increases the optimum payload by 19.8 kg. It also identifies the heatflux and the dynamic pressure constraint to have only secondary influence on the payload, even though heatflux will remain an active constraint even for much higher limits.

The same type of information is also computed for the dual-payload mission. One of the findings is that all cost sensitivities are lower than in the direct GTO case, except the sensitivity with respect to perigee. With a value of $\frac{dm_{pl}}{d\epsilon_{hp2}} = 3.032\ \frac{kg}{km}$ the perigee of GTO still has only a third of the influence of the perigee of the LEO orbit. Based on the POA data, about the same ratio is also found for payload trade-off between the two orbits. Every additional kilogram into LEO is paid with a loss of 335 grams into GTO.

After review, it can be stated that the PSA cost gradients are of good quality within the computed confidence radii and provide reliable material for design evaluation and mission planning. Most sensitivities are primarily linear. Therefore it appears sufficient to limit the analysis efforts for general purposes to first order.

From the data it can also be concluded that the coast-arc is very insensitive. With about 8 grams per additional second in duration, it has a positive, but very limited influence on the payload. The prediction of its duration is, hence, very inaccurate. This depicts a general matter. Analysis results are best when sensitivities are reasonable.

A lesson learned from both, HOPPER and Ariane 5 solution analysis is, not to confuse post-optimality analysis with an iterative optimization. The user needs to bear in mind that the sensitivity gradient produced by the POA algorithm does not necessarily match the search direction of an optimizer, but targets the computation of a feasible and perturbed near-optimum in a most efficient way.
The need for proper grid point placement has a slightly different focus in POA than it has in optimization. Even with a reduced number of grid points the results can be accurate, for as long as the grid covers the critical sectors of the optimal and the predicted solution.

Even though the benefits of post-optimality analysis have been demonstrated for two spacecraft trajectory optimization problems, the developed algorithm is generic and can be applied on a multitude of different problems.
Only by evaluation and validation of post-optimality analysis in practise can the concept prove its capabilities. Hence, its perspective is defined by its employment. This does not necessarily require more sophisticated test cases. But it calls for rapid assessment of neighboring design spaces and the improvement of problem understanding under realistic statements of work.

Further development efforts should focus on visual processing of the computed sensitivities to enhance the data absorption by the user. It will lead to more efficient interpretation of optimal solutions and will streamline the post-processing.
The information could be made available in a graphical environment or be organized in other forms that relieve the user from personal data mining.

Second order sensitivity analysis offers potential for further enhancement. The computation of the Hessian matrix is time consuming and losses in accuracy can motivate critical discussion of the results. Following the concepts in NLP optimization a more coarse approximation of the Hessian should be considered. Interpretation would then be limited to tendency prediction, but could possibly support quick assessment.

Bibliography

[1] *Air University Space Primer*, Air University, Maxwell AFB, AL, USA, 2003.

[2] *ASTOS Conventional Launcher Application Manual*, Tech. Report ASTOS CLA 5.0, Institute of Flight Mechanics and Control, University of Stuttgart, 2004.

[3] *ASTOS Model Library Reference Manual*, Tech. Report ASTOS MLR 5.0, Institute of Flight Mechanics and Control, University of Stuttgart, 2004.

[4] *GESOP Software User Manual - The User Interface Tutorial*, Tech. Report GESOP 4.6.0, Institute of Flight Mechanics and Control, University of Stuttgart, 2004.

[5] U. M. Ascher, R. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Classics in Applied Mathematics, SIAM, 13 (1995).

[6] T. Beltracchi, *Optimization Software Modifiactions to Study Launch Vehicle Sizing/Trajectory Design Problems*, in 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA–94–4403, Panama City, Florida, USA, Sept. 1994.

[7] T. J. Beltracchi and H. N. Nguyen, *Experience with Post Otimality Parameter Sensitivity Analysis in FONSIZE*, in 4th AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, A Collection of Technical Papers - Part I, Cleveland, USA, Sept. 1992, pp. 496–506.

[8] J. T. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming, Advances in Design and Control*, Socienty for Industrial and Applied Mathematics, 2001.

[9] J. T. Betts and S. L. Campbell, *Discertize Then Optimize*, Tech. Report M&CT–TECH–03–001, Phantom Works, Mathematics and Computing Technology, Seattle, WA, USA, 2003.

[10] J. T. Betts and S. O. Erb, *Optimal Low Thrust Trajectories to the Moon*, SIAM Journal on Applied Dynamical Systems, 2 (2003), pp. 144–170.

[11] J. T. Betts and W. P. Huffman, *Estimating Adjoint Variables from a Direct Transcription Solution*, Tech. Report M&CT–TECH–04–001, Phantom Works, Mathematics and Computing Technology, Seattle, WA, USA, 2004.

[12] B. F. Blackwell and K. J. Dowding, *Sensitivity and Uncertainty Analysis for Thermal Problems*, in 4th International Conference on Inverse Problems in Engineering, Rio de Janeiro, Brazil, 2002.

[13] D. Bose, M. Wright, and T. Gökçen, *Uncertainty and Sensitivity Analysis of Thermochemical Modeling for Titan Atmospheric Entry*, in 37th AIAA Thermophysics Conference, AIAA–2004–2455, Portland, USA, 2004.

[14] C. D. Brown, *Spacecraft Mission Design - Second Edition*, AIAA Education Series, American Institute of Aeronautics and Astronautics, Inc., 1633 Broadway, New York, NY 10019, 1998.

[15] A. E. Bryson Jr. and Y.-C. Ho, *Applied Optimal Control*, Hemisphere Publishing Corporation, New York, USA, 1975.

[16] W. Buhl, K. Ebert, and H. Herbst, *Optimal Ascent Trajectories for Advanced Launch Vehicles*, in AIAA 4th International Aerospace Plane Conference, AIAA 1992–5008, Orlando, FL, USA, 1992.

[17] C. Büskens and H. Maurer, *Sensitivity Analysis and Real-Time Optimization of Parametric Nonlinear Programming Problems*, in Grötschel M. et al (Eds.) Online Optimization of Large Scale Systems, 2001, pp. 3–16.

[18] J.-B. Caillau and J. Noailles, *Sensitivity Analysis for Time Optimal Orbit Transfer*, tech. report, ENSEEIHT–IRIT, UMR CNRS 5505, 2001.

[19] A. R. Curtis, M. J. D. Powell, and J. K. Reid, *On the Estimation of Sparse Jacobian Matrices*, Journal of the Institute of Mathematics and Applications, 13 (1974), pp. 117–120.

[20] B. de Bruin, *Lap-Time Optimization*, master's thesis, Institut für Flugmechanik und Flugregelung, Universität Stuttgart, Germany, 2003.

[21] F. Deneu, A. Roenneke, and J. Spies, *Ascent and Entry Trajectory Design for the Hopper Vehicle*, in 54th Internatinoal Astronautical Congress of the International Astronautical Federation, IAC–03–V.4.04, Bremen, Germany, Sept. 2003.

[22] M. M. Diehl, *Real-Time Optimization for Large Scale Nonlinear Processes*, PhD thesis, Ruprecht–Karls Universität, Heidelberg, Germany, 2001.

[23] S. O. Erb, *Trajectory of a Moon Transfer from GTO employing Solar Electric Propulsion*, master's thesis, Institut für Flugmechanik und Flugregelung, Universität Stuttgart, Germany, 2002.

[24] S. O. Erb and S. Weikert, *ATPE - Optimization of Pre-Aerocapture Trajectories*, tech. report, TTI GmbH, ASTOS Solutions, Stuttgart, Germany, 2004.

[25] S. O. Erb and K. H. Well, *ASTRA–übertragung der Methode der Dynamischen Inversion auf die Lageregelung des HOPPER*, Technical Report IFR TR 03–002, Institut für Flugmechanik und Flugregelung, Universität Stuttgart, Stuttgart, Germany, Dec. 2003.

[26] M. C. Ferris, J. Lim, and D. M. Shepard, *An Optimization Approach for Radiosurgery Treatment Planning*, SIAM Journal on Optimization, 13 (2003), pp. 921—937.

[27] S. Ferson, J. Hajagos, and W. Tucker, *Probability Bounds Analysis is a Global Sensitivity Analysis*, in 4th International Conference on Sensitivity Analysis of Model Output, SAMO04–68, Santa Fe, USA, Mar. 2004.

[28] A. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, 1983.

[29] R. Fletcher and S. Leyffer, *User Manual for filterSQP*, Technical Manual, University of Dundee, U.K., 1998.

[30] C. A. Floudas, *Nonlinear and Mixed-Integer Optimization*, Oxford University Press, Inc., New York, 1995.

[31] P. F. Gath, *Design and Development of a Hybrid Trajectory Optimization Software*, PhD thesis, University of Stuttgart, Germany, 2002.

[32] P. F. Gath and A. J. Calise, *Optimization of Launch Vehicle Ascent Trajectories with Path Constraints and Coast Arcs*, Journal of Guidance, Control and Dynamics, 24 (2001), pp. 296–304.

[33] P. F. Gath and K. H. Well, *HISTOS Technical Report 1*, Tech. Report TR–99–002, Institute of Flight Mechanics and Control, University of Stuttgart, June 1999.

[34] ——, *HISTOS Technical Report 2 - Direct Method*, Tech. Report TR–00–002, Institute of Flight Mechanics and Control, University of Stuttgart, June 2000.

[35] ——, *Trajectory Optimization using a Combination of Direct Multiple Shooting and Collocation*, in AIAA Guidance, Navigation and Control Conference, AIAA 2001–4047, Montreal, Quebec, Canada, 2001.

[36] R. Giering and T. Kaminski, *Recipes for Adjoint Code Construction*, ACM Transactions on Mathematical Software, 24 (1998), pp. 437–474.

[37] P. Gill and W. Murray, *The Computation of Lagrance Multiplier Estimates for Constrained Minimization*, Mathematical Programming, 17 (1979), pp. 32–60.

[38] P. Gill, W. Murray, and M. A. Saunders, *User's Guide for SNOPT Version 6, A Fortran Package for Large Scale Nonlinear Programming*, Tech. Report SNOPT 6, Systems Optimization Laboratory, Stanford University, 2002.

[39] ——, *SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization*, SIAM Review, 47 (2005), pp. 99–131.

[40] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, Inc., London, 1981.

[41] M. Grässlin, *Entwurf und Analyse eines Prädiktiven Lenkkonzepts für Rückkehrmissionen Auftriebsgestützter Raumfahrzeuge*, PhD thesis, University of Stuttgart, Germany, 2004.

[42] W. Grimm and K. H. Well, *On Mission and Trajectory Planning in an Air–Combat Processor*, in AIAA Guidance, Navigation and Control Conference, AIAA 1987–2394, Monterey, CA, USA, Aug. 1987, pp. 670–680.

[43] ——, *Nichtlinear Optimierung – Skriptum*, Institut für Flugmechanik und Flugregelung, Universität Stuttgart, 2001.

[44] W. Hallman, *Sensitivity Analysis for Trajectory Optimization Problems*, in 28th AIAA Aerospace Sciences Meeting, AIAA–90–0471, Reno, Nevada, USA, Jan. 1990.

[45] ——, *Optimal Scaling Techniques for the Nonlinear Programming Problem*, in 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA–94–4417, Panama City, Florida, USA, Sept. 1994.

[46] C. R. Hargraves and S. W. Paris, *Direct Trajectory Optimization Using Nonlinear Programming and Collocation*, AIAA Journal of Guidance, Control and Dynamics, 10 (1987), pp. 338–342.

[47] M. Heinkenschloss, M. Ulbrich, and S. Ulbrich, *Superlinear and Quadratic Convergence of Affine–Scaling Interior–Point Newton Methods for Problems with Simple Bounds without Strict Complementarity Assumption*, Mathematical Programming, (1998).

[48] J. Helton, *Sampling Based Methods for Uncertainty and Sensitivity Analysis*, in 4th International Conference on Sensitivity Analysis of Model Output, SAMO04–04, Santa Fe, USA, Mar. 2004.

[49] A. L. Herman and B. A. Conway, *Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules*, AIAA Journal of Guidance, Control and Dynamics, 19 (1996), pp. 592–599.

[50] B. Iooss, N. Devictor, and F. V. Dorpe, *Response Surfaces and Sensitivity Analyses for an Environmental Model of Dose Calculations*, in 4th International Conference on Sensitivity Analysis of Model Output, SAMO04–13, Santa Fe, USA, Mar. 2004.

[51] S. S. Isukapalli, *Uncertainty Analysis of Transport-Transformation Models*, PhD thesis, 1999.

[52] C. T. Kelley and E. W. Sachs, *A Trust Region Method for Parabolic Boundary Control Problems*, SIAM Journal for Optimization, 9 (1999), pp. 1064–1081.

[53] D. Kraft, *A Software Package for Sequential Quadratic Programming*, vol. Forschungsbericht, DFVLR-FB-88-28, DFVLR, Oberpfaffenhofen, Germany, 1988.

[54] J. Lerch, *Optimal Design of a Rocket Launch Vehicle with ASTOS*, master's thesis, Institut für Flugmechanik und Flugregelung, Universität Stuttgart, Germany, 2004.

[55] J. R. Martins, J. J. Alonso, and J. J. Reuther, *Complete Configuration Aero-Structural Optimization using a Coupled Sensitivity Method*, in 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA–2002–5402, Atlanta, USA, Sept. 2002.

[56] J. R. Martins, P. Sturdza, and J. J. Alonso, *The Connection Between the Complex-Step Derivative Approximation and Algorithmic Differentiation*, in 39th AIAA Aerospace Sciences Meeting and Exhibition, AIAA–2001–0921, Reno, USA, Jan. 2001.

[57] C. Oberlin and S. J. Wright, *Active Set Identification in Nonlinear Programming*, Tech. Report 05–01, Computer Science Department, University of Wisconsin-Madison, 2005.

[58] J. R. Olds, *System Sensitivity Analysis Applied to the Conceptual Design of a Dual-Fuel Rocket SSTO*, in 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA–94–4339, Panama City Beach, USA, Sept. 1994.

[59] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Second Edition, New York, 2002.

[60] T. Raible and D. Jacob, *Sensitivity Based Optimization of Two-Stage-to-Orbit Space Planes with Lifting Body and Waverider Lower Stages*, in 12th AIAA International Space Planes and Hypersonic Systems and Technologies, AIAA–2003–6955, Norfolk, USA, Dec. 2003.

[61] F. J. Regan, *Re–Entry Vehicle Dynamics*, AIAA Education Series, American Institute of Aeronautics and Astronautics, Inc., 1633 Broadway, New York, NY 10019, 1984.

[62] S. Scarpetta, *Assessing the Role of Labour Market Policies and Institutional Settings on Unemployment: a Cross-Country Study*, OECD Economic Studies No . 26. 1996/1, 1996.

[63] U. Schöttle, *Flug- und Antriebsoptimierung luftatmender aerodynamischer Raumfahrtträger*, PhD thesis, Institut für Raumfahrtsysteme, Universität Stuttgart, Germany, 1988.

[64] J. Sobieszcanski-Sobieski, *Sensitivity Analysis and Multidisciplinary Optimization for Aircraft Design: Recent Advances and Results*, AIAA Journal on Aircraft, 27 (1990), pp. 1291–1299.

[65] J. Sobieszcanski-Sobieski, J.-F. Barthelemy, and K. M. Riley, *Sensitivity of Optimum Solutions of Problem Parameters*, AIAA Journal, 20 (1982), pp. 1291–1299.

[66] M. Stein, *Large Sample Properties of Simulations Using Latin Hypercube Sampling*, Technometrics, 29 (1987), pp. 143–151.

[67] O. Tabart, *Bahnoptimierung und Bahnregelung für den Wiedereintritt des Raumfahrzeugs HOPPER*, master's thesis, Institut für Flugmechanik und Flugregelung, Universität Stuttgart, Germany, 2004.

[68] J. Telaar, *Entwicklung eines Prädiktiven Lenkverfahrens für Wiederverwendbare Raumtransportsysteme*, PhD thesis, University of Stuttgart, Germany, 2005.

[69] A. Varma, M. Morbidelli, and H. Wu, *Parametric Sensitivity in Chemical Systems*, Cambridge University Press, Cambridge, U.K., 1999.

[70] A. Wächter, *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*, PhD thesis, Carnegie Mellon University, USA, 2002.

[71] M. Wade, *Encyclopedia Astronautica*, Tech. Report Ariane 5 - Component Specifications, www.astronautix.com, 2006.

[72] G. G. Wang, Z. Dong, and P. Aitchison, *Adaptive Response Surface Method - A Global Optimization Scheme for Approximation-based Design Problems*, Journal of Engineering Optimization, 33 (2001), pp. 707–734.

[73] S. Weikert, *Numerische Optimierung der Flugleistungen des Raumtransporters HOPPER*, master's thesis, Institut für Raumfahrtsysteme, Universität Stuttgart, Germany, 2003.

[74] A. Wiegand, P. F. Gath, K. H. Well, and J. Spies, *Ascent and Entry Trajectory Design for the Hopper Vehicle*, in 5th Internatinoal Conference on Launcher Technology, CNES, Madrid, Spain, 2003.

[75] H. Yan, F. Fahroo, and M. Ross, *Accuracy and Optimality of Direct Transcription Methods*, in AAS/AIAA Space Flight Mechanis Meeting, AAS–00–205, Clearwater, USA, 2000, pp. 1613–1630.

Appendix A

The Earth Mars Transfer Parameterization

A.1 List of Optimization Parameters

The following table lists the parameters that constitute the Earth Mars transfer optimization problem along with their optimized values:

	Parameter description	**Phase**	**Optimal value**
p_1	State r, initial value	1	1.0000
p_2	State v_r, initial value	1	0.0000
p_3	State v_{th}, initial value	1	1.0000
p_4	State Φ, initial value	1	-2.1713
p_5	Starting time t_0	1	-0.78677
p_6	Control, grid point 1	2	-0.0078065
p_7	Control, grid point 2	2	-0.0021179
p_8	Control, grid point 3	2	0.010309
p_9	Control, grid point 4	2	0.028005
p_{10}	Control, grid point 5	2	0.050469

p_{11}	State r, initial value	2	1.0318
p_{12}	State v_r, initial value	2	0.091326
p_{13}	State v_{th}, initial value	2	1.0676
p_{14}	State Φ, initial value	2	-1.0906
p_{15}	initial phase time t_1	2	60.192
p_{16}	State r, initial value	3	1.5176
p_{17}	State v_r, initial value	3	0.027834
p_{18}	State v_{th}, initial value	3	0.72585
p_{19}	State Φ, initial value	3	1.3801
p_{20}	Initial phase time t_2	3	273.34
p_{21}	Control, grid point 1	3	0.015417
p_{22}	Control, grid point 1	3	0.0090788
p_{23}	Control, grid point 1	3	0.0045383
p_{24}	Control, grid point 1	3	0.0021798
p_{25}	Control, grid point 1	3	0.0020955
p_{26}	Mission end time t_3	3	308.18

A.2 List of Optimization Constraints

The list of constraints of the Earth Mars transfer optimization problem is given below:

$$
\begin{aligned}
c_1 &= p_1 - 1.0 \\
c_2 &= p_2 - 0.0 \\
c_3 &= p_3 - 1.0 \\
c_4 &= p_4 - \Phi_e \\
c_5 &= p_{11} - \int_{t_0}^{t_1} \dot{r}\ dt \\
c_6 &= p_{12} - \int_{t_0}^{t_1} \dot{v_r}\ dt \\
c_7 &= p_{13} - \int_{t_0}^{t_1} \dot{v_{th}}\ dt \\
c_8 &= p_{14} - \int_{t_0}^{t_1} \dot{\Phi}\ dt \\
c_9 &= p_{16} - \int_{t_1}^{t_2} \dot{r}\ dt \\
c_{10} &= p_{17} - \int_{t_1}^{t_2} \dot{v_r}\ dt \\
c_{11} &= p_{18} - \int_{t_1}^{t_2} \dot{v_{th}}\ dt \\
c_{12} &= p_{19} - \int_{t_1}^{t_2} \dot{\Phi}\ dt \\
c_{13} &= \int_{t_2}^{t_3} \dot{r}\ dt - 1.523396 \\
c_{14} &= \int_{t_2}^{t_3} \dot{v_r}\ dt - 0.0 \\
c_{15} &= \int_{t_2}^{t_3} \dot{v_{th}}\ dt - (v_{th})_f \\
c_{16} &= \int_{t_2}^{t_3} \dot{\Phi}\ dt - \Phi_m
\end{aligned}
$$

Appendix B

Collocation Methods

As can be found in [8].

B.1 Trapezoidal Rule

Variables:

$$x^T = (y_1, u_1, ..., y_m, u_m) \tag{B.1.1}$$

Defects:

$$\zeta_k = y_{k+1} - y_k - \frac{h_k}{2}(f_k + f_{k+1}) \tag{B.1.2}$$

B.2 Classical Runge-Kutta Method

Variables:

$$x^T = (y_1, u_1, \bar{u}_2, ..., \bar{u}_m, y_m, u_m) \tag{B.2.3}$$

Defects:

$$\zeta_k = y_{k+1} - y_k - \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{B.2.4}$$

with

$$k_1 = h_k\ f_k, \tag{B.2.5}$$

$$k_2 = h_k\ f(y_k + \frac{1}{2}k_1, \bar{u}_{k+1}, t_k + \frac{h_k}{2}), \tag{B.2.6}$$

$$k_3 = h_k\ f(y_k + \frac{1}{2}k_2, \bar{u}_{k+1}, t_k + \frac{h_k}{2}), \tag{B.2.7}$$

$$k_4 = h_k\ f(y_k + k_3, u_{k+1}, t_{k+1}), \tag{B.2.8}$$

and

$$\bar{u} = \frac{u_{k+1} - u_k}{2} \tag{B.2.9}$$

B.3 Hermite-Simpson Method

Variables:

$$x^T = (y_1, u_1, \bar{u}_2, ..., \bar{u}_m, y_m, u_m). \tag{B.3.10}$$

Defects:

$$\zeta_k = y_{k+1} - y_k - \frac{h_k}{6}(f_k + 4\bar{f}_{k+1} + f_{k+1}), \tag{B.3.11}$$

with

$$\bar{y}_{k+1} = \frac{1}{2}(y_k + y_{k+1}) + \frac{h_k}{8}(f_k - f_{k+1}), \tag{B.3.12}$$

$$\bar{f}_{k+1} = f(\bar{y}_{k+1}, \bar{u}_{k+1}, t_k + \frac{h_k}{2}). \tag{B.3.13}$$

Appendix C

HOPPER Results

No.	Par.	ϵ_c	$\Delta\epsilon$	$\Delta\epsilon$ [%]	f^* [°]	f^{p1} [°]	f^{p2} [°]	f^{ϵ} [°]	σ [%]
	ϵ_v	-208.55[b]	-62.0	-1.25	-36.139	-34.349	-34.389	-33.874	22.7
	ϵ_v	94.203	+124.0	+2.5	-36.139	-39.717	-39.875	-39.099	-26.2
	ϵ_χ	-2.981[b]	-1.0	-1.0	-36.139	-34.861	-34.833	-34.058	37.3
	ϵ_χ	1.908	+1.80	+1.8	-36.139	-38.438	-38.350	-37.738	-38.3
	ϵ_χ	1.908	+2.50	+2.5	-36.139	-39.333	-39.161	-38.162	-49.4
	ϵ_{cd}	-0.036372	-0.025	-2.5	-36.139	-38.084	-38.091	-37.752	-21.0
	ϵ_{cd}	0.072712[b]	+0.025	+2.5	-36.139	-34.193	-34.200	-33.209	33.8
	ϵ_{cl}	-0.06293[b]	-0.025	-2.5	-36.139	-33.955	-33.914	-33.237	23.3
	ϵ_{cl}	0.031811	+0.025	+2.5	-36.139	-38.322	-38.280	-37.890	-22.3
	ϵ_δ	-0.74883	-0.24	-7.5	-36.139	-37.048	-37.039	-36.905	-17.5
	ϵ_δ	0.74161	+0.24	+7.5	-36.139	-35.229	-35.219	-34.988	20.1
	ϵ_{hfl}	58.67[b]	+22.5	+5.0	-36.139	-37.105	-37.127	-36.869	-35.3

Tab. C.0.1: Convergence radii and cost function prediction for design parameter perturbations

	$(\epsilon)_{c-}$	Cause for $(\epsilon)_{c-}$		$(\epsilon)_{c+}$	Cause for $(\epsilon)_{c+}$	
ϵ_r $[km]$	-27.157	p: 48,	α in phase 2	7.635	p: 48,	α in phase 2
ϵ_δ $[^\circ]$	-0755	p: 141,	γ in phase 3	0.742	c: 40,	heatflux limit
ϵ_v $[\frac{m}{s}]$	-147.47	c: 40,	heatflux limit	94.2	p: 141,	γ in phase 3
ϵ_χ $[^\circ]$	-1.96	c: 40,	heatflux limit	1.91	p: 141,	γ in phase 3
ϵ_{cd} [ ]	-0.036	p: 141,	γ in phase 3	0.044	c: 40,	heatflux limit
ϵ_{cl} [ ]	-0.00195	c_p: 306,	α in phase 5	0.032	p: 141,	γ in phase 3
ϵ_{J_2} [ ]	-0.020	p: 141,	γ in phase 3	0.00028	c_p: 306,	α in phase 5
ϵ_m $[kg]$	-27.0	c_p: 318,	α in phase 5	5827	p: 48,	α in phase 2
ϵ_{hfl} $[\frac{kW}{m^2}]$	-22.3	p: 48,	α in phase 2	57.5	c: 43,	heatflux limit

p: New parameter bound constraint

c: Deactivation of inequality constraint

c_p: Deactivation of bound constraint

Tab. C.0.2: Confidence radii and triggering incidents

Appendix D

Ariane 5 - Dual-Payload Launch

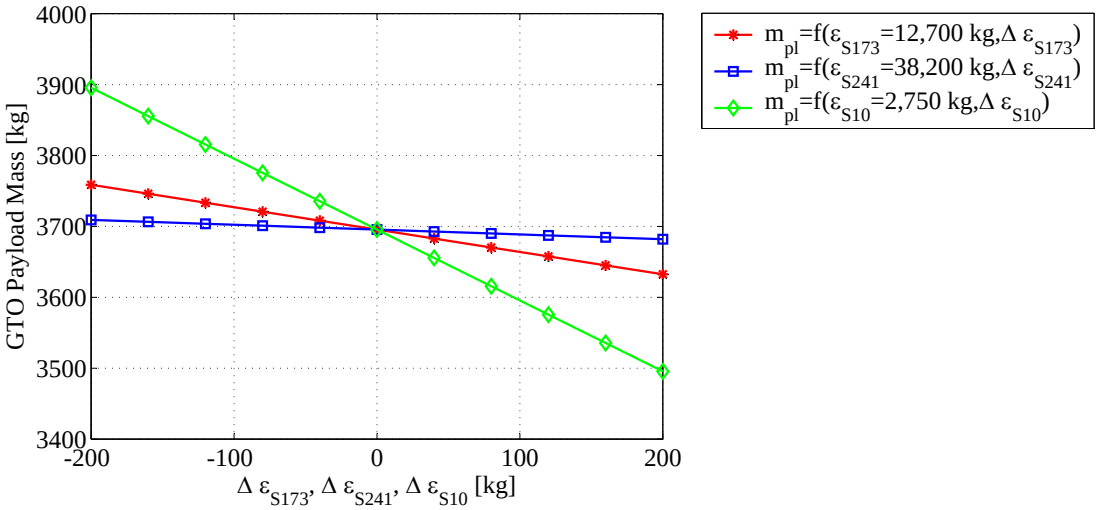


Fig. D.0.1: Dual payload transfer: Structural mass sensitivity

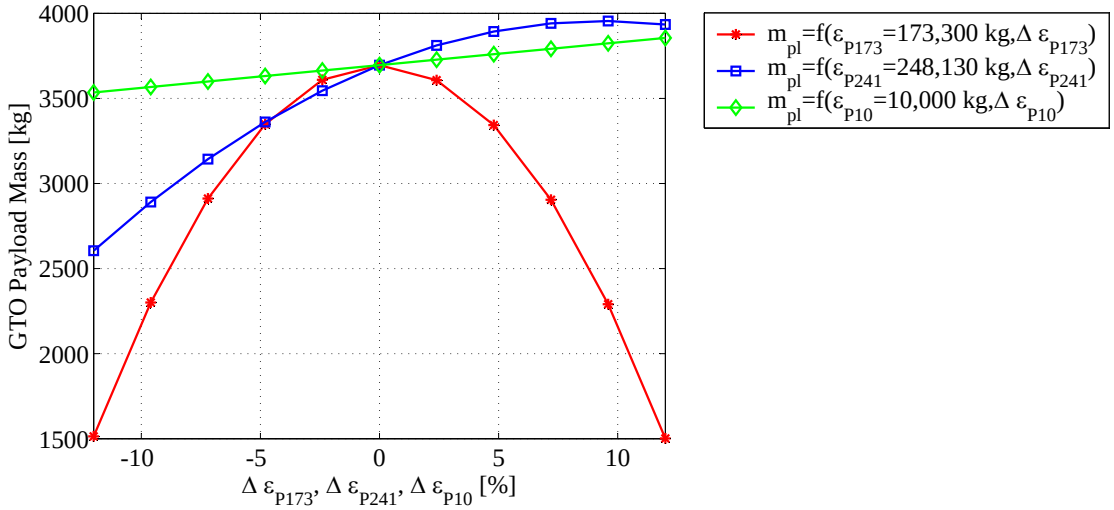


Fig. D.0.2: Dual payload transfer: Propellant sensitivity

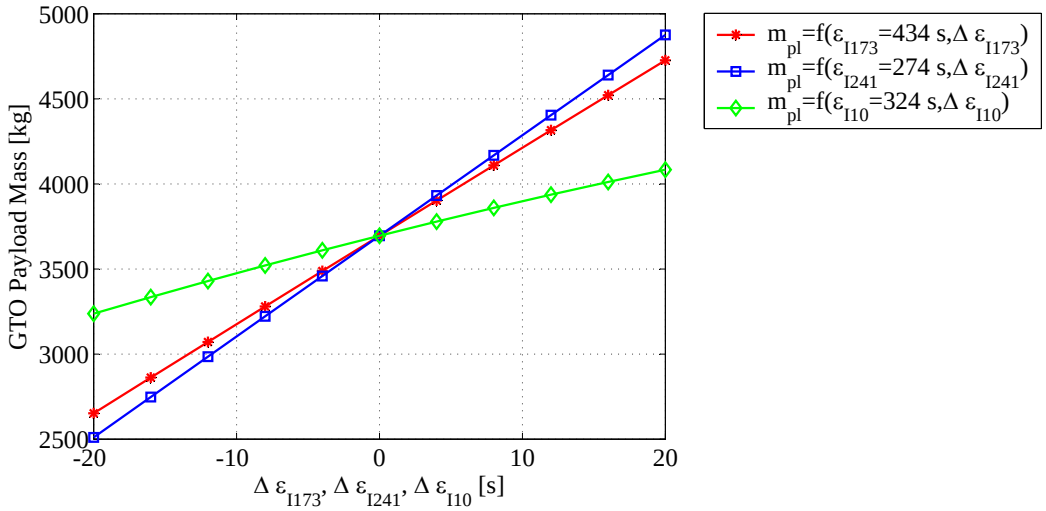


Fig. D.0.3: Dual payload transfer: ISP sensitivity

Curriculum Vitae

Name:	Sven O. Erb
Date of Birth:	19 March 1976
Place of Birth:	Waldshut-Tiengen, Germany
Nationality:	European Union, German
1986 - 1995:	Klettgau Gymnasium Tiengen, Germany
07/1995 - 04/1996:	Basic military service; German-French bi-national brigade.
09/1996 - 05/2002:	Studies in aerospace engineering at the University of Stuttgart, Germany; emphasis on Flightmechanics/Control and Aerodynamics; final degree: Dipl.Ing. Luft- und Raumfahrttechnik.
03/1997 - 12/1997:	Internship at EGL Company, Laufenburg, Switzerland; migration of the documentation system.
11/1998 - 04/1999:	Internship at Raytheon Aircraft, Wichita, USA; sales, market research and customer support.
04/2001 - 11/2001:	Studienarbeit at the Institute of Aerodynamics and Gasdynamics, University of Stuttgart, Germany. "Vergleichende Strömungs- und Schallpegeluntersuchungen an Eckenlavaldüsen unterschiedlicher Kontur in einer Modellmesskammer 1:10 des Stosswindkanals".
11/2001 - 05/2002:	Diplomarbeit at The Boeing Company, Seattle, USA. "Trajectory Optimization of a Moon Transfer from GTO Employing Solar Electric Propulsion".
07/2002 - 02/2006:	Research assistant at the Institute of Flight Mechanics and Control (IFR), University of Stuttgart, Germany. Trajectory optimization, controller development and sensitivity analysis.
09/2002 - 12/2005:	Technical consultant to TTI GmbH-Astos Solutions, Stuttgart, Germany. Trajectory optimization, mission analysis and customer relations.
Since 03/2006:	System engineer, spacecraft technology at Kayser-Threde GmbH, Munich, Germany.