



# **Einsatz moderner Software-Techniken bei der Multidisziplinären Optimierung wiederverwendbarer Raumtransporter**

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

vorgelegt von

**Osman Kalden**

aus Istanbul

Hauptberichter: Prof. Dr. rer. nat. Ernst Messerschmid

Mitberichter: Apl. Prof. Dr.-Ing. habil. Ingolf Grieger

Tag der mündlichen Prüfung: 19.09.2006

Institut für Raumfahrtssysteme  
Universität Stuttgart 2006



# Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Raumfahrtssysteme der Universität Stuttgart.

Ich bedanke mich ganz herzlich bei Herrn Prof. Dr. rer. nat. Ernst W. Messerschmid für die energische, motivierende Unterstützung dieser Arbeit und die Übernahme des Hauptberichts. Ferner danke ich Herrn Prof. Dr.-Ing. habil. Ingolf Grieger sehr für fachliche Anregungen und die Übernahme des Nebenberichts.

Ganz besonders bedanke ich mich bei Herrn Dr.-Ing. Ulrich M. Schöttle für die fachlichen Diskussionen, Anregungen und seine kontinuierliche, freundliche Unterstützung. Von ihm habe ich für meinen weiteren Berufs- und Lebensweg viel gelernt.

Auch meinen Kolleginnen und Kollegen sowie studentischen Mitarbeitern am IRS möchte ich meinen Dank für die schöne Arbeitsatmosphäre und Ihre direkte oder indirekte Unterstützung meiner Arbeit aussprechen.

Auf meinem langjährigen akademischen Weg hat mich meine Frau Aysegül liebevoll begleitet und unterstützt. Dabei musste sie oft auf meine geistige Anwesenheit verzichten und viel Arbeit auf sich nehmen. Dafür danke ich ihr vom tiefsten Herzen.

Auch meinem Vater schulde ich großen Dank für seine Unterstützung und dafür, dass er mir meine Ausbildung ermöglicht hat, die ich mir gewünscht hatte.

# Kurzfassung

In der vorliegenden Arbeit wird das multidisziplinäre Analysewerkzeug RTSOPT für die simultane Flug- und Systemoptimierung wiederverwendbarer Trägersystemkonzepte sowohl verfahrenstechnisch als auch softwaretechnisch grundlegend überarbeitet, aktualisiert und verbessert. Dabei werden das Grundkonzept der Vorgehensweise, die Modellannahmen und die Systemdynamik der Flugsimulation beibehalten. Die erreichte Flexibilität und die Leistungssteigerung werden durch typische Anwendungen des Programmpakets demonstriert. Gegenstand dieser Systemanalysen ist der derzeit aktuelle, teilweise wiederverwendbare Raumtransporter Hopper, ein Leitkonzept des nationalen Technologieprogramms ASTRA.

Die verfahrenstechnische Verbesserung wird durch die Einbindung des zur Zeit leistungsfähigsten Optimierungslösers Sparse Optimal Control Software (SOCS) in RTSOPT erreicht. Durch grundlegende Modifikationen des Programmcodes und die angepasste Problemformulierung wird die Leistungsfähigkeit des Lösers voll ausgeschöpft, was sich in einer signifikanten Beschleunigung der Konvergenz im Vergleich zu früheren Versionen widerspiegelt. Darüber hinaus wird auch die Vorbereitungszeit für eine Optimierungsberechnung drastisch verkürzt. Dies wird dadurch erreicht, dass die Startlösung für das Optimierungsproblem, die bisher durch zahlreiche Simulationsläufe und Sensitivitätsstudien erstellt werden musste, in der aktuellen Version durch Verwendung von SOCS auf der Grundlage der linearen Interpolation der geschätzten Randwerte generiert wird.

Das Modul für die Fahrzeug-Massenanalyse wird ebenfalls aktualisiert. Dazu werden zahlreiche Daten aktueller Trägersubsysteme und die Ergebnisse aus der umfangreichen europäischen Studie FESTIP in die Massen-Datenbasis übernommen.

Die softwaretechnische Verbesserung wird durch den neuen Programmsystementwurf nach Prinzipien des Software-Engineering realisiert. Das neue Konzept berücksichtigt die Rahmenbedingungen des Einsatzumfelds und moderne verfügbare Software-Technologien. Die Hauptqualitätsmerkmale unter diesen Gesichtspunkten sind die Plattformunabhängigkeit, das objektorientierte Entwurf mit verbesserter Erweiterbarkeit und Wartbarkeit, sowie die Flexibilität des Werkzeugs und die fahrzeug- und missionsspezifische Programm-Modifikationen minimiert.

Die Vielseitigkeit des Programmpakets wird zuletzt durch neue Virtuelle-Realität-Visualisierungsmodule für zwei- bzw. dreidimensionale Diagramme und räumliche interaktive Animationen erweitert.

# Summary

This thesis presents a multidisciplinary analysis tool RTSOPT for the simultaneous trajectory and system optimization of reusable launch vehicles. This software package has been developed at the Space Systems Institute, University of Stuttgart over 20 years. In this work the tool has been revised fundamentally, updated and improved both from the algorithmic and software point of view while maintaining the principal methodology, model assumptions and the system dynamics for the flight simulation. The flexibility and performance increase achieved is demonstrated by using this software package to perform system analyses of the partly-reusable launch vehicle Hopper – the central concept of the national technology program ASTRA. Hopper consists of a reusable, unmanned, and winged main stage and an expendable upper stage. The main stage is powered by three cryogenic Vulcain 3 engines – a future version of the Ariane 5 main engine. The upper stage is powered by a cryogenic Vinci engine.

The algorithmic improvement is realized by integration of the presently most advanced optimization algorithm Sparse Optimal Control Software (SOCS) in RTSOPT. By performing fundamental modifications to the program code and the chosen method of problem formulation and implementation, the solver's performance capability was utilized to a maximum degree resulting in a significant convergence speed-up compared to previous RTSOPT-versions. Furthermore the preparation time for an optimization computation has been reduced drastically. This is due to the fact that in the past an initial solution for the optimization problem had to be produced by carrying out numerous simulations and sensitivity studies, whereas with the new version, SOCS is used to generate the initial solution from linear functions interpolating the boundary values estimated by the user. With this solver also a much weaker dependence of the convergence time has been achieved, making it possible to solve typical multidisciplinary mission and system analysis problems with a higher time resolution and with much more optimization parameters without significant convergence rate penalties. This higher performance capacity of RTSOPT can be used in future to challenge more complicated multidisciplinary optimization problems. For example Finite Element Methods for improved vehicle mass analysis and Computation Fluid Dynamics for the aerodynamics analysis can be incorporated in the numerical optimization.

The mass analysis module was also updated. For this purpose numerous data from current launch vehicle subsystems and the results of the detailed European study FESTIP have been included in the mass database. In the application several launch vehicle mass analyses have been carried out to assess the effects of the latest available data. With the updated launch vehicle mass model it was possible to review earlier mass estimations and to derive conclusions concerning the technological developments forecast in the past and the technological state today.

The software improvements are achieved by the new program system design based on software engineering principles. The new concept incorporates the application environment and the available modern software technologies. The main quality parameters in this respect are the platform-independency, the object-oriented design with improved extensibility and the flexibility of the tool which minimizes the need for vehicle- and mission-specific program modifications.

Finally, the versatility of the software package is extended by new Virtual Reality visualization modules for two- and three-dimensional diagrams and three-dimensional interactive animations. For this purpose the Virtual Reality Modeling Language, which is an ISO standard has been used.

# Inhaltsverzeichnis

<b>Symbolverzeichnis</b> .....	<b>9</b>
<b>1 Einleitung</b> .....	<b>12</b>
1.1 Der kommerzielle Raumtransportmarkt.....	12
1.2 Die Ausgangssituation – das Programmsystem RTSOPT .....	16
1.3 Motivation, Zielsetzung und Vorgehensweise.....	19
<b>2 Softwaretechnische Überlegungen</b> .....	<b>22</b>
2.1 Die Programmiersprache.....	23
2.2 Die Visualisierungsmethode .....	31
2.3 Die Benutzerschnittstelle .....	36
2.4 Das Gesamtkonzept.....	42
<b>3 Das Programmpaket RTSOPT</b> .....	<b>44</b>
3.1 Die RTSOPT-Methodik .....	44
3.2 Die Rechenmodelle .....	46
<b>4 Das Optimierungsverfahren SOCS</b> .....	<b>55</b>
4.1 Das Optimierungsproblem .....	55
4.2 Grundlagen der Optimalen Steuerung.....	56
4.3 Einbindung des Optimierungsprogramms SOCS in RTSOPT .....	65
<b>5 Visualisierung</b> .....	<b>68</b>
5.1 Zwei- und Dreidimensionale Plots.....	68
5.2 Formmodelle von Raumfahrzeugen.....	69
5.3 Flugvisualisierung .....	71
<b>6 Anwendungsergebnisse</b> .....	<b>76</b>
6.1 Hopper – Das Fahrzeug und die Transportmission.....	76
6.2 Die Rechenmodelle .....	79
6.3 Aktualisierung der Datenbasis und Anwendungsvergleich der Massenanalyse.....	80
6.4 Leistungsvergleich von Optimierungsverfahren.....	84
6.5 Optimierung der Aufstiegsbahn .....	86
6.6 Optimierung des Flügellastfaktors .....	93
6.7 Optimierung der Gesamtmission Hopper .....	97
6.7.1 Formulierung des Optimierungsproblems.....	98
6.7.2 Analytische Berechnung der Nutzlastmasse .....	101
6.7.3 Exemplarischer Flugverlauf .....	103
6.7.4 Numerische Aspekte.....	107
<b>7 Zusammenfassung und Ausblick</b> .....	<b>110</b>



<b>Literaturverzeichnis .....</b>	<b>114</b>
<b>Anhang A: Marktanalyse.....</b>	<b>121</b>
A.1 Der kommerzielle Raumtransportmarkt.....	121
A.2 Aktuelle Trägerraketen .....	124
A.3 Entwicklungstrends der Satellitenmassen.....	125
A.4 Wiederverwendbare Trägersysteme.....	126
<b>Anhang B: Visualisierung – Anwendungsbeispiele .....</b>	<b>132</b>

# Symbolverzeichnis

## Lateinische Formelzeichen<sup>1</sup>

$A$	Auftriebskraft	(N)
$b$	Sollwertvektor	(-)
$B$	B-Spline-Basisfunktion	(-)
$c$	Beschränkung	(-)
$c_A$	Auftriebsbeiwert	(-)
$F$	Kraft	(N)
$g$	Erdbeschleunigung / Pfadbeschränkung	(m/s <sup>2</sup> ) / (-)
$G$	Jakobi-Matrix	(-)
$h$	Höhe über dem Meeresspiegel / Integrationsschrittweite	(m) / (-)
$I_{sp}$	Spezifischer Impuls	(s)
$J$	Zielfunktion	(-)
$m$	Masse	(kg)
$Ma$	Flugmachzahl	(-)
$n$	Lastvielfachenfaktor	(-)
$p$	Optimierungsparameter	(-)
$\dot{Q}$	Spezifischer Wärmestrom	(W/m <sup>2</sup> )
$r$	Entfernung zum Erdmittelpunkt	(m)
$R$	Radius	(m)
$S$	Schubkraft	(N)
$t$	Flugzeit	(s)
$u$	Steuerfunktion	(-)
$v$	Fluggeschwindigkeit	(m/s)
$W$	Widerstandskraft	(N)
$x$	Variable des Optimierungsproblems	(-)
$y$	Flugzustand	(-)

## Griechische Formelzeichen

$\alpha$	Anstellwinkel	(°)
$\chi$	Flugbahnazimut	(°)

<sup>1</sup> In Gleichungen wird für Vektor- und Matrixsymbole Fettschrift verwendet.

## Symbolverzeichnis

---

$\delta$	Geografische Breite	(°)
$\varepsilon$	Schubwinkel relativ zur Fahrzeuglängsachse	(°)
$\varphi$	Randbedingung	(-)
$\gamma$	Bahnneigungswinkel	(°)
$\lambda$	Geografische Länge	(°)
$\mu_a$	Flugwindhängewinkel	(°)
$\eta$	Drosselfaktor	(-)
$\rho$	Dichte der Atmosphäre	(kg/m <sup>3</sup> )
$\omega_E$	Winkelgeschwindigkeit der Erde	(rad/s)

## Indizes

$I, F$	Anfangs- / Endwert (Initial / Final)
$l, u$	Unterer / oberer Grenzwert (lower/upper)
$L$	Luftatmendes Triebwerk
$R$	Raketentriebwerk
$Ref$	Referenzwert
$SP$	Scheitelpunkt
$k$	Flugphasennummer

## Abkürzungen

ASTRA	Ausgewählte Systeme und Technologien für zukünftige RTS-Anwendungen
FESTIP	Future European Space Transportation Investigations Program
FLPP	Future Launcher Preparatory Programme
FMHR	Freie Molekulare Aufheizrate (Free Molecular Heating Rate)
GESOP	Graphical Environment for Simulation and Optimization
GTO	Geostationäre Transferbahn
GUI	Grafische Benutzeroberfläche (Graphical User Interface)
HTML	Hypertext Markup Language
HTOL	Horizontal startendes und landendes System (Horizontal Take-Off and Landing)
IRDT	Inflatable Re-Entry and Descent Technology
LEO	Low Earth Orbit
NGLT	New Generation Launch Technology
NLP	Nichtlineare Programmierung

NLPQL	Nichtlinearer Sequenziell-Quadratischer Optimierungsalgorithmus
PHIGS	Programmer's Hierarchical Interactive Graphics System
RLV	Wiederverwendbares Trägersystem (Reusable Launch Vehicle)
RTSOPT	Programmsystem für Raumtransportsystem-Optimierung
SLI	Space Launch Initiative
SOCS	Sparse Optimal Control Software
SSTO	Einstufiges System (Single Stage To Orbit)
TAEM	Endanflugphase (Terminal Area Energy Management)
TETRA	Technologies for Future Space Transportation Systems
TSSO	Zweistufig, Suborbital (Two Stage Suborbital)
TSTO	Zweistufiges system (Two Stage To Orbit)
VR	Virtuelle Realität
VRML	Virtual Reality Modeling Language (Virtuelle-Realität-Modellierungssprache)
VTOHL	Vertikal startendes horizontal landendes System (Vertical Take-Off Horizontal Landing)

### Konstanten

$g_0$	Erdbeschleunigung am Boden	(9.80665 m/s <sup>2</sup> )
$\mu_E$	Gravitationsparameter der Erde	(3.986032×10 <sup>14</sup> m <sup>3</sup> /s <sup>2</sup> )
$\omega_E$	Drehgeschwindigkeit der Erde	(7.2921158×10 <sup>-5</sup> rad/s)
$R_E$	Mittlerer Erdradius	(6378.165×10 <sup>3</sup> m)

# 1 Einleitung

Der kommerzielle Satellitentransportmarkt ist ein starker Antrieb für die aktuellen Trägerentwicklungen. Der Konkurrenzkampf um größere Anteile an diesem Markt wird durch die Beteiligung von mehr Nationen mit mehreren Trägersystemen härter. Dies treibt sowohl die Weiterentwicklung existierender Systeme als auch die Entwicklung neuer Trägerraketen an. Parallel dazu ist eine Sättigung bzw. sogar eine sinkende Tendenz des Trägermarktes zu beobachten.

Um die Konkurrenzfähigkeit auf dem derzeitigen Raumtransportmarkt zu wahren und den Erhalt und Zuwachs des Marktes zu gewährleisten, wird von allen Nationen, die Trägersysteme vermarkten, ein Ziel verfolgt: Reduktion der spezifischen Transportkosten. Kurzfristig werden existierende Trägersysteme weiter entwickelt, um die Nutzlastkapazität und die Zuverlässigkeit zu erhöhen. Jedoch wird eine Kostenreduktion von 50% und mehr nur durch wiederverwendbare Systeme erwartet. Mittelfristig werden weltweit teilweise wiederverwendbare Konzepte wie z.B. wiederverwendbare Feststoffbooster untersucht. Langfristig müssen jedoch vollständig neue, voll wiederverwendbare Trägersysteme entwickelt werden.

Der kostspieligen Entwicklung eines wiederverwendbaren Trägersystems müssen Konzeptstudien und Sensitivitätsanalysen vorangehen, um die Entwicklungsrisiken zu mindern. In solchen umfangreichen Studien werden verschiedene Trägerkonfigurationen und -konzepte unter Berücksichtigung der aktuellen Technologien und Technologietrends untersucht. Das Ziel dieser Studien ist es, Trägerkonzepte für die Zukunft zu identifizieren, mit denen die spezifischen Transportkosten wesentlich gesenkt und ein dem zukünftigen Bedarf entsprechender, zuverlässiger und sicherer Zugang zum Weltraum mit hoher Verfügbarkeit ermöglicht werden kann. Sehr wichtige Werkzeuge für solche Projekte stellen hierbei die Analyse-, Simulations- und Optimierungswerkzeuge dar. Daher sind die Verbesserung und die stetige Aktualisierung dieser Programmsysteme für die Untersuchung und den Vergleich verschiedener Trägerkonzepte und die darauf basierende Auswahl eines Leitsystems mit anschließender Detaillierung von zentraler Bedeutung.

Im Rahmen der vorliegenden Arbeit wird das am Institut für Raumfahrtsysteme (IRS) der Universität Stuttgart entwickelte Softwarepaket RTSOPT (Raumtransportsystem-Optimierung) für die multidisziplinäre Optimierung wiederverwendbarer Raumtransporter weiter entwickelt, verbessert und beispielhaft für die Untersuchung des Trägerkonzepts Hopper eingesetzt. Letzteres stellt ein Leitkonzept des nationalen Technologieprogramms TETRA (Technologien für zukünftige Raumtransportsysteme) dar.

## 1.1 Der kommerzielle Raumtransportmarkt

Der kommerzielle Transport von Satelliten zu erdnahen und geostationären Umlaufbahnen stellt mit seinem derzeitigen Wert von etwa 1,6 Milliarden US\$ (Durchschnitt der Jahre 1995-2004) ein großes internationales Geschäftsfeld dar [3], [4]. Trägerbetreiber aus verschiedenen Ländern kämpfen um größere Marktanteile. Während der europäische Trägerbetreiber Arianespace in den Jahren 1995 bis 2000 einen Marktanteil von 57% erringen konnte, ist dieser Anteil in den letzten Jahren bis 2003 auf ca. 45% zurückgegangen. Abbildung 1.1 zeigt den

Verlauf des kommerziellen Raumtransportmarktes und des europäischen Anteils an diesem Markt. Weitere Darstellungen des kommerziellen Raumtransportmarktes sind in Anhang A gegeben.

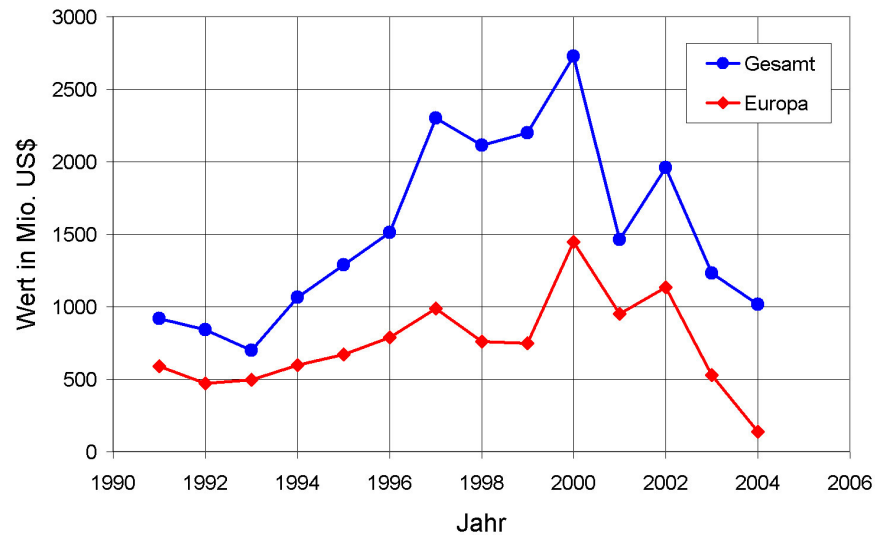


Abbildung 1.1: Weltweite und europäische Einnahmen aus dem Satellitentransport [3], [4].

Die Grafik zeigt einen stetigen Zuwachs bis 2000, einen starken Rückgang danach und eine sinkende Tendenz nach 2002. In den Jahren 1998 – 2000 wurden wegen des Aufbaus von Satellitennetzen Rekord-Startzahlen erreicht. Wie aus dem Diagramm ersichtlich, sank der Wert des Raumtransportmarktes jedoch schließlich von 2,7 Milliarden US\$ in 2000 auf 1,0 in 2004. In diesem Zeitraum ging der europäische Anteil, trotz des Rückgangs des Gesamtmarktes von nur 12%, von 1,5 Milliarden US\$ auf 0,5 zurück. In 2004 ging dieser Anteil durch die Probleme mit der neuen Ariane-5-Trägerversion sogar auf 140 Millionen US\$ (14%) zurück, zeigt aber in 2005 wieder eine aufsteigende Tendenz.

Die Nutzlastkapazitäten der Trägerraketen müssen ständig an die Nutzlastmassen angepasst werden, um Ihre Konkurrenzfähigkeit zu wahren. Die folgende Abbildung 1.2 illustriert die Entwicklung der Satellitenmassen seit 1993 und die Prognose bis 2013. Die Prognose basiert zum Teil auf Auftragsdaten, die internationale Trägeranbieter zur Verfügung gestellt haben.

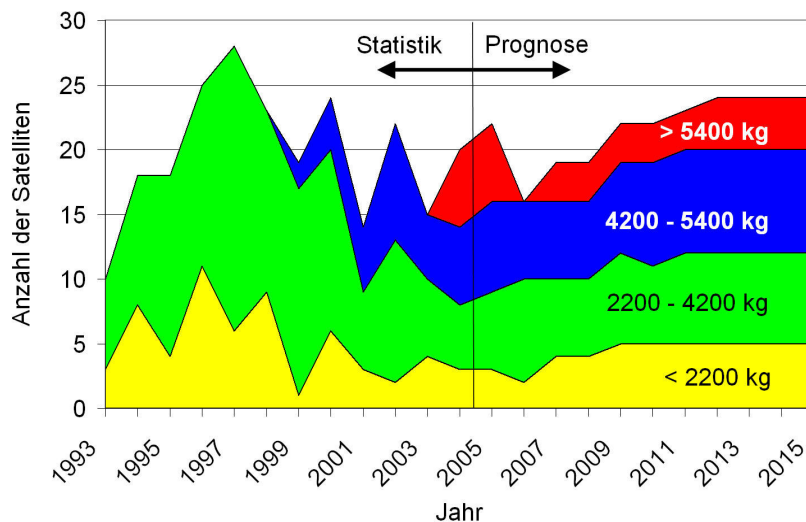


Abbildung 1.2: Entwicklung der Satellitenmassen [5].

Das Bild zeigt eindeutig, dass die Anzahl der schwereren Satelliten auf Kosten der kleineren anwächst. Die hohe Anzahl von Satelliten in der Gewichtsklasse 2-4 t kam durch den Start mehrerer Satellitenkonstellationen zustande. Der Bedarf an kleineren Satelliten bis 4 t bleibt auf einem ungefähr konstanten Wert, während der Bedarf an größeren Satelliten mit bis zu 5,5 t wächst und sogar Schwerlasten mit über 5,5 t Gewicht hinzukommen. Im Anhang A.3 wird zudem gezeigt, dass die durchschnittliche Masse aller gestarteten Satelliten seit 10 Jahren stetig steigt. Diese Entwicklungen bedeuten, dass jede Trägerrakete, deren Nutzlastkapazität nicht weiter ausgebaut wird, einige Jahre später nicht mehr in der Lage sein wird, zwei größere Satelliten gleichzeitig zu starten. Aus diesem Grund war die Entwicklung des Ariane-4-Nachfolgers Ariane-5 sehr wichtig und die Weiterentwicklung der Ariane-5 mit einer zukünftigen Nutzlastkapazität größer als 10 t hat die höchste Priorität, um die Konkurrenzfähigkeit Europas auf dem kommerziellen Raumtransportmarkt zu sichern.

Wie die Trägerentwicklung infolge dieser Markttendenzen verläuft verdeutlicht die folgende Abbildung 1.3. Um einen direkten Vergleich dieser Träger zu ermöglichen, sind hier die geschätzten Startkosten pro kg Nutzlast in die geostationäre Transferbahn für die derzeit wichtigsten Trägersysteme dargestellt. Eine vergleichende Darstellung dieser Träger ist in Anhang A.2 gegeben.

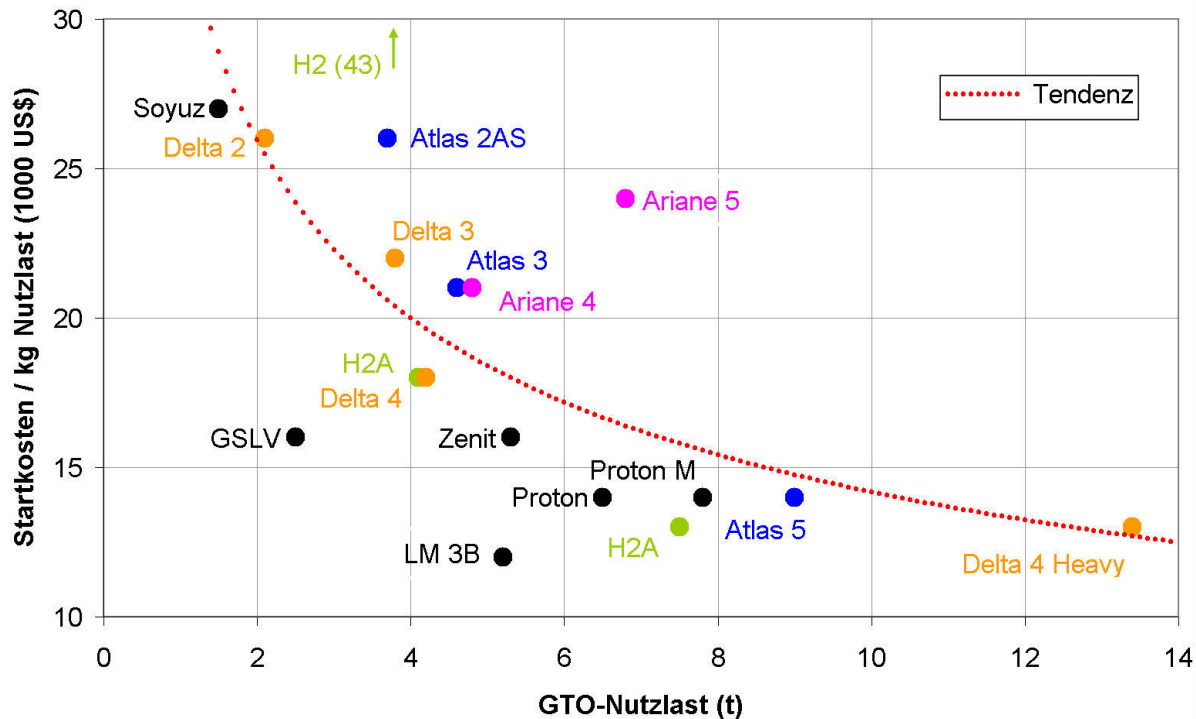


Abbildung 1.3: GTO-Nutzlastkapazitäten und Transportkosten [64].

Die Abbildung lässt einen deutlichen Entwicklungstrend in Richtung höhere Nutzlastkapazitäten und niedrigere Transportkosten erkennen. Die gestrichelte Kurve stellt den allgemeinen Trend dar. Die niedrigsten Transportkosten werden mit dem russischen Proton und dem chinesischen Long March 3B erzielt. In dieser Preisklasse liegen auch die großen amerikanischen Träger Atlas-5 und Delta-4-Heavy, über deren Startkosten jedoch keine zuverlässigen Daten verfügbar sind, bzw. nur die von der Herstellerfirma erwarteten Preise bekannt sind. Die japanische H2 liegt mit 43000 \$/kg außerhalb der Grafik und die anvisierten Transportkosten von ca. 13000 \$/kg mit der größeren Version der H2A scheinen optimistisch.

Die für Doppelstarts von Satelliten konzipierte europäische Trägerrakete Ariane-5 scheint dem weltweiten Trend nicht zu folgen, jedoch müssen hier die Entwicklungskosten und die Anfangsschwierigkeiten der Ariane-5 mit dem Vulkan-2 Triebwerk berücksichtigt werden. Im Weiterentwicklungsprogramm des Trägers ist eine Nutzlaststeigerung auf zuerst 10,5 t, dann auf 12 t geplant. Die erste Steigerung wird durch die neue Oberstufe ESC-A mit dem kryogenen Ariane-4-Oberstufentriebwerk HM7-B erreicht. Die Nutzlastkapazität von 12 t wird dann mit einer neuen Oberstufe und dem neuen, kryogenen Oberstufentriebwerk Vinci erreicht. Daher ist in den nächsten Jahren eine der oben dargestellten Tendenz entsprechende Entwicklung auch bei der Ariane-5 zu erwarten. Ferner ist eine Erweiterung der europäischen Trägerpalette durch das kleinere Trägersystem VEGA und die Vermarktung der Soyuz-Trägerrakete im Gange.

Die Darstellungen bisher zeigen die Entwicklung im kommerziellen Raumtransportmarkt, der Nutzlastmassen und die entsprechende Entwicklung im Trägersektor. Der Markt hat derzeit einen Sättigungszustand erreicht und zeigt sogar eine sinkende Tendenz. Die Fortschritte in der Einwegträger-Technologie zeigt, dass die Transportkosten pro kg Nutzlast sinken aber, wie die Abbildung 1.3 verdeutlicht, dass dieser Verbesserung Grenzen gesetzt sind. Klar ist, dass eine wesentliche Reduktion der Transportkosten z.B. auf 5000 \$/kg oder weniger, nur durch wiederverwendbare Trägersysteme möglich ist. Die Entwicklungskosten hierfür sind zwar höher, aber ihre durch die Wiederverwendung insgesamt niedrigeren Lebenszykluskosten heben diesen Nachteil wieder auf. Niedrigere spezifische Raumtransportkosten würden dann auch das Wachstum des Marktes antreiben. Die weltweiten Aktivitäten zum Thema wiederverwendbare Raumtransportsysteme bestätigen den Stellenwert dieser Thematik. Eine Übersicht dieser Entwicklungen ist im Anhang A.4 gegeben.

Die weltweit durchgeführten aufwändigen Trägersystem-Studien zeigen, dass eine endgültige Entscheidung über zukünftige Trägersysteme noch nicht getroffen werden kann. Dieser Entscheidungsprozess muss, aufgrund der hohen Kosten, Unsicherheiten, der Dynamik im Trägermarkt und in der Technologieentwicklung, ein allmählicher und gestufter Prozess sein und an bestimmten Meilensteinen eine Reorientierung erlauben. Daher ist voraussehbar, dass ein starker Bedarf an weiteren Trägersystem-Analysen und an der Weiterentwicklung von Analyserwerkzeugen bestehen wird. Sowohl die Entwicklungen in den USA als auch der Langzeitplan der europäischen Raumfahrtindustrie [29] bestätigen diese Aussage.

Die Analysen von zahlreichen Trägerkonzepten dienen zum einen zur Beantwortung der Frage "Was für Trägersysteme werden zukünftig gebraucht?" und zum anderen für die Bestimmung der dafür benötigten Technologien. Daher stellen numerische Approximationsmodelle, Simulations- und Optimierungsverfahren einschließlich grafischer Auswertungsprogramme für die Auslegung und Flugleistungsvorhersage neuer Raumtransportsysteme die wichtigsten Analyserwerkzeuge dar. Sie werden eingesetzt, um durch integrierte Missions- und Systemanalysen, Optimierung, Entwurf und Auswertung verschiedener Konzepte Vergleiche durchzuführen, welche die Auswahl- und Entscheidungsprozesse unterstützen. Entwurfs- und Optimierungsmethoden, die verschiedene Fachdisziplinen (z.B. Aerodynamik, Struktur, Flug, Mission) und die Wechselwirkungen zwischen diesen berücksichtigen, werden integrierter Entwurf, Systemanalyse bzw. Multidisziplinäre Optimierung genannt.

In [87] wurde eine Reihe von simultanen Flug- und Fahrzeugoptimierungen durchgeführt und der Einfluss der Missions- und Fluglastforderungen auf den konzeptionellen Fahrzeugentwurf gezeigt. In der Arbeit wurden ein zweistufiger geflügelter HTOL-Träger (Sänger) mit einer



luftatmenden Unterstufe und einer raketenangetriebenen Oberstufe und ein zweistufiger, ebenfalls raketenangetriebener VTOHL-Träger (Ariane-X) mit einer wiederverwendbaren Booster-Stufe und einem geflügelten Orbiter untersucht. Für verschiedene Missionsparameter (Startplatz, Zielbahnhöhe und -inklination, Nutzlast) wurden die Flugleistungen und die Fahrzeugkonfigurationen simultan optimiert. Dabei wurden die Berechnungen mit vorgegebener Nutzlast bzw. Startmasse durchgeführt. Für jede Mission wurden die optimalen Flugbahnen und Fahrzeuge bestimmt und dabei Nutzlaststeigerungen bis zu 30% erreicht.

Solche Analysen, wie z.B. auch in [77] und [85], die das Zusammenspiel von verschiedenen Aspekten wie beispielsweise Treibstoffkomponenten, Strukturmassenanteile verschiedener Stufen, Nutzlastmasse, Entwicklungskostenfaktoren, Größen der Stufen, Leistungsverteilung zwischen den Stufen, bzw. verschiedene Oberstufen-Konfigurationen mit unterschiedlichen Geometrien, Treibstoffen und Triebwerken mit den entsprechenden optimalen Flugleistungen berücksichtigen, zeigen die Bedeutung der Multidisziplinären Analyse. Sie unterstreichen damit auch den Stellenwert von multidisziplinären Optimierungswerkzeugen, die die gekoppelten Wirkungen von verschiedenen Entwurfsparametern und optimalen Flugleistungen numerisch erfassen.

## 1.2 Die Ausgangssituation – das Programmsystem RTSOPT

Für die Missions- und Systemanalyse von Raumtransportsystemen wurden am Institut für Raumfahrtssysteme in mehrjährigen Forschungsarbeiten, Algorithmen und Programmsysteme für die simultane Flug- und Systemoptimierung entwickelt und für Konzeptuntersuchungen zukünftiger Raumfahrtsträger eingesetzt. Das Programmsystem RTSOPT (Raumtransportsystem-Optimierung) wurde in mehr als 20 Jahren konsequent weiter entwickelt. Diese Entwicklung ist im folgenden Kapitel dokumentiert.

RTSOPT fand in vielen nationalen und internationalen Technologieprogrammen Anwendung. Dazu gehören u.a. die Analyse Zukünftiger Raumtransportsysteme (AZURA) [41], Ausgewählte Systeme und Technologien für zukünftige Raumtransportsystem-Anwendungen (ASTRA) [59], [112], [113], [111], Pioneer Pathfinder (USA [61]) [91], Trägerkonzepte Ariane-X, Sänger [41], [84], Raketenträger Kistler (USA [66]) [114] und Technologien für zukünftige Raumtransportsysteme (TETRA) [37].

Neben RTSOPT sollen hier zwei weitere Langzeitprojekte für die Entwicklung von Raumtransporter-Analysewerkzeugen in Deutschland erwähnt werden. Die im Auftrag der ESA am Institut für Flugmechanik und Flugregelung der Universität Stuttgart entwickelte Graphical Environment for Simulation and Optimisation GESOP [110] ist die grafische Arbeitsumgebung für die Aerospace Trajectory Optimisation Software ASTOS (früher ALTOS [121]). ASTOS setzt sich zusammen aus verschiedenen Optimierungsmethoden und einer Modellbibliothek, die u.a. Fahrzeug-, Aerodynamik-, Umwelt- und Missionsbeschränkungsmodelle beinhaltet. Das an der Universität der Bundeswehr in München in Zusammenarbeit mit der Industrie entwickelte Raumtransporter-Entwurfsprogramm Cadehyp [65] wird eingesetzt, um vorwiegend luftatmende Fahrzeugkonzepte zu untersuchen. Während GESOP eine sehr vielseitige, kommerziell vertriebene Software ist, die für die detaillierte Flugsimulation und Bahnoptimierung entwickelt wurde, ist Cadehyp ein Werkzeug, bei dem der Schwerpunkt im Bereich Fahrzeugentwurf und Entwurfsoptimierung bei vorgegebener Flugbahn liegt. RTSOPT unterscheidet sich von diesen beiden Beispielen im Wesentlichen dadurch, dass das Ziel des For-

schungsprojekts die Flugleistungsmaximierung durch simultane Flugbahn- und Systemoptimierung von Raumtransportsystemen betrifft.

Weltweit und insbesondere in den USA werden entsprechend Analysewerkzeuge entwickelt. Aufgrund der ähnlichen multidisziplinären Vorgehensweise und der Ähnlichkeit des untersuchten Trägerkonzepts zu Hopper sei hier [119] genannt. Eine kurze Darstellung dieser Arbeit ist im Anhang A.4 gegeben.

Die Basis für RTSOPT (Flugsimulation, Systemdynamik, Aerodynamik-, Massenmodellierung, Flugsteuermodelle, Flugoptimierung, usw.) wurde in den Arbeiten [100], [94], [99], [95], [97] aufgebaut und durch die Arbeiten [96], [41], [12], [84], [86], [87] weitergeführt.

In [96] wurden numerische Analyseverfahren für die Flug- und Antriebsoptimierung und die Systemanalyse zukünftiger Raumfahrtträger entwickelt und zur Flugleistungsvorhersage unterschiedlicher Fahrzeug- und Antriebskonzepte für eine Transportaufgabe in den erdnahen Orbit eingesetzt. Ein derartiges Analysewerkzeug war bis dahin in Deutschland nicht verfügbar. Auch ein systematischer Vergleich der Transportleistungen ein- und zweistufiger Raumtransporter mit unterschiedlichen Antriebssystemen mit Raketen-, Turboraketen- und Staustrahlkomponenten wurde bis dahin nicht durchgeführt.

In der Arbeit wurden verschiedene Optimierungsverfahren erprobt und ausgewertet. Der erste Lösungsansatz mit einem Konjugationsverfahren im Funktionenraum wurde wegen mangelnder Flexibilität gegenüber wechselnden Missionsverhältnissen, Konvergenzschwierigkeiten und Speicherplatzproblemen zugunsten einer Formulierung der Aufgabe als Parameteroptimierungsproblem aufgegeben. Dieses wurde mit einem Gradienten-Projektions-Algorithmus und einer Sequenziellen Quadratischen Optimierungsmethode (NLPQL) gelöst. Die Parameteroptimierung liefert wegen der ihr inhärenten Parametrisierung der Steuerfunktionen nur suboptimale Lösungen. Diese konnten für die in [96] gelösten Optimierungsaufgaben genügend genau gefunden werden. Für luftatmende Trägerfahrzeuge wurde, durch die direkte Erfüllung der Missionsbeschränkungen für den Auftrieb und die Staudruckbelastung, eine weit reichende Reduzierung der Dimension des Flugführungsmodells erreicht.

In [96] wurde weiterhin gezeigt, dass die Parameteroptimierung die geeignete Vorgehensweise bei der Systemanalyse ist, da sich die Projektierung eines Transportsystems stets iterativ mit allmählich zunehmendem Detaillierungsgrad vollzieht und die Definitions- und Entwurfsphasen durch eine vergleichsweise ungenaue Kenntnis der Antriebs- und Fahrzeugeigenschaften bei wechselnden Missionsforderungen gekennzeichnet sind.

Die Studie Analyse Zukünftiger Raumtransportsysteme AZURA [46] zeigte den Bedarf an einem Analysewerkzeug für die Gesamtoptimierung von Raumtransportsystemen in einer für Konzeptstudien anwendbaren Form. Um diesen Bedarf zu decken, wurde das Analysewerkzeug RTSOPT aufgebaut [41]. Es wurde ein Entwurfsprogramm entwickelt, das die Konzeption von typischen zukünftigen Raumfahrzeugen erlaubt. Empirische Näherungsverfahren für die Aerothermikanalyse wurden in das Entwurfsverfahren integriert. Die Definition der zugrunde gelegten Technologie wurde durch ein statistisches Analyseprogramm vereinheitlicht und erleichtert. Vorhandene sowie die neu erstellten Programm-Module wurden so modifiziert bzw. aufgebaut, dass sie zu einem halbautomatischen Flug- und Systemoptimierungsalgorithmus zusammengefügt werden konnten. Für die einzelnen Programmteile wurde eine Datenbasis geschaffen, die die Bereitstellung aller für die Simulation notwendigen Modelle erlaubt und die als Referenz zur Verifizierung der erzielten Ergebnisse dient.

Diese sequenzielle Flugbahn- und Fahrzeugentwurf-Optimierungsmethode besteht aus einem automatischen Optimierungszyklus, der die Flugleistungsoptimierung, Optimierung der Fahr-

zeugskalierung und die Massenanalyse beinhaltet und einer anschließenden interaktiven Analyse der Fahrzeugkonfiguration und Aerodynamik [101]. Diese Vorgehensweise hat sich sehr gut bewährt und ist bis zur vorliegenden RTSOPT-Version so geblieben (Abbildung 3.1). In [41] wurde das Programmsystem auf verschiedene Varianten von Ariane-X (Ariane-5-Nachfolgerkonzept) erfolgreich angewendet. Die Versuche, RTSOPT auf das luftatmende Trägersystemkonzept Sänger anzuwenden, scheiterten an Konvergenzproblemen, die aus der großen Empfindlichkeit der Flugsteuerung auf Änderungen in den Systemparametern entstanden. Für die Analyse von Sänger wurden daher indirekte Koppelmechanismen eingesetzt.

In [12] wurden die im Rahmen des Sonderforschungsbereiches SFB 259 "Hochtemperaturprobleme Rückkehrfähiger Raumtransportsysteme" an der Universität Stuttgart entwickelten Analysealgorithmen auf aerodynamisch gestützte Raumflugmanöver auftrieberzeugender Fahrzeuge angewendet. Für die missionsspezifische Konfigurationsoptimierung, Massenanalyse, Berechnung der aerodynamischen Fahrzeugeigenschaften, Stabilitätsanalyse, Berechnung der aerothermodynamischen Fluglasten und die Flugleistungsoptimierung wurden geeignete Methoden entwickelt und eingesetzt. Die parametrische Fahrzeugmodellierung wurde aus analytischen bzw. diskreten Flächenmodellen zusammengesetzt. Die Druckverteilung auf der Fahrzeugoberfläche wurde durch flächeninklinationsabhängige Methoden berechnet. Für die Berechnung der laminaren und turbulenten Wärmeströme im hypersonischen Flug wurde eine dreidimensionale Realgas-Methode entwickelt. Bei vorgegebener Druckverteilung wurden reibungsfreie Oberflächenstromlinien mit dreidimensionalen Euler-Gleichungen bestimmt. Der konvektive Wärmestrom wurde mit für das in ein axialsymmetrisches Koordinatensystem transformierte Strömungsfeld bekannten laminaren und turbulenten Wärmelastmodellen bestimmt. Bei der Bestimmung des konvektiven Wärmestroms wurden Modelle entwickelt, die das chemische Gleichgewicht des Gases und die Krümmung der Fahrzeugoberfläche berücksichtigen. Sowohl in [94] als auch in [12] wurde festgestellt, dass die Bestimmung der aerothermodynamischen Fahrzeugbelastungen auf der Basis analytisch vorgegebener Oberflächenfunktionen einen wesentlich stabileren, genaueren und kürzeren Rechenablauf gewährleisten.

Aufgrund der Eigenschaften des generischen Fahrzeugmodells und der erheblichen Zeitvorteile, die sich durch die analytischen Funktionen ergeben, wurden wichtige Voraussetzungen für die Anbindung an Algorithmen zur multidisziplinären Flug- und Systemoptimierung bzw. für die Durchführung einer missionsspezifischen Konfigurationsoptimierung erfüllt.

In dieser Arbeit wurde bei konstant gehaltenen Anströmverhältnissen, für die eine maximale Wärmelast auftritt, der Parametervektor der analytisch definierten Fahrzeugkonfiguration optimiert. Ziel der Optimierung war die Maximierung des Verhältnisses Auftrieb zu Widerstand, erweitert durch die volumetrische Effizienz. Der Untersuchung des Stabilitätsverhaltens und der Aerodynamik des Referenzfahrzeuges folgte die Flugleistungsoptimierung, bei der der Bremsimpuls, Aufstiegsimpuls und der Angleichungsimpuls an die Endbahn minimiert wurden.

Damit wurde eine multidisziplinäre Systemoptimierung unter Berücksichtigung der Disziplinen Fahrzeugkonfiguration, Aerothermodynamik, Stabilität und Flugleistungen durchgeführt. Die Ankopplung des Programms an RTSOPT führte jedoch auf nicht praktikable, lange Rechenzeiten und hat sich darüber hinaus als nicht stabil genug für Optimierungsuntersuchungen von anderen Raumtransportsystemen erwiesen.

Die Hauptmotivation für die Arbeit [84] war in erster Linie, die in [41] festgestellten Konvergenzprobleme bei der Anwendung von RTSOPT auf das Sänger-Konzept zu lösen. Dazu

wurde ein zweistufiger Optimierungsalgorithmus – der Dekompositionsalgorithmus [86] – entwickelt und eingesetzt. Bei dieser Vorgehensweise wird die Gesamtmission in verschiedene Segmente unterteilt, wobei einzelne Flugabschnitte in einer untergeordneten Ebene für gegebene Anfangs- und Endbedingungen getrennt optimiert werden. Anschließend werden diese Randbedingungen, die verschiedene Flugsegmente verknüpfen, in einer übergeordneten Gesamtoptimierung für eine globale Bewertungsfunktion optimal bestimmt. Dieser Ansatz entkoppelt die Flugsteuerparameter von den Systemparametern und trägt entscheidend zur Stabilität des Verfahrens bei. Diese Vorgehensweise bewährte sich bei der Lösung der komplexen Sänger-Transportmission.

Der Dekompositionsalgorithmus erfordert allerdings einen hohen Rechenaufwand, zeitaufwändige Vorbereitungsarbeiten, Analyse der Zwischenergebnisse, neue Problemskalierung und mehrmaligen Neustart des Programms. Um die Rechenzeit zu reduzieren, wurde der Optimierungsalgorithmus parallelisiert. Dies wurde durch die parallele Ausführung der Gradientenberechnung und der eindimensionalen Minimumsuche realisiert. Damit konnte die Rechenzeit um den Faktor 10 reduziert werden.

In [84] wurde des Weiteren das CAD-Werkzeug Pro/ENGINEER [81] mit RTSOPT gekoppelt, um den interaktiven Fahrzeugentwurfsschritt mit CAD-Unterstützung durchzuführen. So wurden die Aufgaben Geometriegenerierung und -modifizierung, Erfassung der geometrischen Abmessungen für die Massenanalyse und die Generierung eines strukturierten Oberflächennetzes für die Aerodynamikanalyse mit diesem Werkzeug durchgeführt. Dazu wurden in dem CAD-Werkzeug Benutzermenüs und -funktionen implementiert.

Neben den algorithmischen Verbesserungen wurde in der Arbeit auch die Benutzerfreundlichkeit des Analyseprogramms verbessert. Die Terminal-Schnittstelle wurde durch eine grafische Benutzeroberfläche mit Menüführung ersetzt.

Mit der Anwendung der simultanen Flugbahn- und Systemoptimierung auf die Missionen eines voll-wiederverwendbaren zweistufigen raketenangetriebenen VTOHL-Trägers Ariane-X und eines luftatmenden HTOL-Systems vom Typ Sänger wurde der Leistungsnachweis von RTSOPT mit dem neuen Dekompositionsalgorithmus erbracht sowie dessen Flexibilität demonstriert. Dabei dürfen allerdings einige Problembereiche und Schwächen nicht übersehen werden, die folgende Aspekte betreffen:

- Konvergenzrate, Konvergenzbereich und Robustheit der numerischen Optimierung
- Vergleichsweise hoher numerischer Aufwand der segmentierten Programmversion (bis zu mehreren Wochen [84], z.T. aufgrund der Verwendung älterer Rechner)
- Abhängigkeit des Softwarepakets von kommerziellen Anwendungen und bestimmten Rechnerplattformen
- Komplexität des Programmcodes und daraus resultierende Schwierigkeiten in der Wartung, Erweiterbarkeit und einfacher Nutzbarkeit

### 1.3 Motivation, Zielsetzung und Vorgehensweise

Die Überwindung der oben genannten Mängel erfordert leistungsfähige Programmsysteme für die simultane Flug- und Systemoptimierung, weil in Konfigurations- und Entwurfsstudien den Einflüssen einer Vielzahl von Systemparametern und Missionsbedingungen nachgegangen werden muss und daher der Beschleunigung einer Problemlösung eine hohe Bedeutung zukommt. Die Steigerung der Leistungsfähigkeiten der numerischen Verfahren und Software-

technologien lässt hier in Verbindung mit den signifikant erhöhten Rechenleistungen im Hardwarebereich der letzten Jahre deutliche Verbesserungen erwarten.

Vor dem Hintergrund der bisher beobachteten Schwächen der früheren RTSOPT-Versionen und der o.g. Entwicklungen ist es das Ziel der vorliegenden Arbeit, die Weiterentwicklung und Verbesserung des Systemanalysewerkzeugs RTSOPT durch den Einsatz fortschrittlicher numerischer Verfahren und moderner Technologien für die Softwareentwicklung einschließlich der Visualisierung zu verfolgen. Teilziele sind dabei

- die Beschleunigung der Konvergenzrate und die Erhöhung der Robustheit der numerischen Optimierung bei gleichzeitiger Erweiterung des Konvergenzbereichs
- die grundlegende Überarbeitung des Programmcodes zur Erhöhung der Einsatzflexibilität durch leichtere Erweiterbarkeit bzw. Wartbarkeit und Plattformunabhängigkeit
- die Erweiterung des Programmsystems durch Visualisierung von Modellen und Ergebnissen für die effizientere Gestaltung der Analyse-, Bewertungs-, und Syntheseaufgaben

Das Grundkonzept der Methodik und die Simulationsmodelle werden in Anlehnung an frühere Arbeiten weiter verwendet. Dabei wird die sequenzielle, iterative Vorgehensweise von RTSOPT, die einen interaktiven Entwurfsschritt und einen automatischen Optimierungsschritt beinhaltet und sich sehr gut bewährt hat, beibehalten. Aus dieser Zielsetzung ergeben sich die nachstehenden verfahrenstechnischen und softwaretechnischen Teilaufgaben.

Die verfahrenstechnische Teilaufgabe beinhaltet die Verbesserung der numerischen Optimierung. Hierfür wurde in RTSOPT ein sehr leistungsfähiges Verfahren SOCS [10] integriert. Hierzu sind umfangreiche Modifikationen des Programmpakets erforderlich. Diese betreffen die Softwarestruktur und den Programmablauf. Mit dem bisher benutzten Optimierungsverfahren NLPQL [92] liegt am Institut für Raumfahrtssysteme viel Erfahrung vor. Für den Einsatz von SOCS musste, über vorhandene Erfahrungen hinaus, das Verständnis und Wissen über das am IRS erstmals eingesetzte Verfahren aufgebaut werden. Das IRS ist in der Tat der erste kommerzielle Kunde der SOCS-Software.

Die softwaretechnische Teilaufgabe umfasst die Entwicklung eines neuen Software-Konzepts (Software Engineering) und den Einsatz aktueller Software-Technologien. Dies umfasst die Identifizierung der Anforderungen an das Systemanalysewerkzeug, die Auswahl der Programmiersprache, der Visualisierungsmethode und die Spezifikation der Benutzerschnittstelle und schließlich die Implementierung der Softwaremodule. Der Softwareentwurf, der vor der Implementierung stattfindet, ist in Kap. 2 dokumentiert.

Der Leistungsnachweis der neuen RTSOPT-Fassung wird schließlich anhand von typischen Anwendungen für die Trägersystemanalyse erbracht. Gegenstand dieser Analysen ist das Trägerkonzept Hopper, das Leitkonzept des derzeitigen nationalen Technologieprogramms ASTRA. Bei der Optimierung der Gesamtmission von Hopper wurden die Restriktion der Bahninklination bei der Stufung zu Gunsten höherer Nutzlastmasse aufgegeben und die Stufungsbedingungen optimiert. Dies ergibt eine maximale Nutzlast von 8770 kg bei einer optimalen Stufungs-Bahninklination von  $14,3^\circ$ . Im Anwendungsteil wird ferner die Flexibilität und Leistungsfähigkeit des verbesserten Werkzeugs RTSOPT demonstriert. Neben der signifikanten Konvergenzbeschleunigung gegenüber früheren Versionen wird auch der Vorbereitungs- und Modellierungsaufwand für Systemanalysen drastisch reduziert.

**Übersicht:**

Das folgende Kapitel 2 behandelt softwaretechnische Überlegungen mit den Teilaspekten Programmiersprache, Visualisierungsmethodik und Benutzerschnittstelle. Kapitel 3 gibt eine Beschreibung des Programmsystems RTSOPT mit Hinweisen zu grundlegenden Annahmen und Simulationsmodellen. Das Kapitel 4 ist dem Optimierungsverfahren SOCS (Sparse Optimal Control Software) gewidmet. Nach der Beschreibung der neuen Visualisierungsmodulen des RTSOPT in Kapitel 5 folgt in Kapitel 6 die Darstellung der numerischen Anwendungsergebnisse, die die Hopper-Trägermission betreffen.

# 2 Softwaretechnische Überlegungen

Für die Verbesserung des bisherigen Programmsystems RTSOPT wird ein Softwarekonzept entwickelt, das die bisher beobachteten softwaretechnischen Probleme und Schwächen des Programmpakets überwindet.

Die Qualität einer Software wird nach [103] anhand der Qualitätsmerkmale Wartbarkeit, Zuverlässigkeit, effiziente Nutzung von Rechnerressourcen und Eigenschaften der Benutzerschnittstelle gemessen. Diese Anforderungen sind Bestandteil des im Folgenden entwickelten Softwarekonzepts für RTSOPT. Bei der Softwarekonzeption müssen die Entwicklungskosten bzw. der Entwicklungsaufwand berücksichtigt werden. Meistens können nicht alle Anforderungen erfüllt werden, weil sie teilweise gegensätzlich sind. So ist z.B. ein für die maximale Rechenleistung optimierter Programmcode schwer lesbar und damit nicht wartungsfreundlich. Andererseits erhöhen sich nach Abbildung 2.1 die Entwicklungskosten für jede weitere Verbesserung exponentiell. Daher müssen bei der Softwareentwicklung Kompromisse zwischen verschiedenen Anforderungen und dem Entwicklungsaufwand gefunden und Schwerpunkte gesetzt werden.

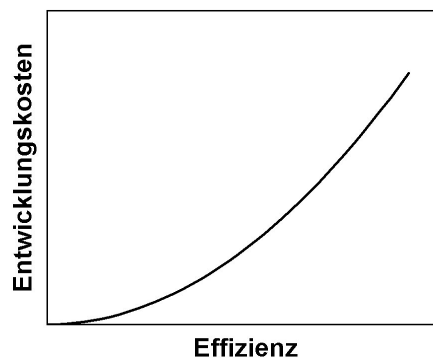


Abbildung 2.1: Kosten der Softwareeffizienz [103].

In diesem Kapitel werden zunächst die Aufgaben des Analysewerkzeuges RTSOPT und die Anforderungen an die Software definiert und danach die softwaretechnischen Lösungen diskutiert und ausgewählt. Anschließend werden die Implementierung und Modultests durchgeführt. Schließlich wird das Softwaresystem integriert, die Funktionsfähigkeit überprüft und seine Leistungsfähigkeit anhand von typischen Anwendungen demonstriert. Diese Vorgehensweise entspricht dem Wasserfall-Modell des "Software Engineering". Die Abbildung 2.2 stellt diese Vorgehensweise schematisch dar.

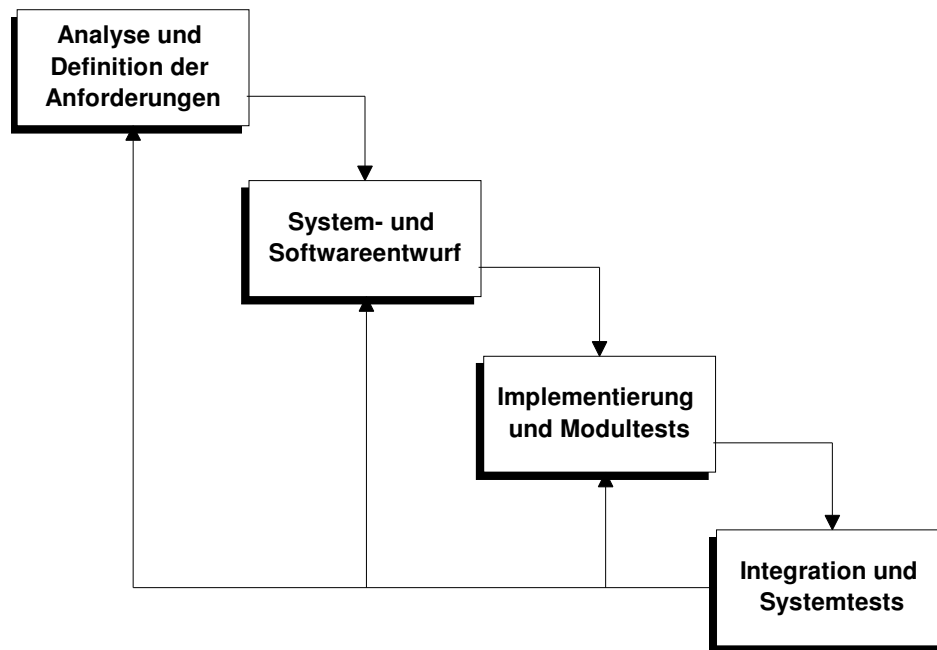


Abbildung 2.2: Wasserfall-Modell der Softwareentwicklung [103].

Die Darstellung illustriert die Stufen und die erforderlichen Iterationsschritte während der Softwareentwicklung. Bei dieser Vorgehensweise werden nach jedem Entwicklungsschritt das Softwaresystem ausgewertet und nach Bedarf vorangehende Entwicklungsschritte aktualisiert bzw. wiederholt. Die Entwicklung wird mit einem Systemtest abgeschlossen, bei dem das Softwaresystem anhand eines realistischen Anwendungsszenarios erprobt wird.

Während der Entwicklung der neuen RTSOPT-Fassung waren solche Iterationen für die Implementierung eines neuen Optimierungsverfahrens und für die Modifikationen bzw. Erweiterungen aufgrund zunehmender Erfahrungen und des Verständnisses des Optimierungsverfahrens erforderlich. So wurden beispielsweise zunächst nur einphasige Flugleistungsoptimierungen implementiert, anschließend schrittweise Optimierungsprobleme mit mehreren Flugphasen und schließlich mit Systementwurfsparameter angegangen.

Nachfolgend werden die bisher beobachteten Softwareprobleme, die Vorteile der Software-Techniken, die bei der Weiterentwicklung von RTSOPT eingesetzt werden und die Vorgehensweise bei der Auswahl dieser Techniken erläutert.

## 2.1 Die Programmiersprache

Das Kernmodul des Programmpakets RTSOPT besteht aus den Komponenten Flugsimulation, Rechenmodelle und numerische Optimierung. In diesem Modul werden die Flugsimulation und die simultane Optimierung der Flugleistungen und des Fahrzeugentwurfs durchgeführt. Vor der Wahl einer bestimmten Programmiersprache für das Programm-Modul werden die Anforderungen an das Softwarepaket aufgestellt. Auf der Basis dieser Anforderungen werden zunächst zwei Arten der Programmierung betrachtet: sequenziell und objektorientiert. Anschließend wird die Programmiersprache für das Programm-Modul gewählt.



### Die Anforderungen an das Softwarepaket

Folgende Anforderungen müssen vom Optimierungsmodul des RTSOPT erfüllt werden:

Hardwareunabhängigkeit: Die dynamische Entwicklung der Leistungsfähigkeit der Computer ermöglicht nicht nur die schnellere Lösung der gleichen Probleme, sondern erlaubt auch die Inangriffnahme neuer und komplexerer Problemstellungen. Um von dieser Entwicklung zu profitieren, müssen Simulations- und Analysewerkzeuge oft auf neue Hardware portiert und neuer Software angepasst werden. Das Problem bei dieser Weiterentwicklung und Anpassung ist der große Aufwand, der damit verbunden ist, die grafische Benutzeroberfläche und grafischen Werkzeuge zur Ergebnisauswertung, die hardware- bzw. betriebssystemspezifischen Funktionen und die Schnittstellen zu externen Anwendungen anzupassen bzw. neu zu kodieren. Der Umstieg von einer Workstation unter UNIX zu einem PC unter Windows kann z.B. folgende Probleme mit sich bringen: Betriebssystemfunktionen, Grafikbibliotheken, Compiler, externe Anwendungen, Programmbibliotheken und die Software-Lizenzbedingungen sind unterschiedlich.

Schwierigkeiten bei der Portierung der früheren Version von RTSOPT von einer Workstation auf einen PC unter Windows waren z.B. die Verwendung von UNIX-spezifischen Dateiverknüpfungen, Systembefehlen und Umgebungsvariablen, die von Windows nicht unterstützt werden.

Die einfache Portierung eines Analysewerkzeuges auf neuere, leistungsfähigere Rechner kann nur durch die Hardwareunabhängigkeit, die zugleich die Unabhängigkeit von einem bestimmten Betriebssystem mit sich bringt, gewährleistet werden.

Erweiterbarkeit, Wartbarkeit: Zum einen, um neue Rechenmodelle und verbesserte mathematische Verfahren, zum anderen, um neue Analysefunktionen im Programm zu integrieren, muss ein Analysewerkzeug ständig weiter entwickelt, verbessert und für neue Aufgabenstellungen erweitert werden. Daher ist der Aufwand bei der Erweiterung und Wartung des Programms ein wichtiges Qualitätsmerkmal der Software. Um diesen Aufwand so gering wie möglich zu halten, müssen Erweiterungen ohne genaue Kenntnisse über den Programmcode und ohne Änderungen in anderen Teilen des Programms möglich sein. Dies wird einerseits durch einen strukturierten, modularen Programmaufbau, der die Abgrenzung von Programm- und Datenmodulen gewährleistet und andererseits durch einen Programmcode realisiert, der selbsterklärend ist. Abbildung 2.3 zeigt beispielhaft zwei selbsterklärende Programmzeilen.

```
atmos(2).Update(altitude,airDensity)
stage(1).engine(2).vacuumIsp = ...
```

Abbildung 2.3: Beispiele für selbsterklärenden Programmcode.

Das erste Beispiel zeigt den Aufruf des Atmosphärenmodells der Flugphase 2, welches die Luftdichte in der Flughöhe liefert. Das zweite zeigt einen Variablenamen, der den spezifischen Vakuumimpuls des zweiten Triebwerkes der ersten Stufe von einem Träger bezeichnet. Ein so kodiertes Programm bedarf weniger Programmdokumentation und sowohl die Einarbeitung in den Programmcode als auch die Erweiterung der Software gestaltet sich einfach.

Modellierung komplexer Datenstrukturen: Die numerische Beschreibung einer Raumtransportmission (Rand- und Flugbeschränkungen), der Fahrzeugstufen (Masse, Aerodynamik, Triebwerke), der Umwelt (Erdforn, Gravitationsfeld, Atmosphäre) und der Flugsteuerung (Schub, Anstellwinkel, usw.) besteht aus Datentabellen und Rechenmodellen (Daten und

physikalische Beziehungen). Abbildung 2.4 illustriert exemplarisch die verschiedenen Phasen einer Raumtransportmission, für deren Beschreibung unterschiedliche Rechenmodelle benötigt werden.

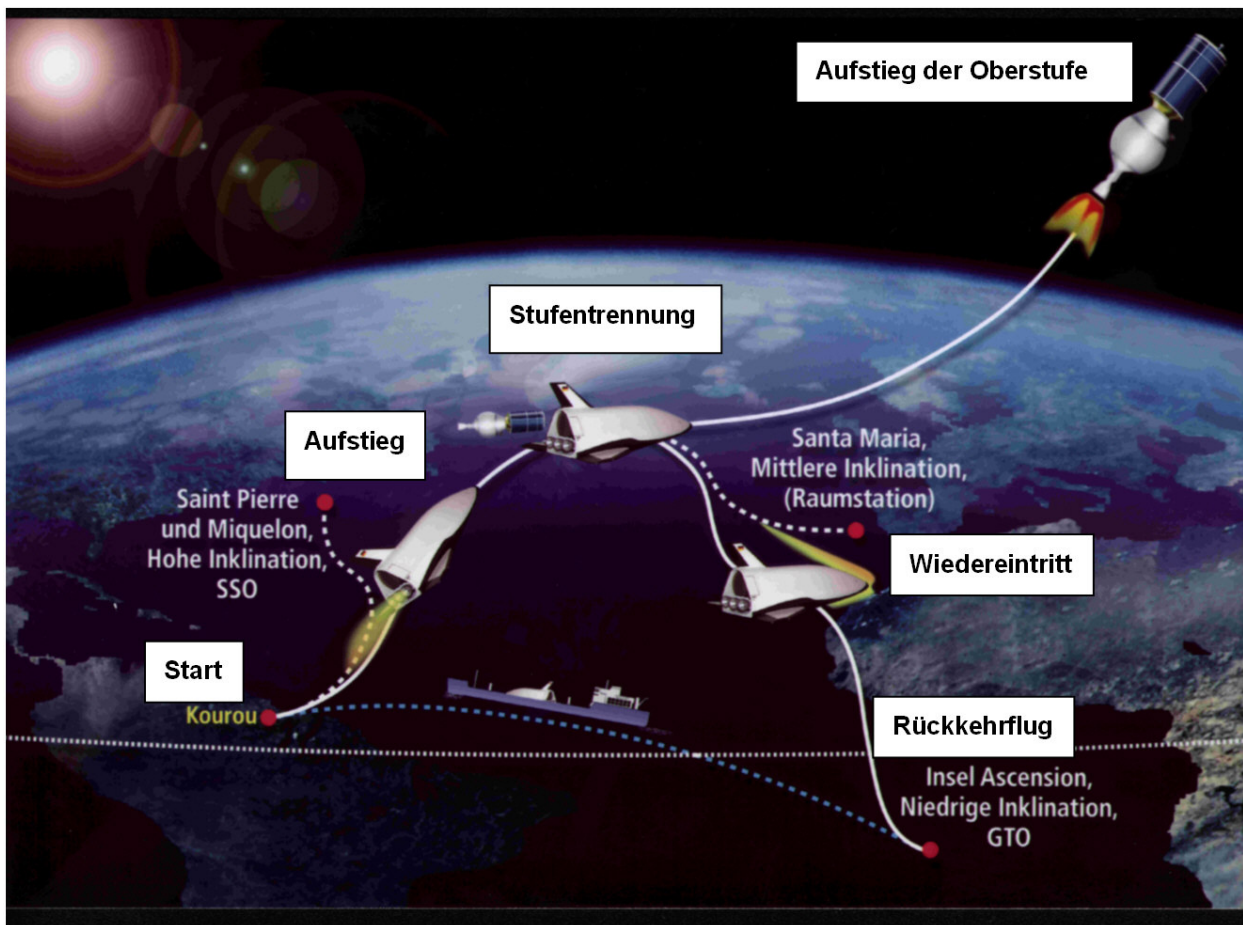


Abbildung 2.4: Flugphasen einer Trägeraufstiegsmission [106].

In der Abbildung ist der Missionsverlauf des Trägerkonzepts Hopper schematisch dargestellt. Aus der Darstellung ist ersichtlich, dass die Transportmission aus verschiedenen Flugphasen (Start, Aufstieg, Stufentrennung, Oberstufenaufstieg, Wiedereintritt und Rückkehrflug) besteht, für die jeweils die Rand- und Flugbeschränkungen definiert und unterschiedliche Rechenmodelle zur Verfügung gestellt werden müssen. Die Simulationsmodelle umfassen die Umwelt-, Fahrzeugstufen-, Triebwerk-, Aerodynamik-, Massen- sowie Dynamik- und Flugsteuerungsmodelle.

Die programminterne Darstellung der Beschreibung einer Raumtransportmission erfordert die Verwaltung einer komplexen, baumförmigen Datenstruktur, die sich aus der Verknüpfung zahlreicher Rechenmodelle sowie Rand- und Flugbeschränkungen ergibt. Abbildung 2.5 stellt die Datenstruktur für die oben beschriebene Trägermission schematisch dar.

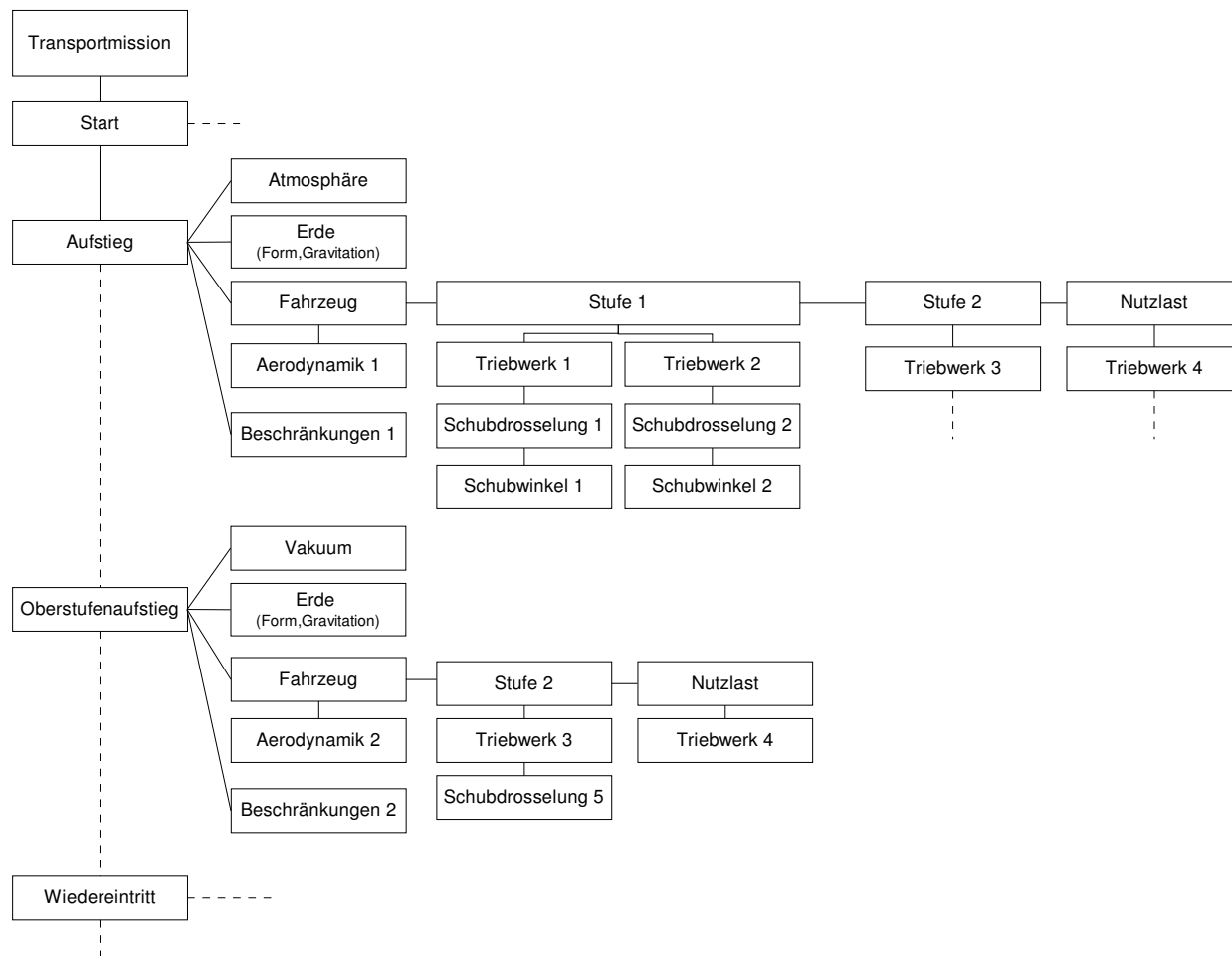


Abbildung 2.5: Baumstruktur der Rechenmodelle.

Die Verwaltung der zahlreichen Modelle wird zusätzlich durch unterschiedliche Modelltypen (Datentabelle, verschiedene Interpolationsmethoden, analytische Beschreibung, Unterprogramm) komplizierter. Aufgrund der hier dargestellten Komplexität der Datenstrukturen und der Anforderung, sehr unterschiedliche Raumtransportmissionen und Trägersysteme ohne Programmcode-Anpassungen analysieren zu können, muss eine Programmiersprache gewählt werden, die eine übersichtliche, flexible und effiziente Implementierung und Verwaltung solcher Datenstrukturen und Modellbäume ermöglicht.

**Rechenleistung:** Die multidisziplinäre, numerische Flug- und Systemoptimierung mit vielen Optimierungsvariablen und Beschränkungen erfordert eine Vielzahl von Simulationen, in denen die o.g. Modelle sehr oft ausgewertet werden. Die Lösung eines solchen Optimierungsproblems stellt ein sehr aufwändiges numerisches Problem dar. Um akzeptable Rechenzeiten zu gewährleisten, muss das Programmpaket die aktuell verfügbare Rechenleistung effizient nutzen.

### Vergleich: Sequenzielle und Objektorientierte Softwareentwicklung

Vor der Wahl einer Programmiersprache muss zunächst die Frage beantwortet werden, ob eine sequenzielle oder objektorientierte Sprache besser geeignet ist, um den oben erläuterten Anforderungen gerecht zu werden.

In der objektorientierten Programmierung werden nach [108] Operationen zum Zugriff und zur Manipulation mit der Beschreibung (Klasse) der Datenstrukturen zusammengefasst (Objekt). Es ist möglich, nahezu beliebige strukturierte Datentypen mit den dazugehörigen Operatoren zu definieren. Damit kann das Softwaresystem als ein möglichst direktes und gegenständliches Modell eines Anwendungsbereichs realisiert werden. Von einer bekannten Basisklasse (Simulationsmodell) können weitere (Triebwerk, Atmosphäre, Atmosphäre-Typ-1, Atmosphäre-Typ-2) abgeleitet werden, welche sowohl die Datenstruktur (Modellname, Typ, Datentabelle) als auch die Operatoren (Load, Init, Update) erben. In diesen abgeleiteten Klassen müssen dann nur noch die spezifischen Erweiterungen implementiert werden. Die objektorientierte Programmierung hat daher den Vorteil, dass zum einen alle Modelle in der Simulation auf die gleiche Weise referenziert werden können und zum anderen die Entwicklung weiterer Rechenmodelle unabhängig vom Rest des Softwaresystems erfolgen kann. Auch die Entwicklung gestaltet sich durch die Wiederverwendung bereits implementierter Klassen produktiver.

Die sequenzielle Programmierung mit C oder Fortran77 ist im wissenschaftlichen Rechnen mehr verbreitet als die objektorientierte Programmierung. Die Kriterien Hardware-unabhängigkeit und gute Rechenleistung können sowohl mit der sequenziellen als auch mit der objektorientierten Variante (typischerweise C++) erfüllt werden. Jedoch haben moderne, objektorientierte Programmiersprachen aufgrund ihrer oben genannten Eigenschaften Vorteile bei der Erfüllung der Kriterien Modulare Erweiterbarkeit, Lesbarkeit und Modellierung komplexer Datenstrukturen. Die folgende Abbildung 2.6 veranschaulicht die Anwendung der objektorientierten Programmierung in einem Flugsimulator:

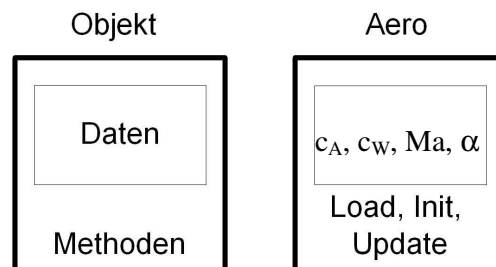


Abbildung 2.6: Schematische Darstellung von objektorientierten Rechenmodellen.

Das Objekt `Aero` beinhaltet sowohl die aerodynamischen Beiwerte in Abhängigkeit von der Machzahl  $Ma$  und des Anstellwinkels  $\alpha$ , als auch die Methoden, mit denen diese Daten eingelesen (Load), initialisiert (Init) und interpoliert (Update) werden. Damit stellt das Rechenmodell bzw. Objekt eine Einheit mit den eingangs erläuterten Vorteilen dar. Abbildung 2.7 demonstriert die Verwendung eines solchen Rechenmodells im Simulationsprogramm.

```
aero(1).Update(machNumber, angleOfAttack, cA, cW)
```

Abbildung 2.7: Verwendung eines objektorientierten Rechenmodells.

Die Methode `Update` des Objekts `aero(1)` (Aerodynamikmodell der Flugphase 1) greift auf die eigenen Daten zu, die in dem Objekt enthalten und für andere Programmteile unzugänglich sind (Datenkapselung) und liefert die aerodynamischen Beiwerte  $c_A$  und  $c_W$  für die aktuelle Flugmachzahl `machNumber` und den Anstellwinkel `angleOfAttack`. Zur Auswertung der Rechenmodelle werden in der Simulation immer die Auswertungsfunktionen der Rechenmo-

dell-Objekte aufgerufen. Da jedes Objekt seine spezifischen Methoden auf seine Daten anwendet, wird immer die richtige Methode die richtigen Daten auswerten, auch wenn es viele verschiedene Modellarten gibt.

Bei dieser Vorgehensweise ist eine detaillierte Kenntnis des Simulatorcodes für die Implementierung eines neuen Modelltyps nicht nötig. Auf der anderen Seite ist für die Benutzung von verschiedenen Modellen eine gute Kenntnis der internen Funktionsweise und der Datenstrukturen der Modelle nicht erforderlich. So können im Simulationsprogramm viele Modelle als eigenständige und unabhängige Einheiten integriert werden. Diese offenkundigen Vorteile des objektorientierten Softwareentwurfs werden in [103] im Detail diskutiert, während die Grundkonzepte der objektorientierten Programmierung in [108] beschrieben sind.

Aufgrund der erläuterten Vorzüge wurde hier die objektorientierte Programmierung der sequenziellen Programmierung vorgezogen.

### **Auswahl der Programmiersprache**

Für die objektorientierte Überarbeitung des RTSOPT-Codes wurden die Programmiersprachen Fortran90, Java, Ada und C++ betrachtet. Fortran90 zählt zwar nicht zu den reinen objektorientierten Sprachen, hat aber durchaus objektorientierte Züge und ermöglicht eine solche Softwareentwicklung. In [21] bis [24] werden wissenschaftliche Hochleistungsrechnung und Implementierung von aus C++ gut bekannten objektorientierten Konzepten wie Polymorphie und Vererbung mit Fortran90 beschrieben. Eine komplette Beschreibung der Programmiersprache wird in [89] gegeben. Im Folgenden werden die Vor- und Nachteile der o.g. Programmiersprachen diskutiert.

Die objektorientierte Programmierung vollzieht mit Verzögerung den Einzug in wissenschaftliche Programme. Der Grund ist die weite Verbreitung der sequenziellen Programmierung mit Fortran77 oder C. Entwickler neuer, numerischer Analyseprogramme müssen – wenn sie objektorientierte Programmierung praktizieren wollen – nicht nur eine neue Programmiersprache erlernen, sondern auch ein neues Softwarekonzept. Hier hat Fortran90 zwei wichtige Vorteile. Erstens können sequenzielle Programme in Fortran77 ohne Änderungen in Fortran90-Programme eingebunden werden. Zweitens wird der noch in der Entwicklung befindliche Nachfolger Fortran2003 eine vollkommen objektorientierte Sprache und kompatibel zu Fortran90/95 sein [31]. Damit ermöglicht Fortran90 einen graduellen Übergang von der sequenziellen zur objektorientierten Programmierung.

Das Analysewerkzeug RTSOPT wird am Institut für Raumfahrtsysteme oft im Rahmen von Studien- und Diplomarbeiten eingesetzt. Die meisten Benutzer sind Studenten ohne gute Programmierkenntnisse. Sie werden häufig erst bei der Implementierung eines neuen Simulationsmodells (z.B. Aerodynamikmodell) mit der Programmierung konfrontiert. Daher muss die Programmiersprache schnell erlernbar sein. Erfahrungsgemäß kann Fortran innerhalb weniger Wochen gelernt werden. Bei den Sprachen C++, Java und Ada müssen eine komplexere Syntax und objektorientierte Konzepte erlernt werden. Objektorientierte Fortran90-Module können dagegen zuerst sequenziell und unabhängig vom Hauptprogramm implementiert und im Anschluss durch geringfügige Anpassung in einer Klasse eingebettet, eingekapselt, in das objektorientierte Hauptprogramm integriert werden. Dieser Schritt kann durch den Betreuer des Codes in kürzester Zeit erledigt werden.

Interpretative Sprachen wie Java werden während der Programmlaufzeit interpretiert bzw. in die Maschinensprache übersetzt und ausgeführt. Dies führt allgemein zu Einbußen in der Rechenleistung. Ein umfangreicher Leistungsvergleich zwischen C, C++, Fortran und Java zeigt jedoch, dass Java Codes nur geringfügig langsamer als Codes in den anderen betrachteten Sprachen sind [13]. Ein Vergleich zwischen Fortran, C++ und Ada zeigt etwas längere Ausführungszeiten bei Ada Codes. Aber die Stärke von Ada liegt mehr in den Bereichen Produktivität und vor allem Sicherheit [122]. Beide Untersuchungen zeigen, dass Fortran Codes meistens etwas schneller sind. Diese Hinweise verdeutlichen, dass die Rechenleistung der Programmiersprache kein ausschlaggebendes Kriterium ist.

Die Parallelisierung des Analysewerkzeugs und sein Einsatz auf Supercomputern ermöglicht eine große Leistungssteigerung. Beispielsweise wurde in [84] durch die Parallelisierung der Gradientenberechnung in RTSOPT eine zehnfache Beschleunigung erreicht. Eine solche Leistungssteigerung gibt dem Benutzer mehr Zeit für die Analyse von Ergebnissen, mehr Freiheit bei Proberechnungen und die Möglichkeit, neue komplexere Problemstellungen zu lösen. Hier hat Fortran90 deutliche Vorteile gegenüber Java, C und Ada. Am Rechenzentrum der Universität Stuttgart ist die Verwendung von Fortran90 im Höchstleistungsrechnen sehr verbreitet und wird sehr gut unterstützt. Fortran-Compiler auf Supercomputer sind sehr weit entwickelt. High Performance Fortran (eine für Supercomputer erweiterte Version) basiert auf Fortran90. Wie objektorientierte Programmierung erhält auch Java den Einzug in den Bereich High Performance Computing [27]. Jedoch ist der Einsatz von Fortran90 mit MPI (Message Passing Interface) [83], eine für den Datenaustausch zwischen den Rechnerknoten auf Parallelrechnern sehr verbreitete und etablierte Vorgehensweise. Daher hat Fortran diesbezüglich einen wichtigen Vorteil gegenüber anderen Programmiersprachen.

Da Analysewerkzeuge Ergebnis jahrelanger kontinuierlicher Entwicklungsarbeit und oft eine Kombination von mehreren Programmpaketen sind, müssen bei der Auswahl der Programmiersprache auch die bisher benutzten Sprachen berücksichtigt werden. Es ist zwar möglich, eine Kombination von mehreren Programmiersprachen zu verwenden, jedoch entstehen bei der Benutzung verschiedener Compiler und Plattformen Inkompatibilitäts- und Anpassungsprobleme. Am IRS wurden die meisten Flug- und Systemoptimierungsprogramme und Rechenmodelle der Simulation in Fortran implementiert. Bei Benutzung von Fortran können diese Modelle und weitere Programm-Module weiter gemeinsam benutzt bzw. übernommen werden.

Um Entwicklungszeit zu sparen, werden in Analyseprogrammen externe Programmbibliotheken eingebunden, die mathematische Verfahren wie numerische Optimierung, Interpolation, Gleichungslöser etc. beinhalten.

In RTSOPT werden Interpolationsverfahren, Differentialgleichungslöser und das Nichtlineare Sequenziell-Quadratische Optimierungsverfahren NLPQL [92] aus der Programmbibliothek IMSL [117] verwendet. IMSL ist für verschiedene Plattformen (z.B. PC, Workstations, Supercomputer) und in verschiedenen Programmiersprachen (Fortran, C, Java) erhältlich und sowohl am IRS als auch am Rechenzentrum der Universität Stuttgart verfügbar. In der neuen Version von RTSOPT wird die IMSL-Bibliothek weiterhin benutzt. Jedoch wird das NLPQL-Verfahren durch die moderne Sparse Optimal Control Software (SOCS) [10] von Boeing ersetzt. Während die Rechenzeit bei NLPQL überproportional mit der Anzahl der Optimierungsvariablen steigt, hängt sie bei SOCS nur von der Anzahl der Zustandsvariablen ab. Damit kann eine erhebliche Leistungssteigerung erwartet werden (was durch die Optimierungsergebnisse in Kapitel 6 bestätigt wird). Dieses Optimierungsverfahren ist in Fortran77 implementiert.

Bei der Entscheidung über die Programmiersprache müssen ferner noch die Auswahl an wissenschaftlichen Programmbibliotheken für die gewählte Sprache und die Verfügbarkeit dieser Bibliotheken auf verschiedenen Plattformen berücksichtigt werden. Da Fortran historisch bedingt und durch seine Eignung für numerische Aufgaben im wissenschaftlichen Rechnen sehr beliebt und verbreitet ist, gibt es hierfür eine sehr große Auswahl an wissenschaftlichen, mathematischen und grafischen Programmbibliotheken. Eine übersichtliche Zusammenstellung einer großen Auswahl solcher Bibliotheken wird beispielsweise in [82] gegeben.

Wie bereits erläutert, ist die Lesbarkeit des Programmcodes ein wichtiger Aspekt. Eine strikte Syntax wie bei Fortran77 (z.B. maximal sechs Zeichen für Namen, Spaltenregel) erschwert eine selbsterklärende und lesbare Art der Kodierung. An dieser Stelle muss aber bemerkt werden, dass Fortran90 oder C++ nicht zu lesbarem Programmierstil zwingen. Hier liegt die Verantwortung beim Programmierer. Jedoch bieten moderne Sprachen wie Java, C++, Ada und Fortran90 sehr viel Flexibilität und keine Einschränkungen bei der übersichtlichen Codegestaltung. Ein selbstdokumentierender Programmcode hat neben seiner besseren Lesbarkeit auch den Vorteil, dass der Code ohne eine ausführliche, aufwendige Programmdokumentation verständlich ist und die Implementierung zügiger erfolgen kann, weil z.B. das Suchen und Verwechseln von Variablenamen entfällt.

Bei der Verwendung weniger weit verbreiteter oder "exotischer" Programmiersprachen wird der Austausch von Erfahrungen und Programmcodes mit anderen Benutzern schwierig. Des Weiteren muss sichergestellt werden, dass der benötigte Compiler auf allen in Frage kommenden Plattformen verfügbar ist. Die hier betrachteten Programmiersprachen sind alle weit verbreitet und verfügbar. Lediglich die Unterstützung von Java und Ada ist auf Supercomputern fraglich.

Nach Berücksichtigung der oben diskutierten Kriterien wird für das Analysewerkzeug RTSOPT die objektorientierte Programmierung in Fortran90 ausgewählt. In der Tabelle 2.1 sind die Vor- und Nachteile der betrachteten Programmiersprachen zusammengefasst. Mit (+) wird ein Vorteil, mit (-) ein Nachteil angezeigt. Der Eintrag (0) bedeutet kein entscheidender Vorteil oder Nachteil.

	Fortran90	C++	Java	Ada
Lernen	+	-	-	-
Leistung	+	0	0	0
Supercomputer	+	-	-	-
Vorhandene Software	+	-	-	-
Bibliotheken	+	0	0	0
Lesbarkeit	+	+	+	+

Tabelle 2.1: Vor- und Nachteile der betrachteten Sprachen.

## 2.2 Die Visualisierungsmethode

Die Visualisierung ist ein unverzichtbares Hilfsmittel bei der Ergebnisaufbereitung und -auswertung einer Simulation und Optimierung von Trägersystemen. Für die Wahl einer oder mehrerer Visualisierungsmethoden müssen die zu erfüllenden Aufgaben und die technischen Anforderungen berücksichtigt werden.

### Die Aufgaben der Visualisierung

Die Aufgaben der Visualisierung betreffen die Darstellung zwei- und dreidimensionaler Plots (z.B. von Rechenmodellen und Simulationsergebnissen), die Darstellung des Raumfahrzeugs und der Flugbahn. Im Folgenden werden diese Teilaufgaben näher erläutert.

2D bzw. 3D Darstellungen: Viele physikalische Modelle (z.B. Aerodynamik-Beiwerte in Abhängigkeit von der Machzahl und dem Anstellwinkel) basieren auf diskreten, mehrdimensionalen Daten, die tabellarisch zusammengefasst sind. Während der Simulation werden solche Modelle durch mehrdimensionale Interpolation zwischen den diskreten Daten ausgewertet. Vor der Verwendung solcher Modelle muss die Qualität der Interpolation durch einen Vergleich zwischen den Datenpunkten und der interpolierten dreidimensionalen Fläche überprüft werden. Dies unterstützt den Benutzer bei der Auswahl einer für die gegebenen Modelldaten am besten geeigneten Interpolationsmethode und bei der schnellen Identifizierung von Fehlern, wie Knicke oder Überschwingverhalten bei der Interpolation. Desgleichen lassen die Zeitverläufe der Zustandsvariablen eine effektive Analyse und Bewertung der Simulationsergebnisse zu.

Darstellung von Raumfahrzeugen: Eine dreidimensionale Modellierung und Darstellung eines Trägersystems ist bei der konzeptuellen Systemauslegung ebenfalls unverzichtbar. Bei der Optimierung einer Trägerkonfiguration werden Subsysteme oder einzelne Entwurfsgrößen wie z.B. die Flügelfläche, die Tankvolumina als zeitunabhängige Parameter in die numerische Optimierung einbezogen. Die dabei notwendige Geometrieskalierung muss aufgrund der beschränkten Geltungsbereiche der Modelle (z.B. Massenmodell, Aerodynamik) und physikalischen Beschränkungen (z.B. Tank-Wand-Abstand) in einem vorgegebenen Bereich eingeschränkt werden. Dabei muss der Benutzer die Änderungen der Geometrie, Überschneidungen zwischen den Subsystemen, die Proportionen und den Optimierungsverlauf verfolgen, analysieren und überprüfen können. Eine detaillierte Beschreibung der Systemskalierung wird in [84] gegeben.

Flugbahndarstellung: Ein wichtiges Hilfsmittel bei der Analyse von Simulationsergebnissen ist die dreidimensionale, animierte Darstellung des Flugverlaufs. Der Nutzen einer solchen Animation wird zusätzlich durch die numerische Anzeige von Flugzustandsdaten erweitert. Besonders bei der Simulation komplizierter, dreidimensionaler Flugmanöver mit Rotationen um alle drei Flugkörperachsen ist eine Animation sehr nützlich, weil sie Bewegungsabläufe leichter beurteilen lässt als z.B. Zeitprofile der verschiedenen Flugvariablen. Nicht unwichtig ist auch die Überzeugungskraft einer solchen Animation auf Außenstehende bei der Präsentation eines Projekts.



### Die Wahl der Visualisierungsmethode

Für die Wahl einer Visualisierungsmethode werden die in Frage kommenden Methoden nachfolgend aufgelistet und ihre Funktionalität bzw. Vor- und Nachteile diskutiert.

Die Visualisierung von Analyseergebnissen kann in einer externen Grafikanwendung durch Importierung von Ausgabedateien in die Anwendung, durch Verknüpfung der Anwendung mit dem Analysewerkzeug oder durch ein im Analysewerkzeug integriertes Grafikprogramm realisiert werden. Die externe Grafikanwendung kann dabei ein kommerzielles oder ein selbst entwickeltes Programmpaket oder eine Kombination sein.

Eine im Analysewerkzeug integrierte Visualisierung bringt viele Probleme mit sich. Das Programmpaket wird sehr komplex, denn die Visualisierungsmethode muss so gewählt werden, dass das komplette Programmpaket plattformunabhängig bleibt. Sie muss kompatibel mit dem Rest des Programmpakets sein. Die Entwicklung der grafischen Funktionen erfordert genaue Kenntnis des gesamten Codes. Das Programmpaket ist dann ohne die grafischen Funktionen nicht einsatzfähig. Besonders bei Verwendung kommerzieller Grafikbibliotheken kann die Portierung des Programmpakets auf eine andere Plattform scheitern, wenn diese Bibliothek für diese Plattform nicht verfügbar ist. Aus diesen Gründen wurde das Visualisierungsmodul des RTSOPT nicht im Programmsystem integriert.

Folgende externe Visualisierungstechniken werden betrachtet:

- Grafikanwendung, entwickelt mit einer kommerziellen Grafikbibliothek
- Grafikanwendung, entwickelt mit Java, OpenGL oder PHIGS
- Kommerzielle Grafikanwendungen
- Virtual Reality Modeling Language (VRML)

Kommerzielle Grafikbibliotheken ermöglichen eine schnelle Entwicklung wissenschaftlicher Grafikpakete. Der entscheidende Nachteil sind die hohen Lizenzkosten im Bereich € 1000 – 5000 [82]. Bei Bedarf an mehreren Lizenzen für mehrere Arbeitsplätze und Plattformen entstehen große Kosten, die sie nur noch für die kommerzielle Softwareentwicklung interessant machen, weil sie einen sehr großen Funktionsumfang und eine einfache Einbindung der Visualisierung in ein Analysewerkzeug ermöglichen.

Die Grafikbibliothek OpenGL [79] ist frei erhältlich, sehr umfangreich und durch die direkte hardwareseitige Unterstützung sehr leistungsfähig. OpenGL ist sehr verbreitet, plattformunabhängig und ihre Spezifikation ist ein Industriestandard [32]. Mit OpenGL und C++ wurde eine Lageanimation für die Unterstützung der Flugdatenanalyse der IRDT-1-Kapsel [38], eine generische dreidimensionale Rückkehrbahndarstellung und parametrische Trägermodelle erstellt. Diese sind in der nächsten Abbildung 2.8 dargestellt.

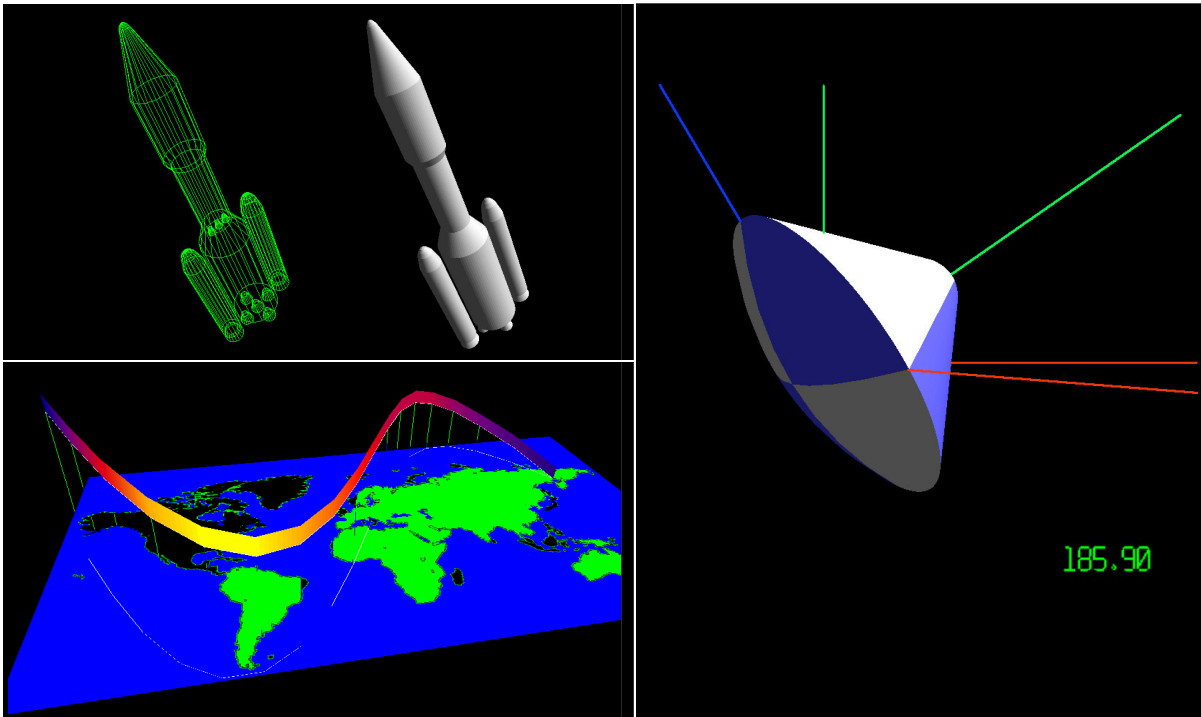


Abbildung 2.8: Animation bzw. Darstellung von Fahrzeugen, Flugbahnen und Kapsel-Taumelbewegungen mit OpenGL.

Die Erfahrung zeigt, dass die Entwicklung kleinerer Grafikpakete mit C++ und OpenGL ohne großen Entwicklungsaufwand möglich ist. Ein entscheidender Nachteil dieser Vorgehensweise ist jedoch der Aufwand bei der programmatischen Modellierung von Raumfahrzeugen. Auch die Entwicklung eines Importfilters für gängige CAD-Dateien ist mit hohem Aufwand verbunden.

Eine Alternative zu OpenGL stellen das Programmer's Hierarchical Interactive Graphics System PHIGS bzw. PHIGS-PLUS (Plus Lumiere and Surfaces) dar. Im Gegensatz zu OpenGL bieten diese Grafikstandards [52], [53] nicht nur Darstellungsfunktionalität, sondern auch Funktionen für die Eingabe, Interaktion, Modifizierung, Ausgabe, Speicherung der Modelle und Verwaltung der Grafikelemente im Speicher [39], [45]. Obwohl PHIGS-PLUS auch Freiformflächen (B-Spline- und Nonuniform B-Spline-Flächen NURBS[39]) unterstützt, ist der Entwicklungsaufwand für das Einlesen von mit CAD-Systemen erzeugten Fahrzeugmodellen, wie bei OpenGL, sehr hoch.

Zum Standardumfang der Programmiersprache Java gehört auch Java3D [109], welches sehr umfangreiche Grafikfunktionen bietet. Da Java eine hardwareunabhängige, interpretative Programmiersprache ist, basiert die Umsetzung der Grafikfunktionen auf der Hardwareseite auf OpenGL. Für Testzwecke entwickelte Java3D-Programme zeigten sehr gute Leistungen unter Windows. Ein Nachteil von Java3D ist jedoch, wie bei OpenGL, die aufwändige Fahrzeugmodellierung. Aufgrund des Entwicklungsaufwandes wurde hier auch auf die Entwicklung eines interaktiven Visualisierungspakets mit Java3D verzichtet.

Kommerzielle Grafikanwendungen und CAD-Systeme sind wegen ihrer Leistungsfähigkeit und ihrem Nutzen bei der technischen Berichterstattung, Modellierung, Systementwurf und

Ergebnisanalyse unverzichtbar. Eine Verknüpfung solcher Programmpakete mit einem selbst entwickelten Analysewerkzeug ist möglich. Eine einfache Methode ist die Kommunikation über Dateien, die zwischen der Anwendung und dem Analysewerkzeug durch Import / Export ausgetauscht werden. Eine anspruchsvolle Vorgehensweise ist die Integration und Kopplung verschiedener, verteilter Anwendungen mit der Common Object Request Broker Architecture CORBA Technologie [78]. CORBA baut auf einem herstellerunabhängigen Standardprotokoll auf, das die Verknüpfung von mehreren Programmen ermöglicht, die auf unterschiedlichen Rechnern, Betriebssystemen und Computernetzen laufen. Eine Anwendung dieser Technologie für die multidisziplinäre Systemanalyse, bei der verschiedene Anwendungen und mit verschiedenen Programmiersprachen entwickelte Werkzeuge auf verteilten Rechnern gekoppelt werden, ist in [30] dokumentiert. Jedoch ist der dafür benötigte Aufwand für die Zwecke der vorliegenden Arbeit nicht berechtigt. Deswegen wird die einfachere, erste Methode bevorzugt.

Bei der Verwendung einer kommerziellen Software muss sichergestellt werden, dass keine Abhängigkeiten bestehen, die den langfristigen Einsatz und die Portierung des Analyseprogramms gefährden. Am IRS wurde das CAD-Programm Pro/ENGINEER [81] verwendet, um Fahrzeugmodelle zu erstellen und zu visualisieren [84]. Nach einer Aktualisierung von Pro/ENGINEER wurde festgestellt, dass die Programmierschnittstelle Pro/DEVELOP [80] geändert wurde und einen neuen C-Compiler benötigte, der wiederum eine Aktualisierung des Betriebssystems erforderte. Vor dem Hintergrund dieser Erfahrung wurde in der vorliegenden Programmversion eine enge Verknüpfung mit externen Programmen vermieden. Darüber hinaus müssen bei Verknüpfung eines Analysewerkzeuges mit kommerziellen Anwendungen auch die Verfügbarkeit der Anwendung auf verschiedenen Plattformen, die Lizenzbedingungen und -kosten berücksichtigt werden.

Virtual Reality Modeling Language VRML ist ein ISO-Standard [118] und dient der Modellierung virtueller Welten mit einer Skriptsprache ähnlich wie Hypertext Markup Language HTML [123], womit Internetseiten erstellt werden. VRML-Modelle sind reine Textdateien und können mit einem kostenlosen Betrachter, meistens eine Browser-Erweiterung, angesehen werden. Im Internet wird eine große Auswahl an VRML-Betrachtern angeboten. Diese Modelle können mit beliebigen Texteditoren bearbeitet werden. Für die Erstellung komplexer Modelle werden auch spezielle VRML-Editoren und -Modelliersoftware angeboten. Die Interaktion wird vom Internet-Browser übernommen und kann durch im Modell in JavaScript implementierte Funktionen erweitert werden. JavaScript ist, einfach ausgedrückt, eine Skriptsprache zur Erweiterung von HTML um Objekte und Funktionen. Obwohl JavaScript und Java ursprünglich unabhängig von einander entwickelt wurden, werden sie durch die gemeinsamen Entwickler (Sun Microsystems und Netscape) immer stärker verknüpft. Eine ausführliche Beschreibung von JavaScript ist in [62] gegeben.

Die Austauschbarkeit von VRML-Modellen ist ein großer Vorteil. Der Austausch ist nicht nur über das Internet möglich sondern auch zwischen verschiedenen Grafikprogrammen. Viele CAD- und Modellierprogramme können VRML importieren bzw. exportieren. Es werden auch viele kommerzielle Konvertierungsprogramme angeboten, die Modelle von VRML in gängige CAD-Formate und zurück konvertieren können (z.B. für das in der Luft- und Raumfahrt verbreitet genutzte CAD-System CATIA). So können vorhandene CAD-Modelle in einer VRML-Animation benutzt werden. Konvertierung von Geometriedaten in VRML-Modelle mit einem einfachen Programm kann auch ohne großen Entwicklungsaufwand realisiert werden.

Da VRML-Dateien reine Textdateien sind, können sie vom Analysewerkzeug aus programmatisch erzeugt werden. Diese Vorgehensweise wurde gewählt, um aus den Simulationsergebnis-

sen automatisch eine dreidimensionale Fluganimation und zweidimensionale Plots zu erstellen. So stehen die grafisch aufbereiteten Ergebnisse während bzw. unmittelbar nach der Berechnung zur Verfügung.

Erfahrungsgemäß ist das Erlernen von VRML-Grundlagen einfach. Die Modifizierung vorhandener Modelle stellt auch für Anfänger, nach einer kurzen Einführung, kein Problem dar. Die komplette VRML-Spezifikation ist in [15] und [118] dokumentiert, und eine ausführliche Einsteiger-Dokumentation ist in [20] gegeben.

VRML ist eine verbreitete Technik in der immersiven Virtuellen Realität (VR). Die immersive Betrachtung bedeutet, dass der Betrachter in die Virtuelle Welt "eintauchen", sich in dieser Welt bewegen, die Modelle aus verschiedenen Blickwinkeln betrachten und mit virtuellen Objekten interagieren kann. Wie in vielen VR-Labors wird diese Technik auch am Höchstleistungsrechenzentrum Stuttgart [40] eingesetzt. Die am IRS erstellten VRML-Modelle können dort in einer vierwandigen VR-Umgebung CUBE [18], [88] ohne Anpassung immersiv betrachtet werden.

In der nachstehenden Tabelle 2.2 sind die Vor- und Nachteile der betrachteten Visualisierungsmethoden zusammengefasst. Mit (+) wird ein Vorteil, mit (-) ein Nachteil angezeigt. Der Eintrag (0) bedeutet kein entscheidender Vorteil oder Nachteil.

	Bibliothek	OpenGL	Anwendung	Java3D	VRML	PHIGS
Plattformunabhängigkeit	0	+	0	+	+	+
Standardisierung	-	+	-	+	+	+
Leistungsfähigkeit	+	+	0	+	+	+
Lernbarkeit	+	-	+	-	+	0
Entwicklungsaufwand	0	-	+	-	+	-
Schnittstellen	+	+	0	+	+	+
Interaktion	+	+	+	+	+	+
Austauschbarkeit	0	-	0	-	+	-
Verfügbarkeit	-	+	-	+	+	-

Tabelle 2.2: Vergleich der betrachteten Visualisierungsmethoden.

Aufgrund der Vorteile von VRML, die im Vergleich mit anderen Visualisierungsmethoden diskutiert wurden, wird in dieser Arbeit VRML als die Hauptvisualisierungsmethode gewählt. Jedoch wird auf professionelle, kommerzielle Grafikanwendungen nicht verzichtet. Der Datenaustausch wird, wie oben begründet, über Textdateien realisiert. Die Ausgabefunktionen wurden so erweitert, dass ein direktes Laden der Flugsimulationsergebnisse in kommerzielle Tabellenkalkulations- bzw. Plotprogramme möglich ist. Die neuen Visualisierungsfunktionen des RTSOPT sind in Kapitel 5 dokumentiert.

### 2.3 Die Benutzerschnittstelle

Die Schnittstellen des Analysewerkzeugs zu externen Grafikanwendungen und zur Visualisierung mit VRML wurden bereits diskutiert. In diesem Kapitel werden die grafische Benutzerschnittstelle (GUI, Graphical User Interface) und die Ein- und Ausgabeschnittstellen beschrieben. Wie bei der Auswahl der Programmiersprache und der Visualisierungsmethode werden zunächst die Anforderungen an die Benutzerschnittstelle aufgestellt und anschließend verschiedene Methoden für ihre Implementierung verglichen.

#### Aufgaben und Anforderungen

Die allgemeine Aufgabe der Benutzerschnittstelle ist es, die Benutzereingaben zu ermöglichen und die mit dem Programm produzierten Ergebnisse darzustellen. Im vorliegenden Fall besteht die Eingabe aus Umweltmodellen, Beschreibung einer Raumtransportmission mit Randbedingungen, Flugbeschränkungen, Flugsteuerung und dem Modell eines Trägersystems mit mehreren Stufen, Triebwerken und aerodynamischen Eigenschaften. Da RTSOPT ein Werkzeug ist, mit dem verschiedene Trägersysteme und Missionen untersucht werden, muss die Benutzerschnittstelle die Eingabe verschiedener Kombinationen von Rechenmodellen und eine variable Anzahl von Flugphasen ermöglichen.

Ein Ziel dieser Teilarbeit ist die Effektivität des gesamten Analysewerkzeugs auch durch die Verbesserung der Benutzerschnittstelle zu steigern. Dabei muss die Benutzerschnittstelle ebenfalls den in Kapitel 2.1 erläuterten softwaretechnischen Anforderungen genügen. Eine empfindliche Schwäche der bisherigen Benutzerschnittstelle war die Plattformabhängigkeit der grafischen Oberfläche. Sie basierte auf AIX Windows Interface Composer AIC [47] und war daher sowohl von dem Rechnertyp als auch von einem bestimmten Betriebssystem abhängig. Eine weitere Schwäche bestand darin, dass viele Benutzereingaben (z.B. Rechenmodelle, Stufentrennung, Flugsteuerung) nur durch Anpassungen und Implementierung neuer Modelle im Programmcode möglich waren. Ferner mussten neue Modelldaten und ihre Interpolation neu implementiert werden. Dadurch entstanden mehrere Programmversionen und die Vorbereitung einer Untersuchung war aufgrund der Einarbeitungszeit in den Programmcode und Implementierung neuer Modelle durch den Benutzer sehr aufwändig.

Vor diesem Erfahrungshintergrund wurde eine neue Softwarelösung entwickelt, um die Benutzerschnittstelle zu verbessern. Im folgenden Kapitel werden das Softwarekonzept und die Vorteile der verbesserten Benutzerschnittstelle von RTSOPT dokumentiert.

#### Die Implementierung der grafischen Benutzeroberfläche

Für die Implementierung der grafischen Benutzeroberfläche von RTSOPT standen Java, Forms Library, Visual C++ und Fortran mit einer kommerziellen GUI-Bibliothek zur Auswahl.

Die Programmbibliothek Forms [124] ermöglicht eine schnelle GUI-Entwicklung in C bzw. C++ auf der Basis von X, dem Standard Netzwerk-Grafiksystem unter UNIX und Linux. In [55] ist eine mit Forms entwickelte GUI für einen Gittergenerator für Strömungssimulation unter Linux und UNIX beschrieben. Jedoch wurde Forms aufgrund der Abhängigkeit von X, fehlender Herstellerunterstützung und der benötigten zusätzlichen direkten X-Programmierung nicht gewählt. Desgleichen wurde auch Visual C++ aufgrund der Plattformabhängigkeit (PC, Windows) nicht berücksichtigt. Die Lösung mit Fortran und einer kommerziellen GUI-Bibliothek wurde wegen den Kosten und der Abhängigkeit von bestimmten plattformspezifischen Lizenzbedingungen verworfen. Diese Nachteile gibt es bei der GUI-Entwicklung mit

Java Foundation Classes JFC nicht. Dies gehört zum Standardumfang der Programmiersprache Java und beinhaltet alle Funktionen und Elemente, die in modernen GUIs verwendet werden [115]. Die Plattformunabhängigkeit von Java basiert auf dem Konzept der Virtuellen Maschine (VM), das in Abbildung 2.9 betrachtet wird.

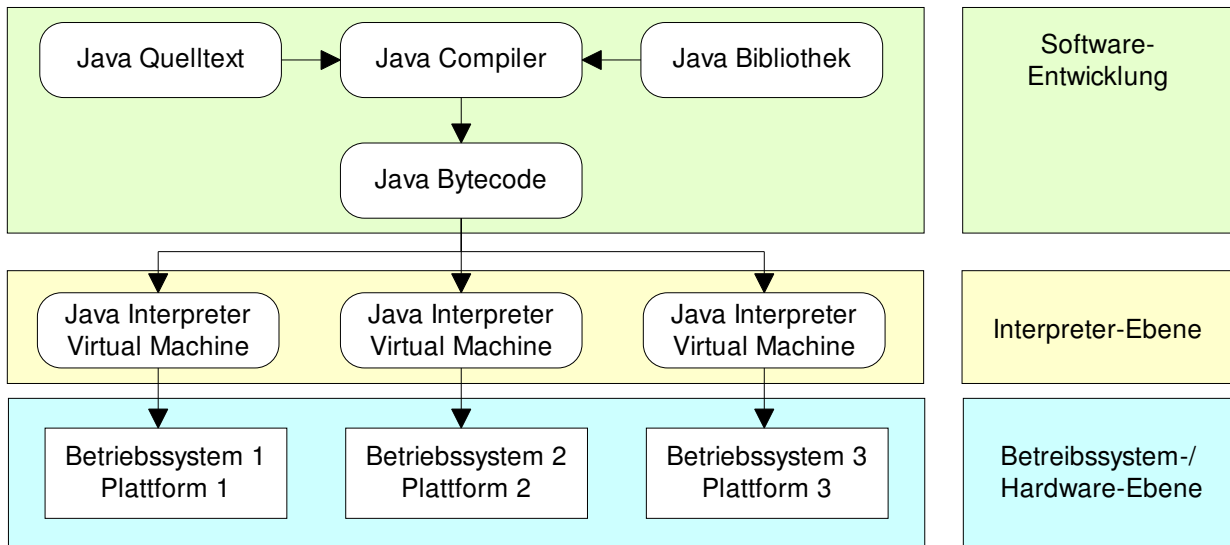


Abbildung 2.9: Schematische Darstellung der Java Virtuellen Maschine.

Die Virtuelle Maschine bezeichnet eine Softwareschicht zwischen Java Programmen und der Hardware. Sie stellt das Softwareabbild eines standardisierten, virtuellen Rechners dar. Die Schnittstelle zwischen der VM und dem kompilierten Java-Programm (Java Bytecode) ist in der Java-Spezifikation festgelegt, so dass alle Java-Programme auf allen virtuellen Maschinen laufen können. Die hardware-spezifische Schnittstelle andererseits wird von jedem Computerhersteller implementiert und zur Verfügung gestellt. Damit sind alle Java-Programme auf allen Plattformen lauffähig. Eine ausführliche Beschreibung sowohl dieses Konzepts als auch der Programmiersprache ist in [43] und [44] gegeben.

### Verbesserungen in der Benutzung

Vom Startdialog von RTSOPT können verschiedene elektronische Formulare gestartet werden, die zur Erstellung von Modell- und Eingabedateien verwendet werden.

Die Eingabefunktionen des Programmsystems RTSOPT wurden völlig überarbeitet und die bisherige programmgesteuerte Problemdefinition durch ein datengesteuertes Konzept ersetzt. Dadurch werden missionsspezifische Programmcode-Modifikationen vermieden. Flugsteuerungsalgorithmen oder Modelle, die zu kompliziert sind, um sie mit einer allgemeingültigen Methode darzustellen, müssen weiterhin im Code implementiert werden. Dies ist aber ohne die o.g. Nachteile auf eine objektorientierte Art weiterhin möglich.

Eine weitere Entwicklung der Eingabefunktionen stellen die Eingabeanalysefunktionen dar. Diese unterstützen unformatierte Eingabedateien (Textdateien) mit beliebig vielen und beliebig positionierten Leer- und Kommentarzeilen und analysieren die Eingabe, um den Benutzer mit präzisen Fehlermeldungen, die die Fehlerbeschreibung, den Dateinamen und die Zeilennummer beinhalten, zu unterstützen. Die Eingabeanalyse überprüft u.a. auch Fehler wie Diskrepanz

## 2 Softwaretechnische Überlegungen

---

zwischen Datei- und Programmversion, fehlende Anführungszeichen, fehlende Daten und physikalisch unsinnige Daten (z.B. negative Dichte).

Die Ausgabefunktionen wurden ebenfalls überarbeitet. Es wird z.B. eine Log-Datei angelegt, die genau dokumentiert, welche Schritte das Programm ausführt. Diese Datei beinhaltet auch Warnungen, Fehlermeldungen, Uhrzeit- und Datums-Angaben. Außerdem werden Modelldateien (z.B. eine optimierte Flugsteuerung) automatisch kommentiert, im Ausgabeverzeichnis gespeichert und können bei einem Neustart bzw. neuen Projekt als Eingabedatei wieder verwendet werden.

Beim Start des Programms lädt die Benutzerschnittstelle projektspezifische Einstellungen, Benutzeroptionen (Dezimalpunkt/-komma, Auswahl eines Zeichens für die Trennung von Datenspalten) und eine Aufgabenliste. Mit Hilfe der Aufgabenliste kann der Benutzer Funktionen wie Massenanalyse, Flugsimulation, Optimierung, Eingabekontrolle, usw. ein- oder ausschalten.

Da alle Ein- und Ausgabedateien Textdateien sind, konnte die Entwicklung der grafischen Benutzerschnittstelle (GUI) in Java unabhängig vom Analysewerkzeug durchgeführt werden. Die GUI dient der einfacheren Erstellung und Bearbeitung der Eingabedateien. Abbildung 2.10 zeigt beispielhaft ein elektronisches Formular von RTSOPT zur Erstellung von Flugphasenmodell-Dateien. In diesen Dateien werden die in dieser Flugphase geltenden Rechenmodelle, Rand- und Flugbeschränkungen und die Flugsteuerung definiert. Eine ausführlichere Beschreibung der Modelldateien ist in Kapitel 3.2 gegeben.

Angle of attack data: relative time [0,1],[deg]	
0.0	12.5
1.0	-6.0

Abbildung 2.10: Ausschnitt aus dem elektronischen Formular für ein Flugphasenmodell.

In der Abbildung sind Angaben über zwei Schubsteuermodelle zu sehen. Ferner sind für eine Anfangsschätzung einer Anstellwinkelsteuerung zwei Datenpunkte (am Anfang und am Ende der Flugphase) definiert. Für die Interpolation der Anstellwinkelsteuerung wurde lineare B-Spline Interpolation gewählt.

Die Eingabedateien sind übersichtlich angelegt, kommentiert und in der Benutzerdokumentation genau beschrieben. Daher können sie problemlos per Hand erstellt werden, und eine Abhängigkeit von der grafischen Benutzeroberfläche besteht nicht. Jedoch steigert diese Schnittstelle die Arbeitseffizienz durch Unterstützung des Benutzers mit Menüs, Optionsschaltern und Eingabefeldern, die fehlerhafte Eingaben verhindern. Bei der datenbankbasierten, statistischen Massenmodellierung, welche die Auswahl zahlreicher Massendaten von Trägersystemen und Subsystemen und die lineare bzw. logarithmische Regression dieser Daten zur Bestimmung von über 100 Koeffizienten, Konstanten und Exponenten erfordert, ist jedoch die grafische Unterstützung des Benutzers und die Automatisierung dieses Vorgangs sehr wichtig. Die folgende Abbildung 2.11 illustriert das grafische Modul zur Massenmodell-Erstellung während der Auswahl von Raketentriebwerken.

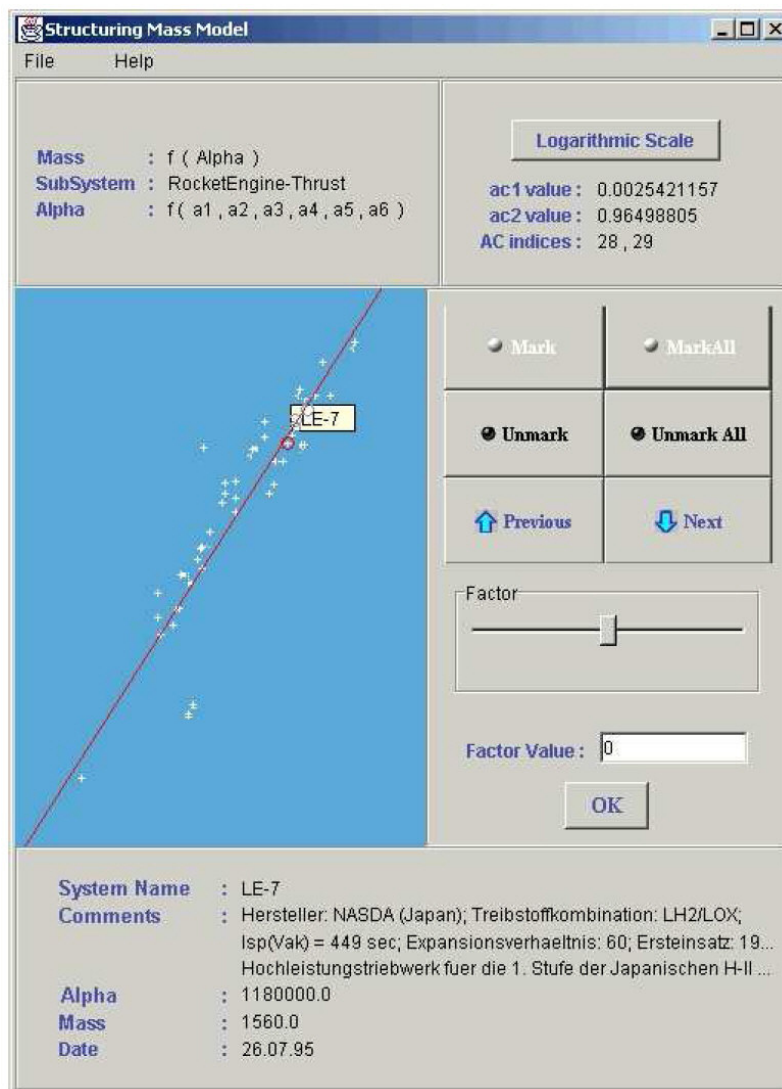
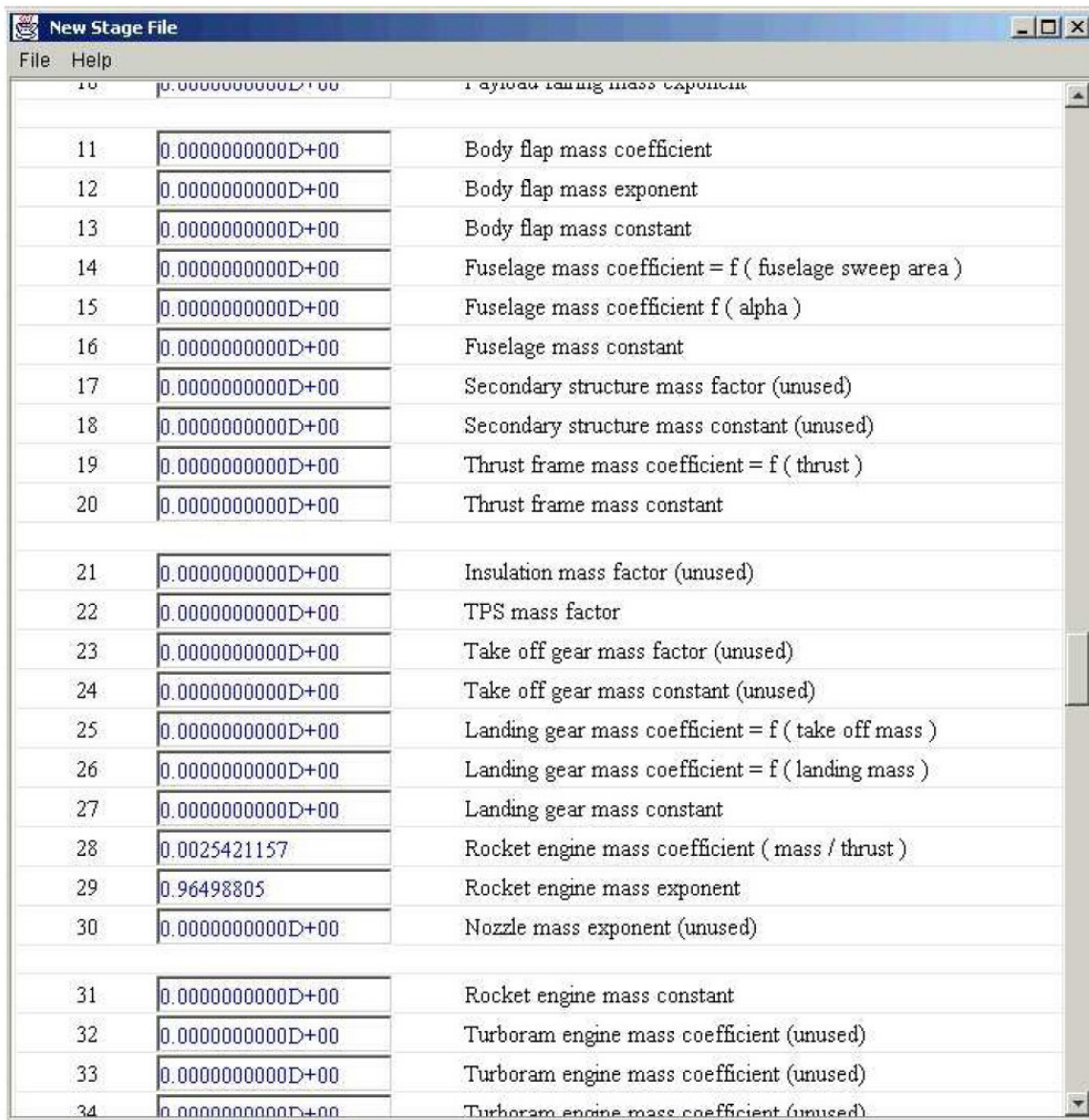


Abbildung 2.11: Die grafische Benutzeroberfläche für die Massenmodellierung.



## 2 Softwaretechnische Überlegungen

Nach dem Laden einer Datei aus der RTSOPT-Massendatenbasis (z.B. Triebwerksmasse in Abhängigkeit vom Schub) werden diese Daten im Grafikbereich links angezeigt. Aus diesen Daten werden diejenigen gewählt, die mit dem untersuchten System vergleichbar sind. Dazu wird ein Faktor gewählt, der den Grad der Technologie (optimistisch/pessimistisch;  $\pm 10\%$ ) definiert. Die lineare bzw. logarithmische Regressionsanalyse erfolgt nach jeder Änderung der Auswahl automatisch. Dabei werden Informationen über den gewählten Datensatz im unteren und das aktuelle Ergebnis im oberen Bereich angezeigt. Das Ergebnis ist die Regressionslinie. Die auf diese Weise bestimmten Korrelationskoeffizienten (Indizes 28, 29) werden automatisch in die Stufenmodelldatei übertragen, die in Form eines elektronischen Formulars dargestellt wird. Die nächste Abbildung 2.12 stellt eine solche Datei während der Erstellung dar.



Index	Value	Description
10	0.0000000000D+00	Body flap mass exponent
11	0.0000000000D+00	Body flap mass coefficient
12	0.0000000000D+00	Body flap mass exponent
13	0.0000000000D+00	Body flap mass constant
14	0.0000000000D+00	Fuselage mass coefficient = f ( fuselage sweep area )
15	0.0000000000D+00	Fuselage mass coefficient f ( alpha )
16	0.0000000000D+00	Fuselage mass constant
17	0.0000000000D+00	Secondary structure mass factor (unused)
18	0.0000000000D+00	Secondary structure mass constant (unused)
19	0.0000000000D+00	Thrust frame mass coefficient = f ( thrust )
20	0.0000000000D+00	Thrust frame mass constant
21	0.0000000000D+00	Insulation mass factor (unused)
22	0.0000000000D+00	TPS mass factor
23	0.0000000000D+00	Take off gear mass factor (unused)
24	0.0000000000D+00	Take off gear mass constant (unused)
25	0.0000000000D+00	Landing gear mass coefficient = f ( take off mass )
26	0.0000000000D+00	Landing gear mass coefficient = f ( landing mass )
27	0.0000000000D+00	Landing gear mass constant
28	0.0025421157	Rocket engine mass coefficient ( mass / thrust )
29	0.96498805	Rocket engine mass exponent
30	0.0000000000D+00	Nozzle mass exponent (unused)
31	0.0000000000D+00	Rocket engine mass constant
32	0.0000000000D+00	Turboram engine mass coefficient (unused)
33	0.0000000000D+00	Turboram engine mass coefficient (unused)
34	0.0000000000D+00	Turboram engine mass coefficient (unused)

Abbildung 2.12: Eine Stufenmodelldatei während der Erstellung.

Diese Stufenmodelldateien werden beim Start von RTSOPT eingelesen und beinhalten alle Daten, die für die Massenschätzung benötigt werden.

Mit Hilfe dieser grafischen Werkzeuge wird die Regressionsanalyse der zahlreichen Einträge in der Datenbasis und die Erstellung von Massenmodellen mit über 200 Parametern erleichtert und die Arbeitseffizienz gesteigert.

Eine Online-Benutzerdokumentation bzw. ein Hilfesystem ist mittlerweile Standard bei moderner Software. Die Online-Dokumentation von RTSOPT wurde wegen der Plattformunabhängigkeit in Hypertext Markup Language HTML geschrieben. Abbildung 2.13 zeigt beispielhaft zwei Seiten dieses Dokuments.

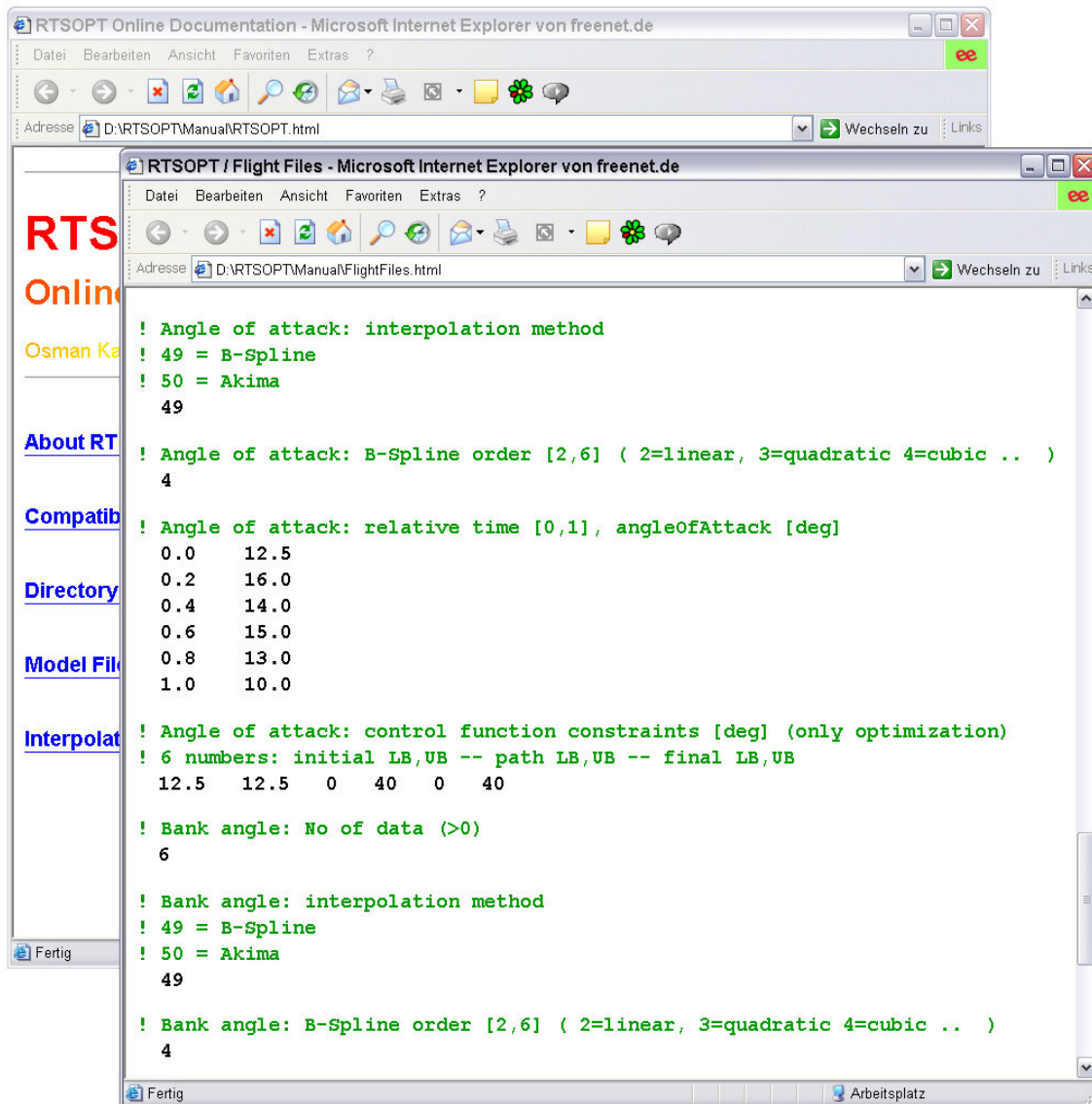


Abbildung 2.13: Die Online-Dokumentation von RTSOPT.

Das Fenster im Hintergrund stellt die Anfangsseite der Dokumentation und im vorderen Fenster ist eine Beispieldatei eines Flugsteuermodells zu sehen.

Die Verzeichnisstruktur der Ein- und Ausgabedateien und die Struktur, Aufbau, Inhalt und Syntax-Regeln der Eingabe- und Modelldateien sind im Benutzerhandbuch von RTSOPT ausführlich dokumentiert [56]. Die Online-Dokumentation dient als eine schnelle Referenz und

beinhaltet Beispiel-Eingabedateien wie z.B. in der Abbildung 2.13, die durch Kopieren, Einfügen und Überarbeiten bei der Erstellung neuer Eingabedateien benutzt werden können und somit eine weitere Steigerung der Effektivität der gesamten Programmnutzung ermöglicht.

### 2.4 Das Gesamtkonzept

Das Gesamtkonzept [60] der vorliegenden RTSOPT-Version sieht eine Trennung zwischen den Programmkomponenten Simulator, Rechenmodelle und Benutzeroberfläche vor und schreibt die Implementierung bestimmter Funktionalitäten der Modelle bzw. Schnittstellen zwischen ihnen vor. Tabelle 2.3 listet die wichtigsten Aufgaben dieser Programmkomponenten auf.

Die Benutzerschnittstelle	Stellt die Simulations- und Modelldaten grafisch dar Stellt Werkzeuge zur Verfügung, mit denen die Modelldaten eingegeben und manipuliert werden können
Der Simulator	Stellt eine zentrale Ausgabeinheit für die Simulator- und Modell-Meldungen zur Verfügung (Logger) Lädt die Simulationsmodelle Verwaltet die Simulationszeit und ruft die Modelle auf
Die Modelle	Stellen Initialisierungs- und Update-Methoden zur Verfügung, welche vom Simulator aufgerufen werden

Tabelle 2.3: Aufgaben der RTSOPT Programmkomponenten.

Diese Vorgehensweise hat den Vorteil, dass diese Komponenten unabhängig von einander entwickelt werden können und der Simulator ohne Programmcodeänderungen verschiedene Modelle laden und ausführen kann. Das RTSOPT-Softwarekonzept hat, trotz der unabhängigen Entwicklung, viele prinzipielle Ähnlichkeiten zu dem von der Europäischen Raumfahrtagentur ESA und den Kooperationspartnern entwickelten Simulation Model Portability (SMP) Standard [28]. Diese Normierung verfolgt die Ziele

- Plattformunabhängigkeit von Modellen
- Wiederverwendbarkeit von Modellen in verschiedenen Simulatoren und Projekten
- Reduzierung der Entwicklungskosten
- Verbesserung der Wartbarkeit

und definiert Richtlinien für die Kodierung von Modellen und Schnittstellen zum Simulator (SMI, Simulation Model Interface). SMP wurde in 2001 veröffentlicht und wird unter der Leitung der ESA durch die Organisation Configuration Control Board weiter entwickelt. In der Anwendung wird dieser Standard für zahlreiche operationelle Simulatoren für aktuelle Raumfahrtmissionen verwendet (z.B. für Rosetta, Mars Express, Venus Express, Radarsat-2, XMM Newton, Cryosat, Artemis, Meteosat, Eureka [116], [33]).

Die Parallele zwischen der neuen RTSOPT-Programmversion und den derzeitigen Simulator-Entwicklungen in Europa belegen die Aktualität und Eignung des in dieser Arbeit gewählten Softwarekonzeptes für ein plattformunabhängiges und flexibles Systemanalysewerkzeug.

In diesem Zusammenhang sei auch auf die Technologie SEDRIS für die Standardisierung und Austauschbarkeit von Umweltmodellen und –daten verwiesen [67]. SEDRIS war ursprünglich die Abkürzung für "Synthetic Environment Data Representation and Interchange Specification". Mittlerweile ist es ein Markenzeichen. SEDRIS verfolgt die folgenden Ziele:

- Bereitstellung einer Methodik für die Aufnahme aller Daten-Elemente und damit verbundener Verknüpfungsinformationen für die vollständige Repräsentation von Umwelt-Modelldaten,
- Bereitstellung eines normierten Austauschmechanismus für die Verteilung von Umweltdaten, Förderung der Datenbasis-Wiederverwendung zwischen heterogenen Systemen,
- Unterstützung einer ganzen Palette von Umweltmodell-Anwendungen, die die Ozeane, die Erdoberfläche, die Atmosphäre und den Weltraum betreffen.

SEDRIS basiert auf fünf Kern-Technologiekomponenten:

DRM	(Data Representation Model)	- Datenmodell & -struktur
EDCS	(Environmental Data Coding Specification)	- Kodierung der Daten
SRM	(Spatial Reference Model)	- Positions-Referenzsysteme
API	(Application Programming Interface)	- Programmierschnittstelle
STF	(SEDRIS Transmittal Format)	- Datenübertragungsformat

In den letzten Jahren wird diese offene technologische Infrastruktur in internationalen Standards verankert (z.B. ISO/IEC 18025:2005(E), ISO/IEC 18041-4:2005, weitere in Bearbeitung) und die Weiterentwicklung durch die Beteiligung der Industrie unterstützt.

## 3 Das Programmpaket RTSOPT

Das seit den 80'er Jahren am Institut für Raumfahrtssysteme der Universität Stuttgart entwickelte Programmpaket RTSOPT (Raumtransportsystem-Optimierung) wird für die Flug- und Systemoptimierung zur Leistungsanalyse unterschiedlicher zukünftiger Raumtransportsystem-Konzepte eingesetzt.

Wie vorstehend beschrieben, wird das Programmsystem nach modernen softwaretechnischen Gesichtspunkten grundlegend überarbeitet, wobei allerdings die Methodik und die mathematisch-physikalischen Modelle der Flugsimulation im Wesentlichen beibehalten werden. In diesem Kapitel kann deshalb unter Hinweis auf die Literatur [96], [41], [12], [84] auf eine nähere Begründung der Annahmen und auf umfängliche mathematische Ableitungen zu Gunsten einer für das Verständnis wichtigen allgemeinen Beschreibung und der Darstellung von Verbesserungsmaßnahmen verzichtet werden. In RTSOPT sind verschiedene Modelle mit unterschiedlichen Detailgenauigkeiten der Umwelt (Erdgestalt, Gravitationsfeld, Atmosphäre) implementiert, die wahlweise je nach Aufgabenstellung aufgerufen werden können. Auf diese Möglichkeiten wird hier ebenfalls nicht eingegangen, da die Leistungsverbesserungen in RTSOPT durch die Programmneufassung und die neue Optimierungsmethode im Vordergrund unseres Interesses stehen.

### 3.1 Die RTSOPT-Methodik

Die multidisziplinäre Analyse und Optimierung eines Raumtransportsystems erfordert die Definition der Transportmission und die Modellierung des Trägersystems, der Umwelt und der Systemdynamik. Die Transportmission beinhaltet die Festlegung der Anfangs- und Endbedingungen und der Flugbeschränkungen wie z.B. maximale Fluglasten. Das Modell des Trägersystems besteht aus dem parametrisierten dreidimensionalen CAD-Konzeptmodell und den Rechenmodellen für die Fahrzeugmasse, die aerodynamischen Eigenschaften des Fahrzeugs und Leistungen der Triebwerke. Da die Systemoptimierung durch die Skalierung geometrischer Parameter der Stufen und Entwurfparameter der Subsysteme erfolgt, basiert das Massenmodell auf diesen Geometrie- und Entwurfsdaten. Schließlich ermöglichen die Umweltmodelle die Berechnung der atmosphärischen Parameter wie z.B. der Luftdichte, Fahrzeugposition und -geschwindigkeit relativ zur Erde und Bewegungsgleichungen zur Berechnung der Systemdynamik während der Flugsimulation.

Um diesen Aufgaben gerecht zu werden, wird in RTSOPT eine sequenzielle, iterative Methodik eingesetzt, die aus einem interaktiven Entwurfszyklus und einem automatischen Optimierungszyklus besteht. Abbildung 3.1 stellt diese Vorgehensweise schematisch dar. Die Datenbasis enthält zahlreiche Fahrzeug-, Triebwerks-, Umweltmodelle und Missionsdefinitionen, die der Benutzer in verschiedenen Untersuchungen verwenden kann. Im interaktiven Zyklus können verschiedene Programme eingesetzt werden, um neue Modelle zu erstellen. Dazu gehören die grafische Benutzeroberfläche des RTSOPT, Programme für die Aerodynamik- und Aerothermodynamik-Analyse und eine Umgebung für die Fahrzeugmodellierung. Diese Software-Werkzeuge werden sowohl für die Erstellung neuer Rechenmodelle als auch für die Auswertung und Überprüfung der Optimierungsergebnisse benutzt.

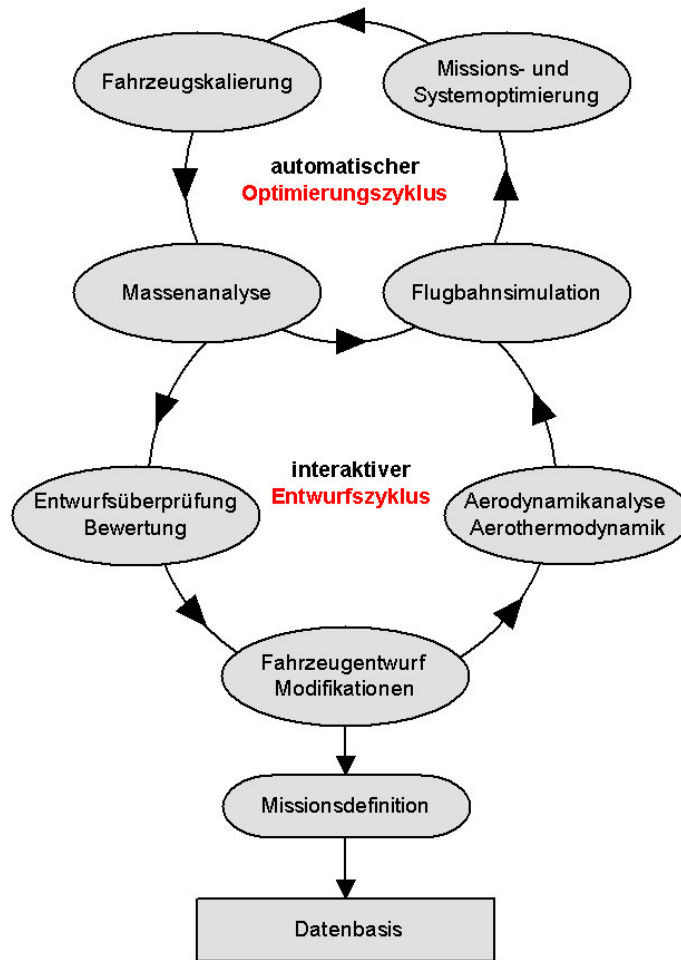


Abbildung 3.1: Schematische Darstellung der RTSOPT-Methodik.

Im automatischen Optimierungszyklus wird ein komplexes, beschränktes, nichtlineares Parameter-Optimierungsproblem mit einem sehr leistungsfähigen und fortschrittlichen Optimierungsverfahren numerisch gelöst. Das Optimierungsmodul des RTSOPT beinhaltet die Programm-Module Flugbahnsimulation, Missions- und Systemoptimierung, Fahrzeugskalierung und Massenanalyse. Im Flugsimulations-Modul werden die vom Benutzer zur Verfügung gestellten Rechenmodelle und die Bewegungsgleichungen ausgewertet. Das Modul Fahrzeugskalierung beinhaltet die Berechnung der Längen, Flächen und Volumina der Subsysteme (Rumpf, aerodynamische Flächen, Tanks) entsprechend den vom Optimierungsverfahren bestimmten Systemparametern (z.B. Aufstiegs-Treibstoffmasse). Das Massenanalyse-Modul beinhaltet die Berechnung der Subsystemmassen und Stufenmassen in Abhängigkeit von den geometrischen Abmessungen und Systemparametern der Subsysteme.

Die Lösung dieses Optimierungsproblems beinhaltet den optimalen zeitlichen Ablauf der Flugsteuerung und der optimalen Systementwurfparameter bzw. geometrischen Skalierungsfaktoren. Sie erfüllt alle Flugbeschränkungen und die Anfangs- und Endbedingungen der Transportaufgabe.

### 3.2 Die Rechenmodelle

Für die Modellierung der Triebwerksleistungen, der Fahrzeugmasse in Abhängigkeit von den Geometrieparametern, der Atmosphäre, Planeteneigenschaften und der Flugsteuerung werden zum einen Rechenmodelle, die auf Interpolation von Datentabellen basieren, zum anderen Rechenmodelle, die auf physikalischen Beziehungen basieren, eingesetzt. Da hier primär der Leistungsvergleich der Optimierungslöser interessiert, werden einfache Modelle der Umwelt verwendet und die optionalen, detaillierteren Varianten nicht aktiviert.

#### Die Bewegungsgleichungen:

Für die Flugsimulation werden Bewegungsgleichungen nach [98] und [100] im rotierenden Relativsystem bei Beschränkung auf die translatorischen Freiheitsgrade in der Form verwendet [84]:

$$\dot{v} = \frac{F_v}{m} + r \cdot \omega_E^2 \cdot \cos^2 \delta \cdot (\sin \gamma - \cos \chi \cdot \tan \delta \cdot \cos \gamma) \quad (3.1)$$

$$\begin{aligned} \dot{\gamma} = & \frac{F_\gamma}{mv} + \frac{r}{v} \cdot \omega_E^2 \cdot \cos^2 \delta \cdot (\cos \gamma + \cos \chi \cdot \tan \delta \cdot \sin \gamma) \\ & + 2 \cdot \omega_E \cdot \cos \delta \cdot \sin \chi + \frac{v}{r} \cdot \cos \gamma \end{aligned} \quad (3.2)$$

$$\begin{aligned} \dot{\chi} = & \frac{F_\chi}{m \cdot v \cdot \cos \gamma} + \frac{r \cdot \omega_E^2 \cdot \cos \delta}{v \cdot \cos \gamma} \sin \chi \cdot \sin \delta \\ & - 2 \cdot \omega_E \cdot \cos \delta \cdot (\cos \chi \cdot \tan \gamma - \tan \delta) \\ & + \frac{v}{r} \cdot \sin \chi \cdot \cos \gamma \cdot \tan \delta \end{aligned} \quad (3.3)$$

$$\dot{\lambda} = \frac{v \cdot \cos \gamma \cdot \sin \chi}{r \cdot \cos \delta} \quad (3.4)$$

$$\dot{\delta} = \frac{v}{r} \cdot \cos \gamma \cdot \cos \chi \quad (3.5)$$

$$\dot{r} = v \cdot \sin \gamma \quad (3.6)$$

$$\dot{m} = -\dot{m}_{Tr} \quad (3.7)$$

wobei  $r$  die Entfernung zum Erdmittelpunkt,  $\lambda$  und  $\delta$  die geographische Länge und Breite,  $\gamma$  und  $\chi$  die Bahnneigung und -azimut,  $v$  die Fluggeschwindigkeit im geozentrischen Horizontalsystem,  $\dot{m}_{Tr}$  den Treibstoffdurchsatz bzw.  $\dot{m}$  die Massenänderungsrate des Fahrzeuges und  $\omega_E$  die Winkelgeschwindigkeit der Erde bezeichnen.

Die auf den Schwerpunkt des Fahrzeuges einwirkenden Gravitations-, Schub- und Luftkräfte werden im bahntangentialen Koordinatensystem aufgestellt, in dem die Lage des Fahrzeuges durch den Anstellwinkel  $\alpha$  und den Flugwindhängewinkel  $\mu_a$  definiert wird [84]:

$$F_v = -g \cdot m \cdot \sin \gamma + S_R \cdot \cos(\alpha + \varepsilon_R) + S_L \cdot \cos(\alpha + \varepsilon_L) - W \quad (3.8)$$

$$F_\gamma = -g \cdot m \cdot \cos \gamma + (S_R \cdot \sin(\alpha + \varepsilon_R) + S_L \cdot \sin(\alpha + \varepsilon_L)) \cdot \cos \mu_a + A \cdot \cos \mu_a \quad (3.9)$$

$$F_\chi = -(S_R \cdot \sin(\alpha + \varepsilon_R) + S_L \cdot \sin(\alpha + \varepsilon_L)) \cdot \sin \mu_a + A \cdot \sin \mu_a \quad (3.10)$$

Hier sind  $S$ ,  $W$  und  $A$  die Schub-, Widerstands- und Auftriebskraft,  $m$  die Fahrzeugmasse,  $g$  die Gravitationsbeschleunigung und  $\varepsilon$  der Schubwinkel gegenüber der Fahrzeuglängsachse (Nullauftriebsrichtung). Die Indizes  $R$  bzw.  $L$  beziehen sich auf einen Raketen- bzw. Luftatmerantrieb. Der resultierende Kraftvektor setzt sich aus den drei Komponenten  $F_v$ ,  $F_\gamma$  und  $F_\chi$  zusammen. Es werden das einfache Newtonsche Gravitationsfeld nach (3.11) und schiebefreie Flugzustände angenommen. Von den in RTSOPT implementierten Modellen für die abgeplattete Erde wird in der vorliegenden Arbeit kein Gebrauch gemacht.

### Das Gravitationsfeld

Die räumliche Bewegung eines Flugkörpers wird maßgeblich durch Gravitationskräfte beeinflusst. In dieser Arbeit wird ein kugelförmiger, rotierender Zentralkörper (Planetenmodell) angenommen, für welchen die Gravitationsbeschleunigung nur eine Funktion der Höhe  $h$  ist:

$$g(h) = g_0 \cdot \left( \frac{R_E}{R_E + h} \right)^2 \quad (3.11)$$

Die Planetenmodell-Datei der Erde beinhaltet die Winkelgeschwindigkeit  $\omega_E$ , den mittleren Radius  $R_E$  und die Erdbeschleunigung  $g_0$ . Durch Austauschen dieser Datei wird die Berechnung von Missionen um andere Himmelskörper ohne Änderungen im Programmcode ermöglicht. Modelle verschiedener Planeten werden in der Modelldatenbank von RTSOPT zur Verfügung gestellt. Darüber hinaus können bei der Untersuchung interplanetarer Missionen verschiedenen Flugphasen jeweils andere Planetenmodelle zugeordnet werden.

### Das Atmosphärenmodell

Für die Flugsimulation (Gleichungen (3.12) bis (3.14)) werden Atmosphärenmodelle benötigt, die die Luftdichte, den Umgebungsdruck, die Temperatur und die lokale Schallgeschwindigkeit in Abhängigkeit von der Höhe und Position darstellen. Ein solches Modell beschreibt eine relativ zur Erde ruhende Lufthülle und stellt die o.g. Atmosphärenparameter, basierend z.B. auf der US Standardatmosphäre [74] bzw. dem Modell Mass Spectrometer Incoherent Scatter Extended MSISE [16] in Form von kontinuierlichen Funktionen oder Datentabellen dar.

### Aerodynamik und Aerothermodynamik

Das Aero- und Aerothermodynamik-Modell hat die Aufgabe, für verschiedene Anströmbedingungen in den Geschwindigkeitsbereichen Unter-, Trans-, Über- und Hyperschall die aerodynamischen Beiwerte zu liefern. Für die Bestimmung der Luftkräfte muss das Modell Änderungen in den aerodynamischen Beiwerten in Abhängigkeit vom Triebwerksstrahl und von der Systemkonfiguration (Stufenzahl), Flugmachzahl und Luftdichte berücksichtigen. Bei mehrstufigen Trägersystemen müssen außerdem unterschiedliche Aerodynamikmodelle für die einzelnen Stufen und Stufenkombinationen zur Verfügung stehen. Daher stellt die Modellierung der Aerodynamik einen sehr aufwändigen Teil der multidisziplinären Systemuntersuchungen dar.



In der Flugsimulation werden die Luftkräfte über die folgende Beziehung berechnet:

$$L = \frac{1}{2} \rho(h) v^2 \cdot c_L(\alpha, \beta, Ma, f) \cdot A_{Ref} \quad (3.12)$$

wo  $\rho(h)$  die Luftdichte in der Flughöhe  $h$ ,  $v$  die Fluggeschwindigkeit,  $c_L$  den jeweiligen Luftkraftbeiwert (Auftrieb, Widerstand, Querkraft),  $\alpha$  den Anstellwinkel,  $\beta$  den Schiebewinkel,  $Ma$  die Machzahl,  $f$  eine Korrekturfunktion und  $A_{Ref}$  die aerodynamische Referenzfläche bezeichnen. Die Kraftbeiwerte im schiefbefreien Flug sind meistens eine Funktion des Anstellwinkels und der Flug-Machzahl.

Die Erstellung von Aerodynamik- und Aerothermodynamik-Modellen mit Hilfe von komplizierten Strömungssimulationsverfahren und Windkanalmessungen sind aufgrund des zeitlichen und finanziellen Aufwandes bei Konzeptuntersuchungen im frühen Projektstadium nicht sinnvoll. Daher werden beim Vorentwurf, wie in [41] beschrieben, einfache bzw. semi-empirische aerodynamische Entwurfsverfahren eingesetzt, die mit fortschreitendem Projektstand durch genauere ersetzt werden. In vielen Fällen kann auf die zahlreichen Aerodynamik-Modelle genauer untersuchter bzw. ausgeführter Systeme zurückgegriffen werden. Solche Modelle werden in der RTSOPT-Datenbasis zur Verfügung gestellt.

Die in RTSOPT eingesetzten Aerodynamik-Modelle beinhalten Datentabellen der aerodynamischen Beiwerte in Abhängigkeit von der Machzahl, dem Anstellwinkel und des Schiebewinkels und Angaben über die Interpolationsmethode und Ordnung der interpolierenden Kurve. In der vorliegenden Programmversion stehen die Interpolationsmethoden Akima (kubisch) und B-Spline (Ordnung zwei bzw. linear bis sechs) zur Verfügung. Mathematische Beschreibungen dieser Interpolationsmethoden sind in [1] und [39], benutzerorientierte und vergleichende Beispiele sind in [56] gegeben. Diese Interpolationsmethoden stehen für alle Rechenmodelle bereit, die auf mehrdimensionaler Interpolation von Datentabellen basieren.

Trägersysteme und Wiedereintrittsfahrzeuge, die mit hypersonischen Geschwindigkeiten fliegen, müssen sehr hohen Wärmelasten standhalten. In RTSOPT wird für die Bahnoptimierungsaufgabe eine semi-empirische Methode zur Bestimmung des Wärmestroms  $\dot{Q}$  am Stau punkt eingesetzt. Die Fahrzeugnase wird näherungsweise durch eine Kugel modelliert, für die eine semi-empirische Lösung existiert:

$$\dot{Q} = c \cdot \sqrt{\frac{R_{Ref}}{R_N} \cdot \frac{\rho(h)}{\rho(0)}} \cdot \left( \frac{v}{v_{Ref}} \right)^n \quad (3.13)$$

wo  $R_N$  den Fahrzeug-,  $R_{Ref}$  einen Referenz-Nasenradius,  $\rho(h)$  und  $\rho(0)$  die Luftdichte in der Flughöhe und auf Meereshöhe,  $v$  die Fluggeschwindigkeit,  $v_{Ref}$  eine Referenzgeschwindigkeit,  $c$  und  $n$  die Korrelationskonstanten bezeichnen. Die Größen  $R_{Ref}$ ,  $v_{Ref}$ ,  $c$  und  $n$  sind modellspezifisch.

Die freie molekulare Aufheizung im Übergangsbereich zum Weltraum ist ein wichtiger Parameter bei der Festlegung des Zeitpunktes für den Abwurf der Nutzlastverkleidung und das Aussetzen einer Nutzlast. Zum Schutz der Nutzlast wird diese Aufheizung beschränkt. Die freie molekulare Aufheizrate wird wie folgt berechnet:

$$\dot{Q}_{FMHR} = \frac{1}{2} \cdot \rho \cdot v^3 \quad (3.14)$$

## Die Triebwerke

Triebwerksmodelle werden eingesetzt, um während der Flugsimulation den Treibstoffdurchsatz, den Schub und die Triebwerksverluste (Impuls-, Höhen-, Drossel- und Schubvektorverluste, usw.) in Abhängigkeit vom Flugzustand und der Flugsteuerung zu berechnen. Bei näher untersuchten Triebwerken, insbesondere bei luftatmenden, basieren diese Modelle auf Triebwerkskennfelder, die in Form von Datentabellen zur Verfügung stehen und mehrdimensional interpoliert werden. Bei Triebwerken, für die keine detaillierten Daten vorliegen, werden einfache physikalische Beziehungen eingesetzt. Gleichungen (3.15) bis (3.17) zeigen beispielhaft die Berechnung der Triebwerksleistungen für ein Raketentriebwerk:

$$S_R = \dot{m}_{Tr} \cdot g_0 \cdot I_{sp}(h, \eta) \quad (3.15)$$

$$I_{sp}(h, \eta) = \left[ I_{sp,v} - \frac{P(h)}{P(0)} (I_{sp,v} - I_{sp,0}) \right] \cdot f(\eta_R) \quad (3.16)$$

$$f(\eta_R) = -0,010 \cdot \eta_R^2 + 0,028 \cdot \eta_R + 0,982 \quad (3.17)$$

wobei  $S_R$  der Raketenschub,  $\dot{m}_{Tr}$  der Treibstoffdurchsatz,  $g_0$  die Erdbeschleunigung,  $I_{sp}(h, \eta)$  der spezifische Impuls,  $I_{sp,v}$  und  $I_{sp,0}$  die spezifischen Vakuum- bzw. Bodenimpulse,  $P$  der Umgebungsdruck in Abhängigkeit von der Flughöhe  $h$ ,  $\eta_R$  der Drosselgrad und  $f(\eta_R)$  eine Drosselverlustfunktion sind. Die Berücksichtigung der Drosselverluste ist optional und kann deaktiviert werden.

In der RTSOPT-Modellbibliothek stehen zudem eine ganze Palette von weiteren Triebwerksmodellen für verschiedene Triebwerkstypen zur Verfügung (Flüssig- und Feststoffrakete, Staustrahltriebwerk, Luftverflüssigungsanlage, usw.). Diese Modelle basieren teilweise auf wenigen Triebwerksparametern wie z.B. spezifischem Impuls, nominellem Treibstoffdurchsatz, bei Luftatmern typischerweise auf Triebwerkskennfeldern in Abhängigkeit von Flugmachzahl, Staudruck, Anstellwinkel und bei Feststoffboostern auf Schub- und Massendurchsatzdaten als Funktion der Brennzeit.

## Die Flugphasen

Um Unstetigkeiten in einer Raumtransportmission wie z.B. Stufentrennung, Abschaltung von Triebwerken, Eintritt in die Atmosphäre, usw. zu berücksichtigen, wird die Mission in mehrere Flugphasen aufgeteilt. Die Transportmission eines ballistischen Trägersystems besteht beispielsweise aus dem Senkrechtstart, einem Nickmanöver, einem anschließenden ballistischen Aufstiegsflug und weiteren gesteuerten Aufstieg nach der Trennung der Feststoffbooster. In der ersten Flugphase sind alle Stufen, nach der Trennung der Feststoffbooster nur noch die übrigen Stufen aktiv, womit sich sowohl die Systemmasse als auch die aerodynamischen Eigenschaften des Fahrzeugs ändern. Während in der ersten Flugphase alle Triebwerke brennen, werden nach der Booster-Trennung nur noch die Triebwerke der Zentralstufe betrieben. Eine weitere Änderung ergibt sich nach dem Aussetzen der Oberstufe, die in einer Höhe gestartet wird, in der die atmosphärischen Effekte bzw. Luftkräfte vernachlässigbar sind. Des Weiteren gibt es signifikante Änderungen in der Flugsteuerung in jeder Flugphase. Während beim Aufstiegsflug eines geflügelten Systems beispielsweise die Anstellwinkelsteuerung und Schubdrosselung sehr wichtig sind, machen sie bei einer ballistischen Flugphase mit abgeschalteten Triebwerken und einem konstanten Anstellwinkel von Null keinen Sinn. Auch die zeitlichen Verläufe der Flugsteuergrößen (z.B. Drosselgrad) können an Flugphasenübergängen Sprünge aufweisen.

Um die Anfangs- und Endbedingungen und die Schnittstellen der Flugphasen sowie jeweils die unterschiedlichen Rechenmodelle für die verschiedenen Flugphasen zu definieren, wurde in der vorliegenden Version des RTSOPT der Begriff Flugphasenmodell eingeführt. Dieses definiert neben den Anfangs- und Endbedingungen der Flugphasen die Fluglastbeschränkungen, die in der Flugphase zu benutzenden Stufen-, Aerodynamik-, Triebwerks- und Planetenmodelle und die Flugsteuerung. Tabelle 3.1 zeigt die in der vorliegenden Programmversion implementierten Flugbeschränkungen:

	Beschränkung
Pfadbeschränkungen	Dynamischer Druck ( $\text{N/m}^2$ )
	Spezifischer Wärmefluss ( $\text{W/m}^2$ )
	Flügelastfaktor (Vielfaches der Erdbeschleunigung)
	Schublastfaktor (Vielfaches der Erdbeschleunigung)
	Spezifische freie molekulare Aufheizrate ( $\text{W/m}^2$ )
	Auftriebskraft (N)
Endbeschränkungen	Bahninklination ( $^\circ$ )
	Dauer der Flugphase (s)
	Machzahl
	Entfernung zum Zielpunkt (m)
	Entfernung zum Ziel-Zylinder (m)
	Steigrate (m/s)

Tabelle 3.1: Rand- und Pfadbeschränkungen der Flugphasen.

Anfangsbedingungen einer Flugphase können zum einen durch die Verknüpfung mit der vorherigen Phase, zum anderen durch die Eingabe des Anfangszustands definiert werden. Neben den in der Tabelle 3.1 aufgeführten Beschränkungen, die durch die Eingabe von unteren und oberen Grenzwerten definiert werden, ermöglicht RTSOPT die Einschränkung aller in der folgenden Tabelle 3.2 aufgeführten Beschreibungsgrößen und Steuerfunktionen. Hierfür werden ebenfalls untere und obere Grenzwerte am Anfang, während und am Ende der Flugphase definiert.

Flugzustände		Steuerfunktionen
Flugzeit (s)	Höhe (m)	Anstellwinkel ( $^\circ$ )
Geschwindigkeit (m/s)	Bahnneigung ( $^\circ$ )	Flugwindhängewinkel ( $^\circ$ )
Bodenstrecke (m)	Azimut ( $^\circ$ )	Schubdrosselung (%)
Geografische Länge ( $^\circ$ )	Integrale Wärmelast ( $\text{J/m}^2$ )	Schubwinkel ( $^\circ$ )
Geografische Breite ( $^\circ$ )	Fahrzeugmasse (kg)	

Tabelle 3.2: Zustandsgrößen und Steuerfunktionen der Simulation.

Falls Randbedingungen einer Flugphase nicht vorgeschrieben werden, sind diese für die Optimierung freigegeben. Sind zum Beispiel die Endhöhe der Freiflugphase eines Trägersaufstiegs zwischen 150 und 200 km begrenzt und die Endbahnneigung  $0^\circ$  vorgegeben, wird die Scheitelhöhe der Bahn optimiert.

### Die Flugsteuerung

Die Flugsteuerung erfolgt über die zeitliche Regelung der in der Tabelle 3.2 aufgeführten Steuergrößen. Zur Modellierung der kontinuierlichen zeitlichen Verläufe der Steuerfunktionen werden diese parametrisiert und die für diskrete Flugzeitpunkte definierten Steuerfunktionswerte in Abhängigkeit von der Flugzeit interpoliert. Abbildung 3.2 veranschaulicht das Modell einer Flugsteuerung.

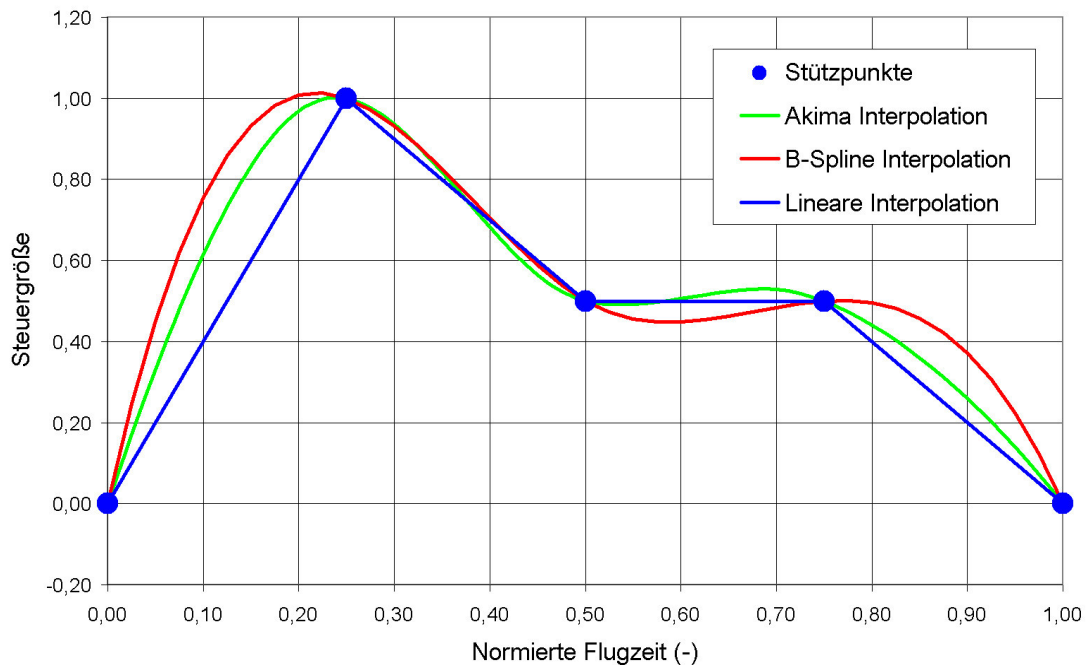


Abbildung 3.2: Stützstellen und interpolierte Darstellungen einer Steuerfunktion.

Während die Markierungen die Stützstellen (Eingabedaten/Gitterpunkte) darstellen, zeigen die Interpolationskurven den kontinuierlichen Verlauf der entsprechenden Steuerfunktion. RTSOPT stellt für die Eingabe der Anfangsschätzung die bereits erwähnten Interpolationsmethoden Akima und B-Spline (Ordnung zwei bis sechs) zur Verfügung. Die Ergebnisse der Optimierungsberechnungen werden immer in kubischer B-Spline-Darstellung ausgegeben. Da vor der Optimierung einer Flugbahn die Dauer der einzelnen Flugphasen nicht bekannt sind, wird in jeder Flugphase eine normierte Zeit eingesetzt, die folgenderweise definiert ist:

$$t_n = \frac{t - t_0}{t_1 - t_0} \quad (3.18)$$

wobei  $0 \leq t_n \leq 1$  die normierte Phasen-Flugzeit,  $t$  die absolute Flugzeit in Sekunden und  $t_0$  bzw.  $t_1$  die absolute Anfangs- und Endzeit einer Flugphase bezeichnen.

#### Das Fahrzeugmodell

Ein Stufenmodell besteht zum einen aus dem Massenmodell, das entweder die Start- und die Leermasse definiert oder die Massen der Subsysteme in Abhängigkeit von den Entwurfsparametern (Dimensionen, maximale Lastfaktoren) darstellt. Weiterhin sind Angaben über die Triebwerksmodelle und deren Anzahl notwendig. Ferner ist das konzeptionelle, dreidimensionale CAD-Modell Bestandteil eines Stufenmodells. Triebwerksmodelle sind externe Rechenmodelle, zu denen Verweise im Stufenmodell angegeben werden müssen. CAD-Modelle sind ebenfalls externe Modelle. Das Stufenmassenmodell ist direkt im Stufenmodell enthalten. Im Folgenden sind die o.g. Modelle erläutert.

#### Das Massenmodell

Die numerische Optimierung des Raumtransporter-Systementwurfs erfolgt durch die Variation und Optimierung von Entwurfsparametern, wie die projizierte Flügelfläche, maximale Lastvielfachen, geometrische Skalierung der Stufen, Tanks, usw. Daher muss ein Massenmodell benutzt werden, das die Systemmasse in Abhängigkeit von solchen Parametern darstellt. In RTSOPT wird ein solches Massenmodell eingesetzt, das auf Subsystem - Massendaten ausgeführt und näher untersuchter Luft- und Raumfahrzeuge basiert.

Die Gesamtsystemmasse ergibt sich aus der Summe von  $N$  Subsystemmassen  $m_i$  in der folgenden Form:

$$m = \sum_{i=1}^N a_i \cdot x_i^{b_i} \quad (3.19)$$

wobei  $i$  den Subsystemindex,  $a_i$  und  $b_i$  die empirischen Massen-Korrelationskoeffizienten und  $x_i$  eine Funktion der Subsystem-Entwurfsparameter bezeichnen. Diese Funktion ist beispielsweise für die Flügelmasse wie folgt definiert:

$$x = \frac{m_0 \cdot n_F \cdot S_S \cdot A_{F.Prj}}{t_w} \quad (3.20)$$

wobei  $m_0$  die Fahrzeugstartmasse,  $n_F$  der maximale Flügellastfaktor,  $S_S$  die strukturelle Flügelspannweite,  $A_{F.Prj}$  die projizierte Flügelfläche und  $t_w$  die Flügelwurzelstärke sind.

Die Massenkorrelationsfaktoren werden durch Regressionsanalyse ausgesuchter Daten berechnet. Die RTSOPT-Datenbasis stellt Massendaten für über 20 Subsysteme zur Verfügung. Sowohl die Massenkorrelationsfunktionen als auch die Datenbasis werden am IRS seit Jahren konsequent weiter entwickelt und erweitert ([90], [99], [84], [102]). In [102] wird eine Übersicht über Datenquellen für Massendaten gegeben. Als ein Standardwerk sei hier [51] erwähnt. Massenmodelle, die auf Finite-Elemente-Analysen und anderen detaillierten Methoden basieren, eignen sich wegen des hohen Aufwandes für die Systemoptimierung in der konzeptionellen Phase nicht. Detaillierte parametrische Massenansätze wie in [42] sind dagegen in ihrem Gültigkeitsbereich stark eingeschränkt.

Für die Massenmodellierung eines Fahrzeuges müssen aus der Datenbasis die Daten der Fahrzeuge gewählt werden, die eine ähnliche Konfiguration haben und vergleichbare Technologien repräsentieren. Daher stellt diese Auswahl einen wichtigen Arbeitsschritt dar, der die Definition eines technologischen Standes bedeutet. Das in der vorliegenden Version von RTSOPT in-

tegrierte grafische Werkzeug, das den Benutzer bei der Ermittlung der Korrelationsfaktoren unterstützt, ist in Kapitel 2.3 beschrieben.

Eine Übersicht des in RTSOPT eingesetzten Rechenverfahrens und die Benutzeranleitung für die Massenmodellierung sind in [56] und [57] gegeben. Die Aktualisierung der Massendatenbank, Datenquellen und genauen Spezifikationen der Datenbankkomponenten und geometrischen Eingabedaten sind in [102] dokumentiert.

### Der konzeptionelle Fahrzeugentwurf

Ein konzeptionelles, dreidimensionales CAD-Modell des untersuchten Raumtransportsystems ist für die Überprüfung der numerisch optimierten Auslegung unverzichtbar. Bei der simultanen Flug- und Systemoptimierung werden die Flugleistungen und Entwurfparameter, wie die projizierte Flügelfläche, Skalierungsfaktoren der Stufen und maximale Lastfaktoren optimiert. Durch die Leistungsoptimierung können der Treibstoffbedarf reduziert und die Dimensionen der Treibstofftanks dem reduzierten Bedarf automatisch angepasst werden. Da die numerische Optimierung Geometriefehler, wie das Durchdringen zweier Tanks ineinander, nicht berücksichtigt, muss nach dem automatischen Optimierungsschritt im interaktiven Entwurfszyklus die neue Auslegung im CAD-System überprüft bzw. angepasst werden.

In der neuen RTSOPT-Version wird durch die Wahl der VRML-Technologie für die Fahrzeugdarstellung erreicht, dass ein beliebiges CAD-System genutzt werden kann. Abbildung 3.3 stellt dieses Konzept schematisch dar.

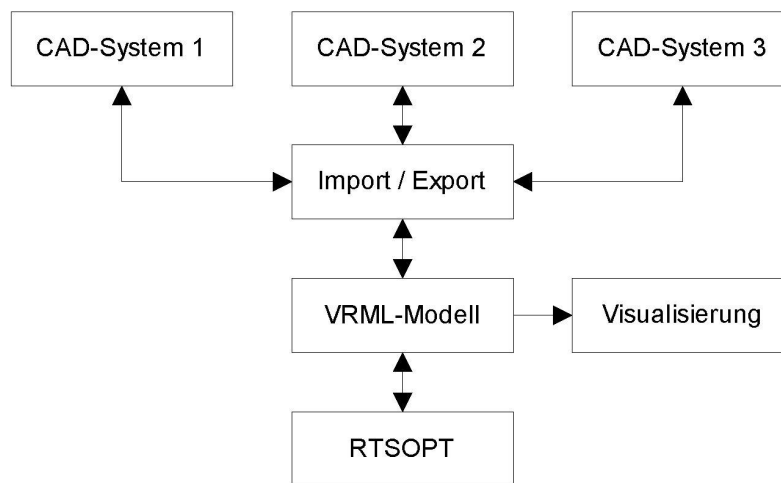


Abbildung 3.3: VRML-Modelle als Schnittstelle zu CAD-Systemen.

Durch diese Verbesserung ist es möglich, sehr leistungsfähige CAD-Systeme, die Flächen- und Volumenberechnungen, die Erstellung von Rechengittern für Strömungssimulationen und Finite-Elemente-Analysen erlauben, in Verbindung mit RTSOPT einzusetzen. Da die dreidimensionale, interaktive, animierte Flugvisualisierung auch auf der VRML-Technologie basiert, kann bei dieser Vorgehensweise das Fahrzeugmodell direkt in die Flugvisualisierung eingebunden werden. Abbildung 3.4 zeigt beispielhaft das VRML-Modell des Trägerkonzepts Sänger bestehend aus einer luftatmenden Unterstufe und der Raketenoberstufe (Horus).



Abbildung 3.4: VRML-Modell von Sänger und Horus.

Das Modell basiert auf den Spanntendaten aus der Arbeit [84]. Diese Daten werden mit einem hierfür entwickelten Programm in ein VRML-Modell konvertiert.

## 4 Das Optimierungsverfahren SOCS

Um die früher beobachteten Konvergenzschwierigkeiten der Optimierung zu überwinden und die Leistungsfähigkeit des Analysewerkzeugs zu erhöhen wurde das bisher in RTSOPT eingesetzte sequenziell-quadratische Optimierungsverfahren NLPQL (Nonlinear Sequential Quadratic Programming [92]), welches Ende der 80'er Jahre entwickelt wurde, in der vorliegenden Arbeit durch den modernen Löser SOCS (Sparse Optimal Control Software [11]) ersetzt. In diesem Kapitel ist zunächst eine Beschreibung der mit RTSOPT zu lösenden Optimierungsprobleme gegeben. Nach der mathematischen Problemformulierung werden anhand der Eigenschaften die Vorteile von SOCS und die Unterschiede zwischen den beiden Verfahren erläutert. Schließlich wird die Implementierung des Optimierungsverfahrens SOCS in RTSOPT beschrieben.

Wesentliche Schwerpunkte der vorliegenden Arbeit sind die Weiterentwicklung und Verbesserung des Systemanalysewerkzeugs RTSOPT. Daher werden in diesem Kapitel Vergleiche beider Optimierungsverfahren anhand von praktischen Anwendungsbeispielen aufgezeigt.

### 4.1 Das Optimierungsproblem

Die Aufgabe der multidisziplinären Optimierung eines Raumtransportsystems ist es, die zeitlichen Verläufe der Flugsteuergrößen (Anstellwinkel, Schubdrosselung, usw.), die Flugphasen - Schnittstellenparameter (Stufungs-Machzahl bzw. -Höhe, Zeit der Triebwerksabschaltung, usw.) und die Werte von Systemparametern (Massenaufteilung der Stufen, Flügelfläche, maximale Flügellast, usw.) zu finden, die eine Optimierungszielfunktion minimieren oder maximieren. Diese Funktion kann bei einem Träger z.B. die maximale Nutzlast bzw. der minimale Treibstoffbedarf, oder bei einem Wiedereintrittsfahrzeug die minimale integrale Wärmelast sein. Teilweise können auch kombinierte bzw. unterschiedliche Bewertungsfunktionen für verschiedene Flugphasen definiert werden.

Die Formulierung dieses Problems als ein Parameter-Optimierungsproblem (Nichtlineare Programmierung bzw. NLP) erfolgt durch die Diskretisierung der Zeit und Modellierung der Flugsteuerfunktionen und Zustandsvariablen mit Stützpunkten, zwischen denen die Funktionswerte mit kontinuierlichen, stückweise stetigen Interpolationspolynomen (Splines) berechnet werden. Diese Methode ist in Kapitel 3.2 beschrieben. Das resultierende NLP-Problem wird nach [11] mathematisch folgenderweise definiert:

Bestimme

den  $N_u^{(k)}$ -dimensionalen Steuervektor  $\mathbf{u}^{(k)}(t)$

und die zeitunabhängigen Parameter  $\mathbf{p}^{(k)}$ ,

um die folgende Zielfunktion zu minimieren:



$$\begin{aligned}
 J = & \phi [ t_I^{(1)}, \mathbf{y}^{(1)}(t_I^{(1)}), \mathbf{u}^{(1)}(t_I^{(1)}), t_F^{(1)}, \mathbf{y}^{(1)}(t_F^{(1)}), \mathbf{u}^{(1)}(t_F^{(1)}), \mathbf{p}^{(1)}, \\
 & \dots, t_I^{(N)}, \mathbf{y}^{(N)}(t_I^{(N)}), \mathbf{u}^{(N)}(t_I^{(N)}), t_F^{(N)}, \mathbf{y}^{(N)}(t_F^{(N)}), \mathbf{u}^{(N)}(t_F^{(N)}), \mathbf{p}^{(N)} ] \\
 & + \sum_{k=1}^N \left\{ \int_{t_I^{(k)}}^{t_F^{(k)}} q^{(k)} [\mathbf{y}^{(k)}(t), \mathbf{u}^{(k)}(t), \mathbf{p}^{(k)}, t] \cdot dt \right\}
 \end{aligned} \tag{4.1}$$

wobei  $N$  die Anzahl und  $q^{(k)}$  die Quadraturfunktionen der Phasen sind. Innerhalb einer Phase  $k$  liegt die unabhängige Variable  $t$  in dem Bereich  $t_I^{(k)} \leq t \leq t_F^{(k)}$ . Die Systemdynamik ist definiert durch einen Satz von Differentialgleichungen in der Form

$$\dot{\mathbf{y}} = \mathbf{f}[\mathbf{y}(t), \mathbf{u}(t), \mathbf{p}, t] \tag{4.2}$$

wobei  $\mathbf{y}$  den  $N_y^{(k)}$ -dimensionalen Zustandsvektor einer Phase bezeichnet. Die Randbedingungen werden spezifiziert für  $t_I$  und  $t_F$  durch Funktionen in der Form

$$\boldsymbol{\varphi}_I[\mathbf{y}(t_I), \mathbf{u}(t_I), \mathbf{p}, t_I] = \mathbf{0} \quad \text{und} \quad \boldsymbol{\varphi}_F[\mathbf{y}(t_F), \mathbf{u}(t_F), \mathbf{p}, t_F] = \mathbf{0} \tag{4.3}$$

Die Lösung muss darüber hinaus algebraische Pfadbeschränkungen in der Form

$$\mathbf{g}_l \leq \mathbf{g}[\mathbf{y}(t), \mathbf{u}(t), \mathbf{p}, t] \leq \mathbf{g}_u \tag{4.4}$$

erfüllen, wobei  $\bar{\mathbf{g}}$  ein Vektor der Dimension  $N_g$  ist.

Zusätzlich können untere und obere Grenzwerte für die Zustandsvariablen, Steuervariablen und zeitunabhängigen Optimierungsparameter vorgeschrieben werden:

$$\begin{aligned}
 \mathbf{y}_l & \leq \mathbf{y}(t) \leq \mathbf{y}_u \\
 \mathbf{u}_l & \leq \mathbf{u}(t) \leq \mathbf{u}_u \\
 \mathbf{p}_l & \leq \mathbf{p} \leq \mathbf{p}_u
 \end{aligned} \tag{4.5}$$

## 4.2 Grundlagen der Optimalen Steuerung

In diesem Kapitel werden die Grundlagen der Theorie der Optimalen Steuerung, die dem Optimierungsverfahren SOCS zugrunde liegen, und die wichtigsten Begriffe und Konzepte beschrieben. Dieses Verfahren wurde im Rahmen der vorliegenden Arbeit in RTSOPT integriert und in der Anwendung erprobt.

Ein nichtlineares Optimierungsproblem wird durch eine finite Anzahl von Variablen  $\mathbf{x}$  und Beschränkungen  $\mathbf{c}$  charakterisiert. Optimale Steuerungsprobleme betreffen dagegen kontinuierliche Funktionen wie  $\mathbf{x}(t)$  und  $\mathbf{u}(t)$ . Dies stellt ein unendlich-dimensionales Problem dar. Um ein solches Problem mit numerischen Verfahren der Nichtlinearen Programmierung zu lösen, muss es in ein Problem mit endlicher Dimension umgewandelt werden (Transkription). Dies geschieht in drei Schritten:

1. Beschreibung des dynamischen Systems durch eine finite Anzahl von Variablen (NLP-Variablen, Beschränkungen und Zielfunktion)
2. Lösung des endlich-dimensionalen Problems mit Parameter-Optimierungsmethoden (NLP-Subproblem)
3. Überprüfung der Genauigkeit der endlich-dimensionalen Approximation und gegebenenfalls Wiederholung der Transkription und Optimierung

Die Differentialgleichungen (4.2) der Systemdynamik können durch Einfach- oder Mehrfach-Schiessverfahren gelöst werden, die in der Abbildung 4.1 veranschaulicht werden.

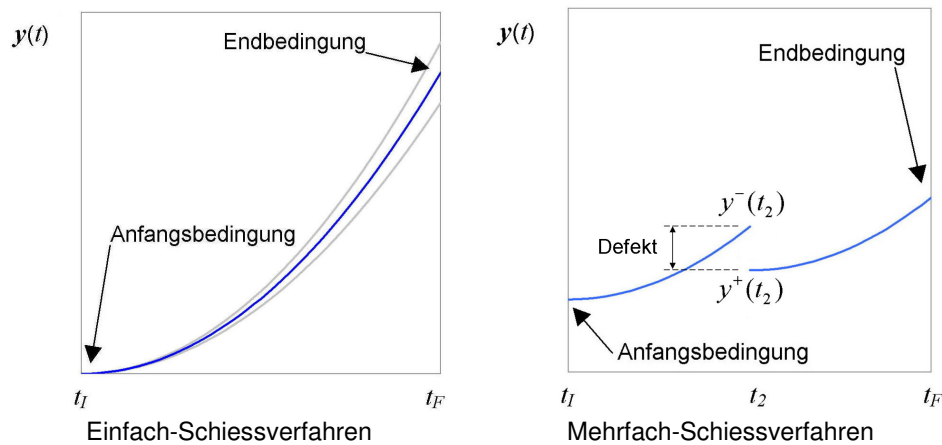


Abbildung 4.1: Schematische Darstellung von Schiessverfahren.

Die mittlere Kurve in der Abbildung links zeigt die Lösung nach zwei Iterationen. Das Einfach-Schiessverfahren besteht aus den folgenden Schritten:

1. Schätzung der Anfangsbedingungen  $\mathbf{x} = \mathbf{y}(t_I)$
2. Propagieren der Differentialgleichungen von  $t_I$  bis  $t_F$  ("Schiessen")
3. Bestimmung der Fehler in den Randbedingungen  $\mathbf{c}(\mathbf{x}) = \mathbf{y}(t_F) - \mathbf{b}$  ( $\mathbf{b}$  = Sollwert-Vektor)
4. Anwendung von NLP zur Bestimmung der Variablen  $\mathbf{x}$ , die die Beschränkungen  $\mathbf{c}(\mathbf{x}) = \mathbf{0}$  erfüllen

Diese Methode hat den Vorteil, dass die Anzahl der Iterationsvariablen gleich der Anzahl der Differentialgleichungen ist und dass das Problem mit dem Newton-Verfahren gelöst werden kann. Der Nachteil ist, dass eine kleine Änderung in den Anfangsbedingungen große Änderungen in den Endbedingungen hervorrufen kann. Dies führt dazu, dass die Beschränkungen  $\mathbf{c}(\mathbf{x})$  hochgradig nichtlinear werden und damit sehr schwierig zu lösen sind. Um diesen Nachteil hoher Sensitivität zu mindern kann das Problem in kleinere Schritte geteilt werden (Mehrfach-Schiessverfahren). In der Abbildung 4.1 ist eine schematische Darstellung des Verfahrens mit zwei Schritten und einem Defekt zu sehen. Um die Kontinuität der Lösung zu gewährleisten, muss eine zusätzliche Bedingung definiert werden:

$$y^-(t_2) = y^+(t_2) \quad (4.6)$$

Das so definierte Problem hat nun im ersten Schritt (Abbildung 4.1) zwei Variablen:

$$\mathbf{x}^T \equiv [y_1, y_2] \quad (4.7)$$

und zwei Beschränkungen

$$c_1(\mathbf{x}) = y^+(t_2) - y^-(t_2) = 0 \quad (4.8)$$

$$c_2(\mathbf{x}) = y(t_F) - b = 0 \quad (4.9)$$

Für  $M-1$  Intervalle stellt sich das Optimierungsproblem wie folgt dar:

$$t_1 = t_1 < t_2 < \dots < t_M = t_F \quad (4.10)$$

NLP-Variable:

$$\mathbf{x}^T = (y_1, y_2, \dots, y_{M-1}) \quad (4.11)$$

Beschränkungen:

$$c(\mathbf{x}) = \begin{bmatrix} y_2 - \hat{y}_1 \\ y_3 - \hat{y}_2 \\ \vdots \\ \boldsymbol{\Phi}[\hat{\mathbf{y}}_M, t_F] \end{bmatrix} = 0 \quad (4.12)$$

mit  $\hat{y}_k$  entsprechend dem Integral der Differentialgleichungen für das Segment  $k$  von  $t_k$  bis  $t_{k+1}$ . Aus dieser Darstellung wird ersichtlich, dass die Problemgröße mit steigender Anzahl von Segmenten wächst. Die Anzahl der NLP-Variablen und -Beschränkungen  $N$  ergibt sich aus der Anzahl der dynamischen Variablen  $N_y$  und der Anzahl der Segmente  $M-1$ :

$$N = N_y(M-1) \quad (4.13)$$

Die Jakobi-Matrix, die für die Newton-Richtungssuche benötigt wird, ist jedoch dünn besetzt. Von den  $[(M-1)N_y]^2$  Werten sind nur  $(M-1)N_y^2$  nicht Null [8]. Das Verhältnis der Matrix-Elemente mit Werten ungleich Null ist  $1/(M-1)$ , was bedeutet, dass die Matrix mit steigender Anzahl von Segmenten immer dünner besetzt wird. Dieses Verhalten kommt daher, dass die Bahnvariablen am Anfang der Trajektorie die Bedingungen am Ende nicht beeinflussen. Dadurch ist es möglich, das Problem mit speziellen Algorithmen zu lösen, die für dünn bzw. diagonal besetzte Matrizen sehr effizient sind.

Für die Bestimmung der optimalen Flugbewegung wird als Teilaufgabe das folgende Anfangswertproblem gelöst:

Berechne  $y(t_F)$  für  $t_1 < t_F$  und den definierten Wert  $y(t_1) = y_1$ , wobei die Lösung (4.15) erfüllen muss. Die Lösung lautet:

$$\mathbf{y}(t_{i+1}) = \mathbf{y}(t_i) + \int_{t_i}^{t_{i+1}} \dot{\mathbf{y}} dt = \int_{t_i}^{t_{i+1}} \mathbf{f}(\mathbf{y}, t) dt \quad (4.14)$$

$$\dot{\mathbf{y}} = \mathbf{f}[\mathbf{y}(t), t] \quad (4.15)$$

Um das Integral zu berechnen, wird das Integrationsintervall in  $k$  Subintervalle geteilt:

$$\tau_j = t_i + h_i \rho_j \quad (4.16)$$

wobei für  $1 \leq j \leq k$

$$0 \leq \rho_1 \leq \rho_2 \leq \dots \leq \rho_k \leq 1 \quad (4.17)$$

In diesem Subintervall wird eine Quadraturformel eingeführt:

$$\int_{t_i}^{t_{i+1}} f(y, t) dt \approx h_i \sum_{j=1}^k \beta_j \hat{f}_j, \quad \hat{f}_j \equiv f(\tau_j, \hat{y}_j). \quad (4.18)$$

Für die Auswertung dieser Approximation werden die Zwischenwerte  $\hat{y}(\tau_j) \equiv \hat{y}_j$  für  $1 \leq j \leq k$  entsprechend bestimmt:

$$\int_{t_i}^{\tau_j} f(y, t) dt \approx h_i \sum_{l=1}^k \alpha_{jl} f_l \quad (4.19)$$

Mit diesen Gleichungen wird eine Klasse von Runge-Kutta-Integrationsverfahren definiert:

$$y_{i+1} = y_i + h_i \sum_{j=1}^k \beta_j f_{ij} \quad (4.20)$$

wobei

$$f_{ij} = f \left[ \left( y_i + h_i \sum_{l=1}^k \alpha_{jl} f_{il} \right), (t_i + h_i \rho_j) \right], \quad 1 \leq j \leq k \quad (4.21)$$

und  $k$  die Runge-Kutta-Stufe bezeichnet,  $\rho_j$ ,  $\beta_j$ ,  $\alpha_{jl}$  sind bekannte Konstanten. Zu dieser Klasse von numerischen Integralberechnungsmethoden gehören z.B. die Euler-, Trapez-, Hermite-Simpson- und die Klassische Runge-Kutta-Methoden ( $k=4$ ). Da in SOCS die Steuerfunktionen ähnlich wie Zustandsvariablen behandelt werden, sind diese Verfahren die Basis für die weiterführenden Darstellungen und die Überleitung zu dünn besetzten Matrizen.

Oft wird die Lösung  $y(t)$  der Gleichung (4.15) über jeden Schritt  $t_i \leq t \leq t_{i+1}$ , durch ein Polynom mit Grad  $k$  bzw. Ordnung  $(k+1)$  approximiert:

$$\tilde{y}(t) = a_0(t-t_i)^0 + a_1(t-t_i)^1 + \dots + a_k(t-t_i)^k \quad (4.22)$$

Dabei werden  $(a_0, a_1, \dots, a_k)$  so gewählt, dass die Approximationsfunktion  $\tilde{y}(t)$  die folgenden Bedingungen erfüllt:

$$\tilde{y}(t_i) = y(t_i) \quad (4.23)$$

$$\frac{d\tilde{y}(\tau_j)}{dt} = f[y(\tau_j), \tau_j] \quad (4.24)$$

Diese Bedingungen werden Kollokationsbedingungen (Erfüllung der Differentialgleichungen) genannt und die resultierende Methode wird als Kollokationsmethode bezeichnet. Die Lösung ergibt ein stückweises Polynom. Typischerweise wird die Approximationsfunktion, statt mit in diesem Beispiel einfachheitshalber benutzten Polynomen, durch B-Splines repräsentiert.

Der Vorteil dieser expliziten Methode ist, dass die Integration nur von den Werten  $y(t_i)$  und  $y(t_{i+1})$  abhängt und damit keine iterative Korrektur benötigt wird. Bei impliziten Methoden dagegen erscheint  $y(t_{i+1})$  als ein unbekannter Term in der Gleichung. Die Trapez-Methode erfordert beispielsweise

$$0 = y_{i+1} - y_i - \frac{h_i}{2} [f(y_{i+1}, t_{i+1}) + f(y_i, t_i)] \equiv \zeta_i \quad (4.25)$$

wobei  $y_i \equiv y(t_i)$  gilt und die Werte von  $t_{i+1}$ ,  $y_i$ ,  $t_i$  und  $f[y_i, t_i]$  bekannt sind und  $\zeta_i$  der Defekt ist. Da der Defekt durch Korrekturiterationen reduziert werden muss, wird eine geschätzte Anfangslösung benötigt. Dabei kann dieser Schritt einen hohen Rechenaufwand bedeuten. Dies gilt insbesondere für "steife" Differentialgleichungen wie im folgenden Beispiel. An Stelle von (4.15) wird die Systemdynamik folgenderweise beschrieben:

$$\dot{y} = f[y(t), u(t), t] \quad (4.26)$$

$$\varepsilon \dot{u} = g[y(t), u(t), t] \quad (4.27)$$

wobei  $\varepsilon$  ein "kleiner" Parameter ist und die Lösung in einem sehr kleinen Bereich  $0 \leq t \leq t_\varepsilon$  schnelle Änderungen aufweist und danach das System sich wie folgt verhält:

$$\dot{y} = f[y(t), u(t), t] \quad (4.28)$$

$$0 = g[y(t), u(t), t] \quad (4.29)$$

Das resultierende System beschreibt eine semi-explizite, differentielle algebraische Gleichung mit den Differentialvariablen  $y(t)$  (Zustandsvariablen) und den algebraischen Variablen  $u(t)$  (Steuerfunktionen). Das Gleichungssystem ist semi-explizit, weil die Differentialvariablen explizit auf der linken Seite der Gleichung sind und die algebraischen Variablen implizit in  $f$  und  $g$  enthalten sind.

Die meisten numerischen Integrationsmethoden können als Ein- oder Mehr-Schritt-Verfahren klassifiziert werden. Mehr-Schritt-Verfahren haben allgemein die folgende Form:

$$y_{i+k} = \sum_{j=0}^{k-1} \alpha_j y_{i+j} + h \sum_{j=0}^k \beta_j f_{i+j} \quad (4.30)$$

wobei  $\alpha_j$  und  $\beta_j$  bekannte Konstanten sind. Für  $\beta_k = 0$  ist die Methode explizit, ansonsten implizit. Zu dieser Methodenklasse gehören beispielsweise die Adams-Methoden. Bei diesen Methoden müssen drei zusätzliche Probleme berücksichtigt werden, die bei Ein-Schritt-Verfahren nicht diskutiert wurden:

1. Die Methode hängt von Informationen an  $(k-1)$  vorherigen Punkten ab und benötigt damit eine Methode (z.B. ein Ein-Schritt-Verfahren wie Euler), um diese Anfangswerte zu produzieren.
2. Die Schrittweite  $h$  in (4.25) ist konstant. Ist eine Schrittweitenanpassung vorgesehen, muss sichergestellt werden, dass die Koeffizientenberechnung effizient und gut konditioniert ist.
3. Ähnliche Überlegungen gelten auch für veränderliche Anzahl von Integrationsschritten  $k$ .

Die dargestellten Verfahren beschreiben verschiedene Methoden für die Transkription eines kontinuierlichen Problems, welches durch differentielle algebraische Gleichungen definiert ist, in ein endlich-dimensionales parametrisiertes Problem, das mit einem NLP-Algorithmus gelöst werden kann, bzw. die Konstruktion eines Funktionsgenerators. Die Aufgabe des Funktionsgenerators ist es, für einen gegebenen Satz von NLP-Variablen  $x$ , die Funktionen  $c(x)$  (Beschränkungen) und die Zielfunktion  $J(x)$  zu berechnen. Die Newton-Methode erfordert die Berechnung der ersten und zweiten Ableitungen (Jakobi- und Hesse-Matrizen respektive). Dies bedeutet, dass der Funktionsgenerator mittels Finite-Differenzen-Approximation numerisch abgeleitet werden muss:

$$G_{.j} = \frac{1}{\delta_j} [c(x + \Delta_j) - c(x)] \quad (4.31)$$

$$G_{.j} = \frac{1}{2\delta_j} [c(x + \Delta_j) - c(x - \Delta_j)] \quad (4.32)$$

Die erste Gleichung beschreibt die Vorwärts- und die zweite beschreibt die Zentrale-Differenzierung angewendet auf die Spalte  $j$  der Jakobi-Matrix  $G$ .  $\Delta_j = \delta_j e_j$  ist ein Einheitsvektor in Richtung  $j$ .

Um die Gradienten auf diese Weise zu berechnen, müssen die Differentialgleichungen für jede Perturbation integriert werden. Die Konsequenz daraus ist, dass für jeden Gradienten und jede NLP-Iteration mindestens  $N$  Trajektorien benötigt werden. Obwohl der Rechenaufwand für genauere Gradientenberechnung hoch ist, überwiegt die Genauigkeitsforderung. Daher wird typischerweise mit Vorwärtsdifferenzen gerechnet und in der Nähe der Konvergenz auf Zentradifferenzen umgeschaltet. Auch die Genauigkeit der Interpolations- und Integrationsmethoden können durch komplexere Verfahren erhöht werden. Komplexere Algorithmen, z.B. mit variabler Schrittweite und Ordnung, führen jedoch dazu, dass sie ein "Rauschen" in die Gradientenberechnung einbringen und damit die Effizienz des Optimierungsalgorithmus beeinträchtigen. SOCS implementiert eine Transkriptionsmethode mit einem Integrations-schritt pro Mehrfach-Schiess-Segment. Diese Vorgehensweise stellt die Konsistenz des Funktionsgenerators sicher. Die Genauigkeitsproblematik wird außerhalb des NLP-Problems gelöst.

Techniken, die bei der Lösung von optimalen Steuerungsproblemen eingesetzt werden, werden in zwei Klassen unterteilt: direkte und indirekte Methoden. Eine direkte Methode bestimmt eine Reihe von Punkten  $x_1, x_2, \dots, x^*$ , die die Zielfunktion  $J(x)$  minimieren. Typischerweise gilt  $J(x_1) > J(x_2) > \dots > J(x^*)$ . Eine indirekte Methode dagegen versucht die Nullstellte der erforderlichen Bedingung  $J'(x) = 0$  zu finden. Während die erste Methode nur die Zielfunktionswerte vergleicht, muss die zweite Methode die Ableitung  $J'(x)$  berechnen und setzt die Differenzierbarkeit der Zielfunktion voraus und ist rechenintensiver. Die Wahl der Lösungsmethode (z.B. Mehrfach-Schiessverfahren) ist jedoch unabhängig von der Wahl der Formulierung (direkt/indirekt).

Die indirekten Methoden haben einige Nachteile (ausführlich beschrieben in [8]), wie z.B. die erneute Bestimmung von partiellen Ableitungsvektoren für jedes neue Problem, ausgesprochen schwierige Bestimmung der korrekten Verknüpfungsbedingungen und die Instabilität der Methode aufgrund ihrer hohen Sensitivität auf die Anfangsbedingungen. Wegen dieser Nachteile wird für den Löser SOCS die direkte Formulierung bevorzugt.

Durch die Diskretisierung eines kontinuierlichen optimalen Steuerungsproblems wird ein NLP-Problem konstruiert, für dessen Lösung die ersten und zweiten Ableitungen der Beschränkungen und der Zielfunktion berechnet werden müssen. Der Großteil der Jakobi-Matrix  $G_{ij}$  wird wie folgt definiert:

$$G_{ij} = \frac{\Delta \text{Defekt}_i}{\Delta x_j}, \quad i = \text{Segment-Index}, j = \text{Gitterpunkt-Index} \quad (4.33)$$

Die Matrix hat  $M$  Zeilen (Anzahl der Defekt-Beschränkungen) und  $N$  Spalten (Anzahl der Optimierungsvariablen). Durch die Diskretisierung wird bewirkt, dass die Änderung einer Variablen nur eine lokale Auswirkung auf die Nachbarbeschränkungen hat. Abbildung 4.2 verdeutlicht schematisch den örtlichen Einfluss der Perturbation einer Optimierungsvariablen anhand einer B-Spline-Kurve:

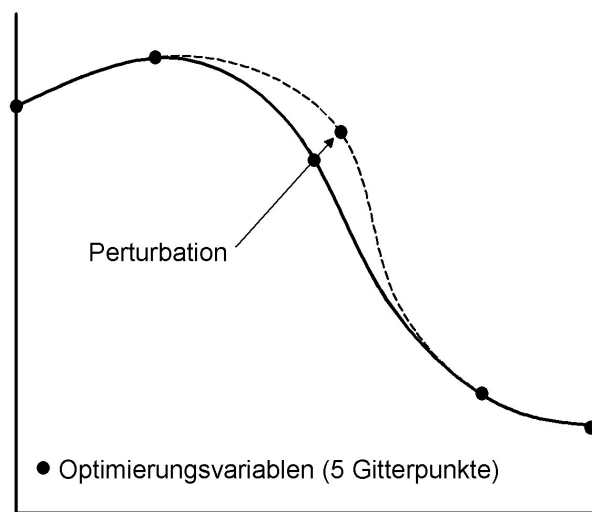


Abbildung 4.2: Sensitivität der Transkription.

Wie aus der Abbildung ersichtlich, hat die Perturbation eine nur lokale Wirkung, und die Ableitungen der meisten Beschränkungen nach der gestörten Variablen sind gleich Null. Dieses Prinzip ist die Grundlage für dünnbesetzte (sparse) Jakobi- und Hesse-Matrizen. In [8] wird die Linear-Tangenssteuerung<sup>2</sup> in Form eines NLP-Problems formuliert und gezeigt, dass durch die

1. lokale Sensitivität der Transkription,
2. Neugruppierung der Beschränkungsfunktionen nach Gitterpunkten (Trennbarkeit der Diskretisierung),
3. Neugruppierung weiterer Terme und Isolierung linearer Terme und
4. Einführung der Separierten Hermite-Simpson-Interpolation

eine Reduktion der Matrix-Elemente, die ungleich Null sind, um über 90% möglich ist.

Die Transkriptionsmethode besteht aus den drei Schritten Diskretisierung, Lösung des NLP-Problems und der Überprüfung der Genauigkeit. Wenn die Genauigkeitsforderungen nicht erfüllt sind, werden diese Schritte wiederholt. In diesem Fall wird eine Gitterverfeinerung (Mesh Refinement) durchgeführt. Hierzu wird zunächst eine Approximation der kontinuierlichen Lösung erstellt.

<sup>2</sup> Eine vereinfachte Version dieses Algorithmus wird von vielen Trägersystemen inklusive des Space Shuttle verwendet.

Für die Zustandsvariable  $y(t)$  wird der folgende Ansatz gewählt:

$$y(t) \approx \tilde{y}(t) = \sum_{i=1}^{N_1} \gamma_i B_i^{(1)}(t) \quad (4.34)$$

Hier sind  $B_i^{(1)}$  die Basisfunktionen der  $C^1$ -stetigen kubischen B-Splines,  $N_1 = 2M$ ,  $M$  =Anzahl der Gitterpunkte und  $\gamma_i$  sind die Koeffizienten, die durch die Hermite-Interpolation der diskreten Lösung definiert sind. Diese Approximation muss die folgenden Bedingungen erfüllen:

$$\tilde{y}(t_k) = y_k, \quad k = 1, \dots, M \quad (4.35)$$

$$\frac{d}{dt} \tilde{y}(t_k) = f_k \quad (4.36)$$

Eine ähnliche Approximation der Steuervariablen  $u(t)$  wird ebenfalls durchgeführt und die beiden Splines wie folgt dargestellt:

$$\begin{bmatrix} y(t) \\ u(t) \end{bmatrix} \approx \begin{bmatrix} \tilde{y}(t) \\ \tilde{u}(t) \end{bmatrix} = \sum_{i=1}^N a_i B_i^{(0)}(t) \quad (4.37)$$

wobei  $B_i^{(0)}$  die Basisfunktionen der  $C^0$ -stetigen kubischen B-Splines sind und  $N = 3M - 2$ . Obwohl sowohl  $y(t)$  als auch  $u(t)$  im  $C^0$ -stetigen kubischen B-Spline-Raum konstruiert werden, sind die Zustandsvariablen  $C^1$ -stetig kubisch und die Steuervariablen  $C^0$ -stetig linear oder quadratisch. Mit diesen Approximationen kann die Reduktion der Ordnung analysiert und anschließend der Diskretisierungsfehler berechnet werden. Für

$$\dot{y} = f[y(t), u(t), t] \quad (4.38)$$

$$0 = g[y(t), u(t), t]$$

und einen Integrationsschritt  $t_k \leq t \leq t_k + h_k$  ergibt sich

$$\begin{aligned} \hat{y}_{k+1} &= y_k + \int_{t_k}^{t_k+h_k} \tilde{f} dt \\ &= y_k + \int_{t_k}^{t_k+h_k} \dot{\tilde{y}} dt - \int_{t_k}^{t_k+h_k} \dot{\tilde{y}} dt + \int_{t_k}^{t_k+h_k} \tilde{f} dt \\ &= y_k + \tilde{y}_{k+1} - \tilde{y}_k - \int_{t_k}^{t_k+h_k} [\dot{\tilde{y}} - \tilde{f}] dt . \end{aligned} \quad (4.39)$$

Per Definition ist  $y_k = \tilde{y}_k$ , somit folgen

$$\tilde{y}_{k+1} - \hat{y}_{k+1} = \int_{t_k}^{t_k+h_k} [\dot{\tilde{y}} - \tilde{f}] dt \quad (4.40)$$

und die Fehlerschranke

$$|\tilde{y}_{i,k+1} - \hat{y}_{i,k+1}| = \left| \int_{t_k}^{t_k+h_k} [\dot{\tilde{y}}_i - \tilde{f}_i] dt \right| \leq \int_{t_k}^{t_k+h_k} |\dot{\tilde{y}}_i - \tilde{f}_i| dt . \quad (4.41)$$



Der absolute lokale Fehler in einem Schritt wird definiert durch

$$\eta_{i,k} = \int_{t_k}^{t_k+h_k} |\boldsymbol{\varepsilon}_i(t)| dt \quad (4.42)$$

und dem Fehler  $\boldsymbol{\varepsilon}$  in der Differentialgleichung in Abhängigkeit von  $t$

$$\boldsymbol{\varepsilon}(t) = \dot{\tilde{\mathbf{y}}}(t) - \mathbf{f}[\tilde{\mathbf{y}}(t), \tilde{\mathbf{u}}(t), t] \quad (4.43)$$

Aus dem absoluten Fehler wird schließlich der relative lokale Fehler wie folgt berechnet:

$$\boldsymbol{\varepsilon}_k \approx \max_i \frac{\eta_{i,k}}{(w_i + 1)} \quad (4.44)$$

mit dem Skalierungsgewicht

$$w_i = \max_{k=1}^M \left[ \left| \tilde{y}_{i,k} \right|, \left| \dot{\tilde{y}}_{i,k} \right| \right], \quad (4.45)$$

der den maximalen Wert für die  $y_i$  bzw.  $\dot{y}_i$  über die  $M$  Gitterpunkte der Phase definiert. Diese Fehlerberechnung reflektiert den Fehler in der Lösung der Differentialgleichungen. Das Programm SOCS implementiert ein ähnliches Abschlusskriterium.

Die beschriebene Methode für die Fehlerberechnung ist die Grundlage für die Konstruktion eines neuen Gitters. Ziel dabei ist die Reduktion der Diskretisierungsfehler. Neue Gitterpunkte werden z.B. durch Teilung der Intervalle hinzugefügt, in denen die Fehler höher sind. Da zusätzliche Gitterpunkte den Rechenaufwand erhöhen, müssen zusätzliche Gitterpunkte nur dann hinzugefügt werden, wenn dies zur Erhöhung der Genauigkeit beiträgt. Anhand von Optimierungen mit Beschränkungen wird in [9] gezeigt, dass die Gitterverfeinerungsmethode in SOCS, im Vergleich zu früheren Methoden, die Genauigkeitsbedingungen mit weniger zusätzlichen Gitterpunkten erfüllt. Dies spiegelt sich, besonders in Problemen mit raschen Zustandsänderungen und aktiven Zustandsbeschränkungen in den niedrigeren Rechenzeiten wider. Eine ausführliche Beschreibung des Gitterverfeinerungs-Algorithmus ist in [8] gegeben.

Die Skalierung eines NLP-Problems hat einen großen Einfluss auf die numerische Stabilität des Verfahrens und die Konvergenzrate der Lösung. In typischen Bahnoptimierungsproblemen können große Unterschiede in den Größenordnungen der Zustandsvariablen auftreten, z.B.  $\pm \pi$  für die Bahnneigung,  $0 - 4 \cdot 10^5$  m für die Höhe. Obwohl es keine universelle, automatische Lösung dafür gibt, kann die spezielle Struktur eines optimalen Steuerungsproblems ausgenutzt werden, um die Problemskalierung zu verbessern. Die in der Software SOCS implementierte Methode [8] erwies sich im Rahmen dieser Arbeit als erfolgreich. In keiner der in Kap.6 dokumentierten Anwendungen mussten benutzerdefinierte Skalierungsfaktoren gewählt werden.

### 4.3 Einbindung des Optimierungsprogramms SOCS in RTSOPT

Der Austausch des Optimierungslösers NLPQL durch SOCS im Programmsystem RTSOPT erforderte eine komplette Überarbeitung und Umstrukturierung von RTSOPT sowie die Implementierung neuer Programm-Module. Da für SOCS viel mehr Eingabedaten benötigt werden und diese ganz anders organisiert sind als es in der NLPQL-Version der Fall ist, war auch eine komplette Überarbeitung der Benutzerschnittstelle, Eingabedateien und Datenstrukturen erforderlich. Darüber hinaus war eine längere Einarbeitungszeit in die Vorgehensweise und Implementierung des Programmpakets SOCS und die Akkumulation praktischer Erfahrungen im Einsatz des Verfahrens notwendig.

Während NLPQL vom Analyseprogramm aus nach einer vollständigen Flugsimulation aufgerufen wird und über eine einfache Schnittstelle verfügt, übernimmt SOCS nach dem Aufruf die Kontrolle über den Programmablauf. Für SOCS müssen Programm-Module zur Verfügung gestellt werden, die den Spezifikationen im Handbuch [11] entsprechen. Diese Module dienen der Berechnung der Bewegungsgleichungen, Beschränkungen und der Definition des Optimierungsproblems. Auch die Simulation und Integration der Flugbahn erfolgt durch SOCS. Die Programmablaufkontrolle wird erst nach dem Abschluss der Optimierungsberechnung dem Hauptprogramm übergeben. Folgende Abbildung 4.3 gibt einen Überblick dieser Programmstruktur.

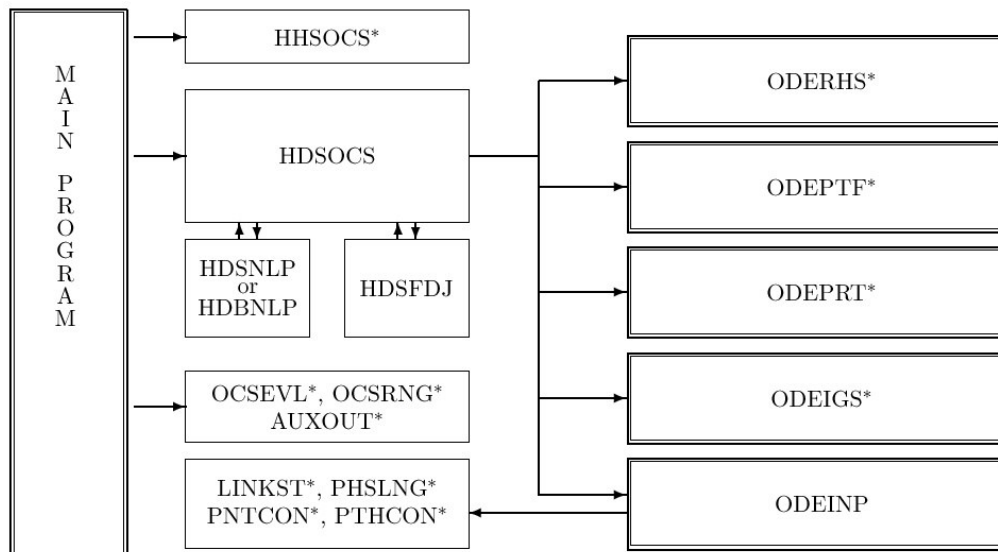


Abbildung 4.3: Programmstruktur bei voller Integration von SOCS im Analysewerkzeug [11].

In der Abbildung sind die vom Analysewerkzeug zu implementierenden Programm-Module dunkel umrahmt. Es ist ersichtlich, dass nach Aufruf der Hauptfunktion HDSOCS die Programmablaufkontrolle auf das Optimierungsmodul übergeht. Die Programm-Module ODE\* (Ordinary Differential Equations) werden vor Anfang bzw. während der Optimierung vom Programmsystem SOCS aufgerufen. Die folgende Tabelle 4.1 gibt eine kurze Übersicht der Unterprogramme, die für den Aufruf durch SOCS zur Verfügung gestellt werden müssen. Eine detaillierte Dokumentation ist in [11] gegeben.

Unterprogrammname	Funktion
ODERHS	Right Hand Side; Berechnung der Ableitungen der Flugzustände, Berechnung der kontinuierlichen Beschränkungsfunktionen für die folgende Eingaben: Flugzeit, Flugphasenindex, Steuerfunktionswerte, zeitunabhängige Optimierungsparameter, Phasenanfangs- und -endzeit
ODEPTF	Point Functions; Berechnung der Randbedingungen und der Zielfunktion für gegebene Flugzeit, Flugzustände, zeitunabhängige Optimierungsparameter für Phasenanfang bzw. -ende
ODEPRT	Zusätzliche benutzerdefinierte Ausgabefunktionen
ODEIGS	Initial Guess; Berechnung (Interpolation) der benutzerdefinierten Anfangsschätzungen aller Flugzustände und Steuerfunktionen für alle Punkte des Anfangsgitters und für alle Flugphasen
ODEINP	Input; Übergabe der gesamten Problemformulierung mit allen Rand- und Pfadbeschränkungen, Verknüpfungsbedingungen und der Zielfunktion

Tabelle 4.1: Die für den Optimierer SOCS zu implementierenden Programm-Module.

Hier ist das Unterprogramm ODEINP das komplexeste. Da die Schnittstelle generisch ist, sowohl die Reihenfolge als auch die Anzahl der zu übergebenden Daten von den vielfältigen Eingabemöglichkeiten des Benutzers abhängen und die Übergabe der Problemformulierung nur über reelle und ganzzahlige Daten bzw. Vektoren erfolgt, ist der Entwicklungsaufwand sehr hoch. Ferner reicht das softwaretechnische Verständnis dieser Schnittstelle nicht aus, weil eine genaue Kenntnis des Optimierungsverfahrens ebenfalls erforderlich ist.

Eine Herausforderung stellt auch die Spezifizierung eines Optimierungsproblems mit allen Flugphasen, Beschränkungen und Parametern beim Aufruf von SOCS dar. Ein Beispiel dafür ist die Optimierung bei gleichzeitiger Beschränkung z.B. des Flügellastfaktors  $n_F$ . Gleichung (4.46) veranschaulicht das Problem:

$$n_F(t) = \frac{A(t)}{G(t)} \leq n_{F.Max} \quad (4.46)$$

Aus der Gleichung ist ersichtlich, dass sowohl der Flügellastfaktor als auch die Auftriebskraft  $A$  und das Fahrzeuggewicht  $G$  Funktionen der Flugzeit sind. Ferner ist der maximal-zulässige Flügellastfaktor  $n_{F.Max}$  ein Optimierungsparameter, dessen Wert während der Optimierung durch den Löser variiert wird. Wie zuvor beschrieben, ist die dynamische Änderung der Beschränkungen nicht möglich. Daher kann diese Flugbeschränkung nicht wie eine gewöhnliche Pfadbeschränkung definiert werden. Folglich musste eine normierte Beschränkungsfunktion eingeführt werden, deren Wert unabhängig vom Flügellastfaktor, kleiner Eins sein muss. Das gleiche Problem entsteht auch dann, wenn die Größe eines Trägersystems optimiert wird, deswegen seine Wiedereintrittsmasse nicht bekannt ist und somit die Anfangsmasse für die Wiedereintrittsflugphase nicht direkt eingegeben werden kann. In RTSOPT mussten daher spezifische Methoden entwickelt werden, um diese Probleme erkennen und durch automatische Definition zusätzlicher Beschränkungsfunktionen lösen zu können.

Unstetigkeiten wie bei einer Triebwerksabschaltung oder Stufentrennung können bei Benutzung von NLPQL einfach im Programmcode realisiert werden. Der Optimierer SOCS lässt dagegen Sprünge in den Flugzuständen nicht zu. Selbst Unstetigkeiten in Rechenmodellen, z.B. durch lineare Interpolation, können zu Konvergenzschwierigkeiten bzw. Erhöhung der Rechenzeit führen. Daher erfordert die Einführung jeder Unstetigkeit in den Flugzuständen, Rechenmodellen und in der Flugsteuerung die Definition einer neuen Flugphase. Dies erfordert wiederum die Neudefinition von allen Beschränkungen, Steuerfunktionen, Optimierungsparametern für jede Flugphase und die Definition von Kontinuitätsbeschränkungen und Verknüpfungsbedingungen zwischen allen Flugphasen.

Aus den geschilderten Gründen wurde die gesamte Benutzerschnittstelle, Eingabestruktur und der Programmablauf von RTSOPT der Optimierungssoftware SOCS angepasst.

Zusammenfassend muss festgestellt werden, dass die Implementierung von SOCS einen zuvor nicht erwarteten hohen und zeitintensiven Aufwand erfordert hat, der allerdings durch die signifikanten Leistungsvorteile in der Anwendung belohnt wird, vgl. Kapitel 6.

## 5 Visualisierung

Durch Einsatz geeigneter Visualisierungsmethoden kann die Analyse verfügbarer Informationen, vor allem wenn die Anzahl der Daten sehr hoch ist, erleichtert werden. Die grafische Aufbereitung von Informationen und die Darstellung großer Datenmengen ermöglichen die Betrachtung vieler Informationen auf einen Blick. Dadurch werden Unstimmigkeiten in den Daten wie z.B. Unstetigkeiten in interpolierten, räumlichen Daten sofort ersichtlich. Durch die Visualisierung können Teilinformationen gefiltert und herausgehoben werden. Darüber hinaus können komplizierte Bewegungsabläufe, die durch Betrachtung der Flugzustandsdaten schwierig nachzuvollziehen sind, durch Animationen veranschaulicht werden.

In diesem Kapitel werden einige Anwendungsbeispiele zur Visualisierung mit VRML erläutert und anhand dieser Beispiele die Vorteile dieser Technologie demonstriert. Diese und die im Anhang B aufgeführten Beispiele entstammen aus den Beiträgen des Autors zu verschiedenen Projekten des Instituts für Raumfahrtssysteme mit externen Kooperationspartnern.

### 5.1 Zwei- und Dreidimensionale Plots

Für die Erstellung von Kurven- und Flächendarstellungen wurden Programme entwickelt, die Simulationsergebnisse und Modelldaten in VRML-Modelle konvertieren. Der Entwicklungsaufwand ist dabei nicht groß, da VRML-Modelle aus wenigen Schlüsselwörtern und Geometriedaten in einer vorgegebenen Struktur bzw. Reihenfolge bestehen. Mit Hilfe solcher Werkzeuge werden aus Optimierungsergebnissen (wie Anstellwinkel und Flugwindhängewinkel als Funktion der Flugzeit) und Modelldaten (wie Auftriebsbeiwert über die Machzahl und Anstellwinkel) sofort VRML-Modelle erstellt und für die Betrachtung durch den Benutzer zur Verfügung gestellt. Abbildung 5.1 zeigt mit diesen Werkzeugen erstellte VRML-Modelle, die im ersten Fall Flugsteuerungen  $\alpha(t)$ ,  $\mu_a(t)$  und im zweiten Beispiel einen Aerodynamikdatensatz  $c_A(\alpha, Ma)$  für das Fahrzeug und die Mission Hopper betreffen.

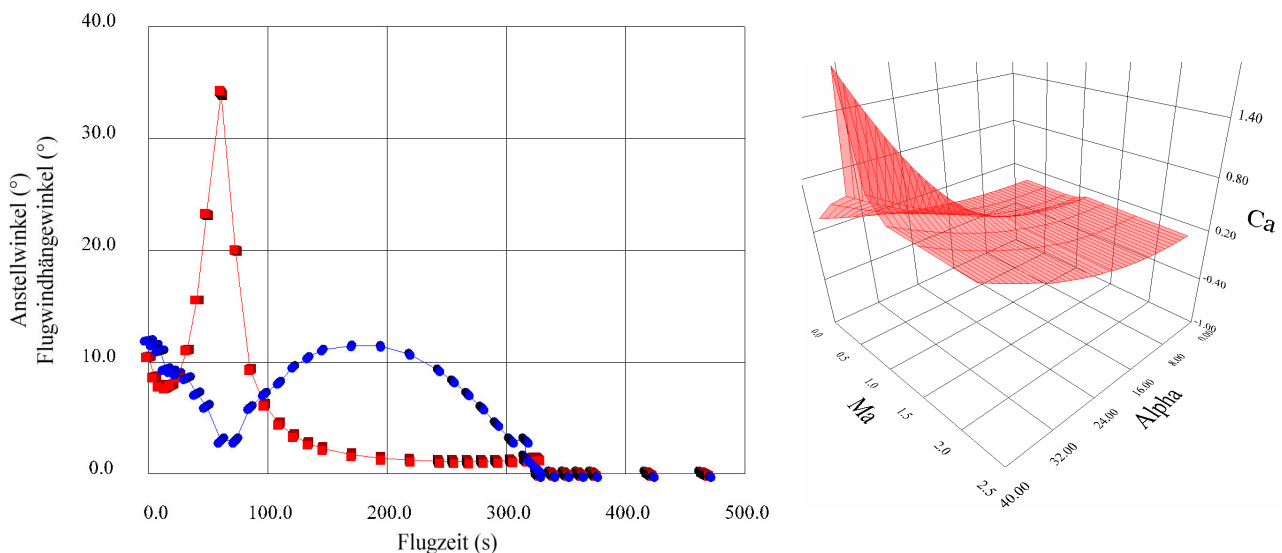


Abbildung 5.1: Mit VRML erstellte Kurven- und Flächendarstellungen von Rechenmodellen.

Die Vorteile dieser Modelle liegen darin, dass sie sehr schnell und automatisch erstellt werden, darüber hinaus plattformunabhängig austauschbar sind und ohne eine kommerzielle Anwendungssoftware betrachtet werden können.

## 5.2 Formmodelle von Raumfahrzeugen

Dreidimensionale Modelle von Raumfahrzeugen dienen der konzeptionellen Auslegung, Entwurfsüberprüfung und Darstellung des Fahrzeuges in der Flugbahnvisualisierung. In RTSOPT wird wegen der in Kapitel 2.2 erläuterten Vorteile die VRML-Technologie eingesetzt. Die Abbildung 5.2 zeigt beispielhaft einige Fahrzeugmodelle, die in verschiedenen Studien am IRS verwendet worden sind.

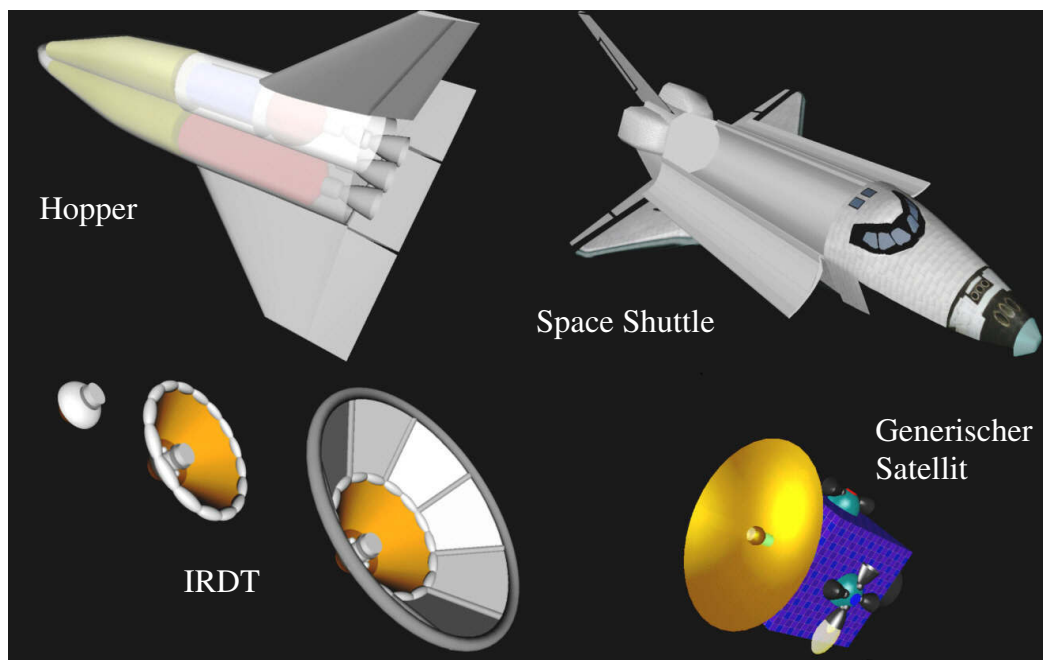


Abbildung 5.2: VRML-Modelle ausgewählter Raumfahrzeuge.

Das Hopper-Modell besteht aus verschiedenen VRML-Modulen, die teilweise mit dem Modellierwerkzeug Cinema4D ([70], [102]) und teilweise durch direkte Kodierung in VRML erstellt und zusammengefügt wurden. Das Satellitenmodell und das Modell der IRDT-Kapsel mit den mehrstufigen, aufblasbaren Brems- und Hitzeschutzschildern wurden vollständig durch direkte Kodierung erstellt. In allen drei Modellen sind Schnittstellen-Programm-Module in JavaScript implementiert, die der Animation von Fahrzeugkomponenten dienen. Über diese Schnittstellen werden die aerodynamischen Steuerflächen von Hopper, die aufblasbaren Stufen der IRDT-Kapsel und die Lageregelungstriebwerke des Satellitenmodells animiert und somit der Informationsgehalt der Flugbahndarstellungen angereichert. Das Space Shuttle Modell wurde von der NASA erstellt und im Internet zur Verfügung gestellt.

Diese Modellierarbeiten zeigen, dass die Fahrzeugmodelle entsprechend den Anforderungen an die Visualisierung in Kapitel 2 mit beliebigen CAD-Systemen und durch direkte Kodierung in VRML erstellt und ohne nennenswerten Aufwand zusammengefügt werden können. Damit bestätigen sich die Vorteile dieser Vorgehensweise. Abbildung 5.3 illustriert eine Auswahl an

parametrisierten generischen Objekten und Raumfahrzeug-Komponenten, die für die schnelle Modellierung verschiedener Raumtransportsysteme verwendet werden.

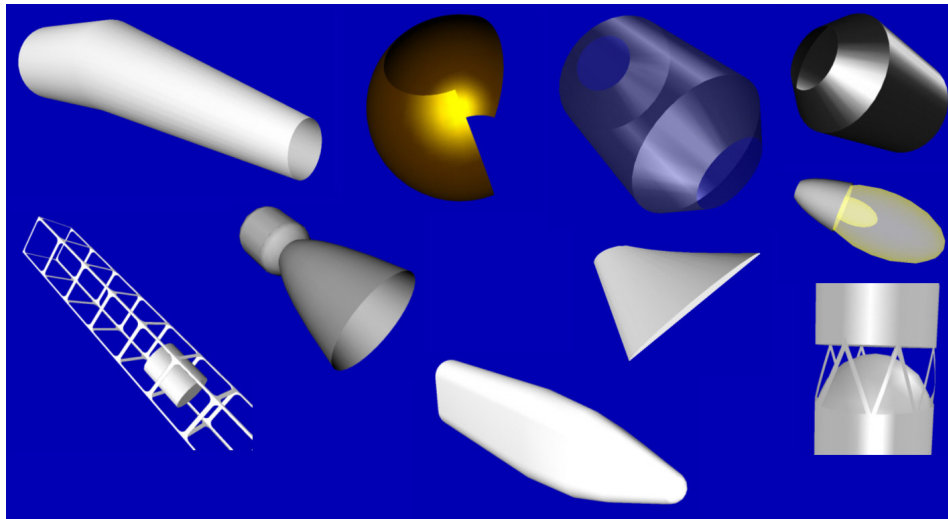


Abbildung 5.3: VRML-Module für Fahrzeugmodellierung.

In der Abbildung sind Fahrzeugkomponenten wie Triebwerke, Stufenadapter, Flügel, Kugel-tanksegmente, zylindrische Körper- und Tankmodule und einige komplexere geometrische Module zu sehen. Diese sind zum größten Teil als VRML-Prototypen erstellt worden. VRML-Prototypen sind Module, die oft JavaScript-Programme beinhalten, die dazu dienen, aus wenigen Geometrieparametern räumliche Objekte zu generieren. Diese Prototypen werden bei der Fahrzeugmodellierung in das Modell eingefügt und durch Eingabe der Prototyp-Parameter wie beispielsweise Rotationswinkel, Radius, Anzahl der Kanten, Koordinaten der Spannten verwendet.

Auch von der Grafiktechnik "Texturabbildung" (Texture Mapping), bei der ein Bild bzw. eine Textur auf eine Oberfläche transformiert wird und die Bestandteil der VRML-Spezifikation ist, wurde intensiv Gebrauch gemacht. Abbildung 5.4 veranschaulicht diese Technik.

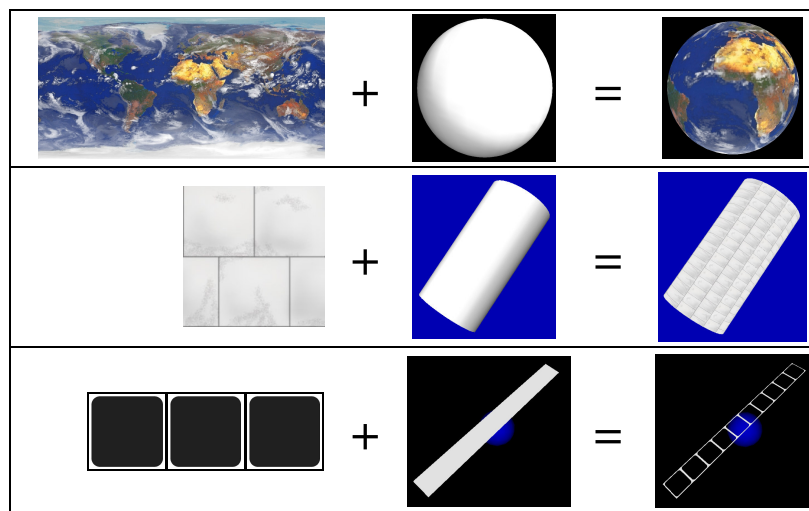


Abbildung 5.4: Texturabbildung auf räumlichen Objekten.

Während im ersten Beispiel eine Karte direkt auf die Oberfläche einer Kugel transformiert wird, wird im zweiten Beispiel das Bild bzw. die Textur vorher skaliert und vervielfacht. Ganz anders ist die Funktion der Textur im dritten Beispiel, wo die Textur die durchsichtigen Stellen des Geometriemodells bestimmt und dadurch die Geometriemodellierung wesentlich vereinfacht. Die erste Methode wird bei der Modellierung von Himmelskörpern und Landkarten benutzt, über denen die Flugbahn dargestellt wird. Während die zweite Methode für die Modellierung von Fahrzeugoberflächen eingesetzt wird ist die dritte Methode in der Modellierung von Gitterstrukturen sehr nützlich. Da VRML die direkte, automatische Texturtransformation von Bilddateien unterstützt, entfallen die aufwändige Dekodierung von Bilddateien und die Berechnung von Texturtransformationen, die bei Visualisierung beispielsweise mit OpenGL erforderlich sind.

Die Fahrzeugvisualisierung dient neben der Auslegung und Flugbahnvisualisierung auch der grafischen Aufarbeitung von Aerodynamikanalyse-Ergebnissen. Abbildung 5.5 zeigt beispielhaft zwei solche Darstellungen, die Ergebnisse von aerothermodynamischen Analysen von Raumfahrzeugen veranschaulichen.

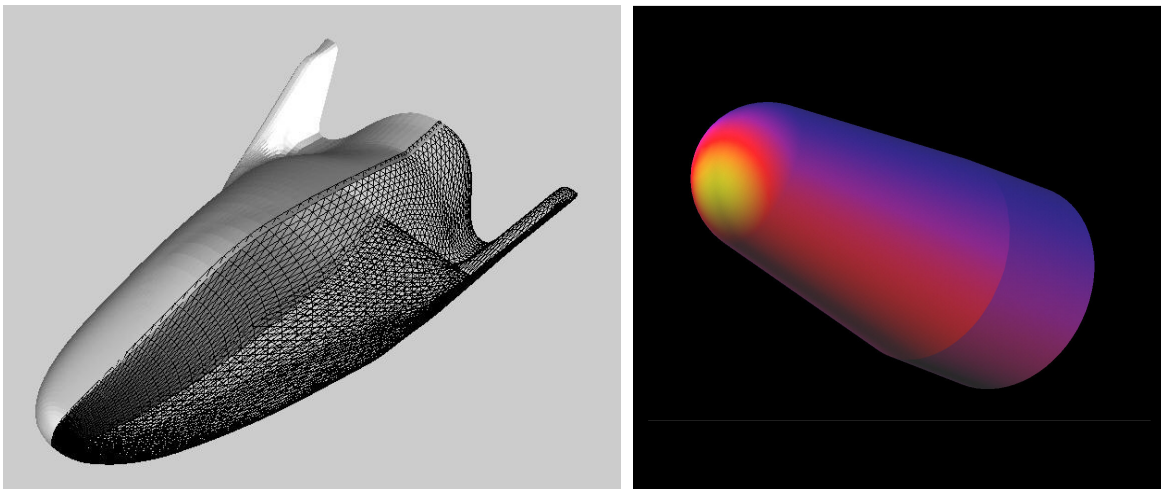


Abbildung 5.5: VRML-Darstellungen aerodynamischer Analysen.

In der Abbildung sind links die Fahrzeugoberfläche und das Rechengitter für eine numerische Strömungssimulation des Raumgleiters X-38 [48] und rechts die Druckverteilung (Ergebnis einer Strömungssimulation) auf einem generischen bikonischen Wiedereintrittskörper [12] zu sehen.

### 5.3 Flugvisualisierung

Für die Flugbahnvisualisierung wurde dem Programmpaket RTSOPT ein im Rahmen der vorliegenden Arbeit entwickeltes Flugvisualisierungsprogramm *VirtualTrajectory* (VT) [58] beigefügt. Mit VT werden Flugsimulationsergebnisse direkt in VRML-Animationen umgewandelt. Das Programm lädt die Simulationsergebnisse (Flugzeit, Bahn- und Lagedaten, Wärmelast), eine Konfigurationsdatei, in der verschiedene Darstellungsoptionen und -einstellungen definiert sind und das Fahrzeugmodell, das als VRML-Modell vorliegt und erstellt eine dreidimensionale, interaktive Animation des Flugverlaufs, die mit einem Internet-



Browser betrachtet werden kann. Des Weiteren können diese Modelle auch in der Virtuellen-Realität-Umgebung mit Laser-Zeiger und 3D-Brille immersiv betrachtet werden. Diese Möglichkeit besteht auch in der CUBE der Visualisierungsabteilung des Höchstleistungsrechenzentrums HLRS der Universität Stuttgart [40]. Abbildung 5.6 zeigt den Aufbau der CUBE und die Betrachtung eines Getriebemodells.

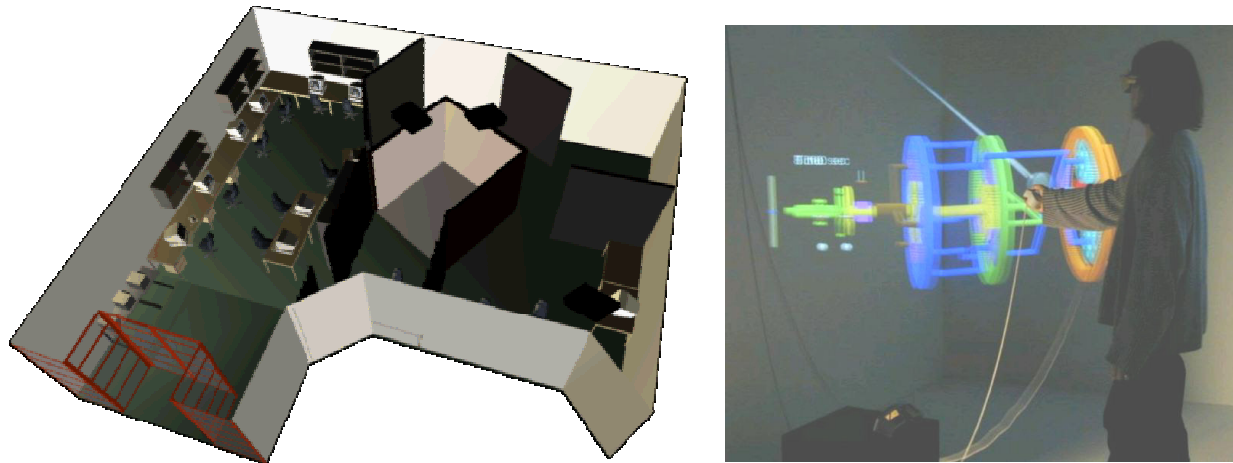


Abbildung 5.6: Die VR-Umgebung in Stuttgart [40].

Die CUBE besteht aus drei seitlichen Projektionswänden und dem Projektionsboden. Die Position und die Blickrichtung des Betrachters werden erfasst und die Bilder für die vier Projektionsflächen entsprechend berechnet. Dabei werden für die beiden Augen des Betrachters perspektivisch unterschiedliche Bilder generiert und auf die Projektionsflächen projiziert (stereoskopische Projektion: Foto rechts). Die mit den schnell wechselnden Bildfolgen synchronisierte 3D-Brille schaltet dabei rechts und links die Sicht abwechselnd ein und aus, so dass jedes Auge nur die entsprechenden Bilder sieht. Ergebnis ist eine sehr realistische dreidimensionale Wahrnehmung der den Betrachter umgebenden virtuellen Welt. In der Abbildung 5.7 sind zwei Benutzer zu sehen, die als Ergebnis einer Missionsanalyse [6] die Nutzlastrückführung von der Internationalen Raumstation ISS mittels der IRDT-Kapsel betrachten.

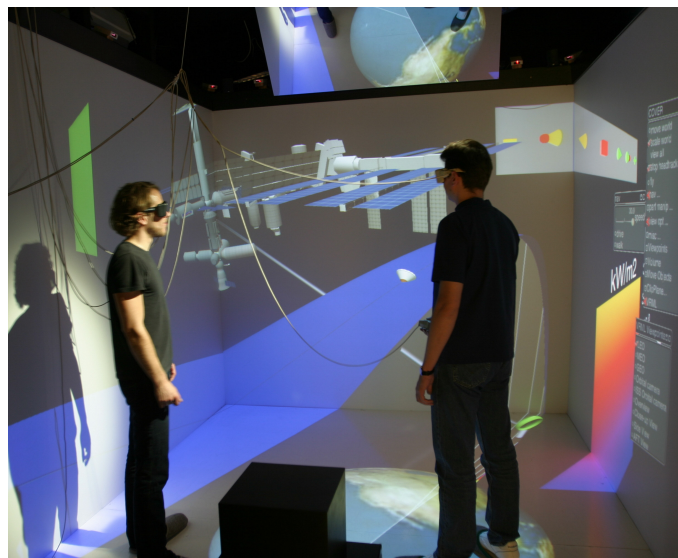


Abbildung 5.7: Immersive Betrachtung einer Flugvisualisierung in der VR-Umgebung [6].

Solche fortschrittlichen Visualisierungstechnologien unterstützen Ingenieure in der Analyse und im Verständnis komplexer Bewegungsabläufe, wie Bahnmanöver, kombinierte Rotationen und Sichtbarkeits- bzw. Abstandsbedingungen. Die Darstellung der Taumelbewegungen für die Flugdatenanalyse der IRDT-Kapsel [38] demonstrierte den Nutzen der Flugvisualisierung. Für diesen Zweck wurde ein Animationsprogramm in C++ mit OpenGL entwickelt. Diese Animation wurde später zum Vergleich auch mit VRML erstellt. Abbildung 5.8 illustriert diese Darstellung und zum Vergleich die OpenGL-Version (kleines Bild).

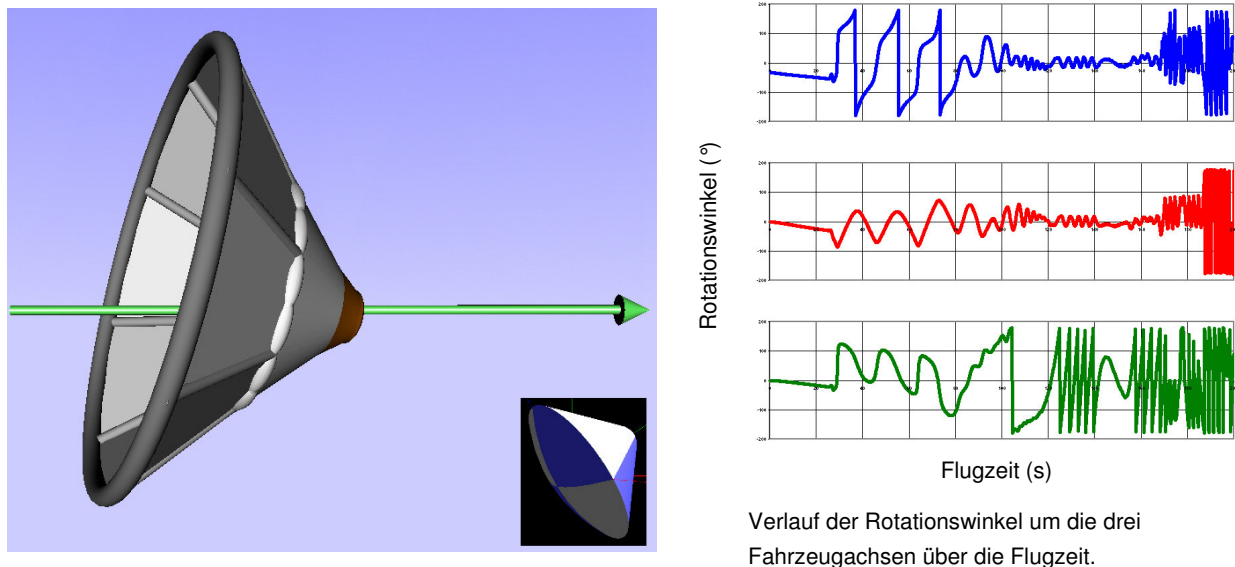


Abbildung 5.8: Animation der Taumelbewegungen der IRDT Kapsel mit VRML.

In der Abbildung ist links ein Standbild aus der Animation der Taumelbewegungen und rechts die Verläufe der Rotationswinkel um alle drei Fahrzeugachsen über die Flugzeit zu sehen. Die Vorstellung der Taumelbewegungen anhand der Zeit-Winkel-Daten und Kurvendarstellungen ist sehr schwierig. Dieses Programm stellt die Taumelbewegungen der Kapsel auf der Basis vorliegender Flugdaten in Quasi-Echtzeit dar. Dabei konnte zweifelfrei eine Störung der Flugbewegung durch eine äußere Einwirkung (Zusammenstoß der Kapsel mit dem Adapter der Trägeroberstufe) festgestellt werden. Durch die Erstellung dieser Animation auch mit VRML wurden die in Kapitel 2 bereits erläuterten Vorteile dieser Technologie demonstriert und bestätigt. Mit VRML konnten sowohl das viel realistischere Fahrzeugmodell als auch die Animation mit wesentlich geringerem Aufwand erstellt werden. Darüber hinaus kann mit dem VRML-Modell auch die Entfaltung der aufblasbaren Schilder dargestellt werden (s. Abbildung 5.2, Seite 69).

Auch in der Analyse der Rückkehrbahn des Raumgleiters X-38 wurden VRML-Animationen eingesetzt, um Flugbahndaten [37] räumlich darzustellen. In der Abbildung 5.9 ist eine mit *VirtualTrajectory* erstellte interaktive Animation in VRML abgebildet. Dargestellt werden die Wärmebelastung, Lage und Bahnverlauf, Flugzeit und die Werte der Zustandsvariablen (z.B. Höhe, Geschwindigkeit). Der Zeitverlauf kann während der Animation interaktiv gesteuert werden.

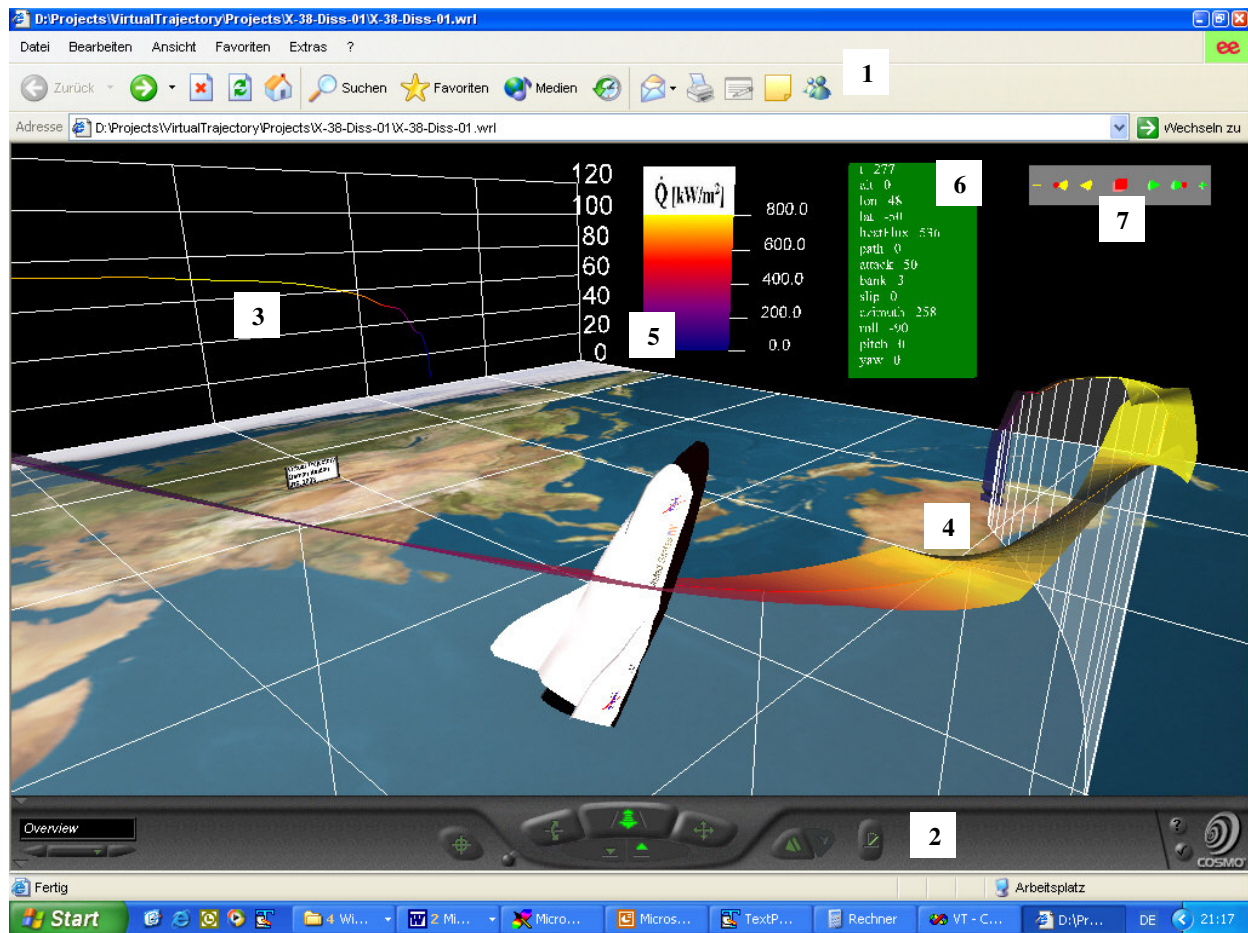


Abbildung 5.9: Visualisierung der X-38 Rückkehrbahn.

In der Abbildung ist die Betrachtung des mit *VirtualTrajectory* erstellten VRML-Modells in Microsoft Internet Explorer (1) mit dem Plug-In Cosmoplayer von CosmoSoftware [17] zu sehen (2). Das X-38-Modell stammt aus den Internet-Seiten der NASA. Die Flugbahn wird über der Erdkarte und auf den senkrechten Ebenen Höhe über Längengrad (3) bzw. Breitengrad dargestellt. Mit einem farbigen, verwundenen Band (4) wird der Flugverlauf visualisiert. Während die Verwindung dieses Bands den Verlauf des Flugwindhängewinkels repräsentiert, entspricht der Farbverlauf dem Wärmelastverlauf. Eine Farbcodetabelle (5) illustriert die Farbwertetabelle für den Wärmefluss. Eine durchsichtige Anzeige (6) listet die numerischen Werte aller Flugzustände (Höhe, Geschwindigkeit, Anstellwinkel, usw.) auf und kann vom Betrachter in einen beliebigen Bereich des Bildes verschoben werden. Schließlich ermöglicht ein Steuerwerkzeug (7) die Steuerung der Flugzeit (z.B. Rückwärts, Pause, Schneller). *VirtualTrajectory* ermöglicht das Ein- und Ausschalten verschiedener Darstellungsoptionen und -elemente. Beispielweise können die Elemente 3-7 in der Abbildung 5.9 ausgeschaltet werden. Die Visualisierung über der Kugelerde ist ebenfalls möglich. Eine solche Visualisierung der in der Abbildung 5.9 dargestellten Flugbahn zeigt die nachstehende Abbildung 5.10.

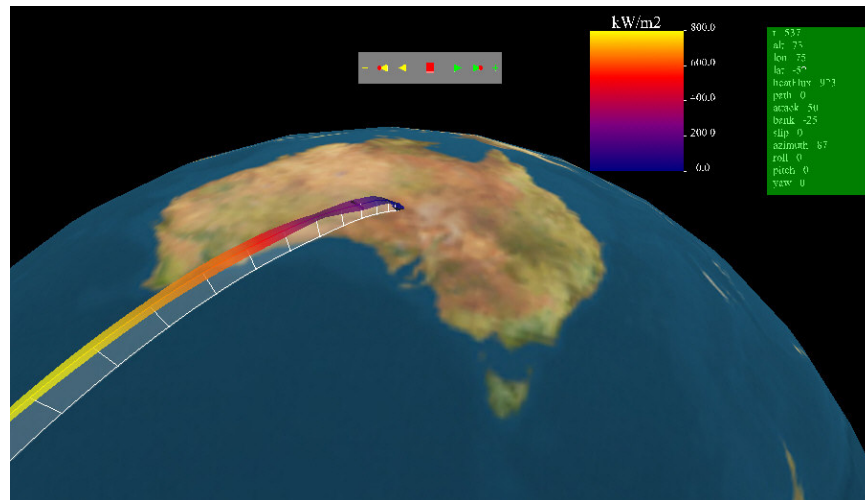


Abbildung 5.10: Darstellung der X-38-Rückkehrbahn.

Bei Raumfahrtmissionen, die z.B. mehrere Umläufe um die Erde beinhalten, ist diese Art der Darstellung sinnvoll.

Durch die Verwendung von VRML wurden die Visualisierungsfähigkeiten des Systemanalyse-Werkzeugs RTSOPT deutlich verbessert und durch neue, animierte, interaktive Darstellungen erweitert.

Die in diesem Kapitel dargestellten VRML-Anwendungen zeigen die Flexibilität, den Nutzen und die Effektivität dieser Technologie in der Analyse von Raumtransportsystemen und -missionen. Weitere Visualisierungsbeiträge des Autors zu Projekten des IRS mit den Kooperationspartnern verdeutlichen die vielseitigen Einsatzmöglichkeiten der VRML-Technologie und sind im Anhang B aufgeführt.

## 6 Anwendungsergebnisse

Die neue RTSOPT-Version wurde für die Flugoptimierung und Analyse des Trägerkonzeptes Hopper von EADS [106], das Leitkonzept des nationalen Technologieprogramms ASTRA ist, eingesetzt. In diesem Kapitel werden das Hopper-Konzept, das Fahrzeug und die Transportmission sowie die in der Analyse eingesetzten Rechenmodelle beschrieben. Anhand der Optimierungsergebnisse werden die Verbesserungen und die Leistungsfähigkeit des Analysewerkzeuges RTSOPT demonstriert.

### 6.1 Hopper – Das Fahrzeug und die Transportmission

Das Hopper-Konzept stammt aus der FESTIP-Studie der ESA [25], [19]. Später wurde das Konzept im Rahmen des nationalen Programms ASTRA aufgegriffen [7], [105], die Transportmission neu definiert, das Fahrzeug modifiziert und im Detail untersucht. Nachfolgend werden das Fahrzeug, die Transportmission und die Rechenmodelle der vorliegenden Anwendungsstudie beschrieben.

Hopper ist ein unbemannter, teilweise wiederverwendbarer Träger mit Raketenantrieb. Das Trägersystem besteht aus der wiederverwendbaren geflügelten Grundstufe und der Oberstufe HUS. Abbildung 6.1 stellt Subsysteme, die Stufen und die Nutzlast des Hopper-Trägersystems dar.

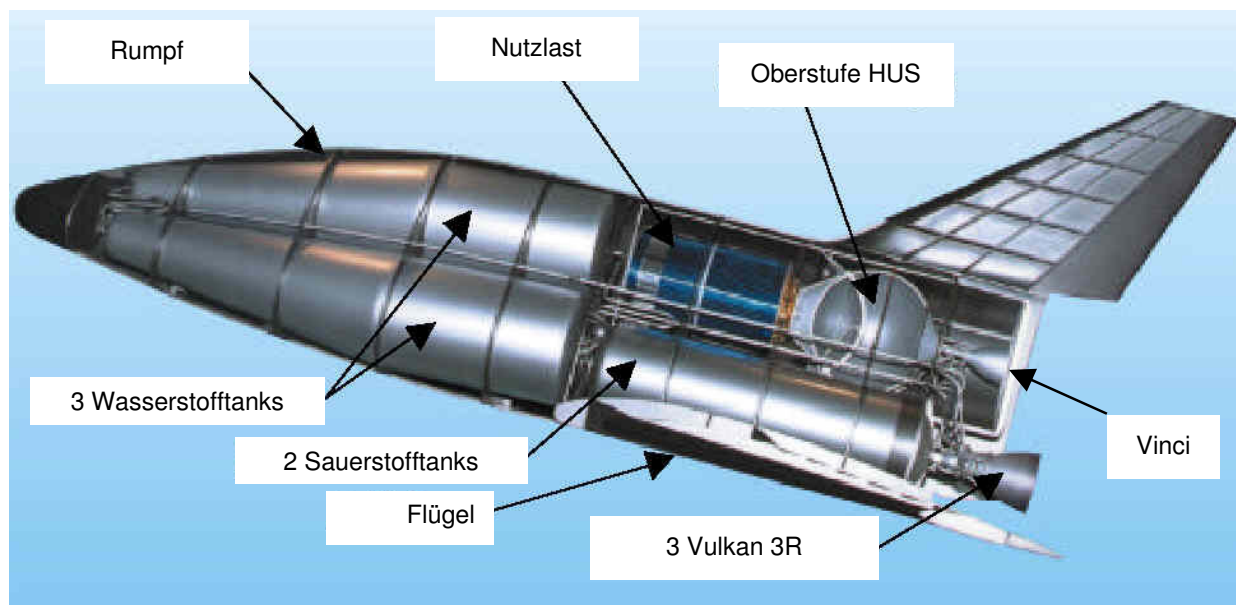


Abbildung 6.1: Das Hopper-Fahrzeug [106].

Das Antriebssystem der Grundstufe besteht aus drei kryogenen Vulkan-3R Triebwerken (eine zukünftige Version des Zentraltriebwerks der Trägerrakete Ariane-5). Die Hopper-Oberstufe HUS wird mit dem ebenfalls kryogenen Vinci 200 Triebwerk (Triebwerk der geplanten ESC-B Oberstufe der Ariane-5) angetrieben. In der Tabelle 6.1 sind einige Daten des Hopper-Fahrzeugs zusammengetragen, während Tabelle 6.2 die Leistungsdaten der Triebwerke zeigt. Eine genaue Beschreibung des Fahrzeugs ist in [106] gegeben.

<b>Nutzlast</b>	
Geforderte Nutzlast (GTO)	7,5 Mg
<b>Oberstufe HUS</b>	
Leermasse	3,9 Mg
Treibstoffmasse	24 Mg
<b>Hopper</b>	
Trockenmasse	59 Mg
Aufstiegstreibstoff	390 Mg
Reserven und Betriebsstoffe	5,6 Mg
Startmasse	490 Mg
Masse nach Abheben von der Schiene	480 Mg
Masse bei Triebwerksabschaltung	98 Mg
Landemasse	60 Mg
Länge	50,2 m
Spannweite	27,2 m
Nutzlastraum-Länge	16,5 m
Aerodynamische Referenzfläche	580,0 m <sup>2</sup>

Tabelle 6.1: Massen und Dimensionen von Hopper [106].

	<b>Grundstufe</b>	<b>Oberstufe</b>
Triebwerk	Vulkan 3R	Vinci 200
Anzahl	3	1
Vakuumschub	1650,6 kN	197 kN
Bodenschub	1351,2 kN	-
Spezifischer Vakuumimpuls	429,1 s	466 s
Spezifischer Bodenimpuls	351,3 s	
Massendurchsatz	392 kg/s	43,32 kg/s

Tabelle 6.2: Leistungsdaten der Triebwerke des Raumtransporters Hopper [106].

Die Transportmission von Hopper beginnt mit der horizontalen Beschleunigung auf Schienen vom Guayana Space Center in Kourou (von wo aus auch die Ariane-5-Rakete startet). Nach [106] reduziert diese Art des Starts die Startmasse, den Schubbedarf und die Startrisiken. Die erste Stufe erreicht bis zum Abschalten der Triebwerke ca. 100 km Höhe und eine suborbitale Geschwindigkeit von etwa 5 km/s. Während des nachfolgenden Freifluges bis zur Gipfelhöhe und dem Zurückfallen in die Atmosphäre muss die Oberstufe mit der Nutzlast freigesetzt werden. Die Oberstufe beschleunigt anschließend zur geostationären Transferbahn. Die suborbitale Unterstufe tritt in die Atmosphäre ein und gleitet etwa 4500 km bis zum Landeort Ascension Island, wo sie horizontal landet. Abbildung 2.4 (S.25) illustriert den Missionsablauf schematisch, wobei auch zwei Missionen zu Zielbahnen höherer Inklination (Sonnensynchrone Umlaufbahn, Raumstation) angedacht sind.

Die nachstehende Tabelle 6.3 gibt eine Beschreibung der Anfangs-, Ziel- und Endbedingungen dieser Transportmission:

<b>Startort: Kourou</b>			
Startmasse	480261 kg	Relativgeschwindigkeit	204,9 m/s
Geografische Länge	-52,799°	Bahnneigung	2,404°
Geografische Breite	5,255°	Anstellwinkel	12,5°
Höhe	26,7 m	Azimut	45°
<b>Zielbahn: GTO</b>			
Perigäumshöhe	180 km	Inklination	7,3°
Apogäumshöhe	35920 km	GTO-Nutzlast	7500 kg
<b>Endbedingungen: TAEM-Schnittstelle</b>			
Entfernung	$\leq 20$ km	Höhe	$\geq 15$ km
Geografische Länge	-14,21°	Machzahl	$\geq 1,5$
Geografische Breite	-8,04°	Fahrzeugmasse	60237 kg
<b>Landeort: Ascension Island</b>			
Geografische Länge	-14.333°	Geografische Breite	-7.9167°

Tabelle 6.3: Anfangs-, Ziel- und Endbedingungen der Transportmission Hopper.

Die Anfangsbedingungen des Aufstiegflugs beschreiben den Zustand von Hopper nach der Beschleunigung auf einer Schiene und dem Abheben von dem Startschlitten. Die Optimierungsberechnungen in dieser Arbeit betrachten die Mission von diesem Anfangszustand bis zur TAEM-Schnittstelle (Terminal Area Energy Management, Endanflug), welche die Anfangsbedingungen Bahnenergie und Position definiert. Diese Bedingungen sind für die Einhaltung der Energie durch Steuerung der Position, Lage, Geschwindigkeit und Flugrichtung durch den Lenkalgorithmus für den automatischen Endanflug und die Landung von Bedeutung.

Aus den Missionsbedingungen und dem Fahrzeugentwurf ergeben sich zahlreiche Flugbeschränkungen. Tabelle 6.4 zeigt eine Zusammenstellung dieser Beschränkungen für die Flugphasen Aufstieg und Wiedereintritt:

<b>Flugbeschränkung</b>	<b>Aufstieg</b>	<b>Wiedereintritt</b>
Fahrzeugmasse		60237 kg
Anstellwinkel	<15° >3° für Höhe < 70km	16° für Ma > 8 16° – 20° für $\gamma > 0$
Bahnneigung		< 1° (leichtes Skipping)
Dynamischer Druck	$\leq 40$ kPa	$\leq 40$ kPa
Auftrieb	$\leq 6600$ kN	
Auftriebsbeschleunigung	$\leq 1,5 g_0$	$\leq 4,5 g_0$ (evtl. 4,8 $g_0$ )
Schubbeschleunigung	$\leq 4,15 g_0$	
Dauer des Freiflugbogens, freie molekulare Aufheizrate	180 s bei $\dot{Q}_{FM} \leq 1340$ W/m <sup>2</sup>	
Staupunktswärmefluss		$\leq 450$ kW/m <sup>2</sup>
Integrale Staupunktswärmelast		$\leq 150$ MJ/m <sup>2</sup>

Tabelle 6.4: Flugbeschränkungen der Hopper-Mission.

## 6.2 Die Rechenmodelle

Die in RTSOPT implementierten Rechenmodelle sind in Kapitel 3.2 erläutert. An dieser Stelle sind die für die folgenden Systemanalysen des Trägersystems Hopper eingesetzten Modelle genannt und spezifischen Erweiterungen beschrieben.

**Die Schwerpunktwanderung:** Infolge des Treibstoffverbrauchs wandert der Schwerpunkt des Trägersystems entlang der Fahrzeuglängs- und -hochachsen. Diese Verschiebungen werden nach [104] mit polynomischen Funktionen des Massenverhältnisses  $m/m_0$  in der Form approximiert:

$$\frac{x_{SP}}{L_{Ref}} = \sum_{i=0}^N \left[ c_i \left( \frac{m}{m_0} \right)^i \right] \quad (6.1)$$

wo  $x_{SP}$  die Schwerpunktsposition entlang der Fahrzeuglängsachse,  $L_{Ref}$  eine Referenzlänge,  $m$  die aktuelle Fahrzeugmasse,  $m_0$  die Startmasse, und  $c_i$  die Korrelationskonstanten sind. Die Schwerpunktsverschiebung entlang der Fahrzeughochachse wird analog berechnet.

Gleiches gilt auch für die Modellierung der Auslenkung  $\varepsilon_{R,0}$  des Schubvektors, die erforderlich ist, um den Schub auf den momentanen Schwerpunkt zu richten [104]. Die Trimmung um den aerodynamischen Bezugspunkt erfordert eine zusätzliche Schubwinkelkorrektur ( $2-7^\circ$ ). Diese wird mit einem Modell nach [104] berechnet, das von der Schwerpunktsverschiebung, dem Auftrieb, Widerstand, Schub und Anstellwinkel abhängt.

**Die Triebwerke:** Die Leistungsdaten der Triebwerke in der Tabelle 6.2 sind die Grundlage für die Triebwerksmodelle. Die Berechnung des Schubes und der Triebwerksverluste basiert auf den Gleichungen (3.15) bis (3.17), die auch die Schubdrosselung berücksichtigen.

**Das Aerodynamikmodell:** Es wird ein Modell eingesetzt, das auf der Interpolation von aerodynamischen Beiwertdaten basiert. Nach der Interpolation dieser Daten in Abhängigkeit von Machzahl und Anstellwinkel mit einer B-Spline-Fläche werden verschiedene Korrekturfunktionen eingesetzt. Die aerodynamischen Beiwertdaten und eine genaue Beschreibung der Korrekturfunktionen sind in [104] gegeben. Das Wärmelastmodell basiert auf der Berechnung des Wärmeflusses am Staupunkt nach einer empirischen Gleichung in der Form (3.13). Die freie molekulare Aufheizrate wird ebenfalls empirisch modelliert (3.14).

**Umweltmodelle:** Es wird eine kugelförmige, rotierende Erde angenommen, bei der die Gravitationsbeschleunigung nur eine Funktion der Flughöhe ist. Die Atmosphäre wird entsprechend der US Standardatmosphäre [74] modelliert.

**Die Flugsteuerung:** Der Flug wird über die Schubdrosselung, den Anstellwinkel und den Flugwindhängewinkel gesteuert. Die Anfangsschätzung für die Optimierung wurde sowohl für die Flugzustände als auch für die Steuerfunktionen durch lineare Interpolation zwischen den geschätzten Anfangs- und Endwerten modelliert.



### 6.3 Aktualisierung der Datenbasis und Anwendungsvergleich der Massenanalyse

Die Massenanalyse in RTSOPT basiert auf einer Massendatenbank, die Subsystemmassen von Luft- und Raumfahrzeugen in Abhängigkeit von Subsystem-Entwurfsparametern enthält. Zu diesen zahlreichen Parametern gehören z.B. Flächen, Längen, Lastvielfache und Triebwerksmassendurchsätze. Durch die Auswahl von Daten aus der Datenbank werden Technologieannahmen getroffen. Anschließend wird eine Regressionsanalyse dieser Daten durchgeführt, aus der sich die Technologiekoeffizienten ergeben. Diese Daten werden dann für die Massenschätzung des betrachteten Systems herangezogen. Außerdem können einzelne, bekannte Subsystemmassen auch vorgegeben werden.

Mit dem Ziel, die Massendatenbank zu aktualisieren und die Massenanalyse mit der vorliegenden RTSOPT-Version zu vergleichen, wurde am IRS eine Untersuchung durchgeführt [102]. Die Teilaufgaben waren wie folgt:

1. Aktualisierung der Massendatenbank
2. Erstellung eines Hopper Fahrzeugmodells mit VRML
3. Massenschätzung für Hopper, Sänger und Ariane-X
4. Vergleich der Ergebnisse mit der alten und neuen Datenbasis

Für die Datenaktualisierung wurde in erster Linie auf die in der FESTIP-Studie erarbeiteten Daten und Prognosen zurückgegriffen. Darüber hinaus wurden zahlreiche Daten aus der Literatur, dem Internet und Broschüren von Herstellerfirmen gesammelt und in die Datenbank eingegeben. Dabei wurden über 300 Quellen für geeignete Daten untersucht. Die Triebwerks-Massendatenbank wurde beispielsweise um 33 Einträge erweitert. In [102] wird auch eine Übersicht von zusätzlichen Datenquellen gegeben, die für die Ergänzung der Datenbank genutzt werden können. Nachfolgend sind die Vorgehensweise und die Ergebnisse dieser Untersuchung dokumentiert.

Entsprechend dem neuen RTSOPT-Softwarekonzept wurde das Fahrzeugmodell von Hopper in VRML erstellt. Dabei kamen die neuen VRML-Prototypen aus der RTSOPT-Modellbibliothek zum Einsatz. Freiformflächen wie z.B. der Rumpf, die Flügel und Ruder wurden mit dem Modellierwerkzeug Cinema4D erstellt und in VRML-Modelle konvertiert. Schließlich wurden diese Modelle zusammengefügt. Die Abbildung 6.2 zeigt zwei Darstellungen dieses Modells:

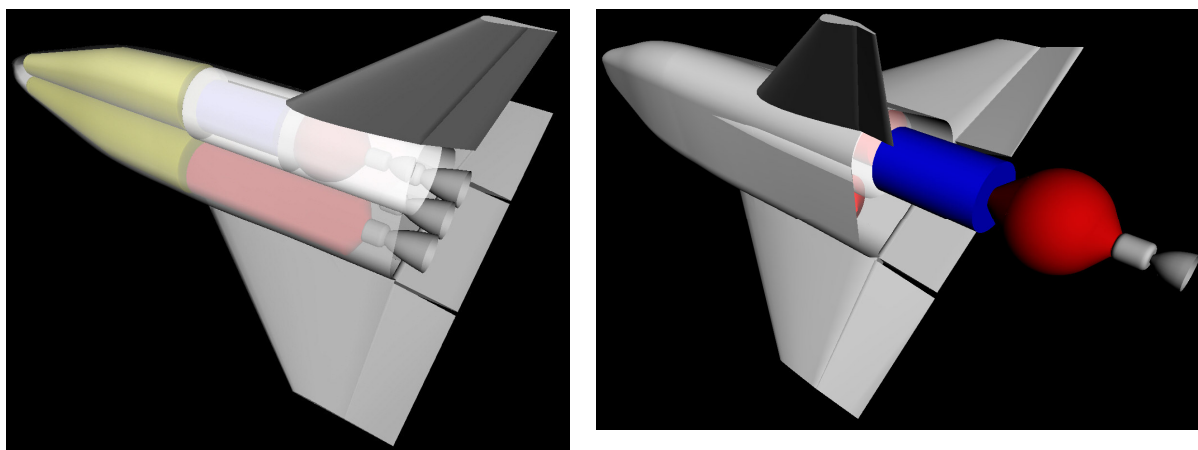


Abbildung 6.2: Das VRML-Modell von Hopper.

In der Abbildung ist der Rumpf durchsichtig dargestellt, um einen Einblick in die inneren Module zu erlauben. Für realistische Darstellungen kann jeder Komponente eine Materialeigenschaft zugeordnet werden. Beispiele hierfür sind bereits früher in der Abbildung 5.3 gegeben. An bewegliche Komponenten (Oberstufe, Ruder) wurden Translations- und Rotationsknoten angeschlossen, so dass eine interaktive Animation dieser Teile möglich ist. Darüber hinaus hat das Modell den Vorteil, dass durch die Eintragung von dreidimensionalen Skalierungsfaktoren für alle Module eine einfache Darstellung von durch die Optimierung skalierten Fahrzeugen möglich ist. In der Abbildung rechts ist die Durchdringung des Rumpfes durch den offensichtlich zu stark vergrößerten Nutzlastcontainer zu beobachten (Darstellung ohne Triebwerke). Mit dieser Vorgehensweise wurde das neue Modellierkonzept von RTSOPT erprobt und seine Eignung und Vorteile für die Systemanalyse demonstriert.

Die für die Massenschätzung des Raumtransporters Hopper benötigten geometrischen Daten, wie z.B. Flächen, Volumina, Massendurchsätze, wurden aus der Literatur entnommen, analytisch berechnet bzw. approximiert [102]. In der früheren RTSOPT-Version [84] wurden diese Daten im CAD-Werkzeug Pro/ENGINEER [81] berechnet. Jedoch hatte diese enge Verbindung von RTSOPT mit dem CAD-System zur Folge, dass diese Verknüpfung nach Aktualisierungen der Software und des Betriebssystems nicht mehr aufrechterhalten werden konnte. Mit der jetzigen Vorgehensweise ist der Einsatz eines beliebigen CAD-Werkzeugs möglich.

Nach der Bestimmung der geometrischen Daten und der Subsystem-Entwurfsparameter wurde eine Regressionsanalyse der Massendatenbank durchgeführt. Dabei wurde die in den Abbildungen Abbildung 2.11 und Abbildung 2.12 dargestellte grafische Benutzeroberfläche von RTSOPT eingesetzt. Mit den gewählten Technologieannahmen und den Systemparametern wurden Modelle der Hopper-Stufen erstellt.

Im dritten Teil dieser Anwendung wurde die Massenschätzung für die folgenden fünf Trägerstufen durchgeführt:

Trägerstufe	Vorgabe (kg)
1. Hopper Hauptstufe	$m_N = 34200$
2. Hopper Oberstufe	$m_0 = 34200$
3. Ariane-X Unterstufe	$m_0 = 368200$
4. Ariane-X Oberstufe	$m_0 = 115000$
5. Sänger Oberstufe	$m_0 = 115000$

Tabelle 6.5: Untersuchte Trägersysteme und Iterationsvorgabe der Massenanalyse [84],[106].

Der Zweck dieser Analysen ist der Vergleich des Massenmodells vor und nach der Aktualisierung der Datenbasis und die Beantwortung der Frage, ob die etwa zehn Jahre früheren Technologieannahmen inzwischen erreicht worden sind. Die Trägersysteme Ariane-X und Sänger wurden in [84] untersucht und für Hopper liegen aktuelle Ergebnisse aus [106] vor, daher bieten sich diese Systeme für einen Anwendungsvergleich bzw. eine Validierung des Modells an. Eine Abhängigkeitsanalyse der Massenschätzung zeigt nach [57], dass die Massen der meisten Trägersubsysteme von der Startmasse des Fahrzeuges abhängen. Die Startmasse ist jedoch die Summe der Subsystemmassen, zu denen auch die Nutzlast gehört. Daher gibt es bei der Massenanalyse nur eine iterative Lösung, für die eine Anfangswertvorgabe erforderlich ist. Folglich muss für die Massenschätzung entweder die Nutzlastmasse  $m_N$  oder die Fahrzeugstartmasse  $m_0$  vorgegeben werden (zweite Spalte der Tabelle 6.5).

## 6 Anwendungsergebnisse

Die Berechnung der Systemmassen wurde zuerst ohne Berücksichtigung der aktualisierten Daten und anschließend mit den aktualisierten Daten durchgeführt. Für die Ariane-X- und Sänger-Stufen wurden die Massen dieser Systeme nur mit der aktualisierten Datenbank abgeschätzt und für den Vergleich auf die Ergebnisse aus [84] zurückgegriffen. Die folgende Abbildung 6.3 stellt die Auswirkungen der Aktualisierung der Massendatenbank auf die verschiedenen Subsysteme grafisch dar:

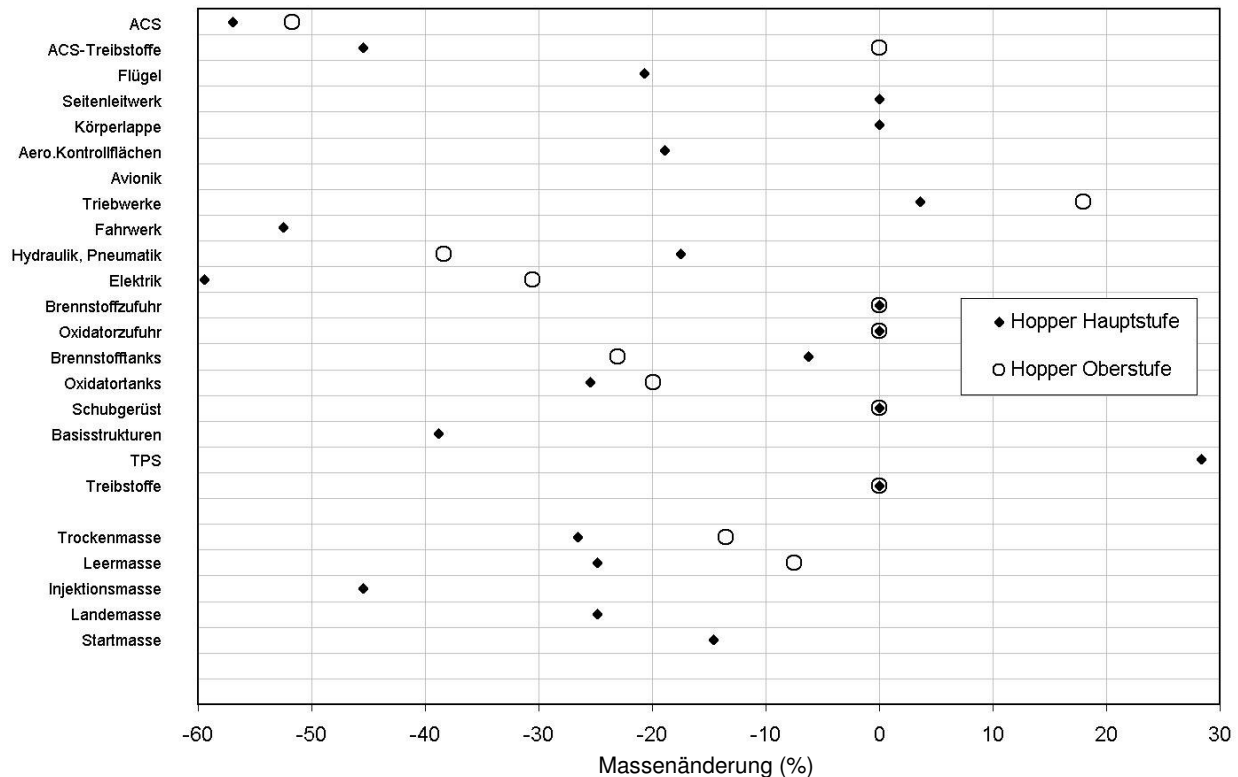


Abbildung 6.3: Massenanalyse für den Raumtransporter Hopper [102].

Die Darstellung zeigt für jedes Subsystem, welche Massenänderung (in %) die Aktualisierung der Massendatenbank mit vorwiegend FESTIP-Daten bewirkt hat. Während die Änderungen bei den meisten Subsystemmassen im Bereich  $\pm 20\%$  liegen, beträgt dieser Wert bei manchen Subsystemen, wie z.B. für Hydraulik, Pneumatik, Treibstoffzufuhrsystem und Elektronik bis zu  $-80\%$  und für die Lageregelung und Fahrwerk etwa  $-50\%$ .

Bei dieser Untersuchung sind die Hitzeschutzschildmassen bei allen Systemen höher ausgefallen als bei der Schätzung mit der alten Datenbank. Dies ist darin begründet, dass die früher angenommene Massenreduktion bei Hitzeschildmaterialien auf  $8 \text{ kg/m}^2$  nicht in dem Maße erreicht wurde. Für niedrige Wärmebelastungen, wie z.B.  $450 \text{ kW/m}^2$  für das suborbitale Hopper, sind jedoch niedrigere Dichten realistisch (Annahme von Astrium:  $5,86 \text{ kg/m}^2$ ).

Die Darstellung der Subsystemmassen von Hopper lässt erkennen, dass die neuen Massendaten insgesamt niedrigere Systemmassen erwarten lassen. Bei der Massenschätzung für Hopper mit der neuen Datenbank wurden höhere Massen lediglich für das Oberstufentriebwerk Vinci und den Hitzeschutzschild berechnet. Die höhere Triebwerksmasse kommt der realen Masse ( $550 \text{ kg}$ ) viel näher, ist jedoch deswegen trotzdem zu niedrig, weil das Modell die Massenerhöhung infolge einer ausfahrbaren Düse nicht berücksichtigt. Im Endergebnis zeigt die berechnete

Startmasse von Hopper (494824 kg) sehr gute Übereinstimmung (0,8% höher) mit der von Astrium berechneten Startmasse von 490700 kg [106].

Eine weitere Berechnung wurde durchgeführt, bei der der Vorteil von verdichteten Treibstoffen (Matsch) untersucht wurde. Dazu wurde die Massenschätzung mit flüssigen Treibstoffen durchgeführt. Dies ergab eine Erhöhung der Leermasse um lediglich 556 kg und der Startmasse um 481 kg. Aus diesem Grund ist es fraglich, ob die aufwändigere Handhabung (Lagerung, Kühlung, Betankung, Isolierung, Enttanking) von verdichteten Treibstoffen sich lohnt. Setzt man allerdings die Technologie dafür voraus und geht von keiner erhöhten Systemmasse (durch Pumpen, Vorheizern, Filter, Isolierung) aus, ist der Vorteil offensichtlich.

Die Ergebnisse lassen für Hopper erkennen, dass durch die Aktualisierung der Datenbank die Massenschätzung für Hopper um etwa 15% leichter ausfällt. In der Abbildung 6.4 sind die Verschiebungen der Subsystemmassen der Träger Ariane-X und Sänger infolge der Aktualisierung der Massendatenbank dargestellt.

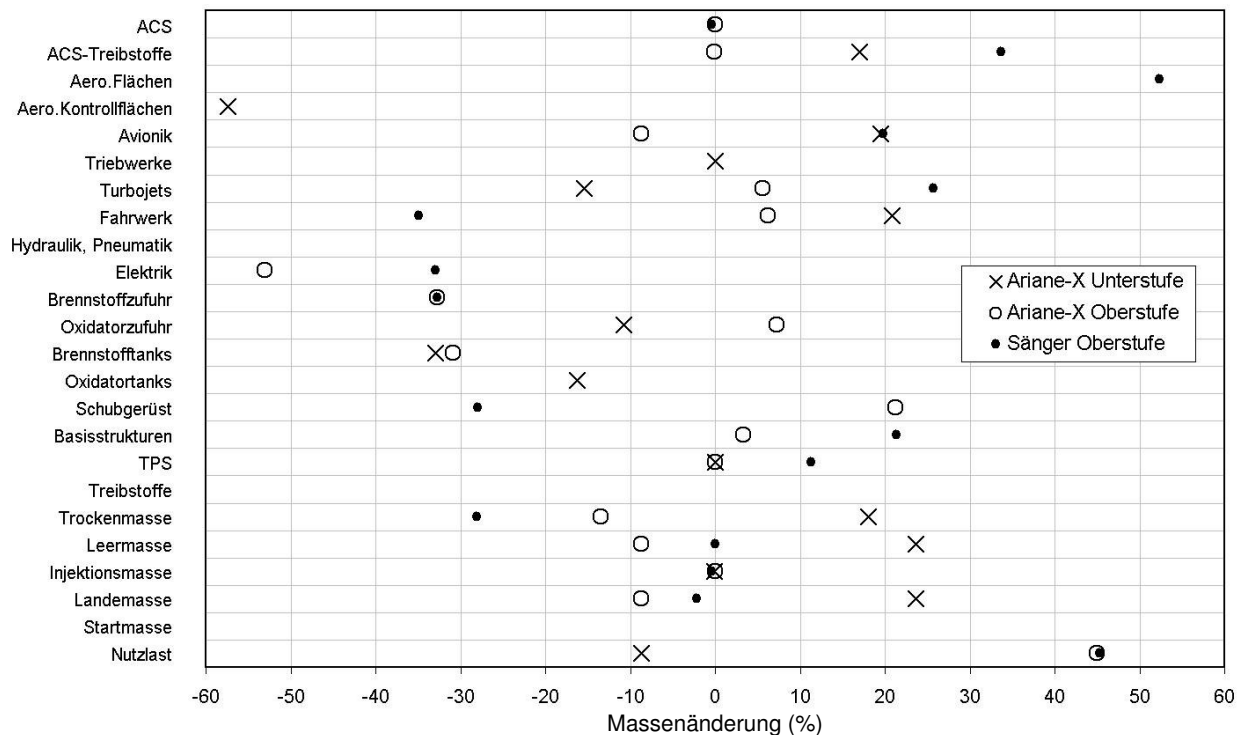


Abbildung 6.4: Massenanalyse für Ariane-X und Sänger Oberstufe.

Die Vergleiche lassen keinen allgemeinen Trend erkennen. Die meisten Massenverschiebungen liegen im Bereich  $\pm 20\%$ . Ein Grund dafür ist, dass die Datenbank aufgrund fehlender neuer Daten für manche Subsysteme nicht erweitert werden konnte. Bei den Oberstufen ist eine Massenreduktion in den elektrischen und Brennstoffzufuhr-Systemen und der Trockenmasse ersichtlich. Betrachtet man die Stufenmassen, hat sich die Masse der Ariane-X-Unterstufe erhöht und die Oberstufenmassen sind niedriger ausgefallen. Die höhere Systemmasse trotz einer Datenaktualisierung bedeutet, dass die früher getroffenen Annahmen zu optimistisch waren. Bei den Oberstufen dagegen ist der positive Einfluss von Entwicklungen in der Leichtbautechnologie sichtbar.

Genaue Aussagen über die Entwicklung von Subsystemmassen können nur für einzelne Subsysteme gemacht werden, für die zahlreiche ältere sowie aktuelle Daten vorliegen. Die Schät-

zung hängt auch sehr stark von den technologischen Annahmen bzw. der Auswahl und der Analyse zugrunde liegenden Daten sowie Definition von Baugruppengrenzen (z.B. Triebwerk / Treibstoffzufuhrsystem / Aufhängung) ab. Eine ausführliche Beschreibung dieser Anwendungsbeispiele und die Ein- und Ausgabedaten sind in [102] gegeben.

Die Ergebnisse der betrachteten Anwendungsbeispiele können wie folgt zusammengefasst werden:

- Für die Untersuchung konnte das aktualisierte RTSOPT-Massenmodell eingesetzt und während der Anwendung entsprechend den Benutzererfahrungen verbessert sowie seine Leistungsfähigkeit demonstriert werden. Darüber hinaus wurden die Algorithmen und Definitionen der Systemparameter genauer spezifiziert und dokumentiert [102], [57].
- Das Fahrzeugmodell von Hopper wurde modulweise mit einem CAD-Programm und mit VRML modelliert, mit Hilfe einer einfachen VRML-Datei assembliert und in räumlichen, interaktiven Animationen von Flugsimulationsergebnissen eingesetzt.
- Die Massendaten aus der FESTIP-Studie repräsentieren fortschrittliche bzw. optimistische Technologien im Vergleich zur früheren RTSOPT-Datenbasis-Version, die Daten ausgeführter bzw. im Detail untersuchter Systeme bis Anfang der 90'er Jahre beinhaltet.
- Mit RTSOPT durchgeführte Massenanalysen produzieren konservative Ergebnisse, die der zur Zeit aktuellen Technologie bzw. etwas pessimistischeren Annahmen entsprechen
- Bis auf einige Differenzen in den Subsystemmassen stimmt die Massenschätzung für Hopper mit den Angaben von Astrium [106] überein. Hier ist eine detaillierte Aufschlüsselung der Massen nach Komponenten nicht möglich, jedoch sind die Differenzen in den Leer-, Injektions- und Landmassen geringer als 6%, und die mit RTSOPT berechnete Startmasse ist nur um 0,8% höher.
- Mit verdichteten Treibstoffen kann die Startmasse um etwa 500 kg reduziert werden. Es stellt sich jedoch die Frage, ob dieser Vorteil durch den anspruchsvolleren und bisher nicht demonstrierten Betrieb der Treibstoffe in diesem Zustand wieder verloren geht.

### 6.4 Leistungsvergleich von Optimierungsverfahren

Für einen Leistungsvergleich zwischen verschiedenen Optimierungsverfahren wurden in der durch das IRS und das Institut für Flugmechanik und Flugregelung der Universität Stuttgart gemeinsam betreuten Diplomarbeit [120] der Aufstieg und der Rückflug des Raumtransporters Hopper modelliert und unter Verwendung von GESOP [49], [50] die Nutzlast zur Geotransferbahn optimiert. Vergleichsrechnungen wurden für die Aufstiegsphase mit 120 bzw. 480 und für die gesamte Mission (Aufstieg und Rückflug) mit 840 Optimierungsparametern durchgeführt. Dabei wurden die Beschränkungen der Wärmelast und Querbeschleunigung aufgegeben, weswegen die Nutzlastmassen hoch ausgefallen sind. Da hier der Leistungsvergleich verschiedener Optimierungsverfahren bei der Lösung gleicher Probleme im Vordergrund steht, wird auf die detaillierte Darstellung der Optimierungsaufgaben verzichtet. Die Ergebnisse dieser Untersuchung zeigen die Überlegenheit der für RTSOPT gewählten Optimierungs-Software SOCS gegenüber anderen aktuellen Verfahren.

GESOP bietet eine Auswahl von Optimierungsverfahren<sup>3</sup>:

1. PROMIS (Parameterized Trajectory Optimization by Direct Multiple Shooting), ein Mehrfach-Schiessverfahren [93]
2. TROPIC (Trajectory Optimization by Direct Collocation), ein Kollokationsverfahren [54]
3. SOCS (Sparse Optimal Control Software) [10], vgl. Kap. 4

Die folgende Tabelle 6.6 gibt eine Übersicht der Resultate:

Anzahl der Parameter	Mission	Optimierer	Rechenzeit (s)	Nutzlast (kg)
120	Aufstieg	PROMIS	14	7917
		TROPIC	10	7918
		SOCS	14	7958
480	Aufstieg	PROMIS	63	7917
		TROPIC	51	7919
		SOCS	37	7931
840	Aufstieg und Rückflug	PROMIS	240 (Abbruch)	7912
		TROPIC	181	7919
		SOCS	61	7931

Tabelle 6.6: Optimierervergleich mit unterschiedlicher Parameterzahl [120].

Es zeigt sich, dass SOCS bei hoher Anzahl von Optimierungsparametern schneller konvergiert als die anderen verwendeten Verfahren. Die Abbildung 6.5 stellt diese Ergebnisse grafisch dar. Die Kurven interpolieren die Daten:

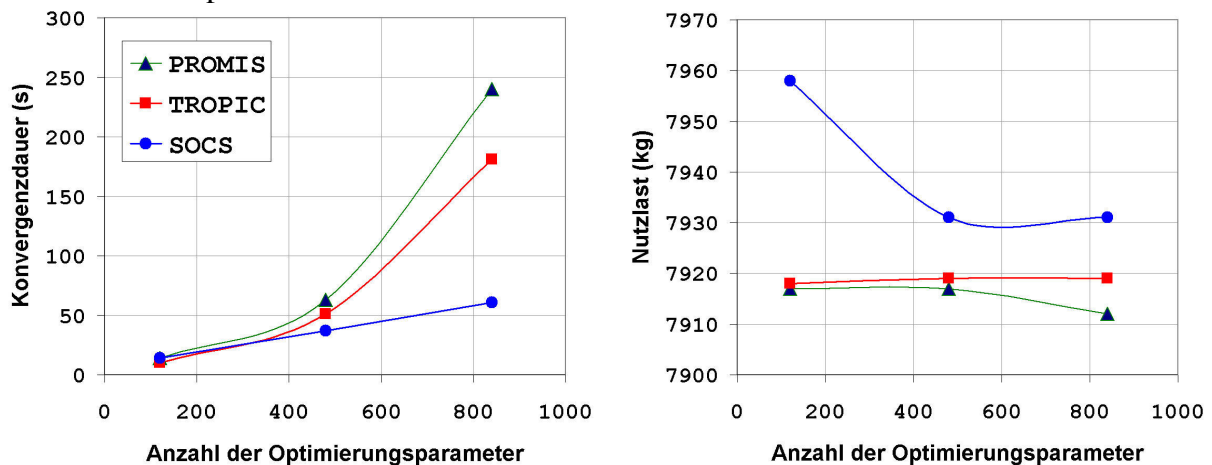


Abbildung 6.5: Konvergenzrate, Nutzlast und Anzahl der Optimierungsparameter.

Aus der Grafik wird ersichtlich, dass die Konvergenzrate des Optimierungsverfahrens SOCS eine viel schwächere Sensitivität auf die Zahl der Optimierungsparameter aufweist als die beiden anderen Verfahren. Darüber hinaus erzielt SOCS ein geringfügig besseres Ergebnis, bei der die maximale Nutzlast 10-30 kg höher liegt. Die Abnahme der Nutzlast im letzten Fall ist auf die Berücksichtigung des Rückkehrflugs zurück zu führen.

<sup>3</sup> Ab Version 4.5.3 auch CAMTOS (Collocation and Multiple Shooting Trajectory Optimization Software)

## 6.5 Optimierung der Aufstiegsbahn

Die Aufstiegsphase der Hopper-Mission wurde mit RTSOPT optimiert und die Ergebnisse der Optimierungsverfahren SOCS und NLPQL verglichen. Ziel der Optimierung ist die Minimierung des Aufstiegstreibstoffes. In der nachstehenden Tabelle 6.7 sind die Beschränkungen für die Flugzustände, -lasten und Steuergrößen, die das Optimierungsproblem definieren, zusammengefasst. Die Geschwindigkeit, Höhe und Bahninklination im Scheitelpunkt der Aufstiegsbahn werden entsprechend der nominellen Bahn nach [104] vorgegeben.

Zustandsbeschränkungen			Flugbeschränkungen
Anfangswert	Zustand	Endwert	Flügelast $\leq 1,5$ g
480 Mg	Masse	$> 60$ Mg	Axiale Beschleunigung $\leq 4,15$ g
204,9 m/s	Geschwindigkeit	$= 5000$ m/s	Dynamischer Druck $\leq 40$ kN/m <sup>2</sup>
(-52,8°,5,26°)	Länge, Breite	Frei	
26,7 m	Höhe	$= 150$ km	Steuerbeschränkungen
2,404°	Bahnneigung	Frei	$-10^\circ \leq$ Anstellwinkel $\leq 40^\circ$
45° N-O	Azimut	Frei	$-90^\circ \leq$ Flugwindhängewinkel $\leq 90^\circ$
	Bahninklination	$= 7,3^\circ$	$90\% \leq$ Schubdrosselung $\leq 100\%$

Tabelle 6.7: Randbedingungen und Steuerbeschränkungen [104].

Die Beschränkung der axialen Beschleunigung ist durch das Drosseln der drei Triebwerke um jeweils maximal 10% nicht einhaltbar. Daher wird der Aufstiegsflug in drei Flugphasen geteilt. In der ersten Flugphase werden alle drei Triebwerke betrieben, in der zweiten brennen nur zwei und in der dritten Flugphase sind alle Triebwerke abgeschaltet. In den ersten zwei Phasen können alle Triebwerke um 10% gedrosselt werden. Die Dauer der Flugphasen sind nicht beschränkt und werden damit automatisch als Optimierungsparameter definiert und optimiert. In früheren Versionen von RTSOPT waren für die Definition dieses Problems Programmcode-Anpassungen, Entwicklung und Implementierung von Regelalgorithmen zur Einhaltung von Fluglastbeschränkungen notwendig. Bei der in dieser Arbeit verbesserten RTSOPT-Version gestaltete sich die Problemdefinition sehr einfach, da diese sich aus der Eingabe der o.g. Rand- und Flugbeschränkungen ergibt. Damit konnte auch die Effektivität und Flexibilität der neuen Benutzerschnittstelle verbessert werden.

Die nachstehenden Abbildungen Abbildung 6.6 und Abbildung 6.7 stellen die Bodenspur und den Höhen- bzw. Geschwindigkeitsverlauf der optimierten Aufstiegsbahn dar.

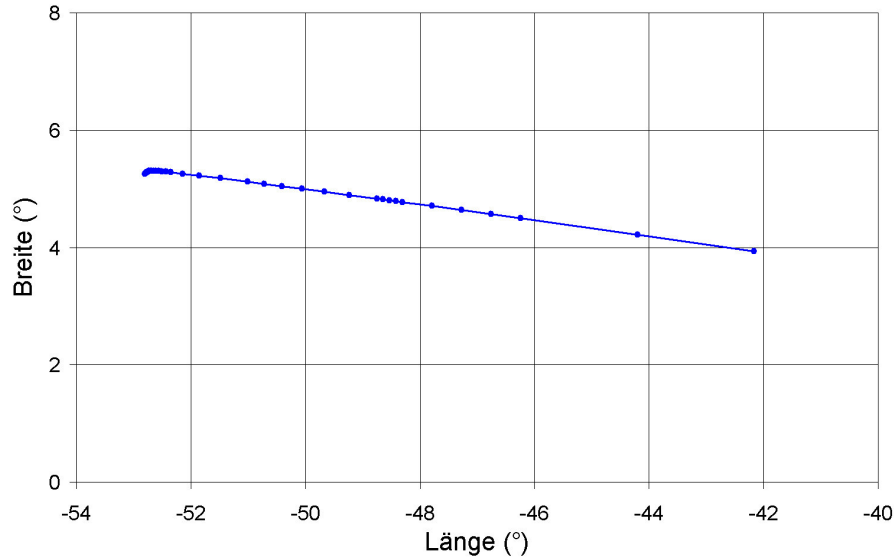


Abbildung 6.6: Die Bodenspur des Aufstiegsflugs.

Der Startazimut von  $45^\circ$  N-O und die geforderte Zielinklination von  $7,3^\circ$  machen einen markanten Kurvenflug am Anfang der Aufstiegsphase erforderlich, um die Flugrichtung anzupassen.

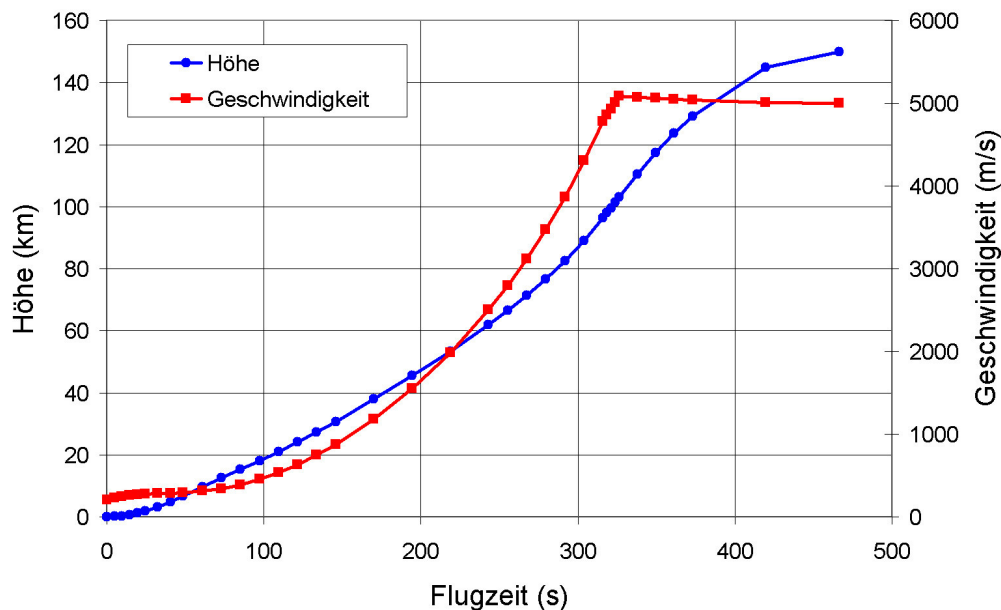


Abbildung 6.7: Höhen- und Geschwindigkeitsverlauf der Aufstiegsbahn.

Hopper steigt angetrieben bis zu einer Höhe von 100 km. Nach dem Abschalten der Triebwerke erfolgt ein ballistischer Aufstiegsflug. Am Ende des Steigflugs werden die geforderten Endgeschwindigkeit (5 km/s) und -höhe (150 km) im Rahmen der Rechnergenauigkeit erreicht. Die folgenden Abbildungen Abbildung 6.8 und Abbildung 6.9 zeigen den Verlauf der Bahnneigung, des Azimut und der optimalen Steuerung.



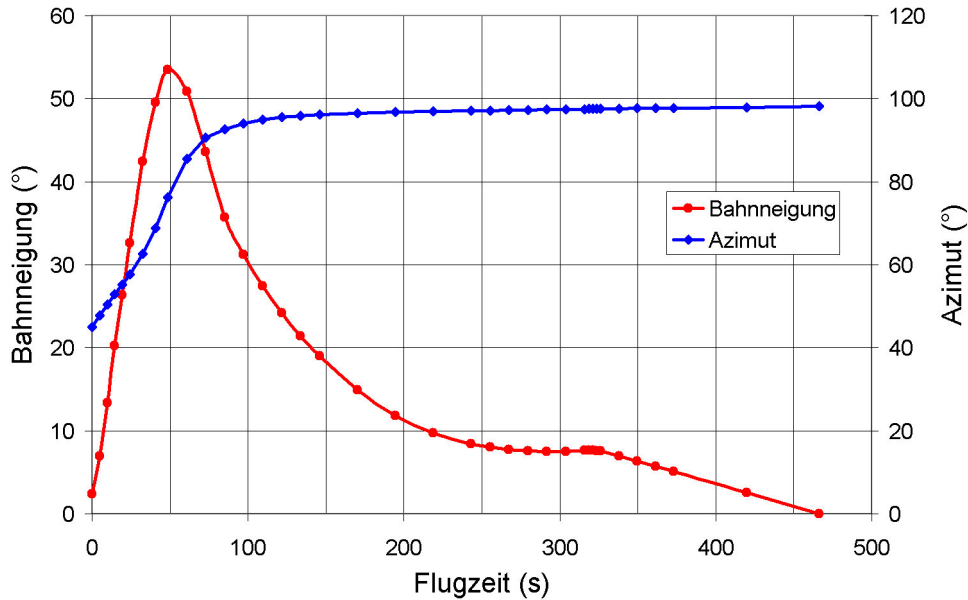


Abbildung 6.8: Verlauf von Bahnneigung und Azimut der optimalen Aufstiegsbahn.

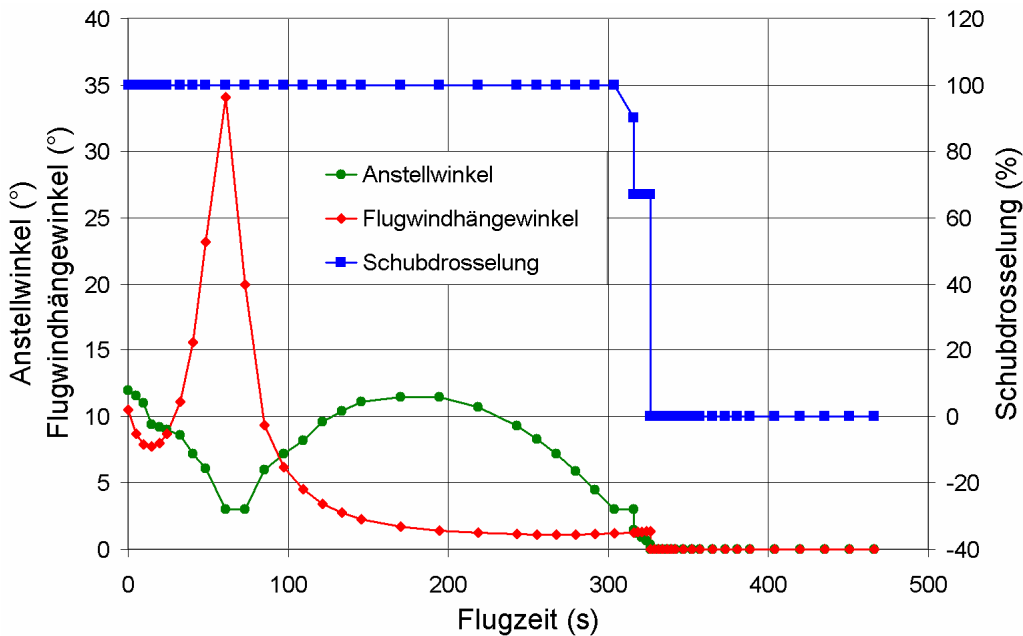


Abbildung 6.9: Verlauf der optimalen Flugsteuerung.

Die Markierungen in den Abbildungen Abbildung 6.8 und Abbildung 6.9 zeigen die Diskretisierung des Parameter-Optimierungsproblems. Durch die gewählte Unterteilung der Flugsteuerung ergibt sich ein Optimierungsproblem mit 117 Parametern. Die kontinuierliche Modellierung der Zustands- und Steuerverläufe erfolgt in kubischer B-Spline-Darstellung. In den Abbildungen sind die Punkte mit Geraden verbunden.

Abbildung 6.9 zeigt die optimale Steuerung des Anstell- und Flugwindhängewinkes und der Schubdrosselung. Um die maximal zulässige axiale Beschleunigung von 4,15 g nicht zu überschreiten, werden nach 304 Sekunden die Triebwerke in 12 Sekunden graduell auf 90% gedrosselt. Danach ist eine weitere Verringerung des Schubes nur noch durch die Abschaltung eines Triebwerkes möglich, die bei  $t = 329$  s erfolgt. Danach laufen die zwei übrigen Triebwerke mit vollem Schub bis zur Abschaltung zum Zeitpunkt  $t = 326$  s. Daran schließt sich eine ballistische Flugphase an, die bis zur Erfüllung der Zielbedingungen der Mission bei  $t = 466$  s dauert. In der Abbildung ist der deutliche Anstieg des Flugwindhängewinkels bis zu  $35^\circ$  in der Kurvenflugphase zu erkennen. Um genügend Auftrieb bei niedrigen Geschwindigkeiten zu erzeugen, werden Anstellwinkel von  $3^\circ$ - $12^\circ$  gesteuert. Im Kurvenflug wird mit niedrigem Anstellwinkel und hohem Flugwindhängewinkel geflogen. Die Anstellwinkelbeschränkung von minimal  $3^\circ$  (vgl. Tabelle 6.4) wird einmal im Bereich bei  $t = 60$  s und einmal bei  $303$  s aktiv und wird exakt eingehalten. Diese Beschränkung entfällt nach der Abschaltung der Triebwerke.

Die folgende Abbildung 6.10 stellt den Verlauf des Flügellastfaktors und der Anstellwinkelsteuerung der optimalen Lösung grafisch dar.

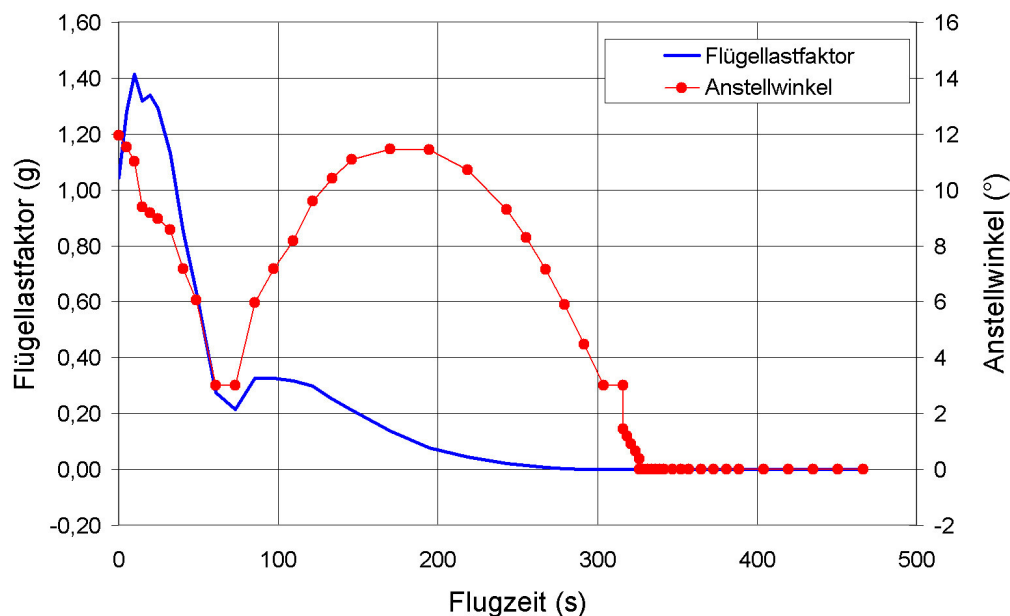


Abbildung 6.10: Verlauf des Flügellastfaktors und Anstellwinkelsteuerung.

Der Verlauf des Flügellastfaktors für die optimierte Aufstiegsbahn zeigt nach Abbildung 6.10, dass die Flügellast über die gesamte Flugzeit unter der maximalen Grenze von 1,5 g für das Lastvielfache liegt. Aus der Abbildung ist ferner ersichtlich, dass die höchsten Flügellastwerte in den ersten 25 s des Fluges auftreten, wo das Fahrzeug noch aufgrund der vollen Treibstoff-tanks sehr schwer ist.

Die nächste Abbildung 6.11 illustriert den Verlauf der zugehörigen Geschwindigkeit und des dynamischen Drucks während der ersten 100 s des Aufstiegsflugs.

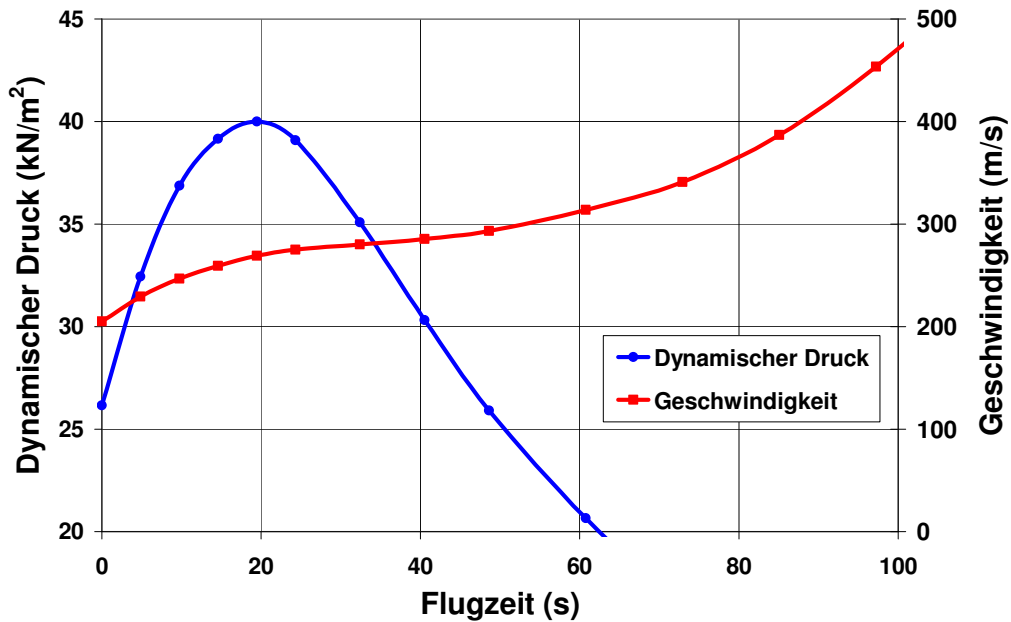


Abbildung 6.11: Verlauf der Geschwindigkeit und des dynamischen Drucks.

Der maximal zulässige dynamische Druck beträgt  $40 \text{ kN/m}^2$ . Man erkennt, dass dieser Wert durch eine Reduktion der Beschleunigung über die Wahl hoher Anstellwinkel eingehalten wird, die den Widerstand erhöhen. Die Verminderung der Beschleunigung spiegelt sich in der Abflachung der Geschwindigkeitskurve in den ersten 40 Sekunden des Fluges wider.

Für die Bahnoptimierung mit dem Löser SOCS wurden lineare, konstante und wenig realistische Anfangsschätzungen für die Steuerverläufe benutzt. Abbildung 6.12 betrachtet einen Vergleich zwischen der Anfangsschätzung (Benutzereingabe), der Anfangslösung (erste Iteration) und der optimalen Lösung. Die Anfangslösung erfüllt erfahrungsgemäß die Bewegungsgleichungen mit etwa 1% oder schlechter und die Flugbeschränkungen mit etwa 10% Genauigkeit.

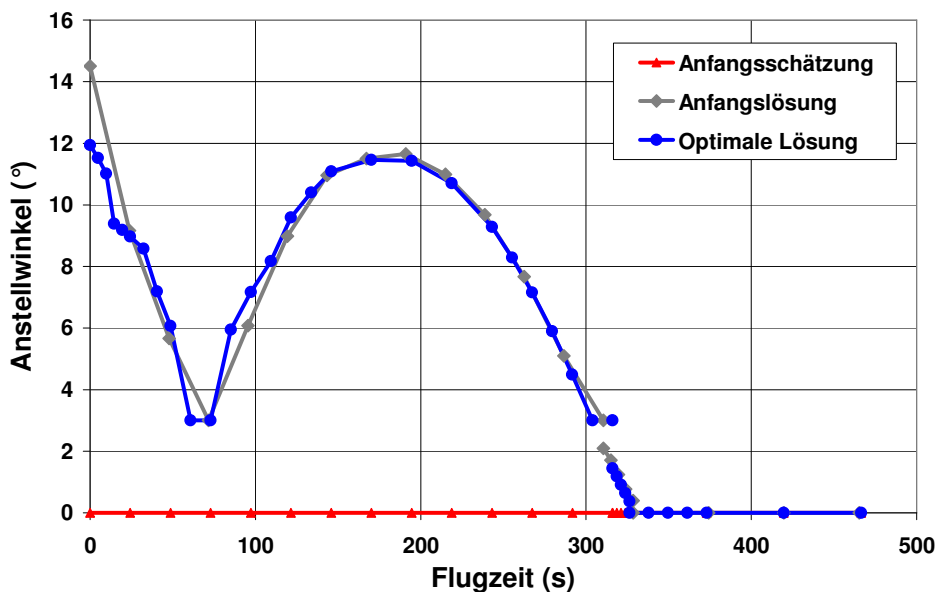


Abbildung 6.12: Anfangsschätzung und optimale Lösung mit SOCS.

Die Markierungen in der Abbildung stellen die Gitterpunkte dar (Parameter des Optimierungsproblems). In den Anfangsschätzungen wurden vier bis 14 Gitterpunkte verwendet. Die Anfangslösung konnte mit SOCS automatisch generiert werden. Der Vergleich mit der optimalen Lösung verdeutlicht, wie nah die Anfangslösung zur optimalen Lösung bereits ist. Aus der Abbildung ist ersichtlich, dass die optimale Lösung mehr Gitterpunkte beinhaltet (höhere Zeitauflösung) als die Anfangsschätzung. In Abhängigkeit von der Problemkomplexität und Anzahl der Gitterpunkte in der Anfangsschätzung, führt SOCS mehrere Gitterverfeinerungen durch. In Flugbereichen, wo die Sensitivität auf die Flugsteuerfunktionen hoch ist bzw. wo entlang Fluglastbeschränkungen geflogen werden muss, werden zusätzliche Gitterpunkte eingefügt. Im vorliegenden Fall wurde die Anzahl der Gitterpunkte der ersten Flugphase verdoppelt. Eine gute Demonstration dieses Merkmals des Optimierungsverfahrens ist auch in der Darstellung der Bodenspur in der Abbildung 6.6 zu sehen. Die Zeitauflösung der Steuerfunktionen wurde dort in der sehr steuerungssensitiven Kurvenflugphase sogar verdreifacht.

Die Tabelle 6.8 dokumentiert die bei der Lösung des Optimierungsproblems mit SOCS erzielte Konvergenzleistung.

Toleranz der Differentialgleichungen*	Anzahl der Gitterverfeinerungen	Anzahl der Parameter	Rechenzeit (s)	Ergebnis: Endmasse (%)
$1 \times 10^{-3}$	2	117	11	100.00
$1 \times 10^{-2}$	1	117	9	100.20
Anfangslösung	0	57	4	100.89

\* Zielfunktion-Toleranz =  $10^{-5}$ , Beschränkungs-Toleranz =  $1,49 \times 10^{-8}$ , kein Einfluss auf die Rechenzeit dieser Parameter beobachtet.

Tabelle 6.8: Konvergenzverhalten des Lösers SOCS.

Für einen direkten Vergleich mit der NLPQL-Lösung wurde das Optimierungsproblem mit gleichen Modellen, Annahmen und Beschränkungen auf demselben Rechner (2,4 GHz Pentium-IV-PC unter Windows) gelöst. Abbildung 6.13 zeigt für das Programm NLPQL einen Vergleich zwischen der Anfangsschätzung und der optimalen Lösung.

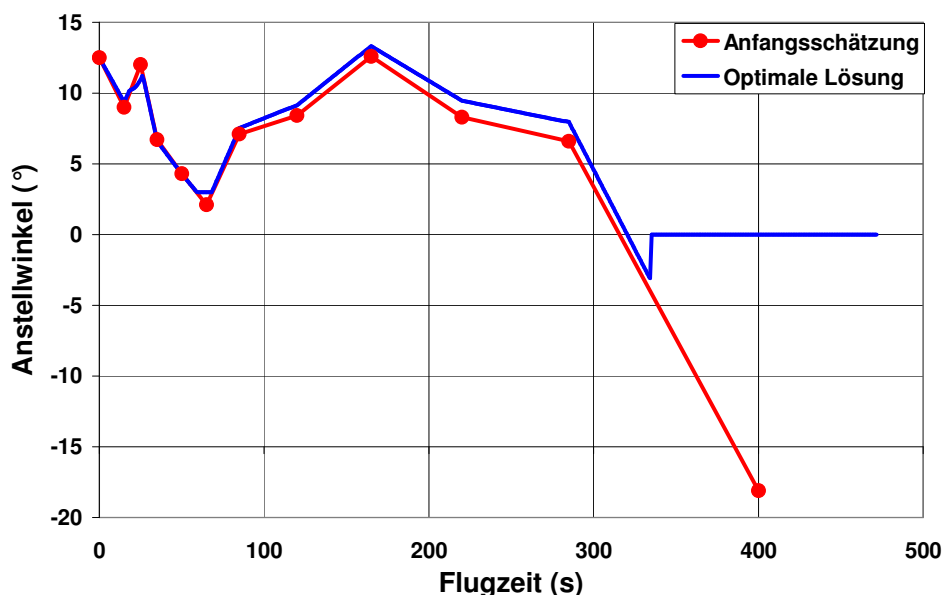


Abbildung 6.13: Anfangsschätzung und optimale Lösung mit NLPQL.

Aus der Abbildung ist ersichtlich, dass die Anfangsschätzung und die optimale Lösung sehr nahe beieinander liegen, d.h. die Anfangsparameter der Lösung schon recht nahe kommen. Die Generierung solcher Anfangsschätzungen ist aufwändig und erfordert eine Reihe von Sensitivitätsanalysen und Simulationen sowie die Anpassung der zeitlichen Diskretisierung entsprechend den Beobachtungen. Um den Konvergenzbereich des Verfahrens NLPQL zu untersuchen, wurde die Anfangsschätzung durch Skalierung degradiert. Dabei wurde eine Steigerung der Konvergenzzeit beobachtet. Für einen Skalierungsfaktor von 0,4 verdoppelt sich die Rechenzeit und für Skalierungsfaktoren unter 0,4 konnte keine Konvergenz mehr erzielt werden. Die Degradierung der Anfangsschätzung führte außerdem auch zur Verschlechterung des Optimierungsergebnisses, die sich in der geringeren Nutzlastmasse widerspiegelt. Die Abbildung 6.14 stellt diese Zusammenhänge grafisch dar.

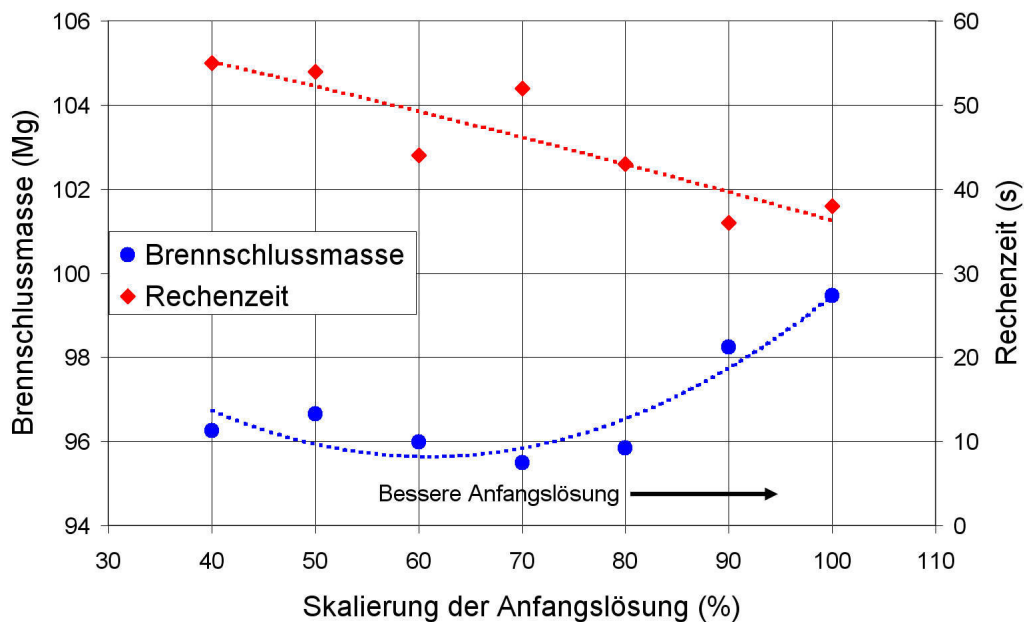


Abbildung 6.14: Konvergenzverhalten des Optimierers NLPQL.

Aus der Abbildung ist ersichtlich, dass trotz der Verschlechterung der Anfangslösung Konvergenz erzielt werden kann, dass aber sowohl das Ergebnis als auch die Rechenzeit darunter leiden. Allerdings kann der Konvergenzbereich durch verbesserte Anfangslösungen und durch richtige Problemskalierung vergrößert werden, aber dies ist mit Aufwand verbunden [96], [111].

Für die Lösung mit NLPQL waren außerdem die Implementierung von Regelalgorithmen erforderlich, um zwischen den Optimierungsparametern, die in der Abbildung 6.13 markiert sind, die Flugbeschränkungen einhalten zu können.

Die Optimierung mit NLPQL konvergierte mit 16 Optimierungsparametern in 38 s wobei mit dem Optimierer SOCS Konvergenz mit 117 Optimierungsparametern in 11 s erzielt wurde. Damit war die Optimierung mit SOCS mit siebenfacher Anzahl von Optimierungsparameter etwa vier mal schneller als mit NLPQL. Kleinere geforderte Toleranzen bei SOCS erhöhten die Anzahl der Gitterverfeinerungen, die die Rechenzeit ohne Verbesserung des Ergebnisses steigerte.

Die Ergebnisse demonstrieren die signifikante Leistungssteigerung von RTSOPT durch Verwendung des Optimierungsverfahrens SOCS. Damit konnte nicht nur eine vielfache Reduzierung der Konvergenzzeit erzielt, sondern auch die sehr schnelle, automatische Generierung einer Anfangslösung (im untersuchten Fall in vier Sekunden) realisiert werden. Damit entfällt dieser aufwändige Arbeitsschritt in zukünftigen Untersuchungen. Darüber hinaus wurde infolge einer höheren Zeitauflösung ein besseres Ergebnis erzielt als mit dem bisher benutzten Verfahren NLPQL. So konnte mit der SOCS-Lösung eine Nutzlaststeigerung gegenüber der NLPQL-Lösung von ca. 7% erreicht werden.

## 6.6 Optimierung des Flügellastfaktors

Das Programm RTSOPT wurde unter Verwendung des Optimierers SOCS beispielhaft eingesetzt, um den Auslegungsparameter maximaler Flügellastfaktor  $n_{F.Max}$  und die Aufstiegsbahn von Hopper simultan zu optimieren. Ziel der Optimierung ist die Maximierung der Nutzlast der ersten Stufe, die aus dem Satelliten, den Treibstoffen sowie der Struktur der Oberstufe besteht. Dazu wird bei konstanter Startmasse die Endmasse der Grundstufe minimiert. Der Flügellastfaktor  $n_F$  ist folgenderweise definiert:

$$n_F(t) = \frac{A(t)}{G(t)} \leq n_{F.Max} \quad (6.2)$$

wobei  $A$  die Auftriebskraft und  $G$  das aktuelle Fahrzeuggewicht bezeichnen. Um die Zunahme der Flügelmasse in Abhängigkeit vom maximalen Flügellastfaktor zu berücksichtigen, wurde mit RTSOPT ein Massenmodell entsprechend der Beschreibung in Kap. 3.2 erstellt. Die Flügelmasse wird mit der folgenden Korrelationsgleichung berechnet:

$$m_F = a \cdot \left( \frac{m_0 \cdot n_F \cdot S_S \cdot A_{F.Prj}}{t_W} \right)^b \cdot 10^{-6} \quad (6.3)$$

wobei  $m_F$  die Flügelmasse,  $m_0$  die Startmasse des Fahrzeuges,  $S_S$  die strukturelle Spannweite,  $A_{F.Prj}$  die projizierte Flügelfläche und  $t_W$  die Flügelwurzelstärke bezeichnen. Die Korrelationskoeffizienten  $a$  und  $b$  wurden in [102] für  $S_S=13,84$  m,  $t_W=2,09$  m und  $A_{F.Prj}=403,15$  m<sup>2</sup> statistisch ermittelt ( $a=0,0096$ ,  $b=0,6161$ ). Die nachstehende Abbildung 6.15 betrachtet das mit den Korrelationen aus Kap. 3.2 definierte Flügelmassenmodell.

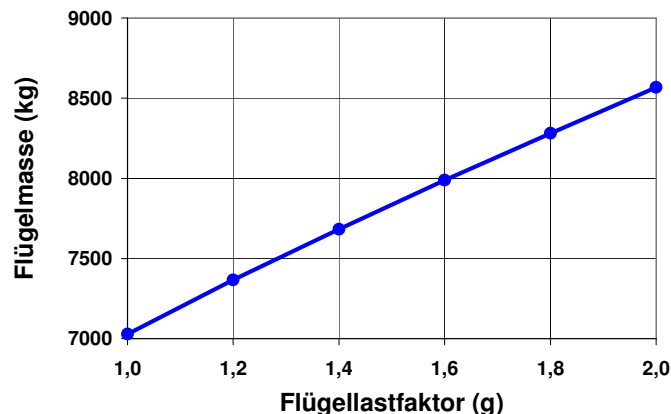


Abbildung 6.15: Flügelmasse des Raumtransporters Hopper in Abhängigkeit vom Flügellastfaktor.

Für diese Optimierungsberechnung gelten die gleichen Anfangs- und Endbedingungen der Mission und die Steuerungs- bzw. Fluglastbeschränkungen wie in Kapitel 6.5. Zusätzlich zu der Optimierung der Flugsteuerung (Schubdrosselung, Anstell- und Flugwindhängewinkel, Abschaltzeiten der Triebwerke) wird der Flügellastfaktor in dem Bereich 1-2 g optimiert.

Die Bodenspur in der Abbildung 6.6 lässt eine Reduzierung der Geschwindigkeitsverluste bei höheren Flügellasten erwarten, weil dann die Kurvenflugphase mit kleinerem Radius durchgeführt werden kann. Jedoch zeigt die simultane System- und Bahnoptimierung mit RTSOPT, dass der optimale Wert für die maximale Flügellast 1,23 g ist und damit niedriger ist als ursprünglich vorgesehen (1,5 g). Dies bedeutet dass der Flügel leichter ausgeführt werden kann und insgesamt dadurch Transportleistungsverbesserungen für den Raumtransporter Hopper erzielt werden können. Diese Resultate wurden mit einer Sensitivitätsstudie untersucht. Abbildung 6.16 illustriert die Ergebnisse dieser Analyse.

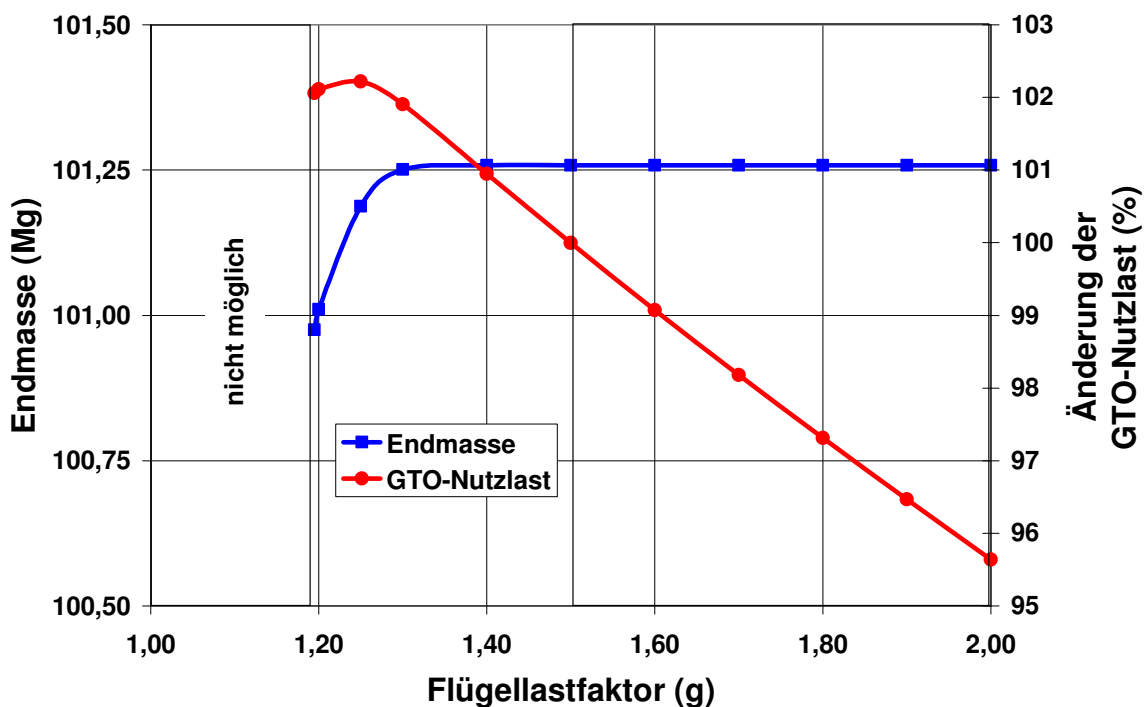


Abbildung 6.16: Endmasse und Nutzlast des Raumtransporters Hopper in Abhängigkeit vom Flügellastfaktor.

Jede Markierung in der Abbildung entspricht einer Bahnoptimierung für einen vorgegebenen Flügelentwurf. Aus der Abbildung ist ersichtlich, dass zulässige Flügelbelastungen über etwa 1,5 g keine Steigerung der Flugleistungen bringen und dass die GTO-Nutzlastkapazität dabei linear abnimmt. Dies liegt daran, dass für höhere Belastungen ausgelegte Flügel schwerer sind und damit die besseren Flugleistungen durch die Flügelmasse wieder verloren gehen. Auf der anderen Seite wird bei Flügellastfaktoren unter 1,2 g der Auftrieb so stark beschränkt, dass der Aufstiegsflug nicht mehr möglich ist. In dem Bereich 1,2 - 1,5 g kann eine optimale Lösung gefunden werden, bei der ein Kompromiss zwischen den Flugleistungen und der Flügelmasse zur maximalen GTO-Nutzlast führt. Dies ist nur möglich, wenn die hochgradig nichtlinear gekoppelten Flugbahn- und System-Einflüsse simultan berücksichtigt werden.

Diese Systemanalyse wurde fortgeführt, indem zunächst die Tankgröße dem Treibstoffbedarf angepasst und schließlich auch der Rumpf entsprechend skaliert wurde. Die folgende Abbildung 6.17 stellt diese Ergebnisse grafisch dar.

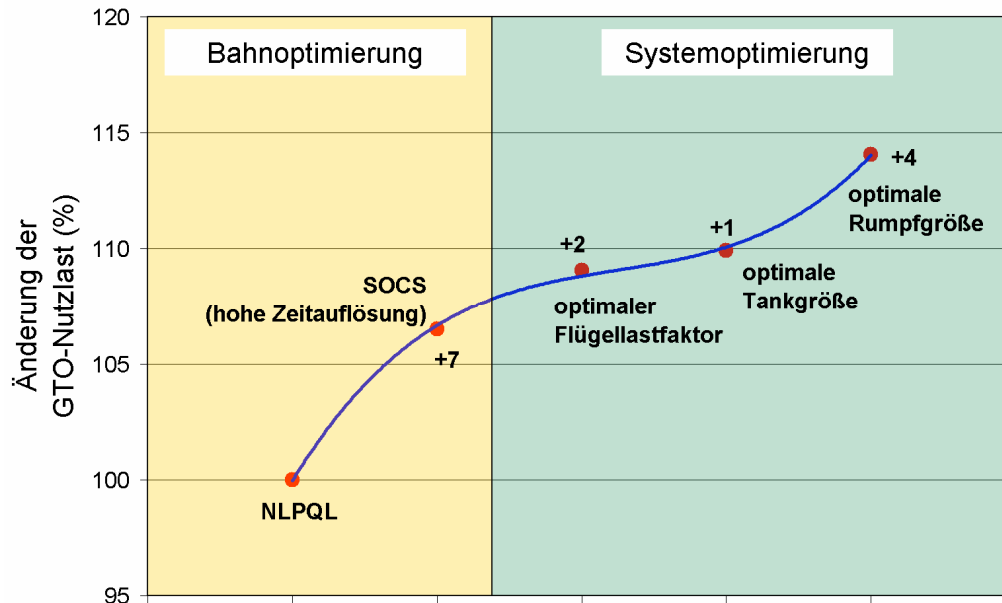


Abbildung 6.17: Nutzlaststeigerung durch die Bahn- und Systemoptimierung.

Wie aus der Grafik ersichtlich, wurde durch die Verwendung des leistungsfähigeren Optimierers SOCS, bei simultaner Optimierung des Flügелentwurfs und entsprechender Skalierung der Tank- und Rumpfgöße mit anschließender Massenanalyse für die betrachtete Hopper-Aufstiegsmission, eine Nutzlaststeigerung von etwa 14% erzielt werden. Dies untermauert den Stellenwert einer simultanen Gesamtsystembetrachtung bzw. -optimierung.

Da hier nur die Aufstiegsmission betrachtet wurde und die Beschränkungen des Aufstiegflugs durch den anschließenden Rückkehrflug nicht berücksichtigt wurden, fällt die Nutzlaststeigerung besonders hoch.

Nachfolgend ist der Flugverlauf des Trägersystems mit optimalem Flügелentwurf dokumentiert. Höhen- und Geschwindigkeitsverlauf der Bahn unterscheiden sich sehr geringfügig von denen der in Kapitel 6.5 ohne Flügeloptimierung berechneten Bahn (vgl. Abbildung 6.7). Ein sehr genauer Vergleich zeigt erwartungsgemäß, dass der Höhenverlauf mit höheren Flügellasten steiler ist als mit niedrigeren. Abbildung 6.18 zeigt die Fluglasten und den optimierten Verlauf der Flugsteuerung.



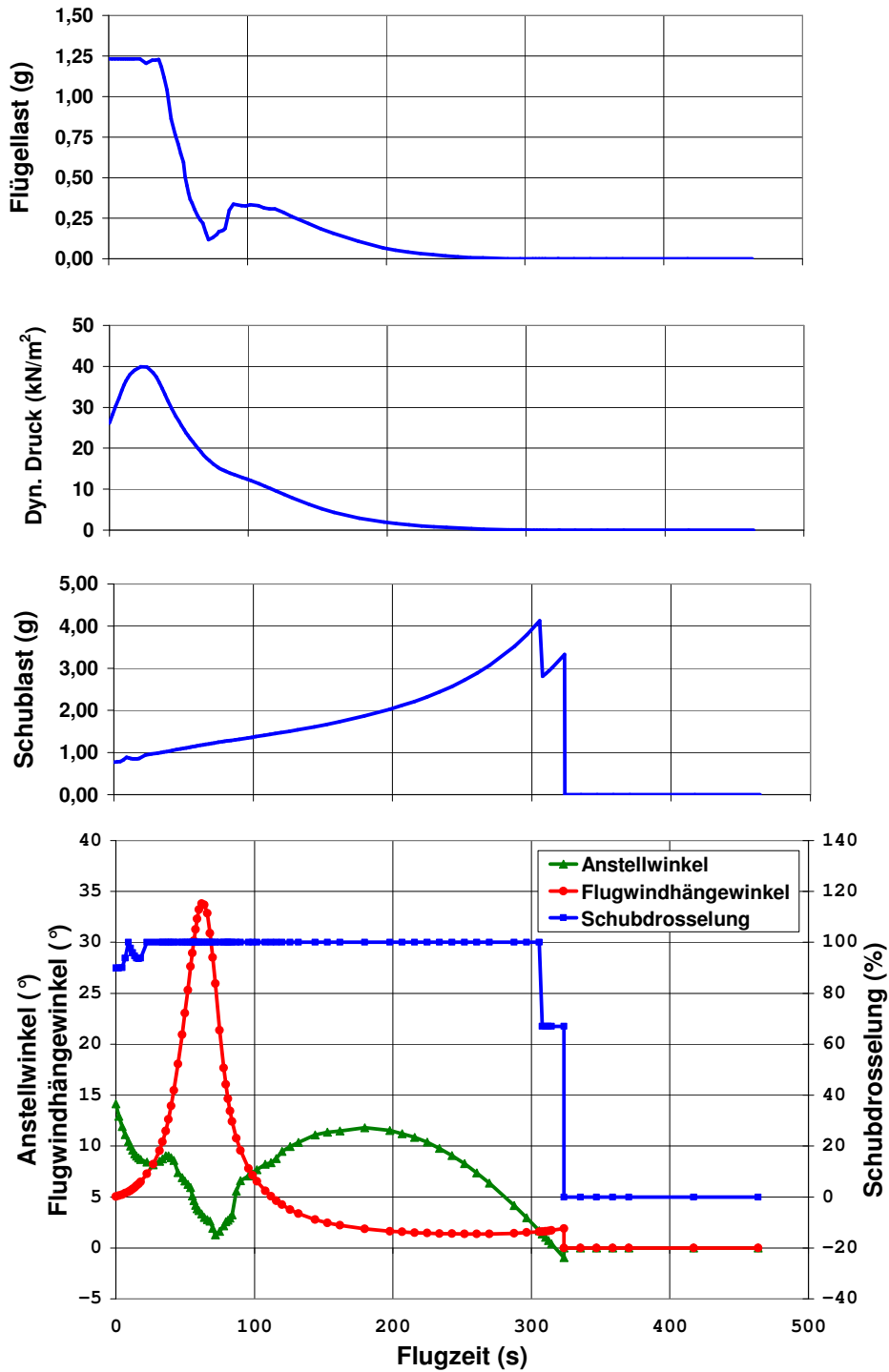


Abbildung 6.18: Fluglasten und die optimierte Flugsteuerung.

Die Darstellung lässt wie zuvor die deutliche Erhöhung der Zeitauflösung durch sukzessive Gitterverfeinerungen erkennen. Dies gilt vor allem in den ersten 100 Sekunden des Fluges, wo entlang der Flügellastbeschränkung von 1,23 g geflogen wird und gleichzeitig der dynamische Druck gegen den maximalen Wert von  $40 \text{ kN/m}^2$  steigt. Jedoch werden beide Beschränkungen durch die genaue Flugsteuerung eingehalten. In der Abbildung ist ferner zu sehen, dass bei etwa 25 s der maximale dynamische Druck erreicht wird. Gleichzeitig wird sowohl der Schub

gedrosselt als auch der Anstellwinkel reduziert, wodurch eine kurze Verminderung der Flügelast erreicht wird und beide Beschränkungen eingehalten werden. Ferner ist beim Vergleich der Schubdrosselung und Schublast eindeutig zu sehen, dass beim Erreichen der maximal zulässigen Schublast von 4,15 g eine neue Flugphase eingeleitet wird, in der nur noch zwei Raketentriebwerke betrieben werden.

Die vorliegenden Optimierungsergebnisse wurden nach mehreren automatischen Gitterverfeinerungen mit 240 Parametern erzielt. Dabei wurden die gleichen, linearen und konstanten Anfangsschätzungen für die Steuergrößen eingesetzt. In der Tabelle 6.9 werden die mit SOCS erzielten Konvergenzraten ohne bzw. mit Einbeziehung der Systemoptimierung verglichen.

Optimierungsaufgabe	Bahnoptimierung	Simultane Bahn- und Systemoptimierung
Anzahl der Optimierungsparameter	117 (100 %)	240 (205 %)
Rechenzeit	11 s (100 %)	15 s (136 %)

Tabelle 6.9: Konvergenzrate ohne bzw. mit Einbeziehung der Systemoptimierung.

Der Löser SOCS konvergierte mit mehr als doppelt so vielen Optimierungsparametern in nur 36% längerer Rechenzeit und zeigte die erwartete schwache Abhängigkeit der Rechenzeit von der Anzahl der Optimierungsparameter. Dabei wurde bei dem zweiten Optimierungsproblem zusätzlich zur Flugsteuerung auch der maximale Flügellastfaktor, ein Systementwurfsparameter, optimiert und die Änderung der Strukturmasse durch die iterative Auswertung des Hopper-Massenmodells berücksichtigt. Eine signifikante Erhöhung der Rechenzeit aufgrund des zusätzlichen zeitunabhängigen Optimierungsparameters wird nicht beobachtet.

Die Ergebnisse der Untersuchungen zeigen eine signifikante Leistungssteigerung durch die Implementierung von SOCS im Analysewerkzeug RTSOPT, wobei sowohl eine verbesserte Flexibilität der Anwendung als auch deutlich kürzere Rechenzeiten erzielt werden.

## 6.7 Optimierung der Gesamtmission Hopper

In diesem Abschnitt wird die Optimierungsaufgabe auf die Gesamtmission Hopper ausgedehnt, die zusätzlich zum Trägeraufstieg den direkt anschließenden Wiedereintritts- und Rückkehrflug zu einem Ziellandeplatz beinhaltet. Ziel der Optimierungsaufgabe ist die Maximierung der Nutzlast zur geostationären Transferbahn. Dabei gelten die in der Tabelle 6.3 dokumentierten Anfangs-, Ziel- und Endbedingungen der Mission und die in der Tabelle 6.4 angegebenen Flugbeschränkungen. Der Flug wird nach dem Start in Kourou bis zur TAEM-Schnittstelle (Terminal Area Energy Management) in der Nähe des Landeplatzes Ascension Islands im Atlantik betrachtet. Das Fahrzeug und die Transportmission sind bereits in Kapitel 6.1 und die in der Simulation verwendeten Rechenmodelle in Kapitel 6.2 beschrieben.

### 6.7.1 Formulierung des Optimierungsproblems

#### Randbedingungen und Beschränkungen

Für die Definition des Optimierungsproblems wird der Flug in vier Flugphasen mit unterschiedlichen Rechenmodellen und Beschränkungen eingeteilt:

1. Aufstieg bis zur Abschaltung eines der drei Triebwerke
2. Weiterer Aufstiegsflug mit zwei Raketentriebwerken
3. Freiflugphase bis zum Scheitelpunkt der Aufstiegsbahn und Aussetzen der Nutzlast
4. Wiedereintritt und Rückkehrflug bis zur TAEM-Schnittstelle

Die Verwendung des Programms SOCS erfordert die Spezifizierung der folgenden Beschränkungstypen [11]:

Randbeschränkungen, die am Anfang (Anfangsbeschränkungen; Initial Constraints) und am Ende (Endbeschränkungen; Final Constraints) einer Flugphase einzuhalten sind, werden durch untere und obere Grenzwerte (Lower/Upper Bounds) festgelegt. Wenn die unteren und oberen Grenzen gleich sind, ergibt sich daraus eine Gleichheitsbedingung. Beispiele dafür sind die Bahninklination, Distanz zu einem Ziel und Steiggeschwindigkeit am Ende einer Flugphase. Die Anfangsbedingungen für eine Flugphase können auch mit den Endbedingungen der vorangehenden Flugphase verknüpft werden. Diese Bedingungen werden durch RTSOPT automatisch generiert und dadurch die Kontinuität der Flugzustände über die Phasengrenzen sichergestellt. Wenn die Übergangskonditionen und –zeit nicht beschränkt sind, werden sie in die Optimierung einbezogen. Diese Möglichkeit wurde benutzt, um die Abschaltzeit eines Triebwerks zu optimieren, welches beim Erreichen der oberen Schubbeschleunigungsgrenze abgeschaltet wird. Die beschränkte Triebwerksdrosselung (maximal um 10%) macht die Abschaltung eines Triebwerks erforderlich, da sonst die maximale Schubbeschleunigung überschritten wird.

Pfadbeschränkungen (Path Constraints) gelten für die gesamte Dauer einer Flugphase. Sie werden verwendet, um Belastungsgrenzen wie etwa den Staupunktswärmefluss, die Auftriebsbeschleunigung etc., die von Flugzustandsvariablen abgeleitet werden, zu berücksichtigen.

Die Zustandsbeschränkungen (State Constraints) und die Steuerbeschränkungen (Control Function Constraints) beinhalten die Anfangs-, Pfad- und Endbeschränkungen für die zeitlichen Verläufe der Zustandsvariablen bzw. der Flugsteuerungen. Alle Parameter, die keiner Gleichungsrestriktion unterworfen sind, wie z.B. Flugphasendauer, Anstellwinkel, Endgeschwindigkeit und –höhe, werden automatisch als Optimierungsparameter definiert und in die Optimierung einbezogen.

Die Tabelle 6.10 gibt eine Übersicht der Beschränkungen für die gesamte Mission entsprechend der SOCS-Nomenklatur auf. Die Phasenverknüpfungsbedingungen sind nicht gesondert aufgeführt. Die Optimierungsparameter sind mit dunklerem Hintergrund hervorgehoben. Leere Zellen weisen darauf hin, dass die jeweilige Größe für die Optimierung frei ist.

<b>Flugphase 1: Aufstieg</b>			
<b>Zustand</b>	Anfang	Pfad	Ende
Relativgeschwindigkeit (m/s)	204,9		
Geographische Länge (°)	-52,799		
Geographische Breite (°)	5,255		
Flughöhe (m)	26,7		
Bahnneigung (°)	2,404		
Azimut (°)	45		
Fahrzeugmasse (kg)	480261		
<b>Pfadbeschränkung</b>		Obergrenze	
Schubbeschleunigung (m/s <sup>2</sup> )		$\leq 4,15 g_0$	
<b>Steuerfunktion</b>	Anfang	Pfad	Ende
Anstellwinkel (°)	12,5	3 – 40	
Flugwindhängewinkel (°)		-90 – 90	
Schubdrosselung (%)		90 – 100	
<b>Flugphase 2: Aufstieg mit einem abgeschalteten Triebwerk</b>			
<b>Pfadbeschränkung</b>		Obergrenze	
Schubbeschleunigung (m/s <sup>2</sup> )		$\leq 4,15 g_0$	
<b>Steuerfunktion</b>	Anfang	Pfad	Ende
Anstellwinkel (°)		0 – 40	
Flugwindhängewinkel (°)		-90 – 90	
Schubdrosselung (%)		60 – 66,7	
<b>Flugphase 3: Freiflug bis zum Scheitelpunkt der Aufstiegsbahn</b>			
<b>Zustand</b>	Anfang	Pfad	Ende
Bahnneigung (°)			0
Flughöhe (km)			140-150
Relativgeschwindigkeit (m/s)			4500 – 6000
<b>Flugphase 4: Vom Scheitelpunkt bis zur TAEM-Schnittstelle</b>			
<b>Zustand</b>	Anfang	Pfad	Ende
Flughöhe (km)			$\geq 15$
Fahrzeugmasse (kg)		60237	
<b>Pfadbeschränkung</b>		Obergrenze	
Wärmestrom (kW/m <sup>2</sup> )		450	
<b>Endbeschränkung</b>		Obergrenze	
Bodenstrecke zum Ziel (km)			$\leq 20$
Machzahl			$\geq 1,5$
<b>Steuerfunktion</b>		Pfad	
Anstellwinkel (°)		0 – 40	
Flugwindhängewinkel (°)		-90 – 90	

Tabelle 6.10: Problemformulierung für die Gesamtflugoptimierung für Hopper.

Die Dauer der einzelnen Flugphasen sind nicht beschränkt und sind damit Optimierungsparameter. Die optimale Dauer für die erste Flugphase ergibt sich aus den Beschränkungen für die Triebwerksdrosselung und die maximale Schubbeschleunigung. Sobald diese Beschränkungen nicht gleichzeitig erfüllt werden können, endet diese Flugphase und die zweite Flugphase beginnt, in der ein Triebwerk abgeschaltet ist.

Da das Aerodynamikmodell nur Daten für einen Anstellwinkelbereich von 0 bis 40° enthält, ist die Anstellwinkelsteuerung in der gesamten Optimierung entsprechend beschränkt.

Ausgewählte Flugbeschränkungen aus der Tabelle 6.4 sind in der folgenden Tabelle 6.11 zur leichteren Bezugnahme nochmals aufgeführt und für den Aufstiegsflug bzw. Rückflug mit A1 bis A4 bzw. W1 bis W4 bezeichnet:

Flugbeschränkung		Aufstieg		Wiedereintritt
Anstellwinkel	A1	<15°	W1	>16° für Ma > 8
	A2	>3° für Höhe < 70km	W2	16° – 20° für $\gamma > 0$
Bahnneigung			W3	< 1° (leichtes Skipping)
Auftrieb	A3	≤ 6600 kN		
Auftriebsbeschleunigung			W4	≤ 4,5 g <sub>0</sub> (evtl. 4,8 g <sub>0</sub> )
Dauer des Freiflugbogens, freie molekulare Aufheizrate	A4	180 s bei $\dot{Q}_{FMHR} \leq 1340 \text{ W/m}^2$	W5	$\dot{Q} \leq 450 \text{ kW/m}^2$

Tabelle 6.11: Flugbeschränkungen für die Optimierung.

Die Beschränkungen A1, A3, W1, W4 und W5 wurden ohne Definition von Optimierungsbeschränkungen erfüllt. A2 wird durch die Beschränkung in der gesamten ersten Flugphase erfüllt. Für Höhen über 70 km wird diese Beschränkung nicht aktiv. Gibt man diese Beschränkung auf, erhöht sich die GTO-Nutzlast minimal um 5 kg. A4 wird durch die Beschränkung der Scheitelpunkthöhe erfüllt. Diese Höhe wurde durch eine Reihe von Optimierungen ermittelt. W2 wurde frei optimiert, der optimale Anstellwinkel liegt zwischen 13 und 13,5°. Mit der Beschränkung W3 konnte keine Lösung gefunden werden und musste deswegen aufgegeben werden. Nach [104] ist leichtes Skipping erlaubt und der maximale Wert in der optimalen Lösung beträgt etwa 1,8°. Die Forderung der Bahninklination von 7,3° im Scheitelpunkt kann ebenfalls nicht erfüllt werden. Daher wurde diese Bedingung aufgegeben und eine notwendige Inklinationsänderung durch die Oberstufe vorgesehen.

In der zweiten Aufstiegsflugphase, in der eines von drei Raketentriebwerken abgeschaltet ist, wird die Schubdrosselung im Bereich 60 – 66,7 % beschränkt. Dies entspricht 2/3 des verfügbaren Schubes mit drei Triebwerken bei 90 – 100 % Schubdrosselung. Damit ist eine kontinuierliche Problemformulierung möglich und diese Vorgehensweise erfordert keine Implementierung spezieller Beschränkungen, missionsspezifischer Änderungen im Programmcode und zusätzliche Flugphasen für die Abschaltung des Triebwerks.

## Annahmen

Bei der Optimierung wird angenommen, dass beide Stufen soviel Treibstoff mitführen, wie für das Erreichen der Zielbedingungen benötigt wird. Die Oberstufenstrukturmasse wird mit einer linearen Beziehung nach [104] aus dem Treibstoffbedarf für die Bahnmanöver berechnet. Der Treibstoffbedarf ergibt sich aus der notwendigen Inklinationsänderung und dem Übergang zur Geotransferbahn.

Für die Initialisierung der Optimierung werden Anfangslösungen für die Zustandsvariablen und die Steuerfunktionen benötigt. Diese werden für alle Gitterpunkte und Flugphasen eingegeben. Hierfür wurden für drei Flugphasen Anfangsgitter mit jeweils 10 Punkten gewählt, die über die Flugzeit gleich verteilt sind. Für die Freiflugsphase wurden fünf Gitterpunkte definiert. Die Anfangsschätzungen der Zustandsvariablen an den Gitterpunkten wurden durch lineare Interpolation zwischen den geschätzten Anfangs- und Endwerten berechnet.

## Die Zielfunktion

Ziel der mehrphasigen Optimierung ist die Maximierung der Nutzlast zur Geotransferbahn. Dazu wird am Ende der Flugphase drei (am Scheitelpunkt) die Nutzlast analytisch berechnet. Dabei werden die durch die Oberstufe zu leistende Inklinationskorrektur und der Geschwindigkeitsbedarf für den Bahntransfer vom Scheitelpunkt zum GTO berücksichtigt. Diese Berechnung ist nachfolgend beschrieben.

### 6.7.2 Analytische Berechnung der Nutzlastmasse

Da die aerodynamischen Kräfte während des Oberstufenfluges vernachlässigbar klein sind, wird der Oberstufenflug analytisch berechnet. Die notwendigen Berechnungsgrundlagen können beispielsweise [72] oder [73] entnommen werden. Im Einzelnen läuft die Berechnung der Nutzlast im GTO wie folgt ab:

Es wird angenommen, dass der Transfer der Oberstufe vom Bahnscheitelpunkt der Hopper-Grundstufe (Index SP) zur Geotransferbahn durch zwei Bahnmanöver erfolgt.

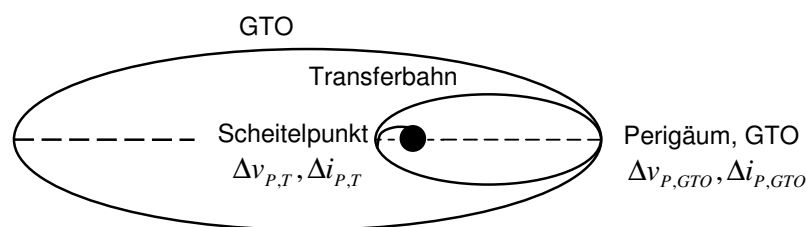


Abbildung 6.19: Der Transfer vom Scheitelpunkt zum GTO.

Dazu muss die Oberstufe zunächst im Scheitelpunkt bzw. im Perigäum der Transferbahn eine Inklinationsänderung und Beschleunigung leisten ( $\Delta v_{P,T}, \Delta i_{P,T}$ ) und anschließend im Apogäum der Transferbahn entsprechend dem Perigäum der geostationären Transferbahn GTO das restliche  $\Delta v$ -Geschwindigkeitsinkrement  $\Delta v_{P,GTO} + \Delta i_{P,GTO}$  aufbringen.

Zunächst wird die tangentielle Rotationsgeschwindigkeit der Erde am Scheitelpunkt berechnet:

$$v_{E,SP} = r_{SP} \omega_E \cos \delta \quad (6.4)$$

wobei  $r_{SP}$ ,  $\omega_E$  und  $\delta$  der Abstand vom Erdmittelpunkt, die Winkelgeschwindigkeit der Erde und der Breitengrad sind. Da die Bewegungsgleichungen in der Optimierung im Relativsystem (Index r) aufgestellt sind, muss der Zustand im Scheitelpunkt in das Inertialsystem (Index i) transformiert werden:

$$v_{i,SP} = \sqrt{v_{r,SP}^2 + v_{E,SP}^2 + 2v_{r,SP}v_E \sin \chi_{r,SP} \cos \gamma_{r,SP}} \quad (6.5)$$

$$\gamma_{i,SP} = \arcsin\left(\frac{v_{r,SP} \cdot \sin(\gamma_{r,SP})}{v_{r,SP}}\right) \quad (6.6)$$

$$\chi_{i,SP} = \arcsin\left(\frac{v_{r,SP} \cdot \sin(\chi_{r,SP}) \cdot \cos(\gamma_{r,SP}) + r_{SP} \cdot \omega_E \cdot \cos(\delta_{SP})}{v_{i,SP} \cdot \cos(\gamma_{i,SP})}\right) \quad (6.7)$$

$$i_{i,SP} = \arccos(\cos(\delta_{SP}) \cdot \sin(\chi_{i,SP})) \quad (6.8)$$

Die hier dokumentierten Berechnungen sind für eine inertielle Zielbahninklination von  $7,3^\circ$  gültig. Strebt man eine relative Bahninklination von  $7,3^\circ$  an, reduziert sich die Nutzlast um etwa 100 kg. Dies liegt darin, dass die Inklination im Relativsystem, aufgrund der Erdrotation, einen größeren Betrag hat und deswegen eine größere Inklinationsänderung benötigt wird. Die folgenden Berechnungen werden alle im geozentrischen Inertialsystem durchgeführt. Daher wird der Index i weggelassen. Aus der Vis-Viva-Gleichung ergibt sich die Geschwindigkeit im Perigäum der Transferbahn:

$$v_{P,T} = \sqrt{\mu_E \left( \frac{2}{r_{SP}} - \frac{2}{r_{SP} + r_{P,GTO}} \right)} \quad (6.9)$$

Im Perigäum der Transferbahn wird 65% der gesamten, zum Erreichen der Zielinklination von  $i_{Ziel} = 7,3^\circ$  erforderlichen Inklinationsänderung vorgenommen:

$$\Delta i_{P,T} = 0,65 \cdot (i_{Ziel} - i_{SP}) \quad (6.10)$$

Die Zahl 0,65 ergibt sich aus der Näherungsformel für die optimale Aufteilung der Inklinationsänderung nach [111]:

$$\left. \frac{\Delta i_{P,T}}{\Delta i_{HUS}} \right|_{Opt} = \left( \frac{v_{P,T} v_{i,SP} (v_{P,GTO} - v_{A,T})}{v_{A,T} v_{P,GTO} (v_{P,T} - v_{i,SP})} + 1 \right)^{-1} \quad (6.11)$$

deren Gültigkeit für den untersuchten Fall mit einer Sensitivitätsanalyse überprüft wurde.

Damit kann der Geschwindigkeitsbedarf wie folgt berechnet werden:

$$\Delta v_{P,T} = \sqrt{v_{P,T}^2 + v_{i,SP}^2 - 2v_{P,T}v_{i,SP} \cos \Delta i_{P,T}} \quad (6.12)$$

Mit der Vis-Viva-Gleichung werden anschließend die Bahngeschwindigkeiten am Apogäum der Transferbahn und am Perigäum des GTO ermittelt:

$$v_{A,T} = \sqrt{\mu_E \left( \frac{2}{r_{P,GTO}} - \frac{2}{r_{SP} + r_{P,GTO}} \right)} \quad (6.13)$$

$$v_{P,GTO} = \sqrt{\mu_E \left( \frac{2}{r_{P,GTO}} - \frac{2}{r_{P,GTO} + r_{A,GTO}} \right)} \quad (6.14)$$

Schließlich wird der Geschwindigkeitsbedarf für das zweite Bahnmanöver mit der restlichen Inklinationskorrektur zum GTO mit  $7,3^\circ$  Inklination bestimmt.

$$\Delta v_{P,GTO} = \sqrt{v_{A,T}^2 + v_{P,GTO}^2 - 2v_{A,T}v_{P,GTO} \cos \Delta i_{A,T}} \quad (6.15)$$

Die Startmasse der Hopper-Oberstufe wird aus der Differenz der Gesamtmasse bei Abschaltung der Triebwerke und der Leermasse der Grundstufe berechnet. Die erforderliche Treibstoffmasse  $m_{Tr,HUS}$  der Oberstufe ergibt sich aus der Ziolkowski-Gleichung:

$$m_{0,HUS} = m_{MECO} - m_{Net} \quad (6.16)$$

$$m_{Tr,HUS} = m_{0,HUS} \cdot \left[ 1 - e^{-\frac{\Delta v}{c_{e,HUS}}} \right] \quad (6.17)$$

Die Abhängigkeit der Oberstufenettomasse von der Treibstoffmasse wird nach [104] mit einer linearen Beziehung erfasst und schließlich die Nutzlastmasse ermittelt:

$$m_{Net,HUS} = 2780 + 0,04834 \cdot m_{Tr,HUS} \quad (6.18)$$

$$m_N = m_{0,HUS} - m_{Tr,HUS} - m_{Net,HUS} \quad (6.19)$$

### 6.7.3 Exemplarischer Flugverlauf

Die Abbildungen Abbildung 6.20 bis Abbildung 6.22 stellen die Ergebnisse der Optimierungsberechnungen für die gesamte Hopper-Mission grafisch dar. Dabei zeigen Abbildung 6.20 die Bodenspur, Abbildung 6.21 die Höhen-, Geschwindigkeits-, Inklinations-, Azimut- und Bahnneigungsverläufe und die Abbildung 6.22 die Fluglasten sowie die optimale Steuerung.



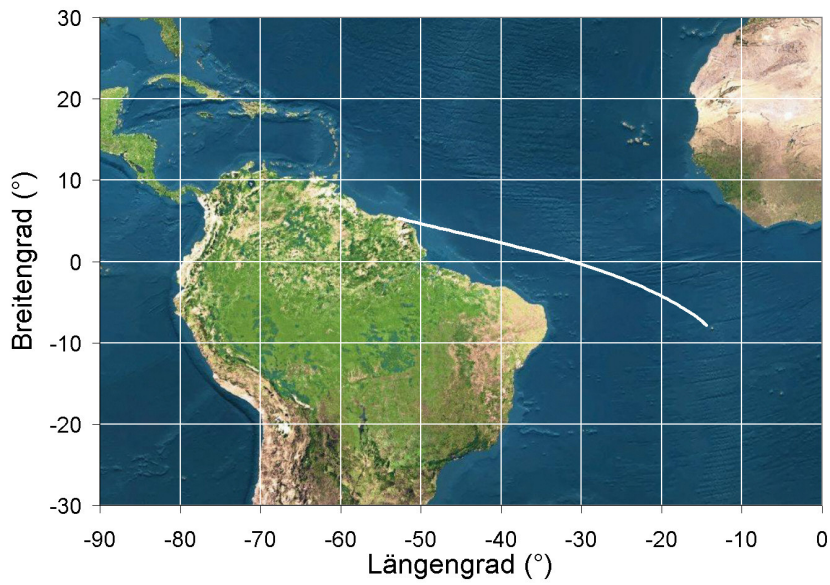


Abbildung 6.20: Bodenspur der Hopper-Gesamtmission.

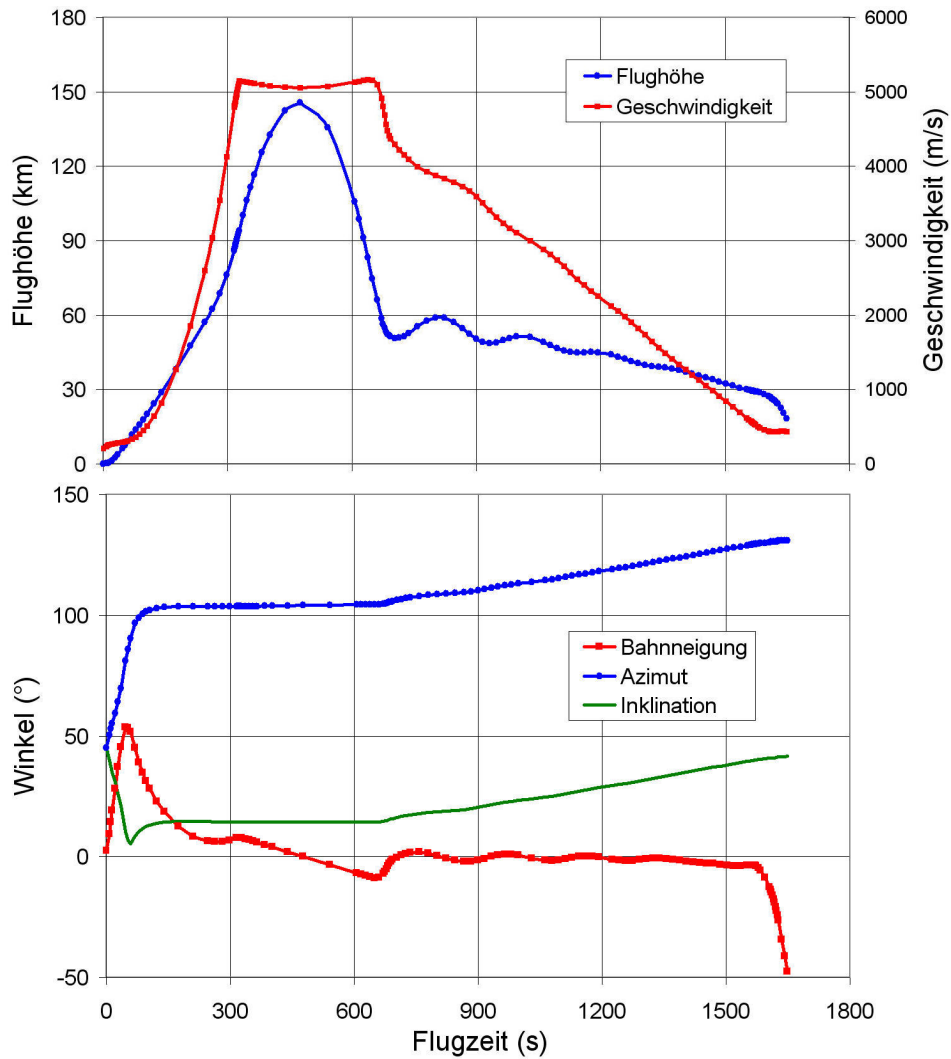


Abbildung 6.21: Flugprofil und Bahnwinkel für die Hopper-Gesamtmission.

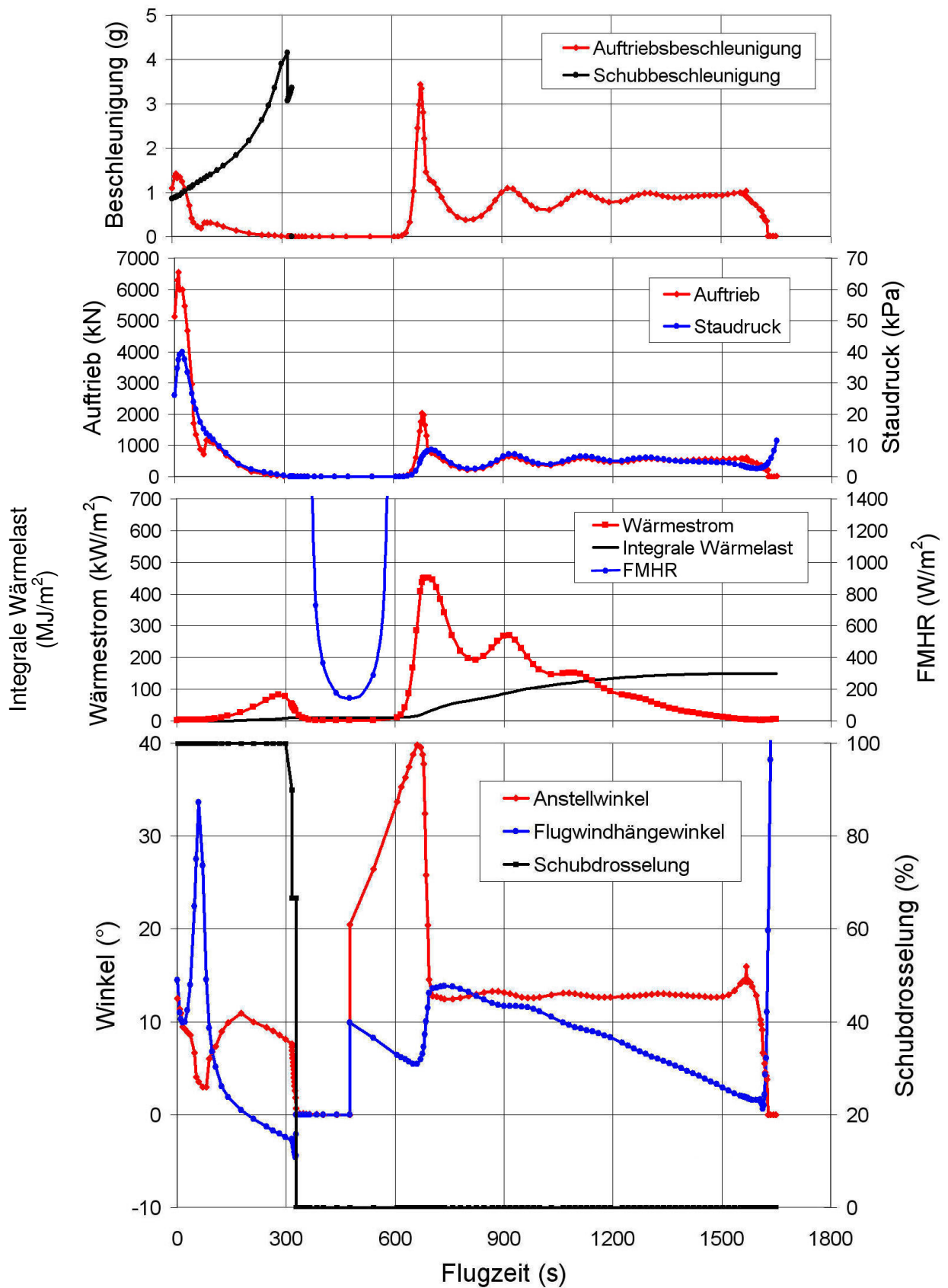


Abbildung 6.22: Die Fluglasten und -steuerung für die Hopper-Gesamtmission.

Die maximale Auftriebsbeschleunigung in der Aufstiegsphase wird 10 s nach dem Start erreicht. Die Beschleunigung beträgt 1,43 g und ist nicht kritisch. Dagegen kommt die Auftriebskraft mit 6543 kN dem maximalen Auftrieb (6600 kN) aufgrund der hohen Fahrzeugmasse sehr nahe. In dieser Phase wird der Anstellwinkel zwar reduziert aber die hohe Dichte der Gitterpunkte in etwa 6 s Abständen ermöglichen eine genaue Steuerung und die volle Ausnutzung des zulässigen Auftriebs. In den ersten 50 s wird 8 km Höhe erreicht, wobei der Auftrieb auf ca. die Hälfte sinkt.

Der maximale dynamische Druck von 40 kPa wird nach 21 s erreicht. Danach wird der Anstellwinkel reduziert, die Geschwindigkeit bleibt fast konstant, der Flugwindhängewinkel wird erhöht und damit der Kurvenflug eingeleitet. Auch in diesem Flugbereich kann die Gitterverfeinerung beobachtet werden. Mit sechs Gitterpunkten zwischen 7 und 35 s kann die Beschränkung exakt eingehalten werden.

Nach einer Flugzeit von etwa 300 s wird die maximale Schubbeschleunigung von 4,15 g erreicht. Durch Schubdrosselung kann diese Beschränkung für 18 s eingehalten werden. Danach muss ein Triebwerk abgeschaltet werden, weil die Vulkan Triebwerke nur um 10% gedrosselt werden können. In der sich anschließenden Beschleunigungsphase, die 12 s dauert, müssen die zwei verbleibenden Triebwerke nicht mehr gedrosselt werden. Die Geschwindigkeit nimmt durch diese Beschleunigung um 360 m/s zu und beträgt beim Abschalten aller Triebwerke 328 s nach dem Start 5152 m/s. Die Abschaltbedingungen unterliegen keiner Beschränkung und sind damit Optimierungsparameter. Sie werden von SOCS so gewählt, dass die am Scheitelpunkt der suborbitalen Bahn berechnete Nutzlast maximiert wird.

Die dritte Phase des Optimierungsproblems endet bei Erreichen des Kulminations- bzw. Scheitelpunktes der Aufstiegsbahn im Freiflug. Die Endbedingung für diese Phase ist der Rückgang der Bahnneigung auf Null. Der Scheitelpunkt im vorliegenden Fall hat eine Höhe von 145,6 km und die geografische Länge  $-41,85^\circ$  und Breite  $2,66^\circ$ . Hier weist Hopper eine Gesamtmasse von 99988 kg inklusive der Oberstufe und Nutzlast auf. Die Oberstufe mit der Nutzlast wird ausgesetzt bei 5057 m/s Geschwindigkeit,  $104,1^\circ$  Azimut und  $14,33^\circ$  relativer bzw.  $13,16^\circ$  inertialer Bahninklination. Damit ergibt sich eine Inklinationsdifferenz von  $5,86^\circ$  von der Zielinklination von  $7,3^\circ$ . Die Inklinationskorrektur und die Beschleunigung auf die GTO-Bahn werden durch die Oberstufe durchgeführt und sind in der Nutzlastberechnung berücksichtigt. Die für diese Anfangsbedingungen analytisch ermittelte maximale GTO-Nutzlast beträgt 8770 kg. Für das Aussetzen der Nutzlast wird gefordert, dass die freie, molekulare Aufheizrate FMHR für mindestens 180 s unter  $1340 \text{ W/m}^2$  liegt. Die hier präsentierte optimale Lösung erfüllt auch diese Beschränkung (vgl. Abbildung 6.22).

Wie die Abbildung 6.22 weiter zeigt, steigt der Wärmefluss im Staupunkt im Rückkehrflug zum Landeziel Ascension Island im Bereich 670 s auf über  $400 \text{ kW/m}^2$ . Das Fahrzeug wird mit dem maximal zulässigen Anstellwinkel von  $40^\circ$  in einer Flughöhe von 58,6 km abgefangen, sinkt weiter bis 50,7 km Höhe und steigt danach kurzzeitig wieder. Während das Fahrzeug steigt wird die Anstellwinkelsteuerung wieder zurückgenommen. Die maximale Wärmelastgrenze von  $450 \text{ kW/m}^2$  wird in dieser ca. 45 s dauernden Flugphase durch die genaue Steuerung des Anstellwinkels mit neun Stützstellen eingehalten. Während im Kurvenflug zum Anfang der Mission die höchste zulässige Auftriebskraft von 6600 kN der Mission erreicht wird, wird hier die höchste Auftriebsbeschleunigung von 3,44 g erreicht (Grenzwert nach [104]: 4,5 evtl. 4,8 g).

Das Abfangen des Hoppers und die Steuerung zur Begrenzung des Staupunkt-Wärmefflusses führen dazu, dass das Fahrzeug wieder steigt. Die Bahnneigung beträgt hier  $1,81^\circ$  und überschreitet die in [104] gegebene Grenze von  $1^\circ$ . Abbildung 6.21 lässt jedoch erkennen, dass die Bahnneigung im weiteren Flugverlauf unter  $1^\circ$  bleibt.

Am Ende des Rückkehrflugs muss die TAEM-Schnittstelle in der Nähe der Ascension Islands erreicht werden. Diese wird definiert durch die geografischen Zielkoordinaten  $14,333^\circ\text{W}$  und  $7,9167^\circ\text{S}$ . Der hier betrachtete Teil der Transportmission endet sobald die Entfernung zum gegebenen Zielpunkt 20 km unterschreitet, die Machzahl mindestens 1,5 und die Flughöhe mindestens 15 km sind. Die optimale Lösung hält die Entfernung und Machzahl exakt ein und erreicht die TAEM-Schnittstelle in 15 km Flughöhe. Die Abbildungen Abbildung 6.21 und Abbildung 6.22 verdeutlichen, dass für die Einhaltung der Machzahl-Bedingung ein steiler Sturzflug mit fast  $90^\circ$  Flugwindhängewinkel und vernachlässigbarem Auftrieb durchgeführt werden muss, um die Zielmachzahl zu erreichen. Dies führt dazu, dass am Ende des simulierten Rückkehrflugs die Bahnneigung  $-48^\circ$  beträgt. Die Machzahlforderung nach [104] erscheint daher wenig sinnvoll.

#### **Anmerkung:**

Die hier berechnete maximale GTO-Nutzlast von 8770 kg ist etwa 5,5% höher als der höchste Wert (8300 kg) aus der Sensitivitätsstudie in [111]. Ein Grund für diesen Unterschied sind die Differenzen in den Umweltmodellen. Während in [111] eine abgeplattete Erde mit positionsabhängigem Gravitationsfeld und das MSISE Atmosphärenmodell verwendet wurde, wurden in dieser Arbeit das Newton'sche Gravitationsfeld, eine kugelförmige Erde und die US-Standardatmosphäre angenommen. Ferner wurde in [111] auch die Startmasse in die Optimierung einbezogen und die TAEM-Schnittstellenbedingungen durch äquivalente Bahnenergie ersetzt, womit ein direkter Vergleich der Ergebnisse nicht möglich ist. Die Differenz zur geforderten Referenz-Nutzlastmasse [106] von 7500 kg kommt dadurch, dass in der hier beschriebenen Lösung die Inklinationvorgabe am Bahnscheitelpunkt von  $7,3^\circ$  nicht eingehalten wurde. Stattdessen wurde diese optimiert ( $14,3^\circ$ ). Die Durchführung der Inklinationsänderung durch die Oberstufe ist offensichtlich vorteilhafter.

#### **6.7.4 Numerische Aspekte**

Für die Praxis ist der Zeitaufwand eines optimalen Entwurfsschritts bedeutsam, da einer Vielzahl von Parametereinflüssen nachgegangen werden muss. Ein Lösungspunkt gibt keine Auskunft zur Systemempfindlichkeit gegenüber Störeinflüssen, Steuer- oder Parameteränderungen innerhalb möglicher Streubreiten. Dies erfordert aufwändige Sensitivitätsstudien (etwa mit Monte-Carlo-Flugsimulationen), die nicht Gegenstand der vorliegenden Arbeit sind. Der interessierte Leser wird diesbezüglich auf die Dissertation Telaar [111] verwiesen. Für solche Untersuchungen sind die Rechenerfordernisse bedeutsam, die nachfolgend betrachtet werden.

Die vorstehenden Optimierungsergebnisse wurden auf einem tragbaren Rechner mit einem 2,66 MHz Pentium 4 Prozessor und 512 MB Arbeitsspeicher erzielt. Einige rechenleistungsrelevante Daten sind in der Tabelle 6.12 zusammengefasst.

Rechenzeit	190 – 200 s (CPU-Zeit: 180 s)
Anzahl der Gitterpunkte	35 am Anfang, 139 in der Endlösung
Anzahl der Gitterverfeinerungen	5
Anzahl der Optimierungsparameter	3 Steuerfunktionen: $3 \times 139 = 417$ + Phasenendbedingungen
Anfangsschätzung	linear interpoliert zwischen geschätzten Anfangs- und Endwerten für alle Flugzustandsvariablen und Steuerfunktionen

Tabelle 6.12: Numerische Aspekte der Gesamtmissionsoptimierung.

Die Stützstellen der Steuerfunktionen werden an allen Gitterpunkten optimiert. Daraus ergeben sich 417 Optimierungsparameter. Da die Verknüpfungsbedingungen zwischen den Flugphasen 1-2, 2-3 und 3-4 nicht festgelegt sind, müssen auch diese optimiert werden. Ebenso müssen die Dauer der einzelnen Flugphasen als zeitunabhängige Optimierungsparameter in die Optimierungsrechnung einbezogen werden.

Die folgende Abbildung 6.23 veranschaulicht die Verteilung der Rechengitterpunkte über die Flugzeit:

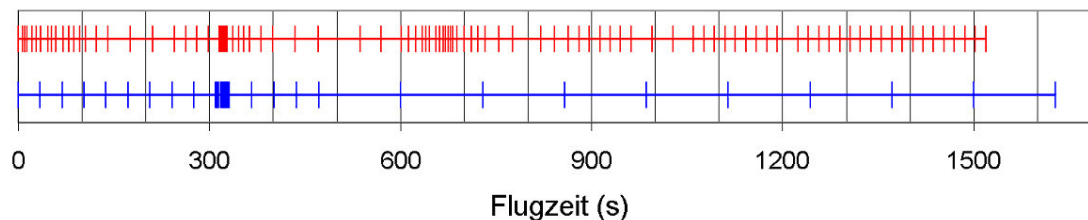


Abbildung 6.23: Verteilung der Gitterpunkte über die Flugzeit.

Die untere Grafik zeigt das Anfangsgitter und die obere das endgültige Rechengitter nach fünfmaliger Verfeinerung. In Flugbereichen, in denen hohe Gradienten der Flugzustandsvariablen auftreten bzw. wo entlang aktiver Flugrestriktionen geflogen wird und daher eine höhere Zeitauflösung für eine genaue Flugsteuerung benötigt wird, verfeinert das Optimierungsverfahren das Rechengitter durch Teilung und erhöht die Stützstellen der Steuerfunktionen und der Zustandsvariablen.

In der Abbildung 6.23 fallen drei Flugsegmente mit besonders hoher Gitterdichte auf. Diese lassen sich folgenderweise erklären:

1.  $0 \leq t \leq 100$  s: In der Anfangsphase der Mission werden die Staudruck- und Auftriebsbeschränkungen aktiv und der Kurvenflug zur Korrektur der Flugrichtung zum Landeziel wird eingeleitet.
2.  $300 \leq t \leq 400$  s: Ohne Schub und ohne signifikante aerodynamische Kräfte ist in der Freiflugphase keine Bahnkorrektur mehr möglich. Kurz vor der Triebwerksabschaltung werden daher die Startbedingungen der Freiflugphase, die die optimalen Kulminationspunktbedingungen bestimmen, justiert.
3.  $600 \leq t \leq 700$  s: Hier findet das erste Abfangmanöver nach dem Wiedereintritt statt und die Wärmeflussbeschränkung am Staupunkt wird aktiv. Das Fahrzeug erfährt während etwa einer halben Minute 97% bis 100% des maximal zulässigen Wärmeflusses.

In allen drei Fällen ist eine zeitlich hoch aufgelöste Anstellwinkel- und Flugwindhängewinkelsteuerung erforderlich.

Die in Kapitel 4.2 beschriebene Transkriptionsmethode beinhaltet nicht nur die Diskretisierung der Steuerfunktionen, sondern auch die der Zustandsvariablen, die als Restriktionen in die Optimierung einbezogen werden. Ein Ziel der Optimierung ist daher die Erfüllung der Bewegungsgleichungen mit einem finiten Relativfehler (Standardeinstellung  $10^{-3}$ ) an allen Gitterpunkten und für alle Zustandsvariablen. Die Betrachtung der numerischen Aspekte der Optimierung schließen daher die Ergebnisse einer zusätzlichen Optimierungsberechnung der gesamten Hopper - Transportmission mit einer vorgegebenen Fehlertoleranz von  $10^{-6}$  ab. Die Ergebnisse dieser Untersuchung sind in der Abbildung 6.24 dargestellt. Die angegebenen Rechenzeiten sind CPU-Sekunden.

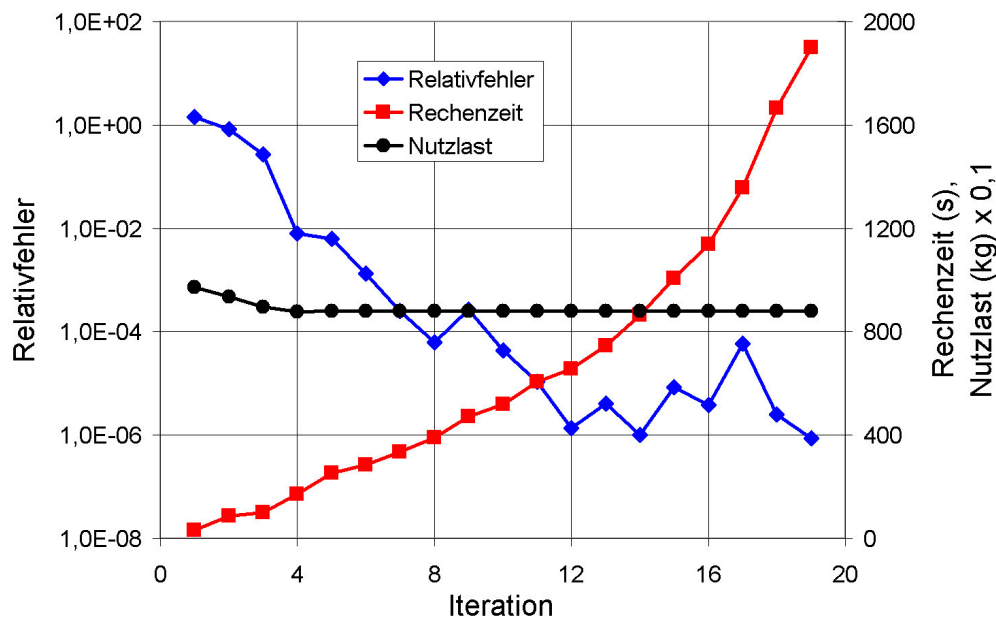


Abbildung 6.24: Rechengenauigkeit und Rechenzeit als Funktion der Iterationszahl.

Wie aus der Abbildung ersichtlich, konnte diese Genauigkeit erreicht werden. Die Optimierung konvergierte nach 19 Gitterverfeinerungsiterationen mit einem Rechengitter von 943 Gitterpunkten und dauerte über 30 Minuten. Betrachtet man die Entwicklung der Zielfunktion bzw. Nutzlastmasse, sieht man, dass sie nach der vierten Iteration fast konstant bleibt. Genauer ausgedrückt, die Nutzlast erreicht nach der fünften Iteration bzw. in vier Minuten schon 99,995% der Masse nach der 19. Iteration.

Aus der praktischen Sicht zeigen diese Ergebnisse folgende Anwendungsvorteile:

1. Die Erfüllung der Differentialgleichungen mit sehr hoher Genauigkeit ist möglich.
2. Die Genauigkeit kann bei Voruntersuchungen, Sensitivitätsanalysen und bei der Suche nach einer Anfangslösung herabgesetzt werden, um die Rechenzeit drastisch zu reduzieren.
3. Die Optimierung der Hopper-Mission mit über 2800 Optimierungsparametern in 30 Minuten zeigt, dass für komplexere Fahrzeuge, Missionen und genauere Simulationsmodelle genügend CPU-Reserven bleiben.

## 7 Zusammenfassung und Ausblick

Das am Institut für Raumfahrtsysteme der Universität Stuttgart für den konzeptionellen Fahrzeugentwurf entwickelte Programmpaket RTSOPT wurde im Rahmen dieser Arbeit grundlegend überarbeitet und verbessert. Die Schwerpunktsetzungen betreffen dabei die simultane Flug- und Systemoptimierung mit einem neuen, leistungsfähigen Löser SOCS (Sparse Optimal Control Software) der Firma Boeing und die Anwendung fortschrittlicher Softwaretechnologien einschließlich moderner Visualisierungstechniken. Daneben werden Modellbeschreibungen aktualisiert und verbessert.

Das neu strukturierte Programm RTSOPT wurde im Anwendungsteil für das Beispiel des Trägerkonzepts Hopper erfolgreich erprobt. Dabei wurden sowohl die Funktionsfähigkeit des implementierten Programmsystems nachgewiesen als auch die Anwendungsvorteile aufgezeigt. Gegenüber früheren Programmversionen wurden mit der vorliegenden Arbeit signifikante Leistungssteigerungen erzielt.

Zur verfahrenstechnischen Leistungssteigerung wurde das in RTSOPT bisher eingesetzte Optimierungsverfahren NLPQL durch den derzeit fortschrittlichsten Löser SOCS ersetzt. Dazu waren umfangreiche Modifikationen des Programmcodes erforderlich. Die in dieser Arbeit gewählte, aufwändige Methode der Integration von SOCS in RTSOPT und die Vorgehensweise bei der Problemformulierung haben den Vorteil, dass die Leistungsfähigkeit des Optimierungsverfahrens dadurch voll ausgereizt wird. Die mit der neuen RTSOPT-Version erzielten Anwendungsergebnisse für die Analyse der Hopper-Mission zeigen, dass diese Vorgehensweise sich bezahlt gemacht hat. Im Folgenden sind die beobachteten Leistungsvorteile dieser Verbesserung zusammengefasst.

Während die Rechenzeit mit NLPQL überproportional mit der Anzahl der Optimierungsparameter steigt, ist sie bei dem Verfahren SOCS nur abhängig von der Anzahl der Zustandsvariablen, deren Anzahl für die mit RTSOPT berechneten, simultanen Flug- und Systemoptimierungs-Problemen konstant ist. Die durchgeführten Optimierungsberechnungen belegen diese schwache Abhängigkeit zwischen der Anzahl der Optimierungsparameter und der Rechenzeit. Neben der Konvergenzzeit spielt auch die Genauigkeit der Lösung eine zentrale Rolle. Infolge der Diskretisierung der Flugzustände und Steuerfunktionen entstehen Ungenauigkeiten bzw. Diskretisierungsfehler. Jedoch ermöglicht die kontinuierliche Darstellung dieser Größen in B-Splines und Hermite Polynome, die automatische Erhöhung der Ordnung bzw. Anzahl der Gitterpunkte (Gitterverfeinerung) und damit der Genauigkeit der Lösung. Das Programm SOCS führt während der Optimierung mehrere Gitterverfeinerungen durch und erhöht damit die zeitliche Auflösung und die Genauigkeit der optimalen Steuerung. Diese Vorgehensweise wurde besonders in sehr sensitiven Flugphasen und in Flugbereichen, wo entlang der Lastbeschränkungen geflogen werden, beobachtet. In den Optimierungsergebnissen spiegelt sich dieser Vorteil durch höhere maximale Nutzlasten gegenüber der NLPQL-Lösung wider. Die hohe Zeitauflösung der Ergebnisse ermöglicht ferner längere Flugphasen entlang von Beschränkungen ohne diese zu verletzen. Daher kann in der vorliegenden RTSOPT-Version, auf die früher benötigten Regelalgorithmen zur Einhaltung von Flugbeschränkungen und Minimierung der Problemdimension ohne Leistungseinbußen verzichtet werden, wodurch die Anpassung des Programmcodes und die Entwicklung solcher Regelalgorithmen für verschiedene Untersuchungen entfallen.

Ein weiterer Vorteil aus der Sicht des Benutzers stellt die automatische NLP-Problemskalierung in SOCS dar. Diese Skalierung hat einen großen Einfluss auf die Konvergenzeigenschaften des Löser. Jedoch ist eine problemspezifische Anpassung der Skalierungsfaktoren aufwändig, da hier keine allgemeingültige Methode existiert. Die in der SOCS-Software implementierte Skalierungsmethode funktionierte für die in dieser Arbeit gelösten Probleme erfolgreich und erforderte keine Eingriffe durch den Benutzer.

Eine Optimierungsberechnung mit NLPQL erfordert eine ausreichend gute Startlösung. Daher mussten bei der bisherigen Vorgehensweise, Sensitivitätsanalysen, Versuche, eine Reihe von Simulationen, Anpassung der Diskretisierung und Änderungen im Programmcode vorangehen. Darüber hinaus hängt die Qualität des Endergebnisses auch von der Qualität der Startlösung bzw. der Nähe der Startlösung zur optimalen Lösung ab. Für eine Optimierung reichen in der neuen RTSOPT-Version dagegen lineare Anfangsschätzungen für die Flugzustände und Steuerfunktionen. Die Anfangslösung wird mit SOCS generiert und die Vorarbeit, die vom Benutzer geleistet werden muss, minimiert.

Die Optimierung der Gesamtmission Hopper wurde in drei Minuten gelöst. Als Anfangsschätzung wurden dabei lediglich die Anfangs- und Endwerte der Flugzustände und Steuerfunktionen verwendet. Für solche Missionsanalysen wurden früher, je nach Problemkomplexität, teilweise mehrere Wochen benötigt.

Die Leistungsfähigkeit des Optimierungsverfahrens ermöglicht die Lösung komplexer, mehrphasiger Optimierungsprobleme mit sehr vielen Optimierungsparametern als ein zusammenhängendes Optimierungsproblem und löst damit die frühere rechenintensive, segmentierte Vorgehensweise ab.

Ferner hat der Optimierungslöser sich als sehr robust erwiesen. Während der Optimierungsrechnungen wurden für die hier dargestellten Arbeiten keine Programmabbrüche beobachtet.

Die hohen Konvergenzraten der Optimierung und die Minimierung der Vorbereitungsarbeiten bzw. Programmcode-Anpassungen ermöglichen dem Benutzer die schnelle Durchführung einer Reihe von Optimierungsberechnungen mit unterschiedlichen Flugbeschränkungen, Randbedingungen und Zielfunktionen, um Sensitivitäten zu bewerten, die kritischen Phasen und Beschränkungen der Mission zu identifizieren und die beste Aufteilung der betrachteten Mission in verschiedene Flugphasen zu finden bzw. ein allgemeines Verständnis für das Optimierungsproblem zu entwickeln.

Neben der numerischen Leistungsfähigkeit eines Analysewerkzeugs spielen die Plattformunabhängigkeit, Wartbarkeit und Erweiterbarkeit des Programmsystems eine zentrale Rolle. Um solche Softwarequalität-Aspekte des Werkzeugs RTSOPT zu verbessern wurde unter Verwendung von Software-Engineering-Methoden ein Softwarekonzept entwickelt. Dieses beinhaltet die objektorientierte Programmierung der Simulations- und Optimierungsmodule in Fortran90, die Steuerung des Programms über Text-Eingabedateien und eine grafische Benutzeroberfläche in Java. Das Konzept sieht ferner eine Trennung zwischen den Programmkomponenten Simulator, Rechenmodelle und Benutzeroberfläche vor und schreibt die Implementierung bestimmter Funktionen bzw. Schnittstellen des Simulators und der Modelle vor.

Die objektorientierte Vorgehensweise bewährte sich sowohl während der Einbindung des software- und verfahrenstechnisch sehr komplexen Programmpakets SOCS in RTSOPT als auch bei der Überarbeitung des Analysewerkzeugs und der am IRS in früheren Arbeiten entwickelten Rechenmodelle der Simulation für die effiziente Implementierung in der neuen Programmfassung.



Die Leistungsfähigkeit und Flexibilität des Analysewerkzeugs wurden durch die einfache Benutzbarkeit des Programmpakets gesteigert. So sind in der neuen RTSOPT-Version keine missionsspezifischen Code-Modifikationen erforderlich.

Auch die Erstellung von Rechenmodellen wurde verbessert. Da das Programmpaket die Interpolation von Datentabellen durch eine Auswahl von generischen Methoden durchführt, besteht im Vergleich zu früheren Versionen keine Notwendigkeit für die Codierung von neuen tabellarischen Rechenmodellen.

Durch das neue Softwarekonzept, welches plattformunabhängige und international normierte Technologien wie z.B. die Programmiersprachen Fortran90 und Java, HTML für die Online-Dokumentation und die Virtuelle-Realität-Modellierungssprache VRML für die Visualisierung verwendet, ist die Portabilität von RTSOPT gesichert.

Zur Aktualisierung der Massenvorhersage für zukünftige Raumtransportsysteme und zur Überprüfung früherer Technologieannahmen wurde das auf der statistischen Analyse von Subsystem-Entwurfsparameter und -Massen basierende Massenmodell des Analysewerkzeugs RTSOPT mit den Ergebnissen der FESTIP-Studie und zahlreichen neuen Daten erweitert und verbessert. Ferner wurde die Benutzung dieser Methode durch eine leicht zu bedienende grafische Benutzeroberfläche vereinfacht.

Durch die grafische Aufbereitung und Visualisierung von Rechenmodellen und Ergebnissen können diese Informationen ausgewertet und deren Analyse für den Benutzer erleichtert werden. Daher wurden die Visualisierungsfähigkeiten der neuen RTSOPT-Version erweitert und durch räumliche und animierte Darstellungen maßgeblich gesteigert.

Für die Visualisierung wurde die VRML-Technologie (Virtual Reality Modeling Language) eingesetzt. VRML sichert den langfristigen Erhalt der Darstellungsmöglichkeiten des RTSOPT und die Erweiterung derselben mit relativ geringem Entwicklungsaufwand.

Es wurden VRML-Anwendungen entwickelt, um Kurvendarstellungen von Simulationsergebnissen und Flächendarstellungen von Rechenmodellen zu generieren. Durch direkte Kodierung in VRML, programmatische Generierung von VRML-Modellen, Modellierung mit verschiedenen CAD-Werkzeugen, Konvertierung anderer Modelle in VRML und Übernahme von Modellen von Internet-Seiten wurden Raumfahrzeugmodelle erstellt. Darüber hinaus wurden VRML-Prototypen mit JavaScript entwickelt, um verschiedene Fahrzeugkomponenten parametrisch zu modellieren. Diese Module wurden schließlich zusammengefügt und kombiniert. Für die Modellierung von Planeten, Oberflächenstrukturen und Raumfahrzeugteilen in VRML wurde von der Technik "Texturabbildung" Gebrauch gemacht. Rechengitter für die numerische Strömungsanalyse wurden mit Drahtgittermodellen und Druckverteilungen auf Raumfahrzeugen durch Farbinterpolation dargestellt. Flugsimulationsergebnisse wurden über flacher Erdkarte oder über der Kugelerde dargestellt. Durch Animation, Darstellung der Wärmelast durch Farbkodierung und Animation von Fahrzeugteilen mittels in den Modellen integrierter JavaScript Programme wurden Flugverläufe visualisiert. Schließlich wurden schwierig vorstellbare Abläufe wie Taumelbewegungen, Sichtbarkeitsbedingungen und Abdeckungsgebiete von Antennen mit VRML-Animationen visualisiert.

Die multidisziplinäre System- und Flugoptimierung, die mehrere Fachgebiete einschließt, ist ein anspruchsvolles Forschungsgebiet. Der Bedarf für leistungsfähige Analysewerkzeuge motiviert weitere Entwicklungsanstrengungen auf diesem Gebiet. Bei der Leistungsfähigkeit der im Rahmen der vorliegenden Arbeit erstellten Programmversion von RTSOPT und dem schnellen Anstieg der Rechenleistungen und Rechnerressourcen bei gleichem Rechnerpreis bleiben genügend Rechenzeitreserven, um komplexere Missions- und Systemanalyseprobleme anzugehen. So können in zukünftigen RTSOPT-Versionen CFD-Werkzeuge für die Entwurfsaerodynamik- und FEM-Methoden für die Massenanalyse eingesetzt und in die Optimierungsberechnungen eingebunden werden.

## Literaturverzeichnis

- [1] Akima, H., "A Method of Univariate Interpolation that has the Accuracy of a Third-Degree Polynomial", ACM Transactions on Mathematical Software, Vol. 17, No.3, pp. 341-366, September 1991.
- [2] Arianespace, "Ariane and Soyuz to Launch two Australian Satellites from Guiana Space Center", Arianespace Press Releases, Berlin, May 2004.
- [3] Associate Administrator for Commercial Space Transportation, Federal Aviation Administration, "Commercial Space Transportation: Year in Review", Washington D.C., 8 Reports from 1997 to 2004.
- [4] Associate Administrator for Commercial Space Transportation, Federal Aviation Administration, "Commercial Space Transportation: Quarterly Launch Report", Washington D.C., 39 Reports from 1st Quarter 1996 to 3rd Quarter 2005.
- [5] Associate Administrator for Commercial Space Transportation, Federal Aviation Administration, "2004 Commercial Space Transportation Forecasts", FAA, USA, May 2004.
- [6] Bechtold, R., "Landegebietsanalyse und Flugvisualisierung einer Rückkehrkapsel", Studienarbeit, IRS-03-S-16, Institut für Raumfahrtssysteme, Universität Stuttgart, Juli 2003.
- [7] Berge, K.D., Brücker, H., Noack, E., Püttmann, N., "German RLV Activities – TETRA / X-38 & ASTRA", 1st AIAA/ IAF Symposium on Future Reusable Launch Vehicles, Huntsville, Alabama, USA, April 2002.
- [8] Betts, J.T., "Practical Methods for Optimal Control Using Nonlinear Programming", ISBN 0-89871-488-5, Society for Industrial and Applied Mathematics, Philadelphia, 2001.
- [9] Betts, J.T., Biehn, N., Campbell, S.L., "Computational Experience with exploiting Order Variation in Mesh Refinement for Direct Transcription Methods", Proc. IMACS 2000, Lausanne, 2000.
- [10] Betts, J.T., Huffman, W.P.: "Application of Sparse Nonlinear Programming to Trajectory Optimization", Journal of Guidance, Vol.15, No.1, 1/2 1992.
- [11] Boeing Company, "SOCS Release 6.1", M&CT-TECH-01-014, Boeing Company, 2003.
- [12] Bregman, E.R., "Flugoptimierung von aerodynamisch gestützten Orbittransferfahrzeugen unter besonderer Berücksichtigung der aerothermodynamischen Fluglasten", Dissertation, Institut für Raumfahrtssysteme, Universität Stuttgart, Stuttgart, April 1994.
- [13] Bull, J.M., Smith, L.A., Pottage, L., Freeman, R., "Benchmarking Java against C and Fortran for Scientific Applications", Proceedings of ACM Java Grande/ISCOPE Conference, June 2001.
- [14] Bundesministerium für Forschung und Technologie, "Review of the German Hypersonic Research and Technology Programme", Proceedings, Bonn, April 1991.
- [15] Carey, R., Bell, G., Marrin, C., "ISO/ICE 14772-1:1997 Virtual Reality Modeling Language (VRML97)", www.vrml.org, The VRML Consortium Incorporated, 1997.
- [16] Chabrillat, S., "Optimisation and use of Hedin's Atmospheric Empirical Model MSIS", Aeronomica Acta B-No 55 ISSN 0065-3713, Institut D'Aeronomie Spatiale de Belgique, Brüssel, 1995.
- [17] Cosmo Software, Computer Associates, <http://ca.com/cosmo/>, Computer Associates, June 2003.

- 
- [18] Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE", ACM SIGGRAPH '93 Proceedings, pp. 135-142, 1993.
- [19] Daimler-Benz Aerospace, "FESTIP System Study", Daimler-Benz Aerospace, Space Infrastructure, Ottobrunn, August 1998.
- [20] Däßler, R., "Das Einsteigerseminar, VRML", ISBN 3-8287-1082-4, bhv Verlag, Kaarst, 1999.
- [21] Decyk, V.K., Norton, C.D., Szymanski, B.K., "Expressing Object-Oriented Concepts in Fortran90", ACM Fortran Forum, vol. 16, no. 1, pp. 13-18, April 1997.
- [22] Decyk, V.K., Norton, C.D., Szymanski, B.K., "High Performance Object Oriented Scientific Programming in Fortran 90", Proc. Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, SIAM Press, MN, March 1997.
- [23] Decyk, V.K., Norton, C.D., Szymanski, B.K., "How to Express C++ Concepts in Fortran 90", Scientific Programming, vol. 6, No. 4, pp. 363-390, Winter 1997.
- [24] Decyk, V.K., Norton, C.D., Szymanski, B.K., "How to Support Inheritance and Run-Time Polymorphism in Fortran 90", Computer Physics Communications, Vol. 115, pp. 1-9, December 1998.
- [25] Dujarric, C., "Possible Future European Launchers – A Process of Convergence", Third European Symposium on Aerothermodynamics for Space Vehicles, ESTEC, 1998.
- [26] Dumbacher, D., "NASA's Second Generation Reusable Launch Vehicle Program, Introduction, Status and Future Plans", 38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, AIAA 2002-3613, Indianapolis, Indiana, USA, July 2002.
- [27] Edinburgh Parallel Computing Centre: "Java Grande at EPCC", [www.epcc.ed.ac.uk/javagrande/](http://www.epcc.ed.ac.uk/javagrande/).
- [28] European Space Agency, "Simulation Model Portability Handbook", ESA/ESTEC, October 2000.
- [29] Eymar, P., Cazes, Grimard, M., "EADS Roadmap for Launch Vehicles", 53. International Astronautical Congress, Houston, Texas, October 2002.
- [30] Faden, M., Wolf, K. "Softwareintegration und -kopplung als Basis der Multidisziplinären Simulation", DGLR Jahrestagung, DGLR-JT99-194, Berlin, September 1999.
- [31] Fortran Standards Technical Committee, "Working Draft J3/04-007, J3, [www.j3-fortran.org](http://www.j3-fortran.org), May 2004.
- [32] Fosner, R., "OpenGL, Programming for Windows 95 and Windows NT", ISBN 0-201-40709-4, Addison-Wesley, July 1998.
- [33] Fritzen, P., Lewis, J., Hinüber, A., "S(t)imulating Technologies – Using Today's Technologies to Solve Today's Problems ...", DLR Raumfahrttechnologietage, Köln, Juli 2003.
- [34] Futron Corporation, "China and the Second Space Age", [www.Astroexpo.com](http://www.Astroexpo.com), November 2003.
- [35] Gath, P.F., "CAMTOS – A Software Suite Combining Direct and indirect Trajectory Optimization Methods", Dissertation, Fakultät für Luft- und Raumfahrttechnik und Geodäsie, Universität Stuttgart, November 2002.
- [36] Gockel, W., "Phoenix RLV Flight Test Demonstrator", Deutscher Luft- und Raumfahrtkongress, JT2003-015, München, November 2003.

- [37] Gräßlin, M., "Entwurf und Analyse eines prädiktiven Lenkkonzepts für Rückkehrmissionen auftriebsgestützter Raumfahrzeuge", Dissertation, Institut für Raumfahrtsysteme, Universität Stuttgart, 2004.
- [38] Gräßlin, M., Schöttle, U.M., "Flight Performance Evaluation of the Re-Entry Mission IRDT-1", 52nd International Astronautical Congress, IAF-01-V.3.05, Toulouse, 2001.
- [39] Grieger, I., "Grafische Datenverarbeitung mit einer Einführung in PHIGS und PHIGS-PLUS", ISBN 3-540-53747-3, Springer-Verlag, Berlin, 1992.
- [40] High Performance Computing Center Stuttgart, Visualization Department, Virtual Environments Lab, [www.hlrs.de/organization/vis/velab/](http://www.hlrs.de/organization/vis/velab/), June 2003.
- [41] Hillesheimer, M., "Entwicklung eines Quasi-Expertensystems zur Flugbahn- und Systemoptimierung zukünftiger Raumtransporter", Dissertation, Institut für Raumfahrtsysteme, Universität Stuttgart, Mai 1994.
- [42] Hornik, A., "Beiträge zur Optimierung von Trägerraketen", Dissertation, Technische Universität München, Herbert Utz Verlag Wissenschaft, München, 1995.
- [43] Horstmann, C.S., Cornell, G., "Core Java 2", Volume 1: Fundamentals (5th Edition), ISBN 0130894680, Prentice Hall PTR, NJ 07458, 2002.
- [44] Horstmann, C.S., Cornell, G., "Core Java 2", Volume 2: Advanced Features (5th Edition), ISBN 0130927384, Prentice Hall PTR, NJ 07458, 2001.
- [45] Howard, T.L.J., Hewitt, W.T., Hubbard, R.J., Wyrwas, K.M., "A Practical Introduction to PHIGS and PHIGS PLUS", ISBN 0-201-41641-7, Addison-Wesley, 1991.
- [46] IABG, "AZURA – Analyse zukünftiger Raumtransportsysteme", Berichtsbände I-IV, Nr. B-TK-4013/02, IABG, 1990.
- [47] IBM, "AIXWindows Interface Composer Developer's Guide", Version 1.2, IBM, 1993.
- [48] Infed, F., "Realgas- und Katalyseeffekte beim Wiedereintritt von Raumflugkörpern aus dem erdnahen Orbit unter Berücksichtigung von Wärmeleitung und Strahlungsaustausch an der Oberfläche", Dissertation, Institut für Raumfahrtsysteme, Universität Stuttgart, Mai 2005.
- [49] Institut für Flugmechanik und Flugregelung, "GESOP Software User Manuals – The User Interface Tutorial, GESOP SUM 4.3", Universität Stuttgart, April 2001.
- [50] Institut für Flugmechanik und Flugregelung, "GESOP Software User Manual – Modeling and Coding, GESOP SUM 4.3", Universität Stuttgart, April 2001.
- [51] Isakowitz, S.J., Hopkins, J.P., Hopkins, J.B., "International Reference Guide to Space Launch Systems", 3rd Edition, American Institute of Aeronautics and Astronautics AIAA, Virginia, September 1999.
- [52] ISO/IEC 9592-1, "Information Processing Systems – Computer Graphics – Programmer's Hierarchical Interactive Graphics System (PHIGS), Functional Description", 1989.
- [53] ISO/IEC 9592-4, "Information Processing Systems – Computer Graphics – Programmers Hierarchical Interactive System (PHIGS), Functional Description Part 4 – Plus Lumiere und Surfaces, PHIGS PLUS, 1992.
- [54] Jansch, C., Schnepfer, K., Well, K.H., "Multiphase Trajectory Optimization Methods with Applications to Hypersonic Vehicles", Applied Mathematics in Aerospace Science and Engineering, edited by Miele, A., Salvetti, A., Plenum Press, New York, Chapter 8, 1994.
- [55] Kalden, O., "Konzeption und Entwicklung eines zweidimensionalen Multi-Block-Gittergenerierungsprogramms einschließlich grafischer Benutzeroberfläche", Technische Universität Berlin, Herrmann-Föttinger-Institut für Strömungsmechanik, Juli 1997.

- 
- [56] Kalden, O., "RTSOPT – Benutzerhandbuch", Interner Bericht, IRS-03-IB-06, Institut für Raumfahrtsysteme, Universität Stuttgart, Juli 2003.
- [57] Kalden, O., "RTSOPT – Massenmodell", Interner Bericht, IRS-03-IB08, Institut für Raumfahrtsysteme, Universität Stuttgart, Juli 2003.
- [58] Kalden, O., "VirtualTrajectory VT, Benutzerhandbuch", Interner Bericht, IRS-03-IB07, Institut für Raumfahrtsysteme, Universität Stuttgart, Juli 2003.
- [59] Kalden, O., Schöttle, U.M., "A Software Tool for Analysis of Future Launch Vehicle Concepts", International Astronautical Congress, IAC-03-V.5.08, Bremen, October 2003.
- [60] Kalden, O., Schöttle, U.M., "Entwicklung eines Portablen und Flexiblen Softwarepakets für die Analyse und Optimierung von Raumtransportsystemen", DGLR Jahrestagung, DGLR-JT02-177, Stuttgart, September 2002
- [61] Knapp, M., "Mission Optimization for a Reusable Launch Vehicle System", Pioneer Rocketplane, 9th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, AIAA-99-4932, AIAA, November 1999.
- [62] Koch, S., "JavaScript", ISBN 3-920993-64-0, dpunkt-Verlag, Heidelberg, 1997.
- [63] Kuczera, H., Krammer, P., Sacher, P.W., "Sänger and the German Hypersonics Technology Programme – Status Report 1991", 42nd IAF Congress, IAF-91-198, Montreal, October 1991.
- [64] Leitenberger, B., "Raumfahrt Aufsätze", [www.bernd-leitenberger.de](http://www.bernd-leitenberger.de), Oktober 2004.
- [65] Lentz, S., "Implementierung eines Verfahrens zur LOX-Collection in das bestehende numerische Entwurfswerkzeug Cadehyp", Universität der Bundeswehr München, Vortrag, DGLR-Fachauschuss S4.1 – Raumtransportsysteme, Köln, 2001.
- [66] Lepore, D.F., "Expanding Space Station Utilization through Affordable, Routine, Monthly Access with the K-1", Kistler Aerospace Corporation, International Astronautical Congress, IAC-04-TP05, October 2004.
- [67] Mamaghani, F., Campos, J., Gifford, T., Macchi, W., "SEDRIS™ - Environmental representation Standards for M&S Interoperability", I/ITSEC 2005, Orlando, Florida, USA, 28 November - 1 December 2005.
- [68] Marshall Space Flight Center, "NASA's Space Launch Initiative: The Next Generation Launch Technology Program", NASA Facts, FS-2003-05-63-MSFC, MSFC, NASA, Huntsville, Alabama, USA, May 2003.
- [69] Marshall Space Flight Center, "The Space Launch Initiative: Technology to Pioneer the Space Frontier", Fact Sheet, FS-2003-02-28-MSFC, MSFC, NASA, Huntsville, Alabama, USA, February 2003.
- [70] MAXON Computer Inc., [www.maxon.net](http://www.maxon.net), MAXON Computer Inc., CA91320, USA, 2003.
- [71] McAlister, P., "Orbital Space Plane – How Did We Get Here and Why?", Futron Corporation, Maryland, USA, July 2003.
- [72] Messerschmid, E., Fasoulas, S., "Raumfahrtsysteme, Eine Einführung mit Übungen und Lösungen", 2. aktualisierte Auflage, ISBN 3-540-21037-7, Springer-Verlag, 2005.
- [73] Messerschmid, E., Fasoulas, S., et al., "Grundlagen der Raumfahrtsysteme", Vorlesungsmanuskript, Institut für Raumfahrtsysteme, Universität Stuttgart, 2002.
- [74] NASA, USAF, US Weather Bur., "US Standard Atmosphere", December 1962.
- [75] Noack, E., "Deutsche Interessen und Prioritäten beim Future Launcher Preparatory Programme (FLPP)", Berliner Forum Zukunft der DGAP, DLR Bonn, Berlin, Juni 2004.

- [76] O'Keefe, S., "NASA's Initial FY 2004 Operating Plan", NASA, Washington, DC 20510, March 2004
- [77] Obersteiner, M., Fossati, F., "Future Launch Vehicle Structures – Expendable and Reusable Elements", 53rd International Astronautical Congress, IAC-02-I.7.-V.6.01, Texas, October 2002.
- [78] Object Management Group, "CORBA<sup>®</sup> Basics", [www.omg.org](http://www.omg.org), Object Management Group, Inc., MA, USA, October 2005.
- [79] OpenGL.org, [www.opengl.org](http://www.opengl.org), Juni 2003.
- [80] Parametric Technology Corporation, "Pro/ENGINEER – Pro/DEVELOP User's Guide", Document Number U0A110795, Parametric Technology Corporation, Juli 1995.
- [81] Parametric Technology Corporation, "Pro/ENGINEER<sup>®</sup> User Guides, Parametric Technology Corporation, 1995.
- [82] QT software GmbH, München 80802, <http://www.qtsoftware.de/>, Juni 2003.
- [83] Rabenseifner, R., Rühle, R., "Message Passing Interface (MPI), Course Material, Rechenzentrum der Universität Stuttgart, Institut für Computeranwendungen, IB 159, Stuttgart, Mai 1999.
- [84] Rahn, M., "Eine numerische Methodik zur simultanen Flug- und Systemoptimierung von Raumtransportsystemen", Dissertation, Institut für Raumfahrtsysteme, Universität Stuttgart, März 1998.
- [85] Rahn, M., Möller, C., Gomez, J.A., Rößler, T., "Comparison of Cryogenic Upper Stages for the Ariane 5 Launcher", 52nd International Astronautical Congress, Toulouse, October 2001.
- [86] Rahn, M., Schöttle, U.M., "Decomposition Algorithm for Performance Optimization of a Launch Vehicle", Journal of Spacecraft and Rockets, Vol.33, No.2, pp.214-221, AIAA, Washington DC, 1996.
- [87] Rahn, M., Schöttle, U.M., Messerschmid, E., "Impact of Mission Requirements and Constraints on Conceptual Vehicle Design", Aerospace Science and Technology, No.6, pp.391-401, 1999.
- [88] Rantzau, D., Frank, K., Lang, U., Rainer, D., Wössner, U., "COVISE in the CUBE: An Environment for Analyzing Large and Complex Simulation Data", Proc. 2nd Workshop on Immersive Projection Technology (IPTW '98), Ames, Iowa, 1998.
- [89] Regionales Rechenzentrum für Niedersachsen, Universität Hannover, "Fortran90", 1993.
- [90] Rosin, O., "Beiträge zur Massenmodellierung von Raumtransportsystemen", Diplomarbeit, IRS-91-S10, Institut für Raumfahrtsysteme, Universität Stuttgart, April 1991.
- [91] Roth, G., "Optimierung der Antriebserfordernisse und Entwurf der Oberstufe des Trägerkonzepts Pathfinder", Diplomarbeit, IRS-99-S-25, Institut für Raumfahrtsysteme, Universität Stuttgart, Oktober 1999.
- [92] Schittkowski, K.: "NLPQL – A Fortran Subroutine for Solving Constrained Nonlinear Programming Problems", Anals of Operations Research, Vol.5, pp. 485-500, 1985-1986.
- [93] Schnepfer, K., "ALTOS Software User Manual, PROMIS: Optimization Program", Deutsche Forschungsanstalt für Luft- und Raumfahrt, Institut für Flugsystemdynamik, Februar 1992.
- [94] Schöttle, U.M., "Analysis of the Aeroheating Environment of Future Space Transportation Systems", SFB 259, Universität Stuttgart, Preprint Nr.7, Juli 1990.

- 
- [95] Schöttle, U.M., "Control Models for Ascent Performance Optimization of Airbreathing Launch Vehicles", 17th International Symposium on Space Technology and Science", Tokyo, Japan, May 1990.
- [96] Schöttle, U.M., "Flug- und Antrieboptimierung Luftatmender Aerodynamischer Raumfahrtsträger", Dissertation, Institut für Raumfahrtsysteme, Universität Stuttgart, 1988.
- [97] Schöttle, U.M., "Geflügelte Luftatmende Raumtransporter", Symposium Geflügelte Raumfahrzeuge der Hermann-Oberth-Gesellschaft e.V., Hannover, Mai 1984.
- [98] Schöttle, U.M., "Koordinatensysteme und Transformationen für Bahnuntersuchungen", IRA 75-IB-8, Institut für Raumfahrtantriebe, Universität Stuttgart, Dezember 1975.
- [99] Schöttle, U.M., "Massenanalyse Einstufiger Geflügelter Raumtransporter mit Raketen- und Staustrahlantrieb", IRA-83-IB-2, Institut für Raumfahrtantriebe, Universität Stuttgart, April 1983.
- [100] Schöttle, U.M., "Mathematische Beschreibung der räumlichen Flugkörperbewegung", IRA 79-IB-1, Institut für Raumfahrtantriebe, Universität Stuttgart, überarbeitete Version, 1984.
- [101] Schöttle, U.M., Hillesheimer, M., "Performance Optimization of an Airbreathing Launch Vehicle by a Sequential Trajectory Optimization and Vehicle Design Scheme", AIAA, Guidance, Navigation and Control Conference, New Orleans, USA, August 1991.
- [102] Schultheiß, D., "Aktualisierung und Erweiterung einer datenbasierten Methodik zur Massenabschätzung von Raumtransportsystemen", Studienarbeit, IRS-03-S-05, Institut für Raumfahrtsysteme, Universität Stuttgart, März 2003.
- [103] Sommerville, I., "Software Engineering", ISBN 0-201-56529-3, Addison-Wesley, 1992.
- [104] Spies, J., "ASTRA Systemkonzept 1: Hopper Bahnoptimierung mit ASTOS", Private Mitteilung, März 2002.
- [105] Spies, J., Kuczera, H., "The Sub-Orbital Hopper – one of FESTIP's Preferred Concepts", AIAA 9th International Space Planes and Hypersonic Systems and Technologies Conference, Northfolk, USA, AIAA-99-4945, AIAA, November 1999.
- [106] Spies, J.: "RLV Hopper: Consolidated System Concept", The World Space Congress, IAC-02-V.4.02, Houston, Texas, October 2002.
- [107] Stanley, D., "An Orbital Space Plane to Meet NASA's Future Mission Needs", 54th International Astronautical Congress, IAC-03-V.3.03, Bremen, October 2003.
- [108] Stritzinger, A., "Komponentenbasierte Softwareentwicklung – Konzepte und Techniken für das Programmieren und Modellieren in Smalltalk", Addison-Wesley, München, 1998.
- [109] Sun Microsystems Inc. CA, USA: "The Source for Java™ Technology", <http://java.sun.com/>, June 2003.
- [110] Technology Transfer Initiative, TTI GmbH, Department of Optimization, Guidance & Control, [www.gesop.de](http://www.gesop.de), Stuttgart, 2005.
- [111] Telaar, J., "Entwicklung eines prädiktiven Lenkverfahrens für wiederverwendbare Raumtransportsysteme", Dissertation, Institut für Raumfahrtsysteme, Universität Stuttgart, November 2005.
- [112] Telaar, J., Schöttle, U.M., "Leistungsvergleich NLP-Basierter Lenkkonzepte für Trägeraufstiegsmissionen mit Optimalsteuerungen", Deutscher Luft- und Raumfahrtkongress, DGLR-JT2003-190, München, November 2003.



- [113] Telaar, J., Schöttle, U.M., Röser, H.-P., "Anwendung eines Prädiktiven Lenkverfahrens auf das Raumtransportsystem Hopper", DGLR Jahrestagung, DGLR-JT2004-197, Dresden, September 2004
- [114] Tetlow, M.R., "Commercial Launch Vehicle Design and Predictive Guidance Development", PhD Dissertation, University of Adelaide, 2002.
- [115] Topley, K., "Core JFC", ISBN: 0-13-090581-X, Prentice Hall PTR, NJ 07458, 2002.
- [116] VEGA Group PLC, "Simulations Development", [www.vega-group.com](http://www.vega-group.com), VEGA Group PLC, 2004.
- [117] Visual Numerics Inc., "IMSL C, Fortran, Java<sup>TM</sup> Mathematics & Statistics Libraries", [www.vni.com/products/ims/](http://www.vni.com/products/ims/), 2002.
- [118] VRML Consortium Inc.: "VRML 97", International Standard, ISO/IEC 14772-1:1997.
- [119] Wallace, J., Olds, J., Charania, A.C., Woodlock, G., "A Study of ARTS: A Dual-Fuel Reusable Launch Vehicle with Launch Assist", 39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA-2003-5269, Huntsville, Alabama, USA, July 2003.
- [120] Weickert, S., "Numerische Optimierung der Flugleistungen des Raumtransporters Hopper", Diplomarbeit, IRS-03-S-08, Institut für Raumfahrtsysteme, Universität Stuttgart, März 2003.
- [121] Well, K.H., Markl, A., Mehlem, K., "ALTOS – A Software Package for Simulation and Optimization of Trajectories of Launch and Reentry Vehicles", IAF-97-V404, IAF, Turin, 1997
- [122] Windholz, R., "Ada im praktischen Einsatz", Studienarbeit Nr. 1661, Institut für Informatik, Universität Stuttgart, [www.informatik.uni-stuttgart.de/ifi/ps/Publ/](http://www.informatik.uni-stuttgart.de/ifi/ps/Publ/), Dezember 1997.
- [123] World Wide Web Consortium, [www.w3.org](http://www.w3.org), September 2005.
- [124] Zhao, T.C., Overmars, M., "Forms Library, A Graphical User Interface Toolkit for X", University of Wisconsin-Milwaukee, 1995.
- [125] Zimmermann, F., "Optimierung der seilgestützten Rückkehrmission einer gelenkten Wiedereintrittskapsel", Dissertation, Institut für Raumfahrtsysteme, Universität Stuttgart, September 2001.

## Anhang A: Marktanalyse

Die hier aufgeführten Darstellungen des kommerziellen Raumtransportmarktes basieren auf den Trägerstartdaten in den Berichten der US-Behörde AST/FAA (Associate Administrator for Commercial Space Transportation, Federal Aviation Administration) von 1996 bis 2005 [3], [4]. Die Daten berücksichtigen sowohl Nutzlasten in die erdnahen Umlaufbahnen (LEO) als auch den geostationären Transferorbit (GTO). Kommerzielle Starts werden dadurch definiert, dass die Nutzlast einem wirtschaftlichen Verwendungszweck dient und der Transport für den internationalen Wettbewerb offen sein muss.

### A.1 Der kommerzielle Raumtransportmarkt

Die folgende Abbildung A.1 stellt die Marktanteile verschiedener Trägerbetreiber dar.

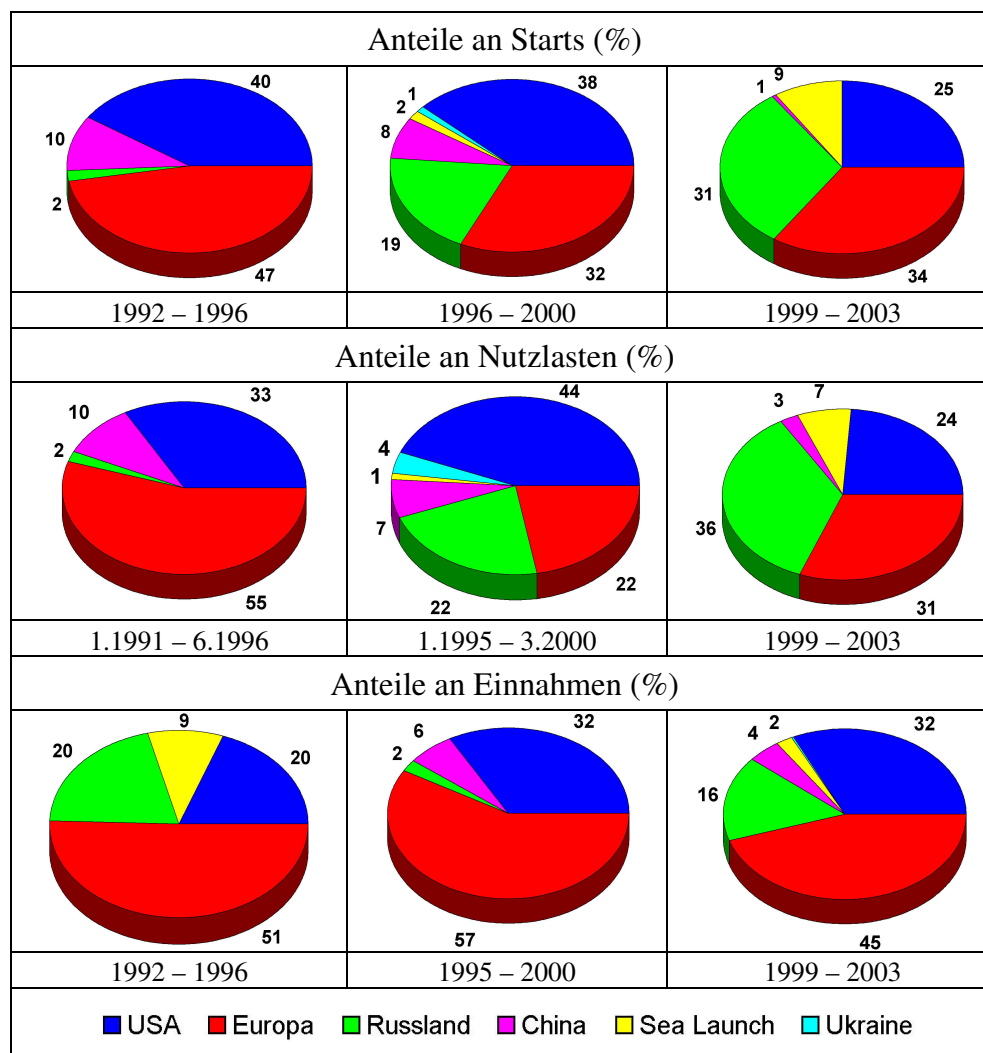


Abbildung A.1: Anteile der Trägerbetreiber am Raumtransportmarkt [3],[4].

**Europa:** Der in Abbildung A.1 beobachtete Rückgang des europäischen Marktanteils in den Jahren 1996-2000 hat seine Ursache darin, dass die verwendeten Ariane-Träger fast ausschließlich für GTO-Nutzlasten eingesetzt wurden und von den zahlreichen Starts von Satellitenkonstellationen nicht profitiert haben. Darüber hinaus wurden mit Ariane-4 und ab 1998 mit Ariane-5 viele Mehrfachstarts durchgeführt, so dass die Startzahlen nicht so hoch sind wie bei kleineren Trägern. Betrachtet man die Anteile an Einnahmen, sieht die Statistik ganz anders aus. Da mit den Ariane-Trägern schwere Satelliten transportiert wurden, erreichte Arianespace mit 1,5 Milliarden US\$ einen Marktanteil von 57% in 2000. Jedoch zeigt die Statistik eindeutig, den Bedarf an einem kleineren europäischen Träger für kleinere Nutzlasten und niedrige Umlaufbahnen, um europäische Anteile auch in diesem Bereich zu sichern.

Bis 2003 ging der europäische Anteil an Einnahmen von 57% auf 45% und an gestarteten kommerziellen Nutzlasten von 55% auf 31% zurück. Der noch hohe finanzielle Anteil der Arianespace basiert auf Doppelstarts bzw. der hohen Nutzlastkapazität der Ariane-5-Trägerrakete. Jedoch wird auch dieser Vorteil durch die neuen amerikanischen Trägersysteme Delta-4 und Atlas-5 gefährdet. Diese Tendenz zeigt den dringenden Bedarf an der Weiterentwicklung der Trägerrakete Ariane-5, um ihre Konkurrenzfähigkeit auf diesem Markt wieder herzustellen und auszubauen. Nach dem Ariane-5-ECA mit der neuen Version der kryogenen Oberstufe der Ariane-4 und 10 t Nutzlastkapazität bei ihrem Erststart am 12.12.2002 scheiterte, kehrte sie erfolgreich am 12.02.2005 zurück zum kommerziellen Betrieb und lässt optimistisch in die Zukunft schauen.

**USA:** Die Statistik von 1992 bis 2003 zeigt auch den Rückgang der Marktanteile der USA. Jedoch wird die Trägerentwicklung in den USA sehr stark von militärischen Anwendungen angetrieben, wie z.B. die Entwicklung der militärischen Delta-4 Rakete. Für die Raumfahrt steht, im Vergleich zu Europa, ein viel größeres Budget zur Verfügung. Das staatliche Raumfahrtbudget der USA beträgt 26 Milliarden US\$ [34]. Allein das Budget der NASA beträgt über 15 Milliarden US\$, wovon die Hälfte für Raumflugkapazitäten vorgesehen sind [76]. Darüber hinaus besteht in den USA ein großer militärischer Satellitenstartmarkt. So konnten amerikanische Trägerbetreiber trotz der sinkenden kommerziellen Marktanteile mit drei weiteren bzw. neu entwickelten Trägern Delta-3, -4 und Atlas-3, -5 vier erfolgreiche Erststarts zwischen 1998 und 2002 durchführen und damit auch direkte Ariane-5-Konkurrenten auf den Trägermarkt bringen.

Das Space Shuttle wird nur noch für die Unterstützung der Arbeiten an der Internationalen Raumstation eingesetzt und ist damit nicht am Start von kommerziellen Satelliten beteiligt. Nach dem Unfall beim Wiedereintritt von Columbia am 01.02.2003 wurden zahlreiche Modifikationen durchgeführt, um die Sicherheit des Systems zu erhöhen. Schließlich wurde der Träger am 26.07.2005 für die nächste ISS-Mission gestartet und kehrte erfolgreich zurück.

**Russland:** Die Abbildung A.1 verdeutlicht den Zuwachs der Marktanteile russischer Träger. Dies geht offensichtlich auf Kosten der europäischen und amerikanischen Marktanteile. Hierbei ist zu bemerken, dass diese Träger, wie die folgende Tabelle A.1 zeigt, von internationalen Firmen vermarktet werden. Auch Arianespace wird zukünftig die Soyuz-Rakete vermarkten und von Kourou aus starten. Der erste Start mit dem Kommunikationssatelliten Optus D2 ist in 2007 geplant [2].

Träger	Anbieter	Beteiligte Länder	Seit
Proton	ILS	Russland, USA	1995
Soyuz	Starsem	Europa, Russland	1996
Rockot	Eurockot	Deutschland, Russland	1995
Zenit	Sea Launch	Norwegen, Russland, Ukraine, USA	1995

Tabelle A.1: Internationale Unternehmen, die russische Träger vermarkten.

**China:** Auf den ersten Blick scheinen die chinesischen Träger vom kommerziellen Markt zu verschwinden, weil die Anteile an Starts und Nutzlasten in den Jahren 1992-2003 stark zurückgegangen sind (Abbildung A.1). In den ersten kommerziellen Starts erwiesen sich die chinesischen Trägerraketen als unzuverlässig und verloren rasch Startaufträge. Bei Betrachtung der Anteile an Einnahmen jedoch, ist ein Anteil von durchschnittlich 5% in den Jahren 1995 bis 2003 zu beobachten. Dies liegt zum einen daran, dass die Zuverlässigkeiten und die Nutzlastkapazitäten der chinesischen Träger gestiegen sind. Die markanten Zeichen dafür sind z.B. der Träger Long March CZ-3B mit 4.85 t GTO Nutzlastkapazität und der Start des ersten chinesischen Astronauten Yang Liwei am 15.10.2003 mit der Trägerrakete CZ-2F. Die Fortschritte im Raumfahrtprogramm von China und das Raumfahrtetat von 2,2 Milliarden US\$ [34] lassen für die Zukunft stärkere Konkurrenz aus China erwarten.

**Sea Launch / Ukraine:** Wie aus der Abbildung A.1 ersichtlich, ist die Beteiligung der Ukraine mit der Zenit-Rakete am kommerziellen Satellitenstartmarkt sehr klein. Nach der Gründung des internationalen Unternehmens Sea Launch jedoch, das die Zenit-Rakete von einer umgebauten Ölplattform direkt am Äquator startet, ist der Anteil auf 7% von allen kommerziellen Starts gestiegen. Von diesem idealen Startplatz hat die Zenit-Rakete mit der russischen Block-DM-Oberstufe eine GTO-Nutzlastkapazität von 6 t und ist damit ein ernst zu nehmender Konkurrent, wie die 14 Starts in dem Zeitraum 1999-2004 belegen.

**Indien:** Die indischen Trägersysteme Geostationary bzw. Polar Satellite Launch Vehicle (GSLV und PSLV) werden noch für Teststarts und staatliche Aufträge eingesetzt. Allerdings ist zukünftig damit zu rechnen, dass auch diese Träger für den Start kleinerer, kommerzieller Satelliten verfügbar sein werden.

**Japan:** Die hohen Startkosten und die technischen Probleme, insbesondere mit dem sehr fortschrittlichen zentralen Hauptstromtriebwerk LE-7, verhinderten den kommerziellen Erfolg der japanischen Trägerrakete H2. Ihr Betrieb wurde nach dem letzten Start in 1999 eingestellt. Der Nachfolger H2-A basiert auf weniger anspruchsvoller Technologie und wurde für geringere Startkosten und höhere Zuverlässigkeit entwickelt, steht aber bisher nicht international zur Verfügung. H2-A startete 15 Monate nach dem Fehlstart in 2004 erfolgreich am 26.02.2005.

## A.2 Aktuelle Trägerraketen

Die nächste Tabelle A.2 listet die derzeit wichtigsten Trägersysteme mit den wichtigsten Daten auf.

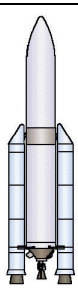

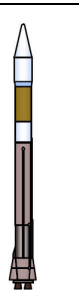
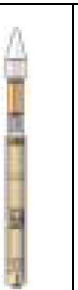



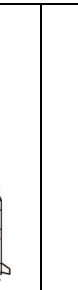
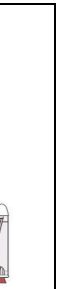
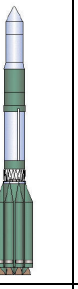


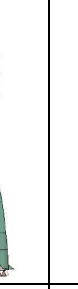
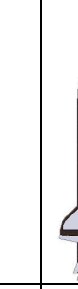



									
Träger	<b>Ariane-5</b>	<b>Atlas-2AS<sup>4</sup></b>	<b>Atlas-3</b>	<b>Atlas 5</b>	<b>Delta-2</b>	<b>Delta-3</b>	<b>Delta-4</b>	<b>GSLV</b>	<b>H2A</b>
Land	Europa	USA	USA	USA	USA	USA	USA	Indien	Japan
GTO <sup>5</sup>	6,8	3,7	4,6	9,0	2,1	3,8	4,2-13,4	2,5	4,1-7,5
LEO <sup>6</sup>	18	8,6	10,7	20,1	6,0	8,3	8,1-23	5	10-17
ER % <sup>7</sup>	82	100	100	100	99	33	100	100	83
SK <sup>8</sup>	150-180	90-105	90-105	75-170	50-60	75-90	75-170	35-45	75-95
									
Träger	<b>LM 2F</b>	<b>LM 3A</b>	<b>LM 3B</b>	<b>Proton</b>	<b>PSLV</b>	<b>Rockot</b>	<b>Soyuz</b>	<b>Zenit 3SL</b>	<b>Space Shuttle</b>
Land	China	China	China	Russl.	Indien	Russl.	Russl.	Ukraine	USA
GTO	3,5	2,5	5,2	6,5-7,8	0,8	-	1,5	5,3	5,8
LEO	9,5	7,2	11,2	19,8-21	3,7	1,8	4,1-5,5	13,7	24,4
ER %	100	100	80	93	100	100	97	91	98
SK			50-70	90-112	20-25	12	30-50	75-95	600

Tabelle A.2: Aktuelle Trägersysteme auf dem kommerziellen Markt<sup>9</sup> [3],[4],[51].

Aus der Zusammenstellung der aktuellen Trägersysteme werden mehrere Entwicklungstendenzen ersichtlich. Die Nutzlastkapazitäten der Träger werden durch Weiter- und Neuentwicklungen erhöht (z.B. Atlas). Durch modularen Aufbau (z.B. Delta-4) bzw. durch die Kombination von mehreren, gemeinsamen Triebwerken und Stufen werden Trägerfamilien erstellt (z.B. Long March 2A/C/D/E/F, 3, 3A/B, 4, 4B) und für eine ganze Palette von Nutzlastgrößen angeboten. Um einen direkten Vergleich dieser Träger zu ermöglichen, sind in der Tabelle A.2, die geschätzten Startkosten pro kg Nutzlast in die geostationäre Transferbahn für die derzeit wichtigsten Trägersysteme dargestellt.

<sup>4</sup> Zum letzten Mal gestartet 31.08.2004.

<sup>5</sup> Nutzlast zur geostationären Umlaufbahn, Tonnen.

<sup>6</sup> Nutzlast zur erdnahen Umlaufbahn, Tonnen.

<sup>7</sup> Erfolgsrate, letzte 10 Jahre.

<sup>8</sup> Geschätzte Startkosten, Millionen US\$

<sup>9</sup> Maßstäbe der Bilder ungleich.

### A.3 Entwicklungstrends der Satellitenmassen

Neben der Entwicklung des kommerziellen Raumtransportmarktes, wird die Trägerentwicklung auch durch die Entwicklungen in den Nutzlasten angetrieben. Diese Entwicklung ist in der folgenden Abbildung A.2 dargestellt.

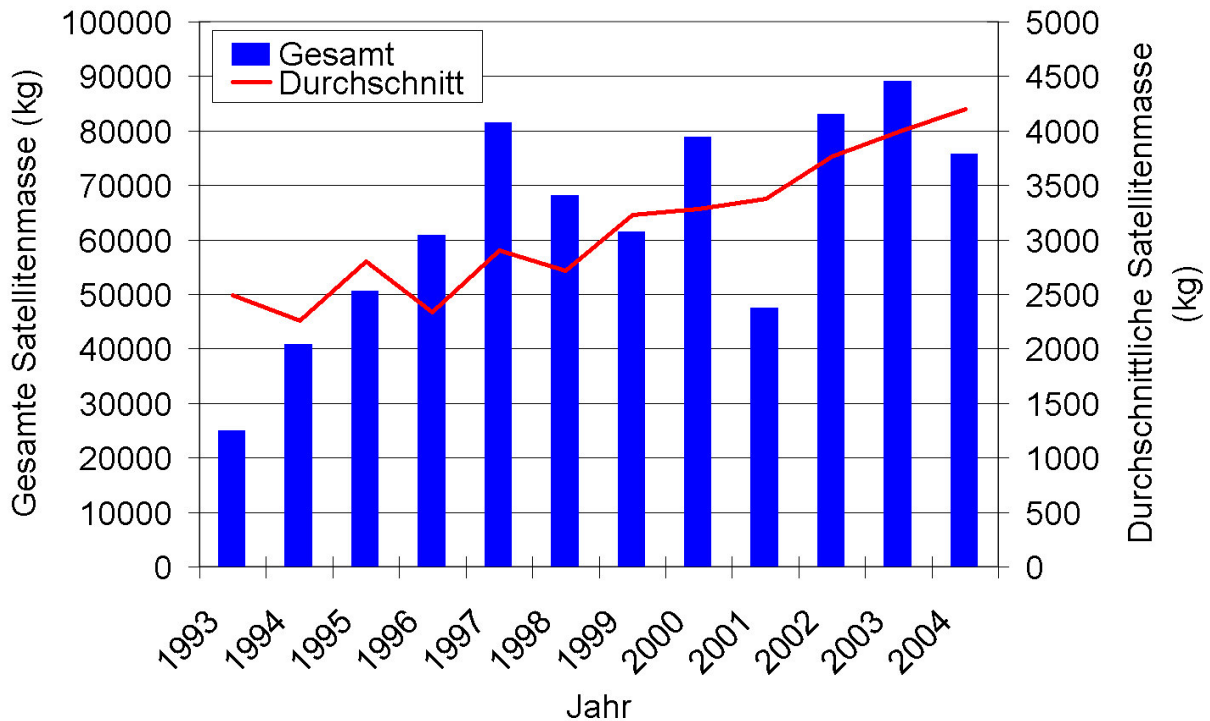


Abbildung A.2: Summe und Durchschnitt gestarteter Satellitenmassen [5].

Die Grafik verdeutlicht, dass die durchschnittliche Masse von Satelliten stetig zunimmt. Darüber hinaus ist in dieser Darstellung gut zu erkennen, dass trotz dem Rückgang der Startzahlen, die Masse, die zur Umlaufbahn transportiert wird, steigt.

## A.4 Wiederverwendbare Trägersysteme

Derzeit ist Space Shuttle das einzige wiederverwendbare Trägersystem im Betrieb. Es ist jedoch ein sehr komplexes, bemanntes, teilweise wiederverwendbares Trägersystem, das eine sehr aufwändige Wartung zwischen zwei Missionen erfordert. Daher sind die Startkosten mit etwa 500 M\$ [51] sehr hoch. Deswegen werden weltweit verschiedene Studien durchgeführt, um ein wiederverwendbares, zukünftiges Trägersystem zu definieren, das die Transportkosten wesentlich reduzieren soll.

Einige wichtige nationale und europäische Studien in diesem Bereich sind das Deutsche Hyperschalltechnologieprogramm [14],[63], Future European Space Transportation Investigations Programme FESTIP [25],[19], Technologien für Zukünftige Raumtransportsysteme TETRA [7], Future Launcher Preparatory Programme FLPP [75] und Ausgewählte Systeme und Technologien für zukünftige Raumtransportsystem-Anwendungen ASTRA [7].

Im Rahmen des Deutschen Hyperschalltechnologieprogramms (1986-1995) wurde das Trägerkonzept Sänger untersucht. In diesem Zeitraum wurden auch verschiedene, teilweise wiederverwendbare Ariane-Nachfolgerkonzepte (z.B. Ariane-X) untersucht. Die nächste Abbildung A.3 zeigt ein CAD-Modell von Sänger und die verschiedenen Ariane-X Versionen.

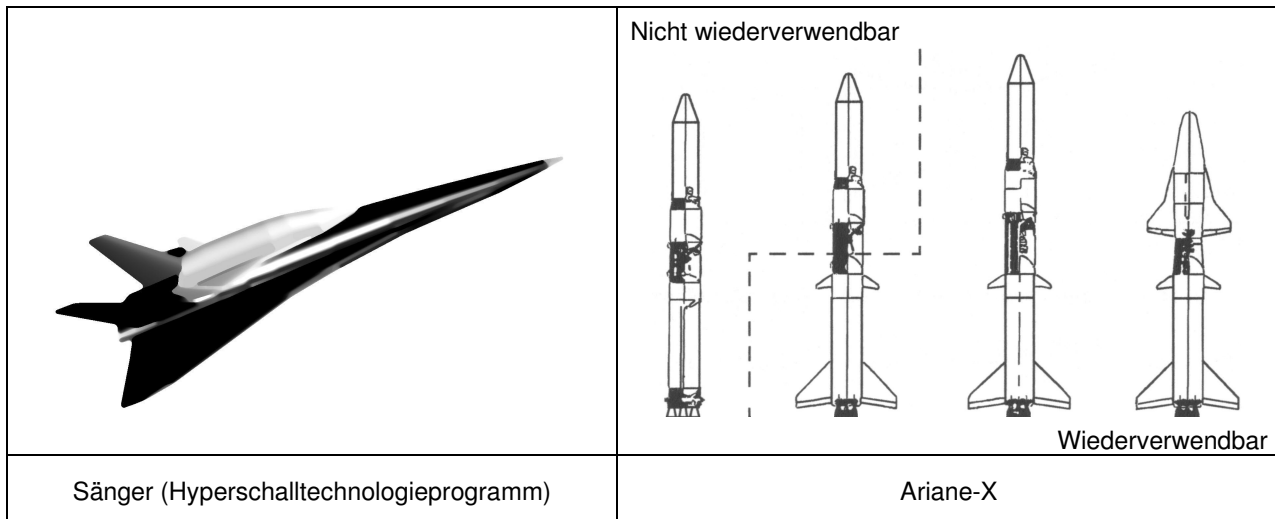


Abbildung A.3: Trägerkonzepte Sänger und Ariane-X [84].

Sänger ist ein voll wiederverwendbarer, horizontal startender und landender Träger. Es besteht aus einer geflügelten Unterstufe mit luftatmenden Triebwerken und einer ebenfalls geflügelten, raketenangetriebenen Oberstufe.

In der Studie FESTIP (1993-1998) wurden zahlreiche Trägerkonzepte untersucht. Diese beinhalten raketen- und luftatmerangetriebene, voll oder teilweise wiederverwendbare, horizontal oder vertikal startende bzw. landende, ein- oder zweistufige und suborbitale Trägerkonzepte. Die nachstehende Abbildung A.4 gibt eine Zusammenstellung der in dieser detaillierten Studie untersuchten Trägerkonzepte.

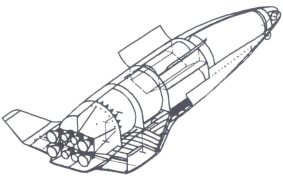
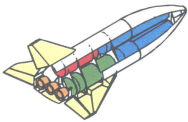
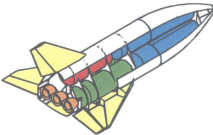
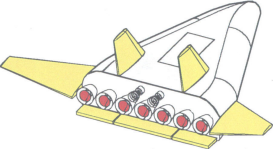
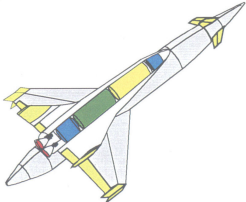
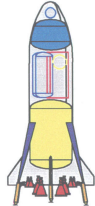
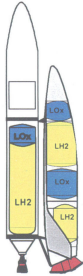
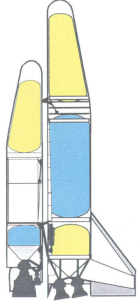
			
VTOHL TSSO FSSC 1 & 2	HTOL SSTO FSSC 4	HTOL Suborbital FSSC 15	VTOHL SSTO FSSC 5
			
HTOL TSTO FSSC 12	VTOL SSTO FSSC 3	VTOHL TSTO <sup>10</sup> FSSC 9	VTOHL TSTO FSSC 7

Abbildung A.4: In FESTIP untersuchte wiederverwendbare Trägerkonzepte [19].

Dabei wurden die Aspekte Antrieb, Aerothermodynamik, Materialtechnologie, Betrieb, Kosten, Subsysteme, Strukturen im Detail analysiert. Die Konzepte FSSC 5 und FSSC 3 sind ähnliche Konzepte, wie die US Projekte X-33 und Delta-X respektive. Aus dem Konzept FSSC 15 entstand später das suborbitale Trägerkonzept Hopper.

Das Projekt TETRA (1997-2003) war ein nationales Programm in Kooperation mit NASA und war u.a. der Technologieentwicklung in den Bereichen Heiße Strukturen, Thermalschutzsysteme, Aerothermodynamik, Lenkung und Regelung bzw. Missions- und Systemanalyse gewidmet. Das Leitkonzept war der Raumgleiter X-38, welcher mehrere atmosphärische Testflüge absolvierte.

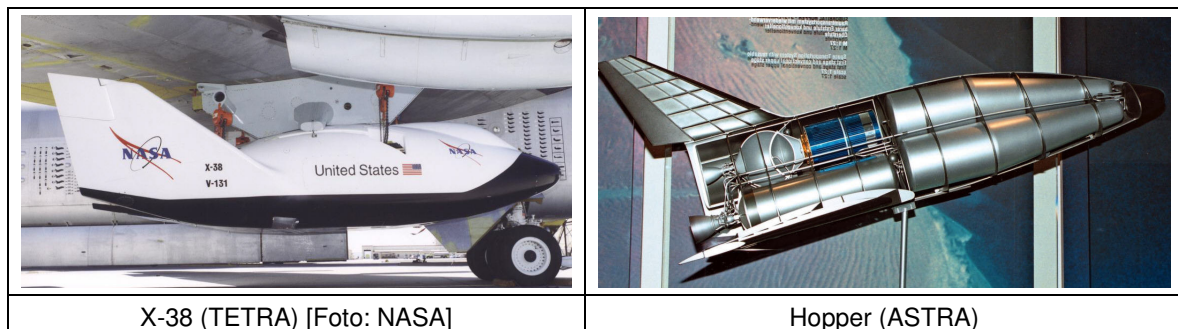


Abbildung A.5: Leitkonzepte aus den Programmen TETRA und ASTRA.

Das Ziel des europäischen Programms FLPP ist es, die technologischen Kompetenzen zu entwickeln und zu bündeln, um eine Entscheidung über ein neues europäisches Trägersystem für den Einsatz ab etwa 2020 zu treffen und die Wettbewerbsfähigkeit im Raumtransportmarkt zu

<sup>10</sup> Teilweise wiederverwendbar.



sichern. Die erste Phase (2004-2006) beinhaltet die Auswahl je eines Systemkonzepts wiederverwendbarer Startfahrzeuge für zwei Referenzmissionen und die Vorentwicklung experimenteller Fahrzeuge. In der zweiten Phase (2006-2009) sollen neue Trägertechnologien durch Boden- und Flugversuche erprobt und eine Missionsvorauswahl für konkurrierende wiederverwendbare und nicht wiederverwendbare Systemkonzepte durchgeführt werden. Schließlich sollen, in der dritten Phase (2009-2011) die Missionsanforderungen bestätigt und festgelegt, Triebwerksversuche abgeschlossen und eine endgültige Empfehlung eines Systemkonzepts ausgesprochen werden. Deutschland ist in der ersten Phase von FLPP nicht involviert. Stattdessen wurde in 2000 das nationale Projekt ASTRA gestartet, mit dem Ziel, Beiträge für den autonomen und wettbewerbsfähigen Zugang Europas zum Weltraum zu leisten und die industriellen und wissenschaftlichen Einrichtungen in Deutschland auf eine kompetente Mitwirkung in den späteren Phasen des FLPP vorzubereiten. ASTRA beinhaltet die Analyse von verschiedenen Trägerkonzepten, Lebenszykluskosten, erforderlichen Technologien für den Entwurf, Entwicklung, Betrieb und Planung und Durchführung von Boden- und Flugexperimenten. Neben einem vertikal startenden Trägerkonzept mit zwei geflügelten Booster-Stufen ist ein Leitkonzept dieser Studie der Hopper, ein suborbitaler, raketenangetriebener, wiederverwendbarer Träger mit einer kryogenen Oberstufe, welches vom FESTIP-Konzept FSSC-15 abgeleitet ist. Die Studie hatte gezeigt, dass für Europa ein suborbitales, horizontal startendes und einstufiges Konzept, unter den untersuchten Varianten, die attraktivste ist [105]. Hierfür wurde im Rahmen des Programms ASTRA der Flugdemonstrator Phoenix entwickelt [36].



Abbildung A.6: Der Technologiedemonstrator Phoenix.

Ziel des Projekts ist die Entwicklung und Erprobung von Technologien und Prozeduren für den Betrieb eines wiederverwendbaren Raumtransporters. Phoenix demonstrierte eine automatische Präzisionslandung erfolgreich am 6. Mai 2004 in Schweden (Abbildung A.6).

Auch in den USA werden zahlreiche groß angelegte Studien durchgeführt, um verschiedene zukünftige Trägersystem-Konzepte zu analysieren. Das zurzeit laufende Programm Space Launch Initiative SLI [69] hat zum Ziel, die Raumtransportausgaben der NASA von derzeit 5 Milliarden US\$ jährlich [69], durch höhere Zuverlässigkeit, Sicherheit und günstigere Betriebskosten zu reduzieren. Das Forschungsvorhaben besteht aus dem New Generation Launch Technology Program NGLT [68] und dem Programm Orbital Space Plane OSP [107].

In [71] wird die Entwicklung und Evolution dieser Forschungsprogramme erläutert. Mitte der 90er Jahre herrschte in den USA der Konsens, dass das Space Shuttle ein hohes Unfallrisiko

trägt und sehr teuer im Betrieb und schwierig zu warten ist, und dass die Nation deswegen einen Nachfolger dafür braucht. Nach einer Ausschreibung wählte die NASA das von Lockheed Martin vorgeschlagene einstufige Konzept Venture Star. Vorher sollte das etwa 1:3 Konzeptfahrzeug X-33 gebaut und damit die Technologie demonstriert werden. Die Abbildung A.7 illustriert einen Größenvergleich zwischen dem Experimentalfahrzeug X-33, VentureStar und dem Space Shuttle.

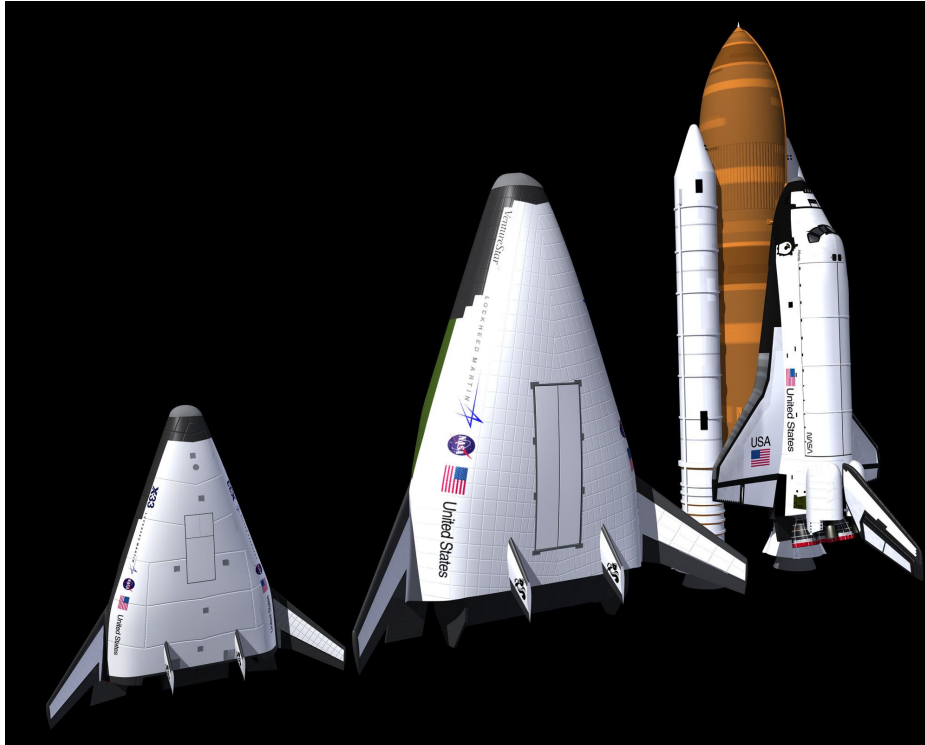


Abbildung A.7: Größenvergleich X33, VentureStar und Orbiter (Quelle: Lockheed Martin).

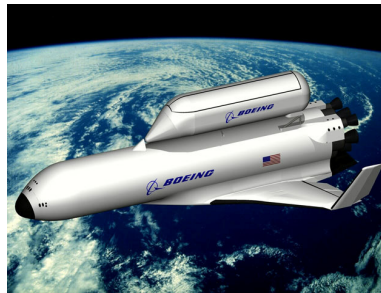
Das sehr ambitioniert ausgelegte Fahrzeug war vollständig wiederverwendbar und einstufig. Um die Strukturmasse zu minimieren wurden die Tanks aus Kohlefaser hergestellt und als Antrieb wurde ein Linear-Aerospire-Raketentriebwerk vorgesehen. Bis Ende 1999 waren 15 Flüge geplant. 2000 sollte dann die Entscheidung für den Bau von VentureStar gefällt werden. Obwohl die Triebwerksentwicklung sehr erfolgreich war, versagte der Kohlefasertank beim Belastungstest. Während der Trägermarkt stagnierte stiegen die Entwicklungs- und Transportkostenschätzungen und schließlich wurde das Programm in 2001 gestrichen. Danach wurde das 4,5 Milliarden US\$ Programm Space Launch Initiative SLI gestartet [26]. Ziel dieses umfangreichen, fünfjährigen Technologieprogramms ist es, Technologien zu entwickeln, die zur Entwicklung eines zukünftigen wiederverwendbaren Raumtransporters mit hoher Sicherheit, Verlässlichkeit und niedrigen Betriebskosten benötigt werden. Zentralpunkt von SLI ist das "2<sup>nd</sup> Generation RLV", was durch die folgenden Eigenschaften definiert ist:

- Privater Betrieb, kommerzielle Verfügbarkeit
- Kosten pro kg Nutzlast: 2000 \$ (aktuell 20 000 \$)
- Wahrscheinlichkeit des Mannschafts-Verlustes: 1/10000 (Space Shuttle ~1/300)
- Signifikante Technologieentwicklung und Bau von drei Fahrzeugen

Geplant war die Entscheidung über den Bau eines Fahrzeuges in 2006 und die Aufnahme des Betriebs in 2010. Die Abbildung A.8 betrachtet einige NGLT-Konzepte, die in den USA von der Industrie vorgeschlagen wurden.



Lockheed Martin



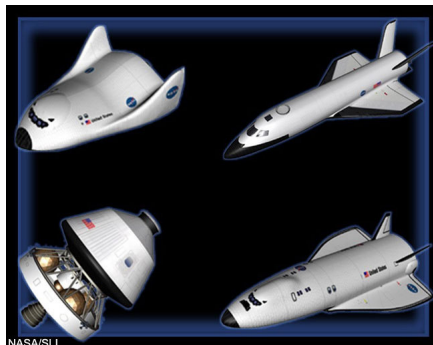
Boeing



Orbital Sciences

Abbildung A.8: Konzeptvorschläge der Industrie für zukünftige RTS, USA.

2003 wurde SLI jedoch neu strukturiert, das "2<sup>nd</sup> Generation RLV" wurde gestrichen und der ISS Mannschaftstransporter Orbital Space Plane (OSP) [107] als Schwerpunkt definiert. OSP soll vorhandene Technologien ausnutzen und eine kommerzielle Verfügbarkeit wird nicht gefordert. Dabei ist die Konfiguration von OSP offen und verschiedene Varianten werden derzeit untersucht und ausgewertet. Die folgende Abbildung A.9 zeigt einige der verschiedenen OSP-Konzepte.



NASA



Orbital Sciences

Abbildung A.9: Verschiedene Orbital Space Plane Fahrzeugkonzepte.

Zum Transport des OSP in die Umlaufbahn wird ein EELV-Träger (Evolved Expendable Launch Vehicle) wie die Delta-4 Rakete benutzt. Space Shuttle wird bis zur Ausmusterung noch parallel eingesetzt. Ein technologischer Durchbruch wird dann durch die kontinuierliche Finanzierung des Programms Next Generation Launch Technology (NGLT) erwartet.

Diese Entwicklung brachte das RLV-Programm der USA, nach sehr hochgesteckten, auf zukünftigen Technologien basierenden Zielen, auf einen Weg, der auf vorhandenen Technologien basiert, finanziell überschaubarer ist, die Projektrisiken minimiert und in erster Linie zur reibungslosen Fortsetzung des ISS-Betriebs dient. Die Entwicklung neuer Technologien für zukünftige wiederverwendbare Raumfahrzeuge wird im Rahmen des NGLT-Programms durchgeführt. Dieses Technologieprogramm beinhaltet die Entwicklung von wiederverwendbaren Raketentriebwerken, hypersonischen luftatmenden Triebwerken und Fahrzeugstrukturen und eine breite Palette an Technologieentwicklungen für verschiedene Fluggerät-Architekturen [68].

Aufgrund der Ähnlichkeit des untersuchten Trägersystems zu Hopper, der multidisziplinären Vorgehensweise und Aktualität soll hier auch die Studie ARTS [119] (Advanced Reusable Transportation System) aus den USA erwähnt werden. Die folgende Abbildung A.10 veranschaulicht das Trägerkonzept und einen Größenvergleich mit Hopper und dem Space Shuttle.

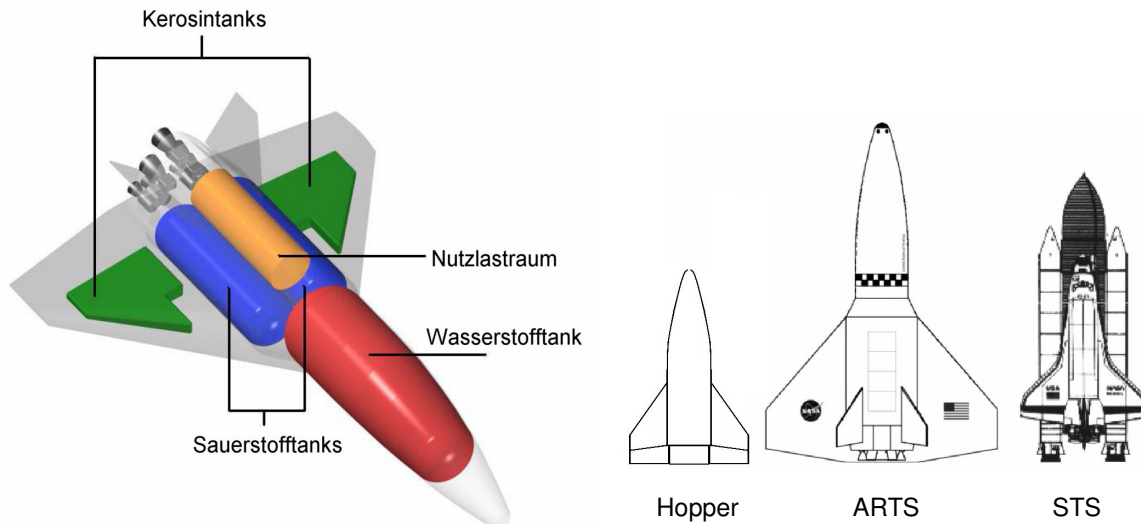


Abbildung A.10: Das Trägerkonzept ARTS – Größenvergleich und Konzeptentwurf [119].

Der unbemannte, autonome und voll wiederverwendbare Raketenträger nutzt bewährte Triebwerke, startet horizontal von einer elektromagnetischen Schiene und setzt seine Nutzlast im Scheitelpunkt (ca. 130 km) seiner suborbitalen Aufstiegsbahn aus. Die folgende Tabelle A.3 gibt einen Überblick der wichtigsten Parameter des Trägers.

Nutzlast	22 t (27,5 t mit 4 SSME )
Zielbahn	200 km Kreisbahn mit 28,5° Inklination
Triebwerke	3 Shuttle Haupttriebwerke (SSME) 2 RD-180
Trockenmasse	142 t
Startmasse	1857 t
Entwicklungskosten	11-12 Milliarden US\$

Tabelle A.3: Eckdaten des Trägerkonzept ARTS.

In dieser multidisziplinären Studie werden die Fachdisziplinen Konzeptioneller Entwurf, Massenanalyse, Aerodynamik, Bahnoptimierung und Schätzung der Entwicklungskosten berücksichtigt. Für die Aerodynamikanalyse werden für die Unter- und Überschall-Flugbereiche zwei Werkzeuge eingesetzt und die ermittelten Daten bei fotografischer Fahrzeugskalierung als konstant angenommen. Die Massenschätzung wird mit einem iterativen Verfahren in Abhängigkeit vom Startgewicht durchgeführt. Die für die Aerodynamik- und Massenanalyse benötigten Geometriedaten werden automatisch im CAD-System produziert. Für die Bahnoptimierung wird das in den USA verbreitete Verfahren POST eingesetzt und dabei der Rückkehrflug nicht betrachtet.

## Anhang B: Visualisierung – Anwendungsbeispiele

Die optimierte Gesamtmission Hopper (Kap.6.7) wurde mit einer VRML-Animation visualisiert. Dabei wurden die Triebwerksstrahlen und -geräuschpegel entsprechend der Schubdrosselung skaliert. Die Abbildung B.1 illustriert diese Animation.

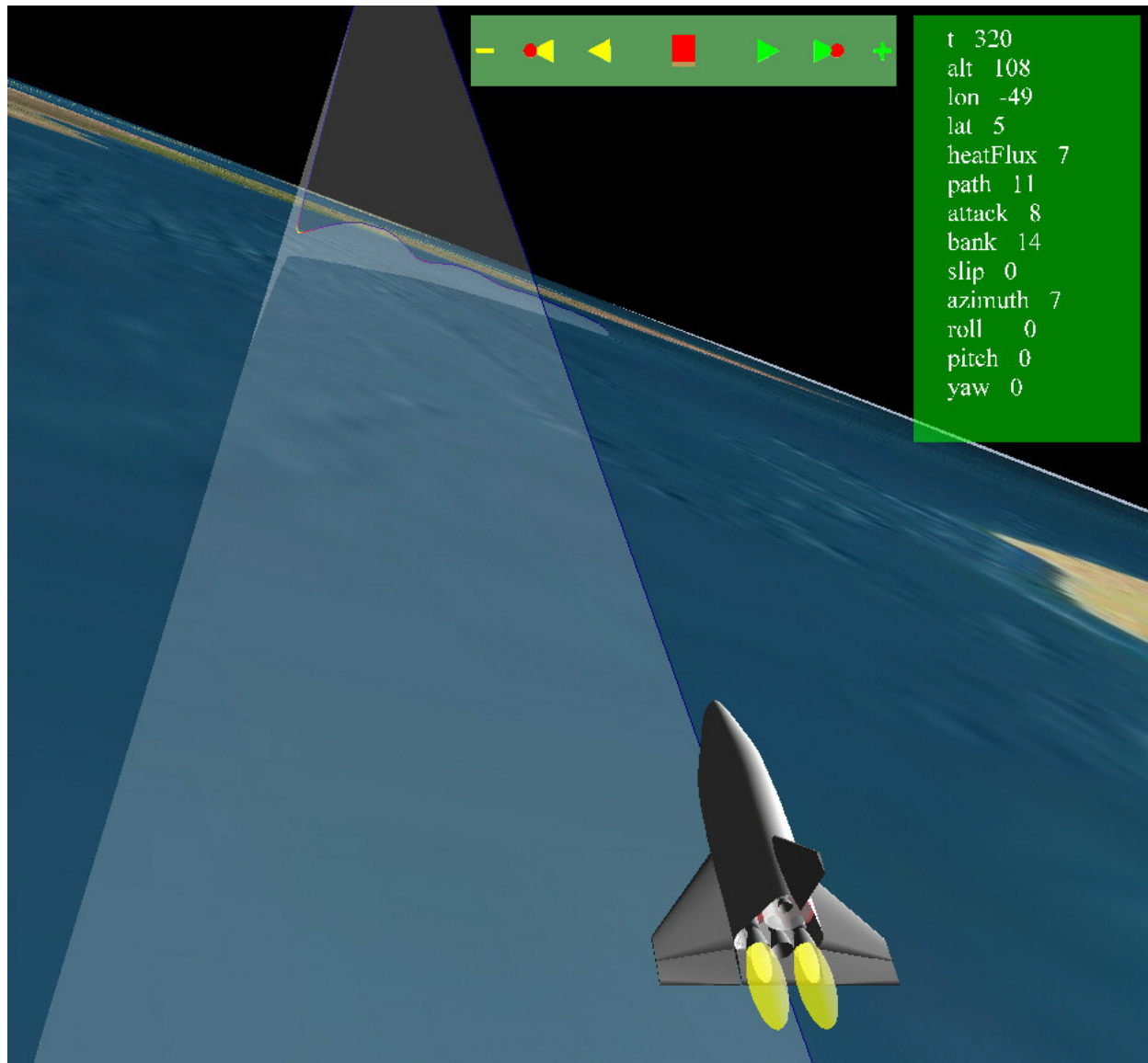


Abbildung B.1: VRML-Animation der gesamten Hopper-Mission.

In der Abbildung befindet sich Hopper im Aufstiegsflug kurz vor dem Abschalten der Triebwerke. Das mittlere Triebwerk ist bereits abgeschaltet, um die axiale Schublastgrenze einzuhalten. Die Anzeige rechts oben gibt den Flugzustand an und im oberen Bildbereich ist die Rückkehrbahn zu sehen.

Die folgenden Bilder zeigen weitere Momentaufnahmen aus Virtuelle-Realität-Animationen von Simulationsergebnissen (Beiträge des Autors zu IRS-Projekten mit externen Kooperationspartnern). Sie demonstrieren die breite Anwendungspalette der Visualisierung mit der VRML-Technologie in Missions- und Systemanalysen von Raumtransportsystemen.

Das erste Beispiel in der Abbildung B.2 visualisiert den spiralförmigen Landeanflug des Raumgleiters X-38 am Gleitschirm und die zweite den Abdeckungsgebiet einer rotierenden Antenne aus dem Orbit (Abbildung B.3).

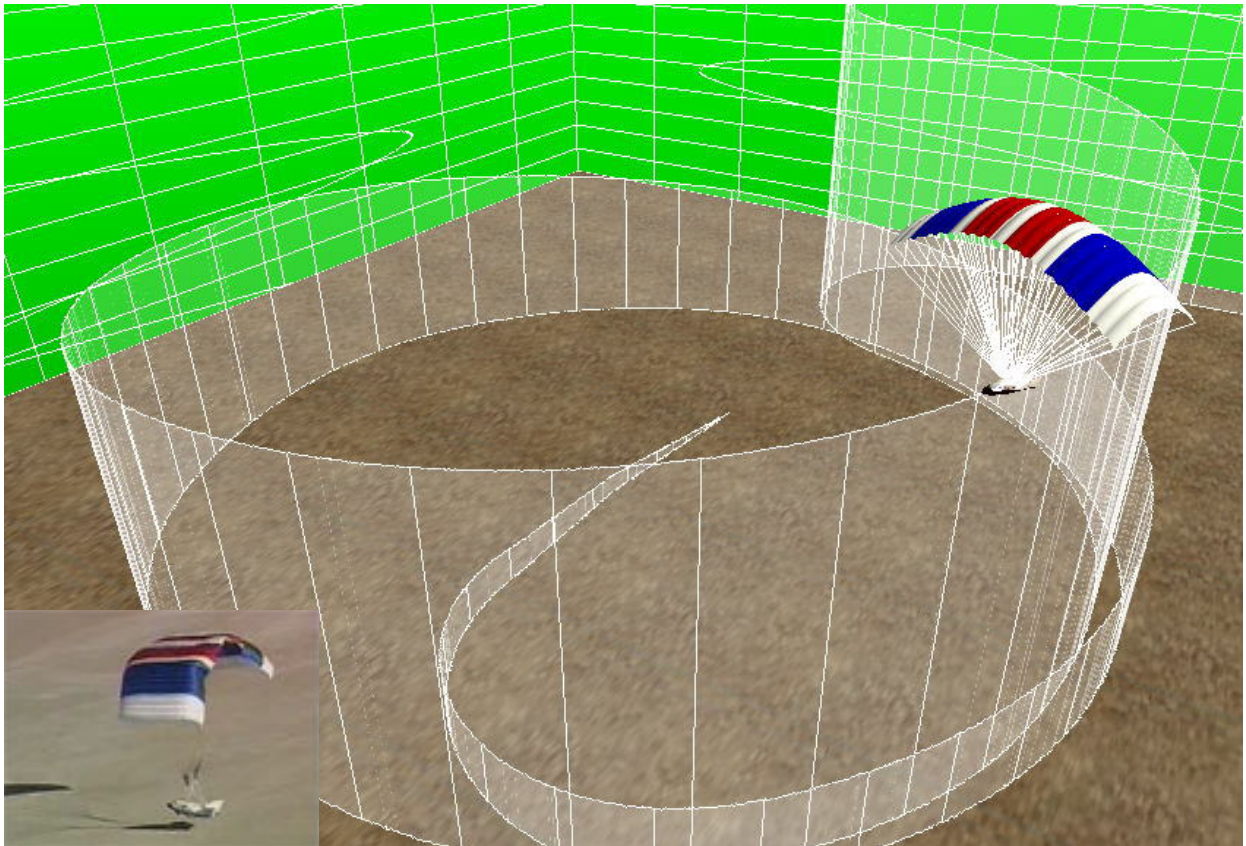


Abbildung B.2: Landeanflug der X-38 am Gleitschirm.  
(Simulationsdaten: DLR, Foto links unten und X-38 Modell: NASA).

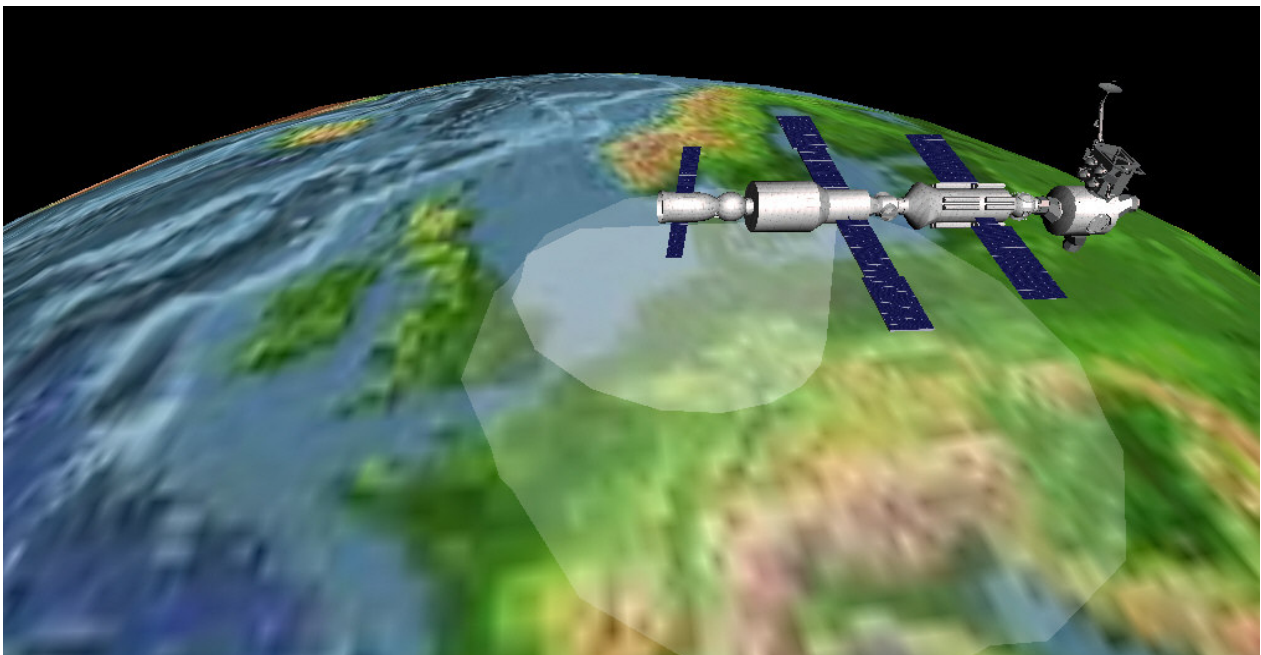


Abbildung B.3: Abdeckungsgebiet einer Antenne aus dem Orbit.

Die folgenden Abbildungen Abbildung B.4 und Abbildung B.5 stellen zwei Nutzlastrückfuhr-Missionen von der Internationalen Raumstation dar. Bei der ersten Anwendung wird der Ausspulgvorgang einer Seilgestützten Rückkehrkapsel und in der zweiten die Sichtbarkeitsanalyse der IRDT-Kapsel dargestellt.

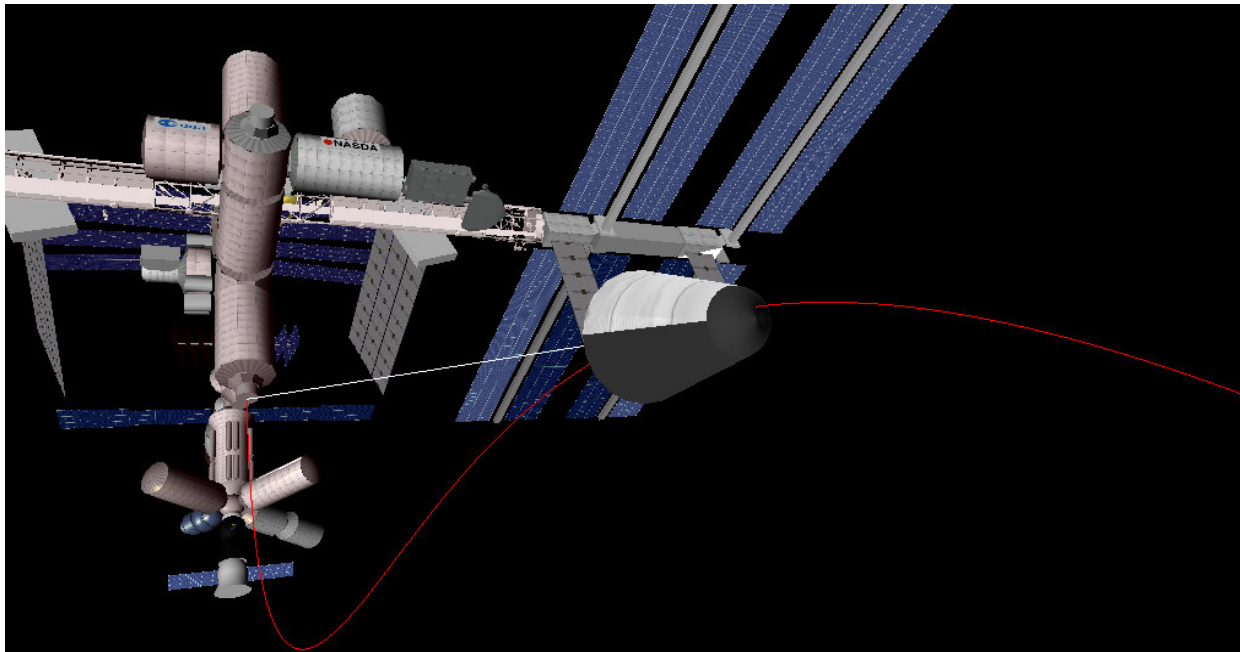


Abbildung B.4: Ausspulgvorgang einer Seilgestützten Rückkehrkapsel (Simulationsdaten [125]).

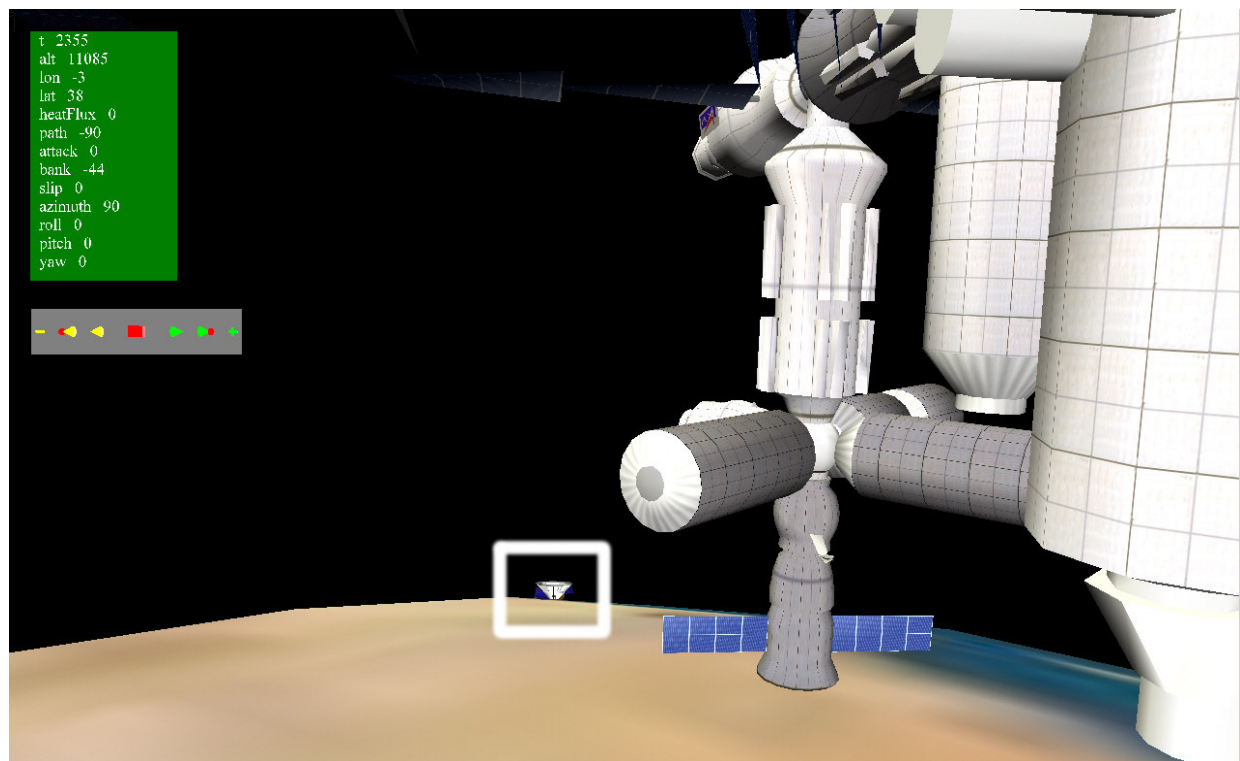


Abbildung B.5: Sichtbarkeitsanalyse der IRDT-Kapsel bei der Nutzlastrückfuhr von der Internationalen Raumstation [6] (ISS-Modell: NASA).

## Lebenslauf

### Persönliche Daten

Name:	Osman Kalden
Geboren:	28.06.1966, Istanbul
Staatsangehörigkeit:	Deutsch, Türkisch
Familienstand:	Verheiratet
Ehefrau:	Aysegül Kalden, Geigerin

### Ausbildung

	07.1985	Schulabschluss, American Robert College, Istanbul
09.1985	– 07.1989	Maschinenbau-Studium Bosphorus Universität, Istanbul Abschluss: Bachelor of Sciences
11.1989	– 02.1998	Luft- und Raumfahrt-Studium Technische Universität Berlin (TUB) Abschluss: Diplom-Ingenieur
12.1992	– 03.1995	Tutor mit Lehraufgaben für die Lehrveranstaltung "Informationstechnik für Ingenieure" Institut für Schiffs- und Meerestechnik, TUB
03.1995	– 09.1996	Tutor mit Lehraufgaben für die Lehrveranstaltung "EDV 1" Hermann-Föttinger-Institut für Strömungsmechanik, TUB

### Berufliche Tätigkeiten

09.1991	– 06.1998	Datenbank-, Softwareentwicklung, EDV-Beratung Vacucast Feinguss GmbH, Berlin
09.1995	– 06.1998	Freiberuflicher Softwareentwickler
06.1998	– 10.1998	Wissenschaftlicher Mitarbeiter Hermann-Föttinger-Institut für Strömungsmechanik, TUB
10.1998	– 01.2004	Wissenschaftlicher Mitarbeiter, Dozent Institut für Raumfahrtssysteme, Universität Stuttgart
02.2004	–	VEGA Informations-Technologien GmbH, Darmstadt



