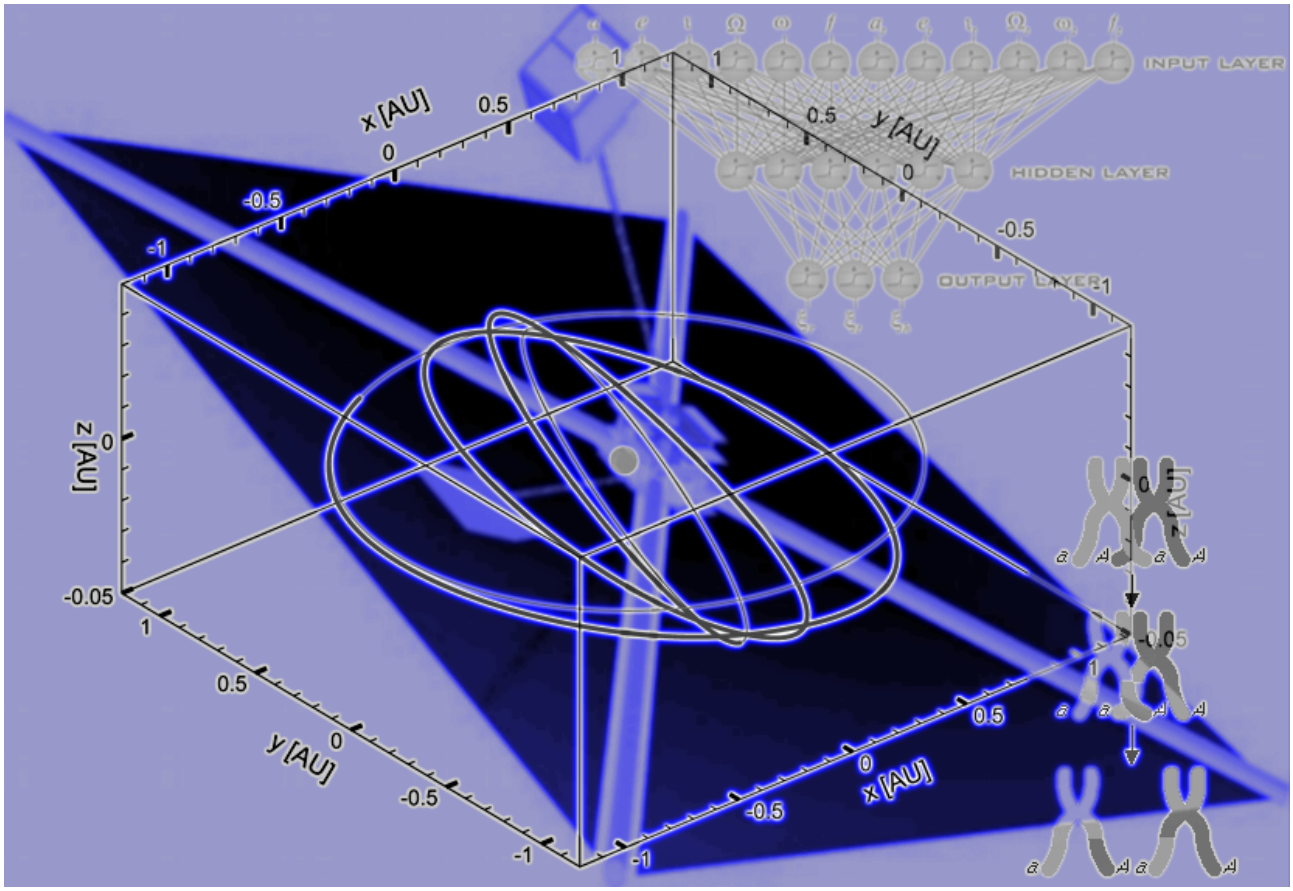
Bernd Dachwald

Low-Thrust Trajectory Optimization and Interplanetary Mission Analysis Using Evolutionary Neurocontrol



Universität der Bundeswehr München
Fakultät für Luft- und Raumfahrttechnik
Institut für Raumfahrttechnik

Low-Thrust Trajectory Optimization and Interplanetary Mission Analysis Using Evolutionary Neurocontrol

Dipl.-Ing. Dipl.-Wirt.Ing. Bernd Dachwald

Vollständiger Abdruck der bei der
Fakultät für Luft- und Raumfahrttechnik
der Universität der Bundeswehr München
zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

eingereichten Dissertation

Vorsitzender	: Prof. Dr.-Ing. Axel Schulte
1. Berichterstatter	: Prof. Dr. rer. nat. Bernd Häusler
2. Berichterstatter	: Prof. Dr. sc. math. Joachim Gwinner

Diese Dissertation wurde am 5. Juni 2003 bei der Universität der Bundeswehr München, 85577 Neubiberg eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 5. Juni 2003 angenommen.

Tag der Prüfung : 2. April 2004

to my family

If the human brain were so simple that we could understand it,
we would be so simple that we couldn't.
Emerson Pugh

Preface and Acknowledgements

This work originates from my research at the "Deutsches Zentrum für Luft- und Raumfahrt"[1] (DLR), "Institut für Weltraumsensorik und Planetenerkundung"[2] (until December 2002), and "Institut für Raumsimulation"[3] (since January 2003), Cologne, in cooperation with the "Institut für Raumfahrttechnik" at the "Universität der Bundeswehr München", Neubiberg.

Parts of this work have been presented in "Performance Requirements for Near-Term Interplanetary Solar Sailcraft Missions" at the 6th International AAAF Symposium on Space Propulsion (2002, Versailles, France) [28], in "Optimization of Interplanetary Rendezvous Trajectories for Solar Sailcraft Using a Neurocontroller" at the AIAA/AAS Astrodynamics Specialist Conference 2002 (Monterey, USA) [26], and in "Verwendung eines neuronalen Reglers und evolutionärer Algorithmen zur Berechnung optimaler interplanetarer Sonnensegler-bahnen" at the Deutscher Luft- und Raumfahrtkongress 2002 (Stuttgart, Germany) [24]. Additionally, parts of this work have been submitted as "Optimization of Interplanetary Solar Sailcraft Trajectories Using Evolutionary Neurocontrol" to the AIAA Journal of Guidance, Control, and Dynamics [22]. Furthermore, parts of this work are intended for presentation in "Interplanetary Mission Analysis for Non-Perfectly Reflecting Solar Sailcraft Using Evolutionary Neurocontrol" at the AAS/AIAA Astrodynamics Specialist Conference 2003 (Big Sky, USA) [25], in "Multiple Rendezvous and Sample Return Missions to Near Earth Asteroids Using Solar Sailcraft" at the 5th IAA International Conference on Low-Cost Planetary Missions (2003, Noordwijk, The Netherlands) [29], and in "Solar Sailcraft of the First Generation" at the 54th International Astronautical Congress (2003, Bremen, Germany) [27, 85].

I especially thank Prof. Dr. Bernd Häusler for his support and time dedication on this work. Moreover, I would like to express my gratefulness towards Prof. Dr. Joachim Gwinner for assuming the co-chair and for providing comments and recommendations throughout this project. My special thanks are directed to Dr. Wolfgang Seboldt who strongly supported my work from the beginning and who always took the time for in-detail discussions of various aspects of this work. I also give thanks to all colleagues who are not named here, but have contributed to this work through discussions, suggestions, and comments. A special thank is due to my wife Bettina. Without her support this research would not have been possible.

[1] German Aerospace Center
[2] Institute of Space Sensor Technology and Planetary Exploration
[3] Institute of Space Simulation

Abstract

Innovative solar system exploration missions require ever larger velocity increments and thus ever more demanding propulsion capabilities. Using for those high-energy missions the state-of-the-art technique of chemical propulsion in combination with (eventually multiple) gravity assist maneuvers results in long, complicated, and inflexible mission profiles. Low-thrust propulsions systems can significantly enhance or even enable those high-energy missions, since they utilize the propellant more efficiently – like electric propulsion systems – or do not consume any propellant at all – like solar sails, that utilize solely the freely available solar radiation pressure for propulsion. Consequently, low-thrust propulsion systems permit significantly larger velocity increments and/or larger payload ratios and/or smaller launch vehicles, while at the same time allowing direct trajectories with reduced flight times, simpler mission profiles, and extended launch windows.

One of the most important tasks during the feasibility analysis and the preliminary design of a deep space mission is the design and the optimization of the interplanetary transfer trajectory. Searching trajectories for low-thrust spacecraft, that are optimal with respect to transfer time or propellant consumption, is usually a difficult and time-consuming task that involves a lot of experience and expert knowledge, since the convergence behavior of traditional optimizers, that are based on numerical optimal control methods, depends strongly on an adequate initial guess, which is often hard to find. Even if the optimizer converges to an "optimal" trajectory, this trajectory is typically close to the initial guess that is rarely close to the (unknown) global optimum.

Within this work, trajectory optimization problems are attacked from the perspective of artificial intelligence and machine learning, which is quite different from that of optimal control theory. Inspired by natural archetypes, a smart method for spacecraft trajectory optimization – that fuses artificial neural networks and evolutionary algorithms to evolutionary neurocontrollers – is developed. Before the novel method is employed for the trajectory optimization and mission analysis of some exemplary deep space missions, its convergence behavior is evaluated and the quality of the obtained solutions is assessed. It is demonstrated, by re-calculating trajectories for several existing low-thrust problems, that this novel method can be applied successfully for near-globally optimal spacecraft steering. Since evolutionary neurocontrollers explore the trajectory search space more exhaustively than a human expert can do by using traditional optimal control methods, they are able to find spacecraft steering strategies that generate better trajectories, which are closer to the global optimum. Using evolutionary neurocontrollers, low-thrust trajectories can be optimized without an initial guess and without the permanent attendance of an expert in astrodynamics and optimal control theory. Their field of application may be extended to a variety of optimal control problems.

Zusammenfassung

Innovative Missionen für die Erkundung des Sonnensystems erfordern immer höhere Geschwindigkeitsänderungen und somit immer bessere Antriebssysteme. Für solch hochenergetische Missionen führt die gegenwärtig übliche Methode – Verwendung chemischer Antriebe in Verbindung mit (eventuell mehrfachen) Gravity-Assist-Manövern – zu langen Missionsdauern und zu komplizierten unflexiblen Missionsprofilen. Durch Niedrigschubantriebssysteme kann die Durchführung von hochenergetischen Missionen erleichtert oder teilweise sogar erst ermöglicht werden, da diese Antriebssysteme den Treibstoff effizienter nutzen – wie z.B. elektrische Antriebssysteme – oder sogar überhaupt keinen Treibstoff benötigen – wie Sonnensegler, die den „kostenlosen" solaren Strahlungsdruck als Antriebsquelle nutzen. Infolgedessen erlauben Niedrigschubsysteme höhere Geschwindigkeitsänderungen und/oder größere Nutzlastverhältnisse und/oder leichtere Trägerraketen. Gleichzeitig erlauben sie schnelle Direktbahnen und einfache Missionsprofile mit ausgedehnten Startfenstern.

Eine der wichtigsten Aufgaben während der Missionsdurchführbarkeitsanalyse und des Grobentwurfs einer interplanetaren Mission ist der Entwurf und die Optimierung der Transferbahn(en). Das Konvergenzverhalten von traditionellen Niedrigschub-Bahnoptimierungsverfahren, die auf numerischen Methoden der optimalen Steuerung basieren, hängt entscheidend von einer guten Startschätzung der Lösung ab, wie sie oft schwer zu finden ist. Selbst wenn der Optimierer gegen eine „optimale" Bahn konvergiert, so befindet sich diese meist nahe an der Startschätzung, welche sich wiederum selten nahe am (unbekannten) globalen Optimum befindet. Dies macht die Suche nach zeit- oder treibstoffoptimalen Niedrigschubbahnen für gewöhnlich zu einer schwierigen und zeitraubenden Arbeit, die viel Erfahrung und Expertenwissen erfordert.

Im Rahmen dieser Arbeit werden Bahnoptimierungsprobleme anders angepackt: aus dem Blickwinkel der Künstlichen Intelligenz und des Maschinenlernens. Inspiriert von natürlichen Vorbildern, wird eine „intelligente" Methode zur Bahnoptimierung von Raumfahrzeugen entwickelt, welche Neuronale Netze und Evolutionäre Algorithmen zu Evolutionären Neuronalen Reglern verbindet. Bevor diese neuartige Methode zur Bahnoptimierung und Missionsanalyse einiger beispielhafter interplanetarer Missionen verwendet wird, wird sie hinsichtlich ihres Konvergenzverhaltens und der Güte der Lösungen geprüft und bewertet. Anhand der Neuberechnung von Bahnen für Niedrigschubprobleme, für die bereits Bahnen in der Literatur vorhanden sind, wird gezeigt, dass die neue Methode erfolgreich für eine beinahe-global optimale Steuerung von Raumfahrzeugen eingesetzt werden kann. Da Evolutionäre Neuronale Regler den Lösungsraum möglicher Bahnen erschöpfender durchsuchen als dies ein menschlicher Experte unter Verwendung traditioneller Methoden der optimalen Steuerung kann, finden sie Steuerstrategien, die bessere – näher am globalen Optimum liegende – Bahnen generieren. Mit Evolutionären Neuronalen Reglern können Niedrigschubbahnen ohne Anfangsschätzung und ohne die ständige Anwesenheit eines Astrodynamik- und Bahnoptimierungsexperten optimiert werden. Ihr Anwendungsbereich ist auf eine Vielzahl von optimalen Steuerungsproblemen erweiterbar.

Contents

List of Figures

List of Tables

List of Symbols, Constants, and Acronyms

Latin Symbols:

a		semi-major axis
	or	acceleration
$\mathbf{a}$		acceleration vector
c		speed of light in vacuum
$\mathbf{c}$		steering law weight vector
$\mathbf{d}$		direction unit vector
e		eccentricity
$e_\diamondsuit$		$\diamondsuit$. RDC epoch
$\mathbf{e}$		unit vector
f		true anomaly
$\mathbf{f}$		thrust unit vector
g_0		Earth standard gravitational acceleration
h		orbital angular momentum per spacecraft unit mass
$\mathbf{h}$		RDC partial / interim solution
$\mathbf{k}$		LSL direction vector
ℓ		EA chromosome length
	or	number of ANN neuron layers
m		spacecraft mass
n_u		number of control variables
n_π		number of ANN parameters
$\mathbf{n}$		solar sail normal vector
$o(S)$		order of schema S
p		probability
	or	semilatus rectum
q		EA population size
r		sun–spacecraft distance
Δr		distance to target body
$\mathbf{r}$		heliocentric position vector
s		solar sail lateral length
	or	specular reflection factor
s_γ		sigmoid ANN transfer function
t		time
$\mathbf{t}$		solar sail transverse unit vector
$\mathbf{u}$		control vector
Δv		velocity relative to target body
$\mathbf{v}_\infty$		hyperbolic excess velocity vector / launch velocity vector
w_{ij}		connection weight between neurons j and i
$\mathbf{x}$		state vector
A		solar sail area
$\mathcal{A}$		action space
B		non-LAMBERTian coefficient
C_3		hyperbolic excess energy / launch energy
E		orbital element
	or	eccentric anomaly
$\mathbf{E}$		vector of orbital elements
$\mathcal{E}$		ecliptic coordinate frame $(\mathbf{e}_r, \mathbf{e}_\varphi, \mathbf{e}_\theta)$
F		force
$\mathbf{F}$		force vector
G		universal gravitational constant
	or	characteristic optical sail film coefficient
H		characteristic optical sail film coefficient
	or	RDC search hyperspace

$\mathcal{J}$ inertial cartesian coordinate frame $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$

$\mathrm{I_{sp}}$ specific impulse

J cost function / fitness function / reward / reinforcement

K characteristic optical sail film coefficient

L local spacecraft steering law to change some osculating orbital element with a maximum rate

$\mathcal{L}$ set of local spacecraft steering laws to change the osculating orbital elements with a maximum rate

M mean anomaly

or blended local spacecraft steering law to change or adjust some osculating orbital element with a maximum rate

$\mathcal{M}$ set of blended local spacecraft steering laws to change or adjust the osculating orbital elements with a maximum rate

N ANN network function

$\mathcal{N}$ set of ANN neurons

$\mathcal{N}_\Diamond$ $\Diamond$. ANN neuron layer

$\mathcal{O}$ orbit coordinate frame $(\mathbf{e}_r, \mathbf{e}_t, \mathbf{e}_h)$

P solar radiation pressure

Q local spacecraft steering law to adjust some osculating orbital element with a maximum rate

$\mathcal{Q}$ set of local spacecraft steering laws to adjust the osculating orbital elements with a maximum rate

S solar radiation flux

or EA schema

S spacecraft steering strategy

SP EA selective pressure

T transfer time

or absolute temperature

$\mathcal{U}$ spacecraft control vector space

V_e exhaust velocity

ΔV velocity increment

W power

$\mathcal{X}$ state space

or ANN input set

ΔX trajectory accuracy

$\mathcal{Y}$ ANN output set

Z orbital element except the anomaly

$\mathbf{Z}$ vector of orbital elements except the anomaly

Greek Symbols:

α absorption coefficient

or sail clock angle

β sail cone angle

γ thrust clock angle

or temperature parameter of the sigmoid ANN transfer function

γ_i temperature parameter of neuron i

δ thrust cone angle

$\delta(S)$ defining length of schema S

$\boldsymbol{\delta}$ RDC delta chromosome

ε emission coefficient

ϵ centerline angle

ζ $\mathcal{E}$-$\mathcal{O}$-rotation angle

η solar sail efficiency parameter

θ elevation angle

θ_i bias of neuron i

ι inclination

κ RDC hyperspace contraction parameter

$\varkappa$ exponent for variation of SEP power with solar distance

λ solar sailcraft lightness number

μ heliocentric gravitational constant

or EA tournament size

ν RDC convergence parameter

$\boldsymbol{\xi}$ EA chromosome / individual / string

$\boldsymbol{\pi}$ ANN parameter vector

ϖ longitude of pericenter

ρ reflection coefficient

σ solar sailcraft loading

or Stefan-Boltzmann constant

τ number of finite time intervals for discretization

or transmission coefficient

φ azimuth angle

χ throttle

ω argument of perihelion

Λ	mean longitude
Ξ	EA population
Ω	longitude of ascending node

Other Symbols and Notations:

$\square$	arbitrary variable
$\dot{\square}$	$d\square/dt$
$\ddot{\square}$	$d^2\square/dt^2$
$\square^*$	locally optimal $\square$
$\square^\star$	optimal $\square$
$\tilde{\square}$	required $\square$
$\underline{\square}$	lower bound
$\overline{\square}$	upper bound
or	discrete $\square$
$\square[t]$	time history / trajectory of $\square$
$\lozenge$	arbitrary index
$\uparrow\!\uparrow$	parallel
$\uparrow\!\downarrow$	anti-parallel
$\langle \ldots \rangle$	EA chromosome
$(\ldots)$	open interval
$[\ldots]$	closed interval

Indices:

0	at 1 AU distance
or	at launch
b	(solar sail) back side
av	available
avg	average
b	back reflection
c	characteristic (solar sail at $r = 1\,\mathrm{AU}$ with $\beta = 90°$)
or	crossover
d	diffuse reflection
dry	dry
eff	effective
f	final
f	(solar sail) front side
m	mutation
s	specular reflection
sc	survival of crossover
sm	survival of mutation
G	(solar) gravitation
P	propellant
PL	payload
PPU	power processing unit
R	reference
SA	sail assembly
SC	spacecraft
SRP	solar radiation pressure
Sys	system
T	target
Tank	tank
$\boldsymbol{\pi}$	parameterized by the ANN parameter vector

Constants:

Universal gravitational constant [1]:

$$G = (6.67259 \pm 0.00030) \cdot 10^{-11} \, \frac{\mathrm{m}^3}{\mathrm{kg\,s}^2}$$

Mass of the sun:

$$M_\odot = (1.98893 \pm 0.00059) \cdot 10^{30} \, \mathrm{kg}$$

Heliocentric gravitational constant [1]:

$$\mu = GM_\odot = (1.32713430018 \pm 0.00000000008) \cdot 10^{20} \, \frac{\mathrm{m}^3}{\mathrm{s}^2}$$

Mean sun–Earth distance (astronomical unit distance) [1]:

$$r_0 = 149\,597\,870\,697(\pm 3)\,\mathrm{m} = 1\,\mathrm{AU}$$

Gravitational acceleration of the sun at r_0:

$$a_0 = \frac{\mu}{r_0^2} = (5.93010834 \pm 0.00000006) \, \frac{\mathrm{mm}}{\mathrm{s}^2}$$

Mean solar radiation flux at 1 AU solar distance (solar constant) [14]:

$$S_0 = 1368 \, \frac{\mathrm{W}}{\mathrm{m}^2}$$

Earth standard gravitational acceleration (source: scientific calculator HP48SX):

$$g_0 = 9.80665 \, \frac{\mathrm{m}}{\mathrm{s}^2}$$

Acronyms:

AAAF	L'Association Aéronautique et Astronautique de France
AAS	American Astronomical Society
AIAA	American Institute of Aeronautics and Astronautics
ANN	Artificial Neural Network
CFRP	Carbon Fiber Reinforced Plastics
CM	Center of Mass
CP	Center of Pressure
DC	Delta Coding
DES	Differential Equation System
DS1	Deep Space 1
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V. (German Aerospace Center)
EA	Evolutionary Algorithm
EC	Evolutionary Computation
ENC	Evolutionary Neurocontrol(ler)
ENEAS	Exploration of Near-Earth Asteroids with solar Sailcraft
ENEASEP	Exploration of Near-Earth Asteroids with SEP spacecraft
EP	Electric Propulsion
ESA	European Space Agency
FAL	Final Accuracy Limit
FEM	Finite Element Method
FPDC	Floating Point Delta Coding
GA	Genetic Algorithm
GTOM	Global Trajectory Optimization Method
IAA	International Academy of Astronautics
IEO	Inner Earth Object
InTrance	Intelligent Trajectory optimization using neurocontroller evolution
ISS	International Space Station
JD	Julian Date
JPL	Jet Propulsion Laboratory (USA)
LDF	Launch Date Fingerprint
LEO	Low Earth Orbit
LSL	Local (spacecraft) Steering Law
LTOM	Local Trajectory Optimization Method
MBB	Messerschmitt-Bölkow-Blohm (Germany)
MGS	Mars Global Surveyor
MJD	Modified Julian Date
NASA	National Aeronautics and Space Administration (USA)
NC	Neurocontrol(ler)
NCE	Neurocontroller Evolution
NEA	Near-Earth Asteroid
NEO	Near-Earth Object
NEP	Nuclear Electric Propulsion
NLP	Nonlinear Programming (method)
NP	Non-deterministic Polynomial
NSTAR	NASA Solar electric propulsion Technology Applications Readiness program
OCM	Orbit Cranking Maneuver
PPU	Power Processing Unit
RDC	Real Delta Coding
RIT	Radio frequency Ion Thruster
RKF	Runge-Kutta-Fehlberg (method)
RL	Reinforcement Learning
SCARLET	Solar Concentrator Arrays with Refractive Linear Element Technology
SEP	Solar Electric Propulsion
SEPTOP	Solar Electric Propulsion Trajectory Optimization Program
SGA	Simple Genetic Algorithm
SPAM	Solar Photonic Assist Maneuver
SR	Sample Return
SRP	Solar Radiation Pressure

1 Introduction

Innovative solar system exploration missions require ever larger velocity increments and thus ever more demanding propulsion capabilities. Using the state-of-the-art technique of chemical propulsion in combination with (eventually multiple) gravity assist maneuvers for those high-energy missions results in long, complicated, and inflexible mission profiles. Low-thrust propulsions systems can significantly enhance or even enable those high-energy missions, since they utilize the propellant more efficiently – like electric propulsion systems – or do not consume any propellant at all – like solar sails, that utilize solely the freely available solar radiation pressure for propulsion. Consequently, low-thrust propulsion systems permit significantly larger velocity increments and/or larger payload ratios and/or smaller launch vehicles, while at the same time allowing direct trajectories with reduced flight times, simpler mission profiles, and extended launch windows, providing more mission flexibility.

1.1 Low-Thrust Trajectory Optimization

One of the most important tasks during the analysis and the design of a deep space mission is the design and the optimization of the interplanetary transfer trajectory. This work deals with the problem of searching optimal interplanetary trajectories for low-thrust spacecraft, where a low but continuous thrust is applied to modify the spacecraft's orbit over an extended period of time. In simple words, a spacecraft trajectory is the spacecraft's path from A (the initial body or orbit) to B (the target body or orbit). In general, optimality can be defined according to several objectives like transfer time or propellant consumption. Since solar sailcraft do not consume any propellant, their trajectories are typically optimized with respect to transfer time alone. Trajectory optimization for electric propelled spacecraft is less straightforward, since transfer time minimization and propellant minimization are sometimes competing objectives, so that one objective can only be optimized at the cost of the other objective.

Spacecraft trajectories are obtained from the (numerical) integration of the spacecraft's equations of motion, which contain terms for the external forces that are acting on the spacecraft (gravitational forces and "disturbing" forces like solar radiation and solar wind) and for the thrust force. Besides the inalterable external forces, the trajectory is determined by the variation of the thrust vector, which is typically described by a (spacecraft) control function. Therefore, the actual optimization problem is to find the optimal spacecraft control function that yields the optimal trajectory.

For spacecraft with high thrust like chemical rockets, optimal interplanetary trajectories can be found relatively easily[1], since only a few thrust phases are necessary. These thrust phases are very short compared to the transfer time, so that they can be approximated by singular events that change the spacecraft's velocity instantaneously while its position remains fixed. In contrast to those high-thrust propulsion systems, low-thrust propulsion systems are required to operate for a significant part of the transfer to generate the necessary velocity increment ΔV. Consequently, the spacecraft control function is a continuous function of

[1] as long as no gravity assist maneuvers are required

time and the dimension of the solution space is infinite. This renders low-thrust trajectory optimization a very difficult problem.

Traditionally, low-thrust trajectories are optimized by the application of numerical optimal control methods that are based on the calculus of variations. All these methods can be generally classified as *local* trajectory optimization methods, where the term optimization does not mean "finding *the best* solution" but rather "finding *a* solution". The convergence behavior of local trajectory optimization methods depends on an adequate initial guess of the solution, which is often hard to find. Mostly, many initial guesses have to be conceived until an adequate one is found and convergence is achieved. Since all tasks require frequent manual interactions and thus the permanent attendance of an expert in astrodynamics and optimal control theory, the search for a good trajectory can become very time-consuming and thus expensive. Even if convergence *is* achieved, a local optimum is typically found, which is close to the initial guess that is rarely close to the (unknown) global optimum.

Emanating from the drawbacks of the traditional local trajectory optimization methods, a *smart global* trajectory optimization method is sought that runs without an initial guess and without the permanent attendance of a trajectory optimization expert.

1.2 Motivation for Evolutionary Neurocontrol

Evolutionary neurocontrollers fuse artificial neural networks with evolutionary algorithms. Like the underlying constructs, they are inspired by the *natural* processes of information processing and optimization. Animal nervous systems incorporate natural evolutionary neurocontrollers to control their actions, giving them marvellous capabilities. One brilliant example for this proposition is the smart flight control system of the housefly. The nervous system of the housefly comprises about 100 000 neurons. This natural neural network manages the flight control of the fly as well as many even more difficult tasks like finding food, finding a mate, producing offspring, etc. As a matter of fact, this flight control system is more ingenious than the one of any aircraft man has ever built. Nature has optimized the fly's neurocontroller with respect to one single objective: survive to produce offspring. This optimization problem is *really* very difficult, much more than finding some optimal spacecraft trajectory. Nature has solved this problem through the recombination and mutation of the fly's genetic material and through natural selection, the famous so-called "survival of the fittest". Fitter flies produce more offspring and there is a high probability that some of them are still smarter than their parents. This very elegant optimization process runs without an initial guess and without employing the calculus of variations! So, if a natural evolutionary neurocontroller can steer a housefly optimally from A to B, why should an artificial evolutionary neurocontroller not be able to steer a spacecraft optimally from A to B?

1.3 Work Objectives and Outline

The primary objective of this work is to develop a smart *global* trajectory optimization method that does not have the drawbacks of the traditional local trajectory optimization methods, as stated above. This novel method, termed "InTrance" (which stands for "Intelligent Trajectory optimization using neurocontroller evolution") employs evolutionary neurocontrol to search

near-globally[2] optimal spacecraft trajectories without an initial guess and without the permanent attendance of a trajectory optimization expert.

Within chapter 2, the simulation model and the propulsion system models are established that are used within this work to simulate the motion of low-thrust spacecraft in interplanetary space.

Within chapter 3, a formal framework for trajectory optimization is given from the perspective of optimal control theory, and it is exemplified how – using local spacecraft steering laws and numerical methods of optimal control – low-thrust trajectories are generally optimized. Special emphasis is given to the problems that local trajectory optimization methods typically encounter.

Within chapter 4, it is demonstrated that trajectory optimization problems can also be attacked from the perspective of artificial intelligence and machine learning. Within this context, the problem of searching an optimal trajectory is equivalent to the problem of searching an optimal spacecraft steering strategy. An artificial neural network can be employed as a neurocontroller to implement such a steering strategy, and an evolutionary algorithm can be used to find the neurocontroller that represents the *optimal* steering strategy that in turn yields the optimal trajectory.

Within chapter 5, the implementation of InTrance is described.

Before InTrance can be considered as a viable low-thrust trajectory optimization tool, its convergence behavior has to evaluated and the quality of the obtained solutions has to be assessed. Within chapter 6, InTrance is applied to re-calculate existing trajectories for several interplanetary low-thrust trajectory optimization problems (rendezvous problems, fly-by problems, orbit transfer problems, solar sails, solar electric propulsion systems). By comparing the obtained InTrance-results with the existing results, it is shown that InTrance is able to find trajectories that are better (faster and fairly accurate for mission feasibility analysis) than the original trajectories, which have been calculated using traditional local trajectory optimization methods.

The secondary objective of this work is to assess the near-term applicability of solar sail propulsion for deep space missions. Within chapter 7, the suitability of InTrance as a tool for mission feasibility analysis and design is assessed. Therefore, InTrance is applied to analyze and design some innovative solar sail missions, and to compare the solar sail mission parameters with those of electric propulsion systems. This is first done for a near-Earth asteroid rendezvous mission, for a near-Earth asteroid sample return mission, and for a mission to rendezvous three different near-Earth asteroids (including a sample return option). Then, two missions are investigated that employ a solar sail for a MESSENGER- and BepiColombo-like mission to Mercury. Finally, it is shown how InTrance can be used to facilitate the feasibility analysis for a piloted Mars mission.

[2] *near*-globally optimal, since for "real-world" optimization problems global optimality can rarely be proved

2 Simulation and Propulsion System Models

Within this chapter, the simulation model is described that is used within this work to simulate the motion of low-thrust spacecraft in interplanetary space (section 2.1). Then, a solar sail model is elaborated (section 2.2) as well as two electric propulsion (EP) system models (section 2.3): a simple nuclear electric propulsion (NEP) system model (section 2.3.2) and a more specific solar electric propulsion (SEP) system model (section 2.3.3).

2.1 Simulation Model

Besides the gravitational forces of *all* celestial bodies and the spacecraft's thrust force, many "disturbing" forces[1] are influencing the motion of spacecraft. Ideally, all these forces have to be considered for a thorough mission analysis, and (ideally!) the theory of general relativity has to be applied to describe the motion of spacecraft. However, to assess the viability of different spacecraft trajectory optimization methods, and for mission feasibility analysis, as done within this work, only "preliminary" trajectory analysis needs to be done, which allows some simplifications:

1. Spacecraft is moving under the sole influence of solar gravitation and – in the case of solar sailcraft – solar radiation. The sun is a point mass and a point light source. All disturbing forces that are small in magnitude compared to gravitation and – in the case of solar sailcraft – solar radiation pressure (SRP) are neglected. Also ignored are the gravitational and radiative forces of other celestial bodies[2]. This simplification results in a limitation of the simulation model to heliocentric trajectories. Since the spacecraft's mass is many orders of magnitude smaller than the sun's mass, one has a *one*-body problem, where the spacecraft is assumed to orbit around the sun's center of mass.[3]
2. The motion of spacecraft can be described by NEWTON's approximation to the field equations of general relativity.
3. The magnitude and direction of the spacecraft's thrust vector can be changed instantaneously.
4. The spacecraft systems (e.g. sail film, solar arrays, electric thrusters, etc.) do not degrade over time.

2.2 Solar Sail Model

Within this section – as a brief introduction to solar sail propulsion – the unique mission capabilities of solar sails are outlined (section 2.2.1) and the DLR solar sailcraft baseline

[1] as caused e.g. by the solar wind and the aberration of solar radiation (POYNTING–ROBERTSON effect)
[2] including the launch and the target body
[3] In a *two*-body problem *both* the spacecraft *and* the sun would rotate about their barycenter.

design is introduced (section 2.2.2). Subsequently, the three different SRP force models that are commonly used to describe the motion of solar sailcraft are elaborated (section 2.2.3). Then, the prevalent solar sail(craft) performance parameters are introduced and utilized for expressing the SRP force that is acting on the sail (section 2.2.4). Finally, the equations of motion for solar sailcraft in interplanetary space are derived (section 2.2.5).

2.2.1 Solar Sail Mission Capabilities

Utilizing solely the freely available solar radiation pressure for propulsion, solar sails provide a wide range of opportunities for innovative interplanetary low-cost missions, many of which are difficult or even impossible for any other type of conventional propulsion system due to their large ΔV-requirement. Within the inner solar system (including the main asteroid belt), solar sailcraft are especially suited for multiple rendezvous and sample return missions due to their (at least in principle) unlimited ΔV-capability. But even missions to the outer solar system may be enhanced by using solar sails, albeit the solar radiation pressure decreases $\sim 1/r^2$. For such missions, solar sailcraft may gain a large amount of energy when first approaching the sun, thereby performing a so-called "solar photonic assist" maneuver that turns the trajectory into a hyperbolic one [51, 52, 80]. Such trajectories allow reasonable transfer times to the outer planets (and to near interstellar space) without the need to perform any gravity assist maneuver. However, without the use of additional propulsive devices and/or an aerocapture maneuver at the target body, only fast fly-bys can be achieved due to the associated large hyperbolic excess velocities. A specific application for advanced solar sailcraft are non-KEPLERian orbits, where the propulsive force is used to cancel out a part of the solar gravitation (figure 2.1) [60]. Solar sails are especially suited for such non-KEPLERian orbits, since they can apply such a force continuously. This allows some exciting unique trajectories. For example, the orbital plane of the sailcraft can be displaced above Earth's orbital plane, so that the sailcraft can stay "fixed" above the Earth[4] at some distance, if the orbital periods

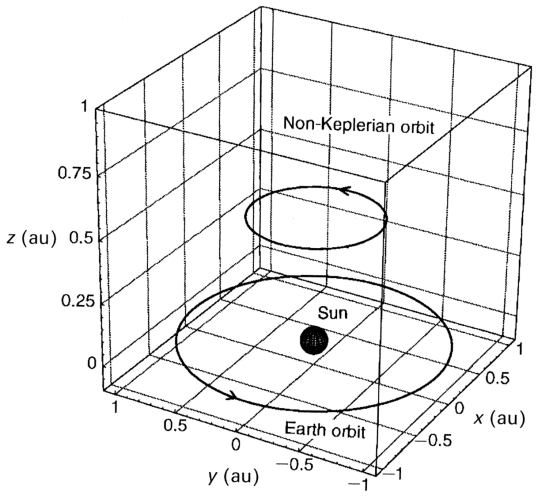


Figure 2.1 Non-KEPLERian orbit (from [60])

[4] in a co-rotating reference frame

are equal. In such orbits, solar sailcraft can be used as communication satellites for high latitudes [38]. In some distant future, laser sailcraft – propelled to relativistic velocities by very powerful lasers – may transcend the limitations of the inverse square law of solar radiation pressure and accomplish interstellar travel (figure 2.2) [37].

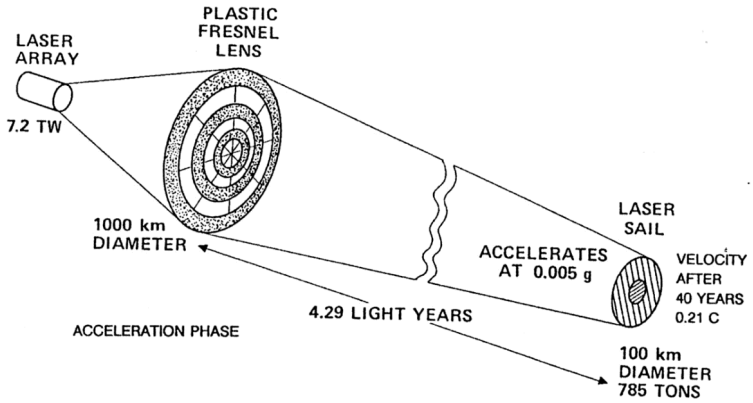
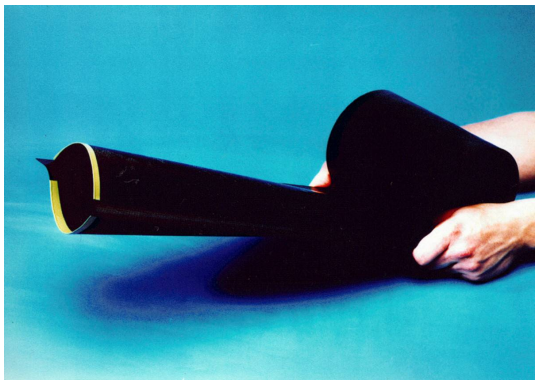


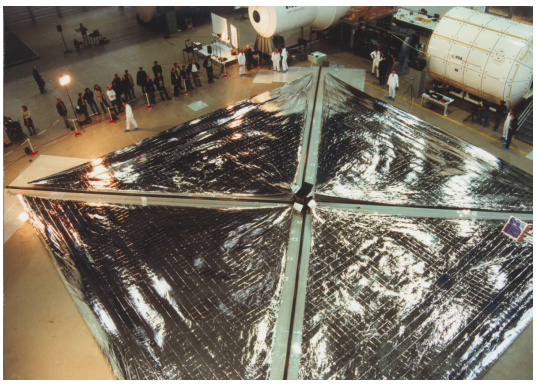
Figure 2.2 Interstellar fly-by using a very advanced laser sail (from [37])

2.2.2 DLR Solar Sailcraft Baseline Design

The DLR solar sail baseline design is a square sail that consists of four CFRP (Carbon Fiber Reinforced Plastics) booms (figure 2.3.1) and of four triangular sail segments made of aluminum-coated plastic film (see figure 2.3.2). The booms consist of two CFRP shells that are bonded at the edges to form a tubular shape , so that they can be pressed flat and rolled up in a central deployment module. During deployment, they unfold automatically and return to their tubular shape with high bending and buckling strength. Subsequently, the four sail segments are deployed by ropes. At DLR, a $(20\,\mathrm{m})^2$ solar sail was successfully deployed in December 1999 on-ground in a simulated gravity-free environment and ambient environmental conditions (figure 2.3.2) [53, 86].



2.3.1: Deployable CFRP boom at DLR



2.3.2: Deployed $(20\,\mathrm{m})^2$ solar sail at DLR

Figure 2.3

According to the DLR baseline design (figure 2.4), the solar sail and the payload micro-satellite are separated by a collapsible control mast, which is housed inside the deployment module in its stowed configuration. This control mast is attached to the deployment module via a two degree of freedom actuator gimbal, which allows to rotate the mast including the attached micro-satellite with respect to the sail. By rotating the control mast, the center of mass (CM) can be offset from the center of pressure (CP). The resulting external torque may be used to rotate the sail about any CM-intersecting axis parallel to the sail plane.[5]

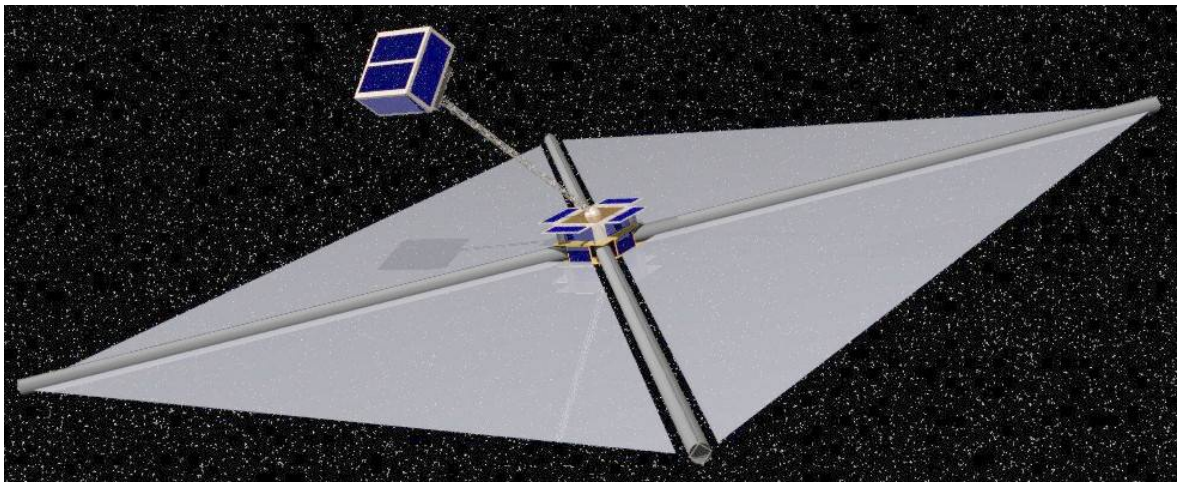


Figure 2.4 DLR solar sailcraft with deployed control mast (artist's view)

2.2.3 Solar Radiation Pressure Force Models

For the optical characteristics of a solar sail, different assumptions can be made, which result in different models for the magnitude and direction of the SRP force acting on the sail. The most simple model assumes a perfectly reflecting sail surface (section 2.2.3.3). Since a real solar sail is *not* a perfect reflector, a thorough trajectory simulation must employ a more sophisticated SRP force model, which takes into account the optical characteristics of the real aluminum-coated sail film (section 2.2.3.4). For *very* preliminary mission feasibility analysis, however, a simplified optical solar sail model may be used (section 2.2.3.5), which facilitates an analytical treatment of solar sail mechanics. All three SRP force models do not take into account the shape of the sail film under load but assume a plane sail surface. During the study for a comet Halley rendezvous mission with a solar sail, which has been performed at NASA/JPL in 1976–77, also a numerical parametric force model has been developed, which takes into account the exact shape of the sail under load [60]. Recently, also numerical analyses to predict the effects of structural wrinkles in the stressed sail film have been performed [69]. However, since those models depend essentially on the actual sail *design*, they are not used within this work.

[5] This propulsionless attitude control concept was originally proposed by Angrilli et al. [7].

2.2.3.1 Solar Radiation Pressure

It can be derived from the laws of quantum mechanics and special relativity,[6] that the **solar radiation pressure** (SRP) due to the momentum transport by solar photons is

$$P = \frac{S}{c} \tag{2.1}$$

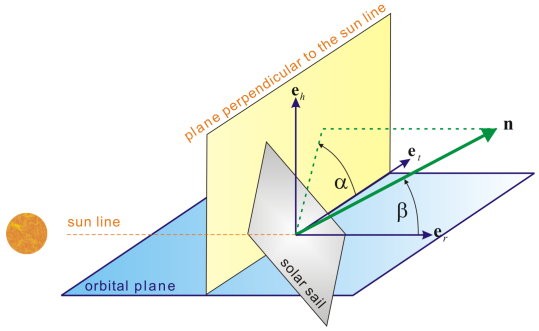
where S is the **solar radiation flux**, and c is the speed of light in vacuum.

Since the radiation flux of a point light source varies with the inverse square of distance, and the mean solar radiation flux at sun–Earth distance $r_0 = 1\,\mathrm{AU}$ is the well-known **solar constant**[7] $S_0 = 1368\,\mathrm{W/m^2}$, the SRP at a distance r from the sun is

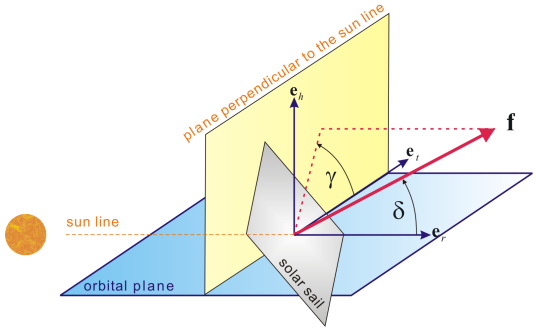
$$P = \frac{S_0}{c}\left(\frac{r_0}{r}\right)^2 \doteq 4.563\,\frac{\mu\mathrm{N}}{\mathrm{m}^2}\cdot\left(\frac{r_0}{r}\right)^2 \tag{2.2}$$

2.2.3.2 Sail Normal Vector and Thrust Unit Vector

For expressing the SRP force exerted on a solar sail, it is convenient to introduce two unit vectors. The first one is the **sail normal vector n**, which is a unit vector that is perpendicular to the sail surface and always directed away from the sun ($\mathbf{n}\cdot\mathbf{e}_r \geq 0$). Its direction, which describes the sail attitude, is – according to figure 2.5.1 – usually expressed by the **sail clock angle** α and the **sail cone angle** β. The second unit vector is the **thrust unit vector f**, which points always along the direction of the thrust force. Its direction is described likewise by the **thrust clock angle** γ and the **thrust cone angle** δ – according to figure 2.5.2.



2.5.1: Definition of the sail clock angle α and the sail cone angle β



2.5.2: Definition of the thrust clock angle γ and the thrust cone angle δ

Figure 2.5

2.2.3.3 Ideal SRP Force Model

The force exerted on a perfectly reflecting solar sail can easily be calculated from figure 2.6. Using $\mathbf{e}_r$ and $\mathbf{e}_{r'}$ as the unit vectors along the direction of the incident and the reflected

[6] see [60] pp. 34–36

[7] the solar constant is actually not a constant but varies approx. 0.1% over days and drifts approx. 0.2-0.6% over centuries [11]

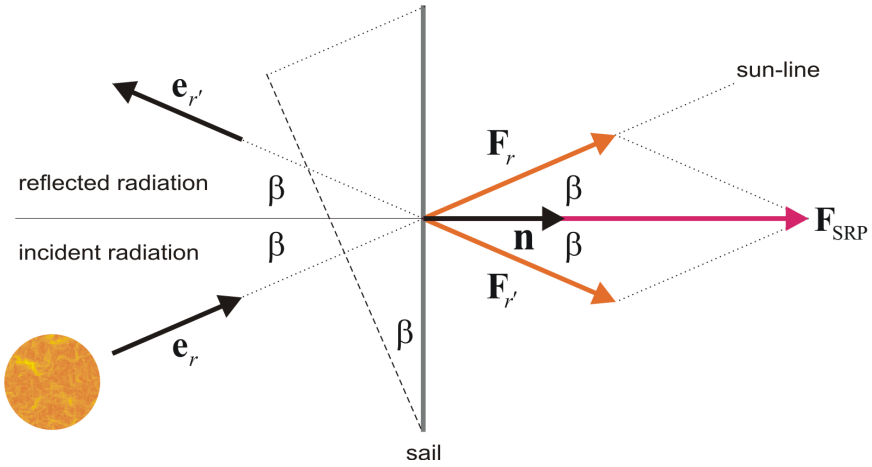


Figure 2.6 SRP force on a perfectly reflecting solar sail

radiation, the force exerted on the sail due to the incident photons is

$$\mathbf{F}_r = PA(\mathbf{e}_r \cdot \mathbf{n})\mathbf{e}_r$$

where $A(\mathbf{e}_r \cdot \mathbf{n})$ is the projected sail area along the $\mathbf{e}_r$-direction. The force exerted on the sail due to the reflected photons is

$$\mathbf{F}_{r'} = -PA(\mathbf{e}_r \cdot \mathbf{n})\mathbf{e}_{r'}$$

Therefore, using $\mathbf{e}_r - \mathbf{e}_{r'} = 2(\mathbf{e}_r \cdot \mathbf{n})\mathbf{n}$, the total SRP force exerted on the sail is given by

$$\mathbf{F}_{\text{SRP}} = \mathbf{F}_r + \mathbf{F}_{r'} = 2PA(\mathbf{e}_r \cdot \mathbf{n})^2\mathbf{n}$$

and, making use of $\mathbf{e}_r \cdot \mathbf{n} = \cos\beta$,

$$\mathbf{F}_{\text{SRP}} = 2PA\cos^2\beta\,\mathbf{n} \tag{2.3}$$

Looking at equation (2.3), one can see that the SRP force exerted on a perfectly reflecting solar sail is always along the direction of the sail normal vector, i.e. $\mathbf{f} = \mathbf{n}$.

2.2.3.4 Standard SRP Force Model

Since a real solar sail is *not* a perfect reflector, a thorough trajectory simulation must consider the optical characteristics of the real sail film, which can be parameterized by the absorption coefficient α, the reflection coefficient ρ, the transmission coefficient τ, and the emission coefficient ε, with the constraint

$$\alpha + \rho + \tau = 1 \tag{2.4}$$

Assuming $\tau = 0$ for the reflecting side of the solar sail, the absorption coefficient is

$$\alpha = 1 - \rho \tag{2.5}$$

Since for a real solar sail not all photons are reflected specularly, the reflection coefficient can be further divided into a coefficient for specular reflection ρ_s, a coefficient for diffuse reflection ρ_d, and a coefficient for back reflection ρ_b, with the constraint

$$\rho_s + \rho_d + \rho_b = \rho \tag{2.6}$$

Assuming $\rho_b = 0$, this can also be expressed by introducing a specular reflection factor s, so that

$$s = \frac{\rho_s}{\rho} \quad \Rightarrow \quad \rho_s = s\rho \quad \text{and} \quad \rho_d = (1-s)\rho \tag{2.7}$$

The emission coefficient ε describes the power that is emitted from a surface of area A at absolute temperature T:

$$W = A\varepsilon\sigma T^4 \tag{2.8}$$

where σ is the STEFAN-BOLTZMANN constant. Using the emission coefficients of the sail's front and back side, ε_f and ε_b, the equilibrium temperature of the sail can be calculated:[8]

$$T = \left(\frac{1-\rho}{\varepsilon_\mathrm{f}+\varepsilon_\mathrm{b}}\frac{c}{\sigma}P\cos\beta\right)^{1/4} \sim \frac{\cos^{1/4}\beta}{r^{1/2}} \quad \Rightarrow \quad r_\mathrm{min} \sim \frac{1}{T_\mathrm{max}^2} \tag{2.9}$$

It can be seen that there is a minimum sun-sail distance r_min, which is inversely proportional to the square of the sail film temperature limit T_max.

It can be shown[8] that, using the optical sail parameters defined above, the SRP force exerted on the solar sail has a normal component $\mathbf{F}_n$ and a transversal component $\mathbf{F}_t$ (see figure 2.7) with

$$\mathbf{F}_n = PA\left((1+s\rho)\cos\beta + B_\mathrm{f}(1-s)\rho + (1-\rho)\frac{\varepsilon_\mathrm{f}B_\mathrm{f} - \varepsilon_\mathrm{b}B_\mathrm{b}}{\varepsilon_\mathrm{f}+\varepsilon_\mathrm{b}}\right)\cos\beta\,\mathbf{n} \tag{2.10a}$$

$$\mathbf{F}_t = PA(1-s\rho)\sin\beta\cos\beta\,\mathbf{t} \tag{2.10b}$$

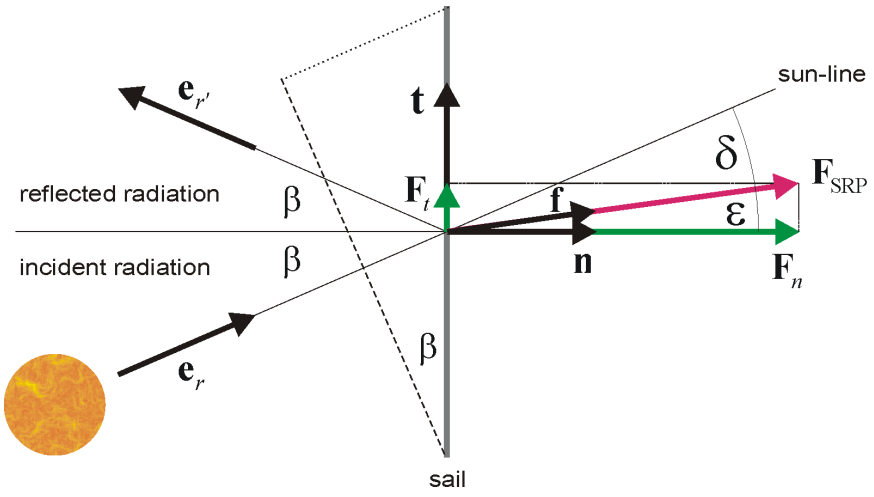


Figure 2.7 SRP force on a non-perfectly reflecting solar sail

[8] see [60] pp. 48–49 for derivation

where $\mathbf{t}$ is a transverse unit vector perpendicular to $\mathbf{n}$ (so that $\mathbf{t} \cdot \mathbf{e}_r \geq 0$), and B_f and B_b are the non-LAMBERTian coefficients of the sail's front and back side. WRIGHT gives values for the optical coefficients for a sail[9] with a highly reflective aluminum-coated front side and a highly emissive chromium-coated back side[10] (table 2.1) [98].

parameter	front side (Al-coated)	back side (Cr-coated)
ρ	0.88	
s	0.94	
ε	0.05	0.55
B	0.79	0.55

Table 2.1 Optical coefficients for an Al|Cr-coated solar sail

Using the values given in table 2.1, three characteristic optical sail film coefficients may be defined to simplify equations (2.10) [78]:

$$G = 1 + s\rho = 1.8272 \tag{2.11a}$$

$$K = B_\mathrm{f}(1-s)\rho + (1-\rho)\frac{\varepsilon_\mathrm{f} B_\mathrm{f} - \varepsilon_\mathrm{b} B_\mathrm{b}}{\varepsilon_\mathrm{f} + \varepsilon_\mathrm{b}} = -0.010888 \tag{2.11b}$$

$$H = 1 - s\rho = 0.1728 \tag{2.11c}$$

so that

$$\mathbf{F}_n = PA\,(G\cos\beta + K)\cos\beta\,\mathbf{n} \tag{2.12a}$$

$$\mathbf{F}_t = PAH\sin\beta\cos\beta\,\mathbf{t} \tag{2.12b}$$

The total SRP force vector may then be written as

$$\mathbf{F}_\mathrm{SRP} = \sqrt{F_n^2 + F_t^2}\,\mathbf{f} = PA\sqrt{(G\cos\beta + K)^2 + H^2\sin^2\beta}\cos\beta\,\mathbf{f} \tag{2.13}$$

and, by defining $Q(\beta) = \sqrt{(G\cos\beta + K)^2 + H^2\sin^2\beta}\cos\beta$, as

$$\mathbf{F}_\mathrm{SRP} = PAQ(\beta)\,\mathbf{f} \tag{2.14}$$

where $Q(\beta)$ depends only on the sail cone angle β and the optical coefficients of the sail film. The angle between $\mathbf{f}$ and $\mathbf{e}_r$ is the thrust cone angle δ and the angle between $\mathbf{f}$ and $\mathbf{n}$ is called **centerline angle** ϵ. It may be calculated via

$$\epsilon = \arctan\left(\frac{F_t}{F_n}\right) = \arctan\left(\frac{H\sin\beta}{G\cos\beta + K}\right) \tag{2.15}$$

Equation (2.15) gives then also the relation for the thrust cone angle:

$$\delta = \beta - \epsilon = \beta - \arctan\left(\frac{H\sin\beta}{G\cos\beta + K}\right) \tag{2.16}$$

[9] JPL square sail and JPL heliogyro

[10] to keep the sail temperature at a moderate limit. Using equation (2.9) and table 2.1, one can easily calculate a maximum equilibrium temperature of $T = 490.5\,\mathrm{K}$ (217.4 °C) at 1 AU for an Al|Al-coated solar sail (for $\mathbf{n} \cdot \mathbf{e}_r = 1$), whereas $T = 313.4\,\mathrm{K}$ (40.2 °C) for an Al|Cr-coated sail.

2.2.3.5 Simplified SRP Force Model

In the solar sail-related literature, a simplified SRP force model is typically employed, which uses an overall **sail efficiency parameter** η. This parameter is intended to encompass the non-perfect reflectivity of the sail as well as the sail deflection/warping under load. Using this parameter, the SRP force acting on the sail is described by

$$\mathbf{F}_{\mathrm{SRP}} = 2\eta P A \cos^2 \beta \, \mathbf{n} \tag{2.17}$$

This SRP force model is widely used, because it allows an easy analytical treatment of solar sail mechanics (since $\mathbf{f} = \mathbf{n}$). However, it provides only a rough approximation of the real sail, as it will be shown in the next section.

2.2.3.6 SRP Force Model Comparison

For the ideal and the simplified SRP force model, the SRP force is always perpendicular to the sail surface, $\mathbf{f} = \mathbf{n}$. This allows an easy analytical treatment of solar sail steering problems. The orbital dynamics of solar sailcraft is in many respects similar to the orbital dynamics of other low-thrust spacecraft. However, as figure 2.8 shows, other low-thrust spacecraft may orient its thrust vector into any desired direction, whereas the thrust vector of solar sailcraft is constrained to lie on the surface of a "bubble" that is always directed away from the sun.[11]

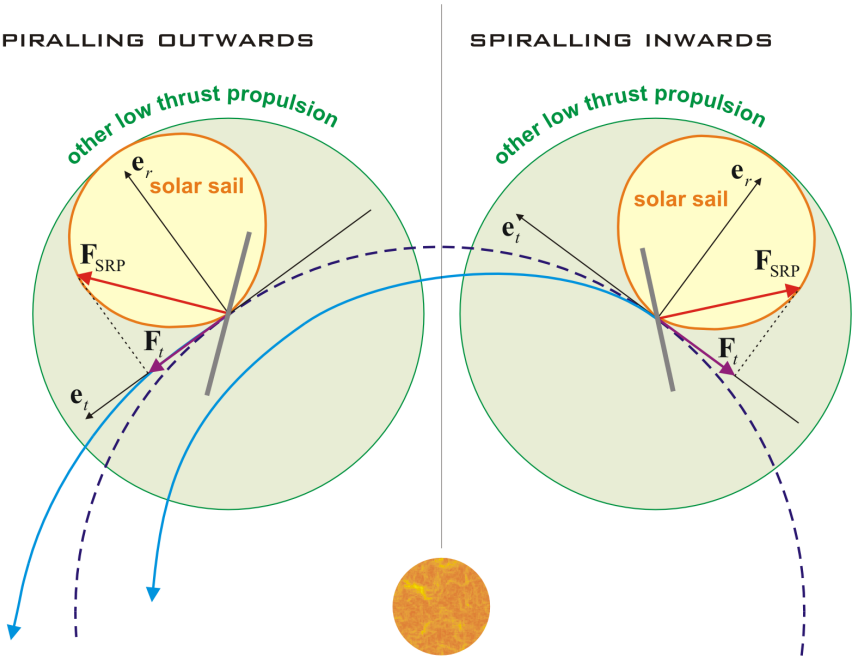


Figure 2.8 Spiralling towards the sun and away from the sun

[11] Nevertheless, by controlling the sail orientation relative to the sun, solar sailcraft can loose orbital angular momentum and spiral inwards – towards the sun – or gain orbital angular momentum and spiral outwards – away from the sun.

If $\mathbf{d}$ denotes the unit vector along the desired (locally optimal) thrust direction, the thrust unit vector $\mathbf{f}$ must point into the direction $\mathbf{f}^*$ that maximizes the SRP force along $\mathbf{d}$; $\mathbf{f}^*$ can be derived analytically from $\mathbf{d}$ (see section 3.2.4.2). The sail normal vector $\mathbf{n}$ – as expressed by the sail clock angle α and the sail cone angle β – is the spacecraft control vector, $\mathbf{u} = (\alpha, \beta)$. The problem of solar sailcraft steering is to determine $\mathbf{n}$ so that $\mathbf{f} = \mathbf{f}^*$. For the standard SRP force model, $\mathbf{n}$ can *not* be calculated analytically from $\mathbf{f}$, since the $Q(\beta)$-expression can not be resolved for β. Hence, the spacecraft control vector that maximizes the SRP force along the desired thrust direction can not be obtained analytically.

Although the ideal and the simplified SRP force model allow the analytical treatment of solar sail steering problems, they misrepresent the normal SRP force component $\mathbf{F}_n$ and completely ignore the transverse SRP force component $\mathbf{F}_t$. In doing so, both models ignore the deviation of the thrust cone angle from the sail cone angle. Figure 2.9 shows how the deviation becomes larger as the light incidence angle increases. As a consequence, the SRP force in the standard SRP force model is not only smaller than in the ideal SRP force model (which is also taken into account by the simplified SRP force model) but also much more constrained in its direction (figure 2.9 shows that there is a maximum thrust cone angle of 55.5° for a sail cone angle of 72.6°).

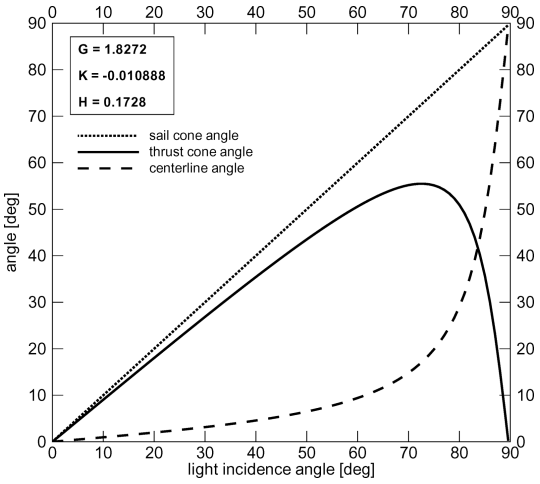


Figure 2.9 Sail cone angle, thrust cone angle and centerline angle for the standard SRP force model

Figure 2.10 shows for each SRP force model the "bubble" on whose surface the SRP force vector tip is constrained to lie (vector tail at origin). From the perspective of trajectory analysis, the simplified SRP force model is equivalent to the ideal SRP force model, since the *shape* of both "bubbles" is identical. A decrease in sail efficiency η can be offset with a proportional increase in sail area, so that both "bubbles" have the same shape *and* size. This equivalency is not the case for the standard SRP force model. Even if the $\cos^2\beta$-"bubble" and the $Q(\beta)$-"bubble" are scaled to the same size, their shape is different.

Since the ideal and the simplified SRP force model are equivalent with respect to trajectory analysis, they can *both* be denoted as models of "**perfect**" **reflection**, whereas the standard SRP force model is a model of **non-perfect reflection**. The associated solar sails will also be denoted as **ideal sails** and **non-ideal sails**.

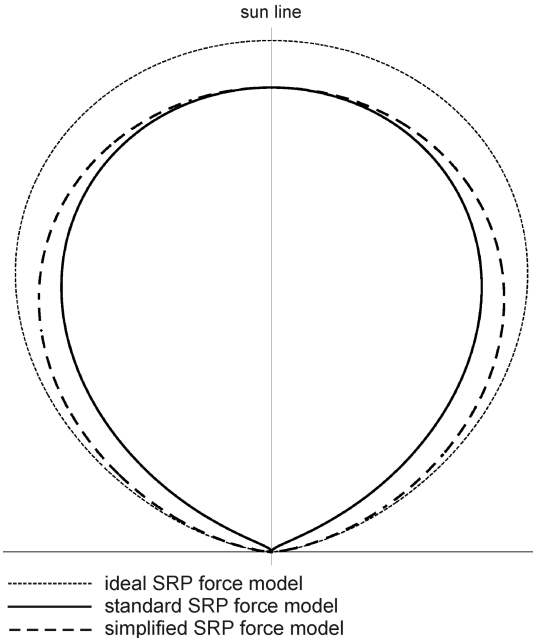


Figure 2.10 SRP force "bubbles" for the different SRP force models

The results that have been obtained within this work (see section 7.1) show that the simplification of taking the non-perfect reflectivity of the sail into account by using an overall efficiency factor η should only be made for *very* preliminary mission feasibility analysis.

2.2.4 Sail Performance Parameters

Within this section, the commonly used solar sailcraft performance parameters are introduced and used for expressing the SRP force acting on the sail.

2.2.4.1 Sail Assembly Loading

The **sail assembly loading**

$$\sigma_{\mathrm{SA}} = \frac{m_{\mathrm{SA}}}{A} \tag{2.18}$$

is defined as the mass of the sail assembly (the sail film and the required structure for storing, deploying and tensioning the sail, index "SA") per unit area. Thus, the sail assembly loading is the key parameter for the efficiency of the solar sail's structural design.

2.2.4.2 Sailcraft Loading

The **sailcraft loading**

$$\sigma = \frac{m}{A} = \frac{m_{\mathrm{SA}} + m_{\mathrm{PL}}}{A} = \sigma_{\mathrm{SA}} + \frac{m_{\mathrm{PL}}}{A} \tag{2.19}$$

is defined as the specific mass of the sailcraft including the payload (index "PL"), where the term payload stands for the total sailcraft except the solar sail assembly (i.e. except the propulsion system).

2.2.4.3 Characteristic Acceleration

Equations (2.14) and (2.17) may also be expressed in terms of the **characteristic acceleration** a_c, which is defined as the SRP acceleration acting on a solar sail that is oriented perpendicular to the sun line ($\mathbf{n} \cdot \mathbf{e}_r = 1$) at sun–Earth distance r_0, where $F_{\mathrm{SRP}} = F_c$, the characteristic SRP force.

For the standard SRP force model, $F_c = P_0 A(G + K)$ and therefore

$$a_c = \frac{F_c}{m} = P_0(G+K)\frac{A}{m} = \frac{P_0(G+K)}{\sigma} \tag{2.20}$$

or, writing $P_{\mathrm{eff},0} = P_0(G+K) = S_0/c \cdot (G+K) \doteq 8.288\,\mu\mathrm{N/m^2}$ for the **effective SRP** at Earth distance,

$$a_c = \frac{P_{\mathrm{eff},0}}{\sigma} = \frac{P_{\mathrm{eff},0}}{\sigma_{\mathrm{SA}} + \frac{m_{\mathrm{PL}}}{A}} \tag{2.21}$$

The SRP force may then be written as

$$\mathbf{F}_{\mathrm{SRP}} = m a_c \Big(\frac{r_0}{r}\Big)^2 \frac{Q(\beta)}{G+K}\,\mathbf{f} = m a_c \Big(\frac{r_0}{r}\Big)^2 Q'(\beta)\,\mathbf{f} \tag{2.22}$$

For the simplified SRP force model, one gets

$$a_c = \frac{F_c}{m} = 2\eta P_0 \frac{A}{m} = \frac{2\eta P_0}{\sigma} \tag{2.23}$$

or, writing $P_{\mathrm{eff},0} = 2\eta P_0 = 2\eta S_0/c$ for the effective SRP at Earth distance, again equation (2.21). The SRP force for the simplified SRP force model may then be written as

$$\mathbf{F}_{\mathrm{SRP}} = m a_c \Big(\frac{r_0}{r}\Big)^2 \cos^2\beta\,\mathbf{n} \tag{2.24}$$

Comparing equation (2.20) and (2.23), one can see that in order to get the same characteristic acceleration for both SRP force models, one has to set $\eta = (G+K)/2 \doteq 0.908$.

2.2.4.4 Lightness Number

Equations (2.14) and (2.17) may also be expressed in terms of the **lightness number** λ, which is defined as the ratio of the SRP acceleration acting on a solar sail that is oriented perpendicular to the sun line ($\mathbf{n}\cdot\mathbf{e}_r = 1$), and the gravitational acceleration of the sun, $a_{\mathrm{G}}(r)$:

$$\lambda = \frac{a_c\big(\frac{r_0}{r}\big)^2}{\frac{\mu}{r^2}} = \frac{a_c r_0^2}{\mu} = \frac{a_c}{a_0} \tag{2.25}$$

with $a_0 = a_G(r_0) = \mu/r_0^2 \doteq 5.930\,\mathrm{mm/s^2}$ as the sun's gravitational acceleration at Earth distance.[12]

Using equation (2.25), the SRP force may be written as

$$\mathbf{F}_{\mathrm{SRP}} = \lambda \frac{\mu m}{r^2} Q'(\beta)\,\mathbf{f} \tag{2.26}$$

for the standard SRP force model and

$$\mathbf{F}_{\mathrm{SRP}} = \lambda \frac{\mu m}{r^2} \cos^2\beta\,\mathbf{n} \tag{2.27}$$

for the simplified SRP force model.

2.2.5 Equations of Motion for Solar Sailcraft

Within this section, the equations of heliocentric translational motion within the one-body simulation model and without disturbing forces are derived for ideal and non-ideal sailcraft. Ignoring second order effects, the acceleration of solar sailcraft is simply obtained by adding the SRP acceleration and the sun's gravitational acceleration

$$\ddot{\mathbf{r}} = \mathbf{a}_{\mathrm{SRP}} + \mathbf{a}_{\mathrm{G}} \tag{2.28}$$

Resolving this equation along the unit vectors of the ecliptic reference frame ($\mathcal{E}$-frame, see appendix A.2) will then give three 2nd order DEs for r, φ, and θ respectively, the DES of motion.

2.2.5.1 Ideal Solar Sail Models

Using equation (2.27), one gets for the ideal/simplified SRP force model

$$\ddot{\mathbf{r}} = \mathbf{a}_{\mathrm{SRP}} + \mathbf{a}_{\mathrm{G}} = \lambda \frac{\mu}{r^2} \cos^2\beta\,\mathbf{n} - \frac{\mu}{r^2}\mathbf{e}_r \tag{2.29}$$

Resolving $\mathbf{n}$ along the $\mathcal{O}$-frame unit vectors (see appendix A.3), one obtains

$$\mathbf{n} = \cos\beta\mathbf{e}_r + \cos\alpha\sin\beta\mathbf{e}_t + \sin\alpha\sin\beta\mathbf{e}_h \tag{2.30}$$

and after transformation into the $\mathcal{E}$-frame

$$\mathbf{n} = \cos\beta\mathbf{e}_r + \cos(\alpha+\zeta)\sin\beta\mathbf{e}_\varphi + \sin(\alpha+\zeta)\sin\beta\mathbf{e}_\theta \tag{2.31}$$

By introducing three dimensionless control functions u_1' to u_3', depending only on the two control variables α and β, and on the local $\mathcal{E}$-$\mathcal{O}$-rotation angle ζ[13]

$$u_1'(\beta) = \cos^3\beta \tag{2.32a}$$

$$u_2'(\alpha+\zeta,\beta) = \cos(\alpha+\zeta)\sin\beta\cos^2\beta \tag{2.32b}$$

$$u_3'(\alpha+\zeta,\beta) = \sin(\alpha+\zeta)\sin\beta\cos^2\beta \tag{2.32c}$$

[12] Since both accelerations have an inverse square variation in r, the lightness of solar sailcraft is – unlike the maximum acceleration – independent of the sun–sail distance.

[13] as defined in appendix A.3

and using equation (2.31), one may write the acceleration of the solar sailcraft in $\mathcal{E}$-frame components as:

$$\ddot{\mathbf{r}} = (\lambda\frac{\mu}{r^2}u_1' - \frac{\mu}{r^2})\mathbf{e}_r + \lambda\frac{\mu}{r^2}u_2'\mathbf{e}_\varphi + \lambda\frac{\mu}{r^2}u_3'\mathbf{e}_\theta \tag{2.33}$$

Expressing $\ddot{\mathbf{r}}$ in polar coordinates (see appendix A.2), one gets three component equations:

$$\ddot{r} - r\dot{\theta}^2 - r\dot{\varphi}^2\cos^2\theta = \lambda\frac{\mu}{r^2}u_1' - \frac{\mu}{r^2} \tag{2.34a}$$

$$2\dot{r}\dot{\varphi}\cos\theta + r\ddot{\varphi}\cos\theta - 2r\dot{\varphi}\dot{\theta}\sin\theta = \lambda\frac{\mu}{r^2}u_2' \tag{2.34b}$$

$$2\dot{r}\dot{\theta} + r\ddot{\theta} + r\dot{\varphi}^2\sin\theta\cos\theta = \lambda\frac{\mu}{r^2}u_3' \tag{2.34c}$$

and after some rearrangement the DES of motion

$$\ddot{r} = r\dot{\theta}^2 + r\dot{\varphi}^2\cos^2\theta - \frac{\mu}{r^2} + \lambda\frac{\mu}{r^2}u_1' \tag{2.35a}$$

$$\ddot{\varphi} = -2\frac{\dot{r}\dot{\varphi}}{r} + 2\dot{\varphi}\dot{\theta}\tan\theta + \lambda\frac{\mu}{r^2}\frac{u_2'}{r\cos\theta} \tag{2.35b}$$

$$\ddot{\theta} = -2\frac{\dot{r}\dot{\theta}}{r} - \dot{\varphi}^2\sin\theta\cos\theta + \lambda\frac{\mu}{r^2}\frac{u_3'}{r} \tag{2.35c}$$

2.2.5.2 Non-Ideal Solar Sail Model

Using equation (2.26), one gets for the standard SRP force model

$$\ddot{\mathbf{r}} = \mathbf{a}_{\mathrm{SRP}} + \mathbf{a}_{\mathrm{G}} = \lambda\frac{\mu}{r^2}Q'(\beta)\,\mathbf{f} - \frac{\mu}{r^2}\mathbf{e}_r \tag{2.36}$$

Resolving $\mathbf{f}$ along the $\mathcal{O}$-frame unit vectors, one obtains

$$\mathbf{f} = \cos\delta\mathbf{e}_r + \cos\gamma\sin\delta\mathbf{e}_t + \sin\gamma\sin\delta\mathbf{e}_h \tag{2.37}$$

and after transformation into the $\mathcal{E}$-frame

$$\mathbf{f} = \cos\delta\mathbf{e}_r + \cos(\gamma+\zeta)\sin\delta\mathbf{e}_\varphi + \sin(\gamma+\zeta)\sin\delta\mathbf{e}_\theta \tag{2.38}$$

where $\gamma = \alpha$ and $\delta = \beta - \arctan\left(\frac{H\sin\beta}{G\cos\beta+K}\right)$.

By introducing three dimensionless control functions u_1' to u_3', depending only on the two control variables α and β, the local $\mathcal{E}$-$\mathcal{O}$-rotation angle ζ, and the optical characteristics of the sail film

$$u_1'(\beta) = Q'(\beta)\cos\delta(\beta) \tag{2.39a}$$

$$u_2'(\alpha+\zeta,\beta) = Q'(\beta)\cos(\alpha+\zeta)\sin\delta(\beta) \tag{2.39b}$$

$$u_3'(\alpha+\zeta,\beta) = Q'(\beta)\sin(\alpha+\zeta)\sin\delta(\beta) \tag{2.39c}$$

one gets again equation (2.33)

$$\ddot{\mathbf{r}} = (\lambda\frac{\mu}{r^2}u_1' - \frac{\mu}{r^2})\mathbf{e}_r + \lambda\frac{\mu}{r^2}u_2'\mathbf{e}_\varphi + \lambda\frac{\mu}{r^2}u_3'\mathbf{e}_\theta$$

and thus again equations (2.35)

$$\ddot{r} = r\dot{\theta}^2 + r\dot{\varphi}^2\cos^2\theta - \frac{\mu}{r^2} + \lambda\frac{\mu}{r^2}u_1'$$
$$\ddot{\varphi} = -2\frac{\dot{r}\dot{\varphi}}{r} + 2\dot{\varphi}\dot{\theta}\tan\theta + \lambda\frac{\mu}{r^2}\frac{u_2'}{r\cos\theta}$$
$$\ddot{\theta} = -2\frac{\dot{r}\dot{\theta}}{r} - \dot{\varphi}^2\sin\theta\cos\theta + \lambda\frac{\mu}{r^2}\frac{u_3'}{r}$$

but with different control functions.

2.3 Electric Propulsion System Models

Within this section, two EP system models are elaborated. The first one is a very simple NEP system with constant power input and constant specific impulse that can be throttled via the propellant mass flow rate (section 2.3.2). The second one is a more specific SEP system with variable power input and variable specific impulse (section 2.3.3). Before this is done, as a brief introduction to electric propulsion in general, the mission capabilities of EP systems are outlined (section 2.3.1).

2.3.1 EP Mission Capabilities

In chemical propulsion systems, chemical reactions release heat that raises the combustion gas temperature to high values. Expanded through a nozzle, the thermal energy is converted into kinetic energy and hence provides thrust. The exhaust velocity is $V_e \approx 3\,000 - 5\,000\,\mathrm{m/s}$ (specific impulse $\mathrm{I_{sp}} = V_e/g_0 \approx 300 - 500\,\mathrm{s}$), being limited by the chemical energy that is stored in the propellant [36]. If this V_e is inserted into the rocket equation

$$\frac{m_f}{m_0} = e^{-\frac{\Delta V}{V_e}} \quad \Leftrightarrow \quad \frac{m_\mathrm{P}}{m_0} = \frac{m_0 - m_f}{m_0} = 1 - e^{-\frac{\Delta V}{V_e}} \tag{2.40}$$

one can see that even for a moderate velocity increment ΔV of about $2V_e$, the propellant mass m_P must be $1 - e^{-2} = 86.5\%$ of the initial mass m_0, where m_f is the final mass.

Innovative solar system exploration missions require an ever increasing ΔV budget. Using for those high-energy missions the state-of-the-art technique of chemical propulsion in combination with (eventually multiple) gravity assist maneuvers results in increasingly long, complicated, and inflexible mission profiles [58].[14] EP systems are a way to overcome the energetic barriers inherent in chemical combustion, since they use e.g. high-voltage electric fields (ion drive) or electromagnetic fields (plasma drive) to ionize and accelerate the propellant (rather than to burn it), which yields exhaust velocities (and specific impulses) that exceed those of chemical rockets by approximately one order of magnitude (see figure 2.11) [58]. Thus, EP systems yield significantly larger ΔVs and/or larger payload ratios and/or smaller launch vehicles. This way, they can significantly enhance or even enable high-energy missions [58]. At the same time, EP systems permit direct trajectories with reduced flight times, simpler mission profiles, and extended launch windows. However, compared to chemical propulsion

[14] An unfortunate example for this statement is the Rosetta mission, which had been intended to fly to comet 46P/Wirtanen with three intermediate gravity assist maneuvers (Mars-Earth-Earth) and a very narrow launch window.

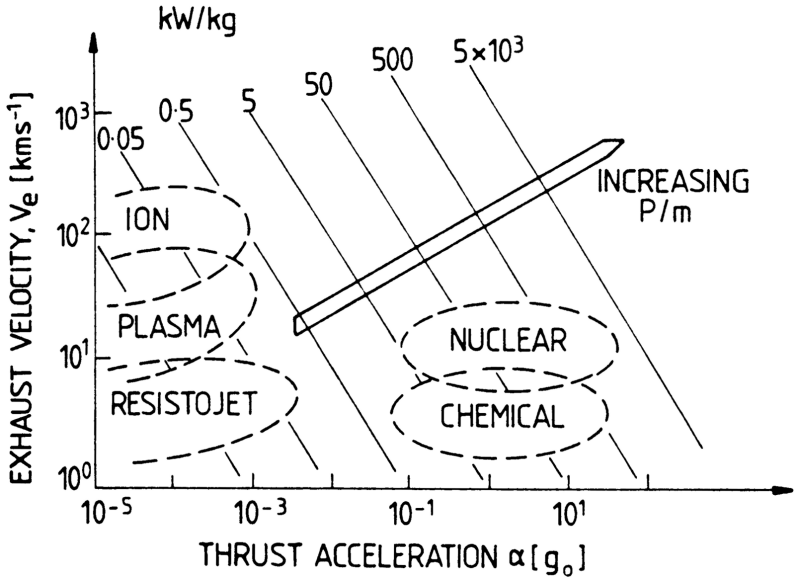


Figure 2.11 Comparative performance of different propulsion systems concerning exhaust velocity V_e, thrust acceleration F/mg_0, and specific power P/m (from [36])

systems, EP systems have two handicaps: ion and plasma thrusters are not able to produce high thrust levels, and they require an adequate source of electric energy (several kilowatts of electric power for several millinewtons of thrust). For interplanetary missions, where high thrust levels can be compensated with longer burn times, the low thrust level is not a serious disadvantage. Also, solar and nuclear high energy power sources with an adequate power to mass ratio have become available today, and the technological readiness of an EP system for primary propulsion has been successfully demonstrated on NASA's Deep Space 1 (DS1) mission[15].

Solar cells can be efficiently used to supply power to the electric thrusters, at least within the inner solar system, since – as for the solar radiation pressure – the solar radiation flux decreases $\sim 1/r^2$. Therefore, like solar sailcraft, SEP spacecraft are especially suited for high-energy missions within the inner solar system. However, unlike solar sailcraft, they do not have an infinite ΔV-capability, which makes them for example unsuitable for missions where a non-KEPLERian orbit has to be maintained for an extended time.

NEP systems are superior to SEP systems in the outer solar system, where the solar radiation flux is not sufficient for power production using solar cells. Providing power and thrust that is independent from solar distance, NEP systems outperform solar sails (and SEP systems) in the outer solar system, if not only a fast fly-by at the target body but a rendezvous is required. Also piloted missions, where thrust levels in the order of several hundred newtons are required, make power generation systems with a power level in the order of several megawatts necessary. For such power levels, the power to weight ratio of nuclear reactors may be superior to that of solar cells.

[15] and on the ARTEMIS satellite

2.3.2 NEP System Model

For the NEP system model, the maximum thrust $F_{\max}$ and the specific impulse $\mathrm{I_{sp}}$ are assumed to be fixed. Then, the maximum propellant mass flow rate $\dot{m}_{\mathrm{P,max}}$ (that is required to generate $F_{\max}$) is

$$\dot{m}_{\mathrm{P,max}} = \frac{F_{\max}}{V_e} = \frac{F_{\max}}{\mathrm{I_{sp}} \cdot g_0} \tag{2.41}$$

For NEP systems, a throttle factor $0 \leq \chi \leq 1$ is used to control the propellant mass flow rate, so that one gets

$$\dot{m}_{\mathrm{P}}(\chi) = \chi \cdot \dot{m}_{\mathrm{P,max}} \tag{2.42}$$

and

$$F(\chi) = \chi \cdot F_{\max} = \chi \cdot \dot{m}_{\mathrm{P,max}} \cdot \mathrm{I_{sp}} \cdot g_0 \tag{2.43}$$

Thus, propellant mass flow rate and thrust vary only with χ. In contrast to solar sails and SEP systems, the thrust is independent of solar distance, which makes NEP systems especially suited for outer solar system missions. Using the thrust unit vector $\mathbf{f}$ to denote the thrust direction, one gets

$$\mathbf{F}(\chi) = \chi \cdot F_{\max}\,\mathbf{f} = \chi \cdot \dot{m}_{\mathrm{P,max}} \cdot \mathrm{I_{sp}} \cdot g_0\,\mathbf{f} \tag{2.44}$$

2.3.3 SEP System Model

Typically, ion thrusters are used for SEP systems. Several technical variants of them have been developed within the last 50 years by competing working groups [20]:

- Radio frequency ion thrusters (RITs, Germany)
- KAUFMAN- or electron bombardment thrusters (USA and UK)
- HALL effect or stationary plasma thrusters (SPTs, Russia and France)
- Electron cyclotron resonance thrusters (ECRs, Japan)

RITs, as they have been developed in Germany (at the Universiät Gießen in cooperation with Astrium GmbH, former MBB) since the 1960's [20], have up to now been used for orbit raising and station keeping[16] but not yet for primary interplanetary propulsion. However, this might change with a larger RIT, which will become operational within the next few years.[17] Within this work, one (section 2.3.3.1) or more (section 2.3.3.2) of NASA's NSTAR[18] thrusters (KAUFMAN type) together with a number of SCARLET[19] solar array wings are considered for the SEP system model, since technical data is easily available for both systems (table 2.2), and since both systems have already been successfully tested on DS1. The NSTAR thruster

[16] RIT-10: $F_{\max} \approx 15\,\mathrm{mN}$, $\mathrm{I_{sp_{max}}} \approx 3400\,\mathrm{s}$, lifetime $> 8\,000$ hours (on ARTEMIS) [2, 56]
[17] RIT-XT: $F_{\max} \approx 200\,\mathrm{mN}$, $\mathrm{I_{sp_{max}}} \approx 5500\,\mathrm{s}$, lifetime $> 15\,000$ hours [56]
[18] NASA Solar Electric Propulsion Technology Applications Readiness Program
[19] Solar Concentrator Arrays with Refractive Linear Element Technology

(figure 2.12) uses a hollow cathode to produce electrons that collisionally ionize xenon. The Xe^+ ions are electrostatically accelerated through a potential of up to about 1300 V and emitted from the 30-cm thruster through a molybdenum grid. A separate electron beam is emitted to produce a neutral plasma beam.

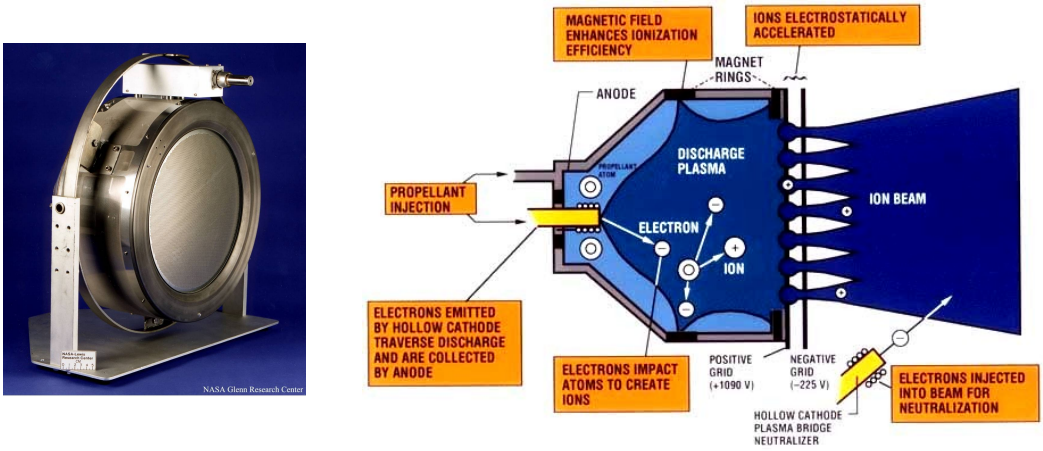


Figure 2.12 NASA's NSTAR ion thruster (from [3])

parameter	symbol	value
NSTAR ion propulsion system:		
minimum PPU input power	$P_{\min}$	0.5 kW
maximum PPU input power	$P_{\max}$	2.0 kW
mass	m_{Thr}	48.0 kg
SCARLET solar array wing:		
power production at 1 AU	$P_{\mathrm{SAW},0}$	2.5 kW
mass	m_{SAW}	27.7 kg

Table 2.2 Technical data for the NSTAR ion propulsion system and the SCARLET solar array wing [68]

Within the SEP system model, the spacecraft is made of five components,

1. a number n_{Thr} of ion thrusters (mass m_{Thr} each)
2. the (empty) propellant tank(s) (mass m_{Tank})
3. propellant (mass m_{P})
4. a number n_{SAW} of solar array wings (mass m_{SAW} each)
5. the payload (mass m_{PL}, including the spacecraft bus).

Hence, the total spacecraft mass is

$$m = n_{\mathrm{Thr}} \cdot m_{\mathrm{Thr}} + m_{\mathrm{Tank}} + m_{\mathrm{P}} + n_{\mathrm{SAW}} \cdot m_{\mathrm{SAW}} + m_{\mathrm{PL}} \tag{2.45}$$

2.3.3.1 Single Thruster

The key parameter for the NSTAR thruster is the input power P_{PPU} that is available to the power processing unit (PPU) of the thruster. This power is delivered by the n_{SAW} solar array wings. The power that is available to the spacecraft at a distance r from the sun is

$$P(r) = n_{\mathrm{SAW}} \cdot P_{\mathrm{SAW},0} \left(\frac{r_0}{r}\right)^{\varkappa} \tag{2.46}$$

where $\varkappa < 2$ may be used to account for an improved solar array performance at low temperatures. If the power P_{Sys} is needed to operate the spacecraft systems, the remaining power that is available to generate thrust is

$$P_{\mathrm{av}}(r) = P(r) - P_{\mathrm{Sys}} \tag{2.47}$$

For SEP systems, a throttle factor $0 \leq \chi \leq 1$ is used to control the PPU power input, so that one gets

$$P_{\mathrm{PPU}}(\chi, r) = \begin{cases} 0, & \text{if } \chi P_{\mathrm{av}}(r) < P_{\min} \\ \chi P_{\mathrm{av}}(r), & \text{if } P_{\min} \leq \chi P_{\mathrm{av}}(r) \leq P_{\max} \\ P_{\max}, & \text{if } \chi P_{\mathrm{av}}(r) > P_{\max} \end{cases} \tag{2.48}$$

For the NSTAR thruster, according to [97], the following polynomial approximation for propellant mass flow rate (in mg/s) and thrust (in mN) can be used in the power range $P_{\min} \leq P_{\mathrm{PPU}} \leq P_{\max}$:

$$\dot{m}_{\mathrm{P}}(\chi, r) = 0.74343 + 0.20951\, P_{\mathrm{PPU}} + 0.25205\, P_{\mathrm{PPU}}^2 \tag{2.49a}$$

$$F(\chi, r) = -3.4318 + 37.365\, P_{\mathrm{PPU}} \tag{2.49b}$$

if P_{PPU} is given in kW (figure 2.13). The minimum and maximum mass flow rate and thrust are then

$$\dot{m}_{\mathrm{P,min}} = 0.91120\,\mathrm{mg/s} \qquad \dot{m}_{\mathrm{P,max}} = 2.1707\,\mathrm{mg/s}$$

$$F_{\min} = 15.251\,\mathrm{mN} \qquad F_{\max} = 71.298\,\mathrm{mN}$$

The specific impulse is variable and depends on solar distance and throttle

$$\mathrm{I}_{\mathrm{sp}}(\chi, r) = \frac{F(\chi, r)}{\dot{m}_{\mathrm{P}}(\chi, r) \cdot g_0} \tag{2.50}$$

The maximum specific impulse is $\mathrm{I}_{\mathrm{sp}_{\max}} = 3231.5\,\mathrm{s}$ for $P_{\mathrm{PPU}} = P_{\mathrm{opt}} = 1.8337\,\mathrm{kW}$ (see figure 2.13). Figure 2.14 shows that – contrary to the NEP system model – both the maximum thrust and the maximum specific impulse of the SEP system decrease rapidly as the spacecraft moves farther from the sun, and drop to zero at the so-called **thrust cutoff distance** (at $r \gtrsim 1.83\,\mathrm{AU}$, if $P_{\mathrm{Sys}} = 1\,\mathrm{kW}$).

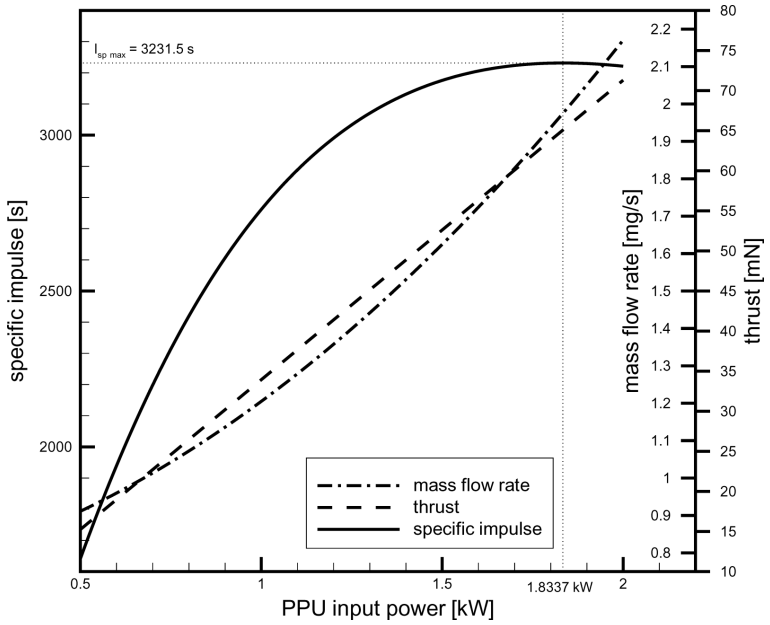


Figure 2.13 Dependence of NSTAR mass flow rate, thrust, and specific impulse on PPU input power

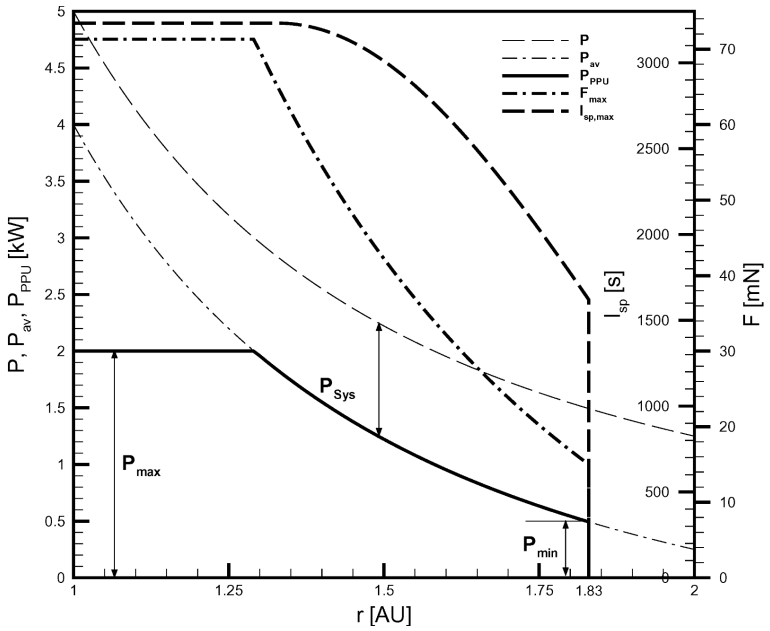


Figure 2.14 Dependence of NSTAR power levels, thrust, and specific impulse on solar distance (for $\varkappa = 2$ and $P_{\mathrm{Sys}} = 1\,\mathrm{kW}$)

2.3.3.2 Clustered Thrusters

Until today, no spacecraft has been built with more than one electric thruster for primary propulsion. However, for faster transfer times and/or heavier payloads, larger thrust levels are required. Those could not only be provided by larger thrusters but also by clustering the currently available thrusters. Another advantage of thruster clustering is the extension of the operational range of the SEP system to larger solar distances (see appendix B).

2.3.4 Equations of Motion for EP Spacecraft

Within this section, the equations of heliocentric translational motion within the one-body simulation model and without disturbing forces are derived for the EP spacecraft models.

Using equations (2.42) and (2.44), one gets for NEP spacecraft

$$\ddot{\mathbf{r}} = \frac{1}{m}\left(\mathbf{F}(\chi) + \mathbf{F}_{\mathrm{G}}(r)\right) = \chi\frac{F_{\max}}{m}\,\mathbf{f} - \frac{\mu}{r^2}\mathbf{e}_r \tag{2.51a}$$

$$\dot{m} = -\chi\dot{m}_{\mathrm{P,max}} \tag{2.51b}$$

and, using the respective equations for $\dot{m}_{\mathrm{P}}(\chi, r)$ and $\mathbf{F}(\chi, r)$, for SEP spacecraft

$$\ddot{\mathbf{r}} = \frac{1}{m}\left(\mathbf{F}(\chi, r) + \mathbf{F}_{\mathrm{G}}(r)\right) = \frac{F(\chi, r)}{m}\,\mathbf{f} - \frac{\mu}{r^2}\mathbf{e}_r \tag{2.52a}$$

$$\dot{m} = -\dot{m}_{\mathrm{P}}(\chi, r) \tag{2.52b}$$

The orientation of the thrust unit vector $\mathbf{f}$ – expressed by the thrust clock angle γ and the thrust cone angle δ – and the throttle χ constitute now the spacecraft control vector, i.e. $\mathbf{u} = (\gamma, \delta, \chi)$.

Resolving equation (2.51a) or (2.52a) along the $\mathcal{E}$-frame unit vectors will give three 2$^{\mathrm{nd}}$ order DEs for r, φ, and θ respectively.

Resolving $\mathbf{f}$ along the $\mathcal{O}$-frame unit vectors, one obtains

$$\mathbf{f} = \cos\delta\mathbf{e}_r + \cos\gamma\sin\delta\mathbf{e}_t + \sin\gamma\sin\delta\mathbf{e}_h \tag{2.53}$$

and after transformation into the $\mathcal{E}$-frame

$$\mathbf{f} = \cos\delta\mathbf{e}_r + \cos(\gamma+\zeta)\sin\delta\mathbf{e}_\varphi + \sin(\gamma+\zeta)\sin\delta\mathbf{e}_\theta \tag{2.54}$$

2.3.4.1 Equations of Motion for NEP Spacecraft

By introducing four dimensionless control functions u_1' to u_4', depending only on the three control variables γ, δ, χ, and the $\mathcal{E}$-$\mathcal{O}$-rotation angle ζ

$$u_1'(\delta, \chi) = \chi\cos\delta \tag{2.55a}$$

$$u_2'(\gamma+\zeta, \delta, \chi) = \chi\cos(\gamma+\zeta)\sin\delta \tag{2.55b}$$

$$u_3'(\gamma+\zeta, \delta, \chi) = \chi\sin(\gamma+\zeta)\sin\delta \tag{2.55c}$$

$$u_4'(\chi) = \chi \tag{2.55d}$$

and using equation (2.31), the spacecraft acceleration may be written in $\mathcal{E}$-frame components as:

$$\ddot{\mathbf{r}} = (\frac{F_{\max}}{m}u_1' - \frac{\mu}{r^2})\mathbf{e}_r + \frac{F_{\max}}{m}u_2'\mathbf{e}_\varphi + \frac{F_{\max}}{m}u_3'\mathbf{e}_\theta \tag{2.56}$$

Expressing $\ddot{\mathbf{r}}$ in polar coordinates (see appendix A.2), one gets – after some rearrangement – the DES of motion for NEP spacecraft:

$$\ddot{r} = r\dot{\theta}^2 + r\dot{\varphi}^2\cos^2\theta - \frac{\mu}{r^2} + \frac{F_{\max}}{m}u_1' \tag{2.57a}$$

$$\ddot{\varphi} = -2\frac{\dot{r}\dot{\varphi}}{r} + 2\dot{\varphi}\dot{\theta}\tan\theta + \frac{F_{\max}}{m}\frac{u_2'}{r\cos\theta} \tag{2.57b}$$

$$\ddot{\theta} = -2\frac{\dot{r}\dot{\theta}}{r} - \dot{\varphi}^2\sin\theta\cos\theta + \frac{F_{\max}}{m}\frac{u_3'}{r} \tag{2.57c}$$

$$\dot{m} = -u_4'\dot{m}_{\mathrm{P,max}} \tag{2.57d}$$

2.3.4.2 Equations of Motion for SEP Spacecraft

For SEP spacecraft, introducing

$$u_1'(\delta) = \cos\delta \tag{2.58a}$$

$$u_2'(\gamma+\zeta,\delta) = \cos(\gamma+\zeta)\sin\delta \tag{2.58b}$$

$$u_3'(\gamma+\zeta,\delta) = \sin(\gamma+\zeta)\sin\delta \tag{2.58c}$$

$$u_4'(\chi) = \chi \tag{2.58d}$$

one obtains likewise the DES of motion for SEP spacecraft:

$$\ddot{\mathbf{r}} = (\frac{F(u_4',r)}{m}u_1' - \frac{\mu}{r^2})\mathbf{e}_r + \frac{F(u_4',r)}{m}u_2'\mathbf{e}_\varphi + \frac{F(u_4',r)}{m}u_3'\mathbf{e}_\theta \tag{2.59}$$

$$\ddot{r} = r\dot{\theta}^2 + r\dot{\varphi}^2\cos^2\theta - \frac{\mu}{r^2} + \frac{F(u_4',r)}{m}u_1' \tag{2.60a}$$

$$\ddot{\varphi} = -2\frac{\dot{r}\dot{\varphi}}{r} + 2\dot{\varphi}\dot{\theta}\tan\theta + \frac{F(u_4',r)}{m}\frac{u_2'}{r\cos\theta} \tag{2.60b}$$

$$\ddot{\theta} = -2\frac{\dot{r}\dot{\theta}}{r} - \dot{\varphi}^2\sin\theta\cos\theta + \frac{F(u_4',r)}{m}\frac{u_3'}{r} \tag{2.60c}$$

$$\dot{m} = -\dot{m}_{\mathrm{P}}(u_4',r) \tag{2.60d}$$

3 Traditional Trajectory Optimization

Within this chapter, the low-thrust trajectory optimization problem is characterized and it is shown how low-thrust trajectory optimization problems are traditionally solved by using local spacecraft steering laws and numerical optimal control methods. Within section 3.1, the low-thrust trajectory optimization problem is formally stated, generally, and from the perspective of optimal control theory. Special emphasis is placed on the potential objectives for trajectory optimization and on the differences between trajectory optimization for high-thrust and low-thrust propulsion systems. It is shown that low-thrust trajectory optimization is equivalent to the search for the optimal spacecraft control function in infinite function space, which is a very difficult problem that can only be solved approximately through discretization. Within section 3.3, a short survey of traditional trajectory optimization methods is presented. All those methods can be generally classified as *local* trajectory optimization methods, where the term optimization does not mean "finding *the best* solution" but rather "finding *a* solution". Their main drawback is the dependence of their convergence behavior on an adequate initial guess of the solution, which is often hard to find. Even if convergence is achieved, a local optimum is typically found, which is close to the initial guess that is rarely close to the (unknown) global optimum. The initial guess is typically generated through the (subsequent or parallel) application of local steering laws, that give the locally optimal thrust direction to change some specific osculating[1] orbital element of the spacecraft with a maximum rate. This is why local spacecraft steering laws are addressed already in section 3.2. Finally, within section 3.4, it is sketched how a *smart global* trajectory optimization should look like.

3.1 The Low-Thrust Trajectory Optimization Problem

In general, a spacecraft trajectory may be defined as an image of some time interval $[t_0, t_f]$ in some six-dimensional **spacecraft state** space[2] $\{\mathbf{x}_{\mathrm{SC}}\} \subset \mathbb{R}^6$. A trajectory is obtained by using some **spacecraft control function** U that maps some input domain[3] onto a **spacecraft control vector** $\mathbf{u} \in \mathbb{R}^{n_u}$, which completely defines the magnitude and the direction of the spacecraft's thrust force $\mathbf{F}$. The trajectory, which is denoted by $\mathbf{x}_{\mathrm{SC}}[t]$ or $\mathbf{x}_{\mathrm{SC}}[t_0, t_f]$, is then obtained from the (numerical) integration of the spacecraft's **equation of motion**,[4] $\dot{\mathbf{x}}_{\mathrm{SC}}(t) = \mathsf{G}(\mathbf{x}_{\mathrm{SC}}(t), \mathbf{u}(t))$, $\mathsf{G} : \mathbb{R}^{6+n_u} \mapsto \mathbb{R}^6$, which is the **dynamic constraint**.

3.1.1 Objectives for Trajectory Optimization

The optimality of spacecraft trajectories may be defined according to several objectives, e.g.:

(1) the transfer time is minimal for a given propellant and payload mass (i.e. for a given launch mass)

---

[1] the instantaneous orbital element that would be obtained if the thrust was turned off

[2] $\mathbf{x}_{\mathrm{SC}}$ is usually the position $\mathbf{r}_{\mathrm{SC}} \in \mathbb{R}^3$ plus the velocity $\dot{\mathbf{r}}_{\mathrm{SC}} \in \mathbb{R}^3$ in some frame of reference, or some set of orbital elements $\mathbf{E} \in \mathbb{R}^6$ (see appendix A)

[3] typically the same time interval $[t_0, t_f]$

[4] if $\mathbf{x}_{\mathrm{SC}} = (\mathbf{r}_{\mathrm{SC}}, \dot{\mathbf{r}}_{\mathrm{SC}})$, the DES of motion with three DEs of 2^{nd} order can be transformed into an equivalent DES with six DEs of 1^{st} order

(2) the required propellant mass is minimal for a given transfer time and launch mass[5]

(3) the required propellant mass is minimal for a given transfer time and payload mass[6]

Even for solar sailcraft, which do not consume any propellant, several objectives may be defined for trajectory optimization, e.g.:

(1) the transfer time is minimal for a given solar sail performance

(2) the required solar sail performance is minimal for a given transfer time

In general, spacecraft trajectories have to be optimized with respect to more than one objectives, e.g. transfer time *and* propellant consumption *and* mission objectives. In practice, such **multi-objective optimization** problems are usually reduced to single-objective problems. This can be done in two ways [42]:

- only one objective is subject to optimization. The other objectives are introduced as constraints, e.g. "maximize the payload mass for a given maximum transfer time and launch mass".
- each objective is associated with a weighting factor and the weighted objectives are combined into a single scalar value. This reduction introduces new parameters in the form of weighting factors. To set a proper combination of values for them, one must be familiar with the relationship between the different objectives to obtain the desired results. Hence, the determination of suitable weighting factors can become an optimization problem of its own.

The result of such objective reduction techniques is a single solution, that does typically not reflect the possible compromises between the different objectives. Probably the best way to tackle multi-objective trajectory optimization problems is to apply the concept of **Pareto-optimality**. According to this concept, every trajectory is Pareto-optimal that is not dominated by some other trajectory, which is better in *all* objectives. Thus, a Pareto-optimal trajectory can only be improved with respect to some single objective at the expense of at least one other objective.[7]

3.1.2 High-Thrust and Low-Thrust Trajectory Optimization

For spacecraft with high thrust like chemical rockets, the problem of finding optimal interplanetary trajectories can be solved relatively easily, as long as no gravity assist maneuvers are required, since only a few thrust phases are necessary (see example below). These thrust phases are very short compared to the transfer time, so that they can be approximated by

[5] this maximizes the payload mass

[6] this also minimizes the launch mass and thus the launcher requirements

[7] Multi-objective trajectory optimization that relies on the concept of Pareto-optimality was done by Hartmann et al. [42], where a "Pareto Genetic Algorithm" was applied in combination with NASA's "Solar Electric Propulsion Trajectory Optimization Program" (SEPTOP) to optimize Earth-Mars trajectories for SEP spacecraft with respect to transfer time *and* payload mass.

singular events that change the spacecraft's velocity $\dot{\mathbf{r}}_{\mathrm{SC}}$ instantaneously while its position $\mathbf{r}_{\mathrm{SC}}$ remains fixed.

> **Example:** Let A be the initial body and B the target body. For an interplanetary fly-by trajectory, only a single thrust impulse at A ($\Delta\mathsf{V}_\mathsf{A}$) is required, so that the solution space of the problem has only three dimensions (e.g. thrust magnitude and two directional angles). In the case of an interplanetary rendezvous problem, another thrust impulse at B ($\Delta\mathsf{V}_\mathsf{B}$) is necessary to "stop" the spacecraft. However, this thrust impulse is completely defined by the velocity vector at B, so that no additional optimization parameters are added to the problem. For the rendezvous problem, an intermediate orbit cranking maneuver (OCM) that adjusts the spacecraft's inclination in the intersection line of A's and B's orbital plane even reduces the dimension of the optimization problem by one. In this case, $\Delta\mathsf{V}_\mathsf{A}$ is within the orbital plane of A, the magnitude and direction of $\Delta\mathsf{V}_\mathsf{OCM}$ are completely defined by A's and B's orbital plane and by the spacecraft's velocity at the OCM-point, and $\Delta\mathsf{V}_\mathsf{B}$ is again defined by the velocity vector at B.

Interplanetary low-thrust missions require the propulsion system to operate for a significant part of the transfer to generate the necessary velocity increment ΔV. Consequently, the thrust vector $\mathbf{F}(t)$ is a continuous function of time and the dimension of the solution space is infinite. $\mathbf{F}(t)$ is manipulated through an n_u-dimensional vector of control variables $\mathbf{u}(t)$ (e.g. throttle χ and two directional angles like γ and δ) that is also a continuous function of time. Thus, the trajectory optimization problem is equivalent to the problem of finding the optimal **spacecraft control function** $\mathsf{U}^\star$ in infinite-dimensional function space. This problem can not be solved except in very rare cases [77]. What can be solved at least numerically, however, is a discrete approximation of the problem (see section 3.1.3.2).

3.1.3 Low-Thrust Trajectory Optimization from the Perspective of Optimal Control Theory

Within this work, three types of trajectory optimization problems are considered: **rendezvous problems**, **fly-by problems**, and **orbit transfer problems**. Within this section, these problems are stated from the perspective of optimal control theory, both in continuous and discrete time. Later on (section 4.1), it will be shown that these problems may also be formulated within a different context: from the perspective of artificial intelligence and machine learning.

3.1.3.1 Problem Formulation in Continuous Time

In terms of optimal control theory, the rendezvous problem RV, the fly-by problem FB, and the orbit transfer problem OT can be stated as follows:

Rendezvous problem (RV) from the perspective of optimal control theory:
Find a spacecraft control function $\mathsf{U} : t \in [t_0, t_f] \mapsto \mathbf{u} \in \mathbb{R}^{n_u}$, which forces the state $\mathbf{x}_{\mathrm{SC}}(t) = (\mathbf{r}_{\mathrm{SC}}(t), \dot{\mathbf{r}}_{\mathrm{SC}}(t))$ of the spacecraft from its initial value $\mathbf{x}_{\mathrm{SC}}(t_0)$ to the state $\mathbf{x}_{\mathrm{T}}(t)$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\mathrm{SC}}(t) = \mathsf{G}(\mathbf{x}_{\mathrm{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{x}_{\mathrm{SC}}(t_f) = \mathbf{x}_{\mathrm{T}}(t_f)$, and at the same time minimizes some cost function J.

Fly-by problem (*FB*) from the perspective of optimal control theory:
Find a spacecraft control function $\mathsf{U} : t \in [t_0, t_f] \mapsto \mathbf{u} \in \mathbb{R}^{n_u}$, which forces the position $\mathbf{r}_{\mathrm{SC}}(t)$ of the spacecraft from its initial value $\mathbf{r}_{\mathrm{SC}}(t_0)$ to the position $\mathbf{r}_{\mathrm{T}}(t)$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\mathrm{SC}}(t) = \mathsf{G}(\mathbf{x}_{\mathrm{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{r}_{\mathrm{SC}}(t_f) = \mathbf{r}_{\mathrm{T}}(t_f)$, and at the same time minimizes some cost function J.

Orbit transfer problem (*OT*) from the perspective of optimal control theory:
Find a spacecraft control function $\mathsf{U} : t \in [t_0, t_f] \mapsto \mathbf{u} \in \mathbb{R}^{n_u}$, which forces the spacecraft's set of orbital elements $\mathbf{Z}_{\mathrm{SC}}(t)$ that defines the size, shape, and orientation of the orbit (e.g. $\mathbf{Z}_{\mathrm{SC}}(t) = (a(t), e(t), \iota(t), \Omega(t), \omega(t))$[8] from its initial value $\mathbf{Z}_{\mathrm{SC}}(t_0)$ to the respective set of orbital elements $\mathbf{Z}_{\mathrm{T}}(t)$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\mathrm{SC}}(t) = \mathsf{G}(\mathbf{x}_{\mathrm{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{Z}_{\mathrm{SC}}(t_f) = \mathbf{Z}_{\mathrm{T}}(t_f)$, and at the same time minimizes some cost function J.

The resulting state function $\mathbf{x}^{\star}_{\mathrm{SC}}[t]$ is the optimal trajectory for the given problem. So, all the trajectory optimization problems stated above are actually problems of finding the optimal spacecraft control function $\mathsf{U}^{\star}$. Both t_0 and t_f can be fixed or free, so that they are part of the optimization problem in the latter case.

If the propellant mass m_{P} is to be minimized,

$$J_{m_{\mathrm{P}}} = \int_{t_0}^{t_f} \dot{m}_{\mathrm{P}} dt = m_{\mathrm{P}}(t_f) - m_{\mathrm{P}}(t_0) \tag{3.1}$$

is an appropriate cost function. If the transfer time T is to be minimized,

$$J_T = \int_{t_0}^{t_f} dt = t_f - t_0 = T \tag{3.2}$$

is an appropriate cost function.[9]

3.1.3.2 Problem Formulation in Discrete Time

Generally, problems *RV*, *FB*, and *OT* can not be solved analytically. This makes a discrete approximation of the problem necessary, which converts the infinite-dimensional problem into a finite-dimensional problem by using some numerical discretization method. If the maximum transfer time interval $[t_0, t_{f,\max}]$ is cut into τ finite elements $\bar{t}_0 = t_0, \ldots, \bar{t}_f = t_f \leq \bar{t}_\tau = t_{f,\max}$ and $\overline{\mathsf{U}} : \{\bar{t}_0, \ldots, \bar{t}_{\tau-1}\} \mapsto \mathbb{R}^{n_u}$ is the discrete spacecraft control function, the optimal solution of the approximate problem is a $n_u f$-dimensional subspace in the $n_u \tau$-dimensional search space, which is usually still a very high-dimensional space. The approximate trajectory optimization problem is then to find an optimal **spacecraft control vector history** $\mathbf{u}^{\star}[\bar{t}] \in \mathbb{R}^{n_u \tau}$, which gives the optimal trajectory $\mathbf{x}^{\star}_{\mathrm{SC}}[t] = \mathbf{x}^{\star}_{\mathrm{SC}}[\bar{t}_0, \bar{t}_f]$.[10] Through discretization, the problem of finding $\mathsf{U}^{\star}$ as an optimal function in infinite-dimensional function space reduces to the problem of finding the optimal control vector history $\mathbf{u}^{\star}[\bar{t}]$ in a finite-dimensional parameter space. The discrete rendezvous problem ($\overline{RV}$) for example may now be stated as follows:

---

[8] of course any other set of equivalent orbital elements may be used alternatively

[9] Propellant mass minimization and transfer time minimization would be equivalent if $\dot{m}_{\mathrm{P}}$ was constant.

[10] Is is to note that only the spacecraft control vector function is discretized, whereas the trajectory is still continuous.

Rendezvous problem ($\overline{RV}$) from the perspective of optimal control theory:
Find a spacecraft control vector history $\mathbf{u}[\bar{t}]$ ($\bar{t} \in \{\bar{t}_0, \ldots, \bar{t}_{f-1}\}$), which forces the state $\mathbf{x}_{\mathrm{SC}}(t) = (\mathbf{r}_{\mathrm{SC}}(t), \dot{\mathbf{r}}_{\mathrm{SC}}(t))$ of the spacecraft from its initial value $\mathbf{x}_{\mathrm{SC}}(\bar{t}_0)$ to the state $\mathbf{x}_{\mathrm{T}}(\bar{t})$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\mathrm{SC}}(t) = \mathsf{G}(\mathbf{x}_{\mathrm{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{x}_{\mathrm{SC}}(\bar{t}_f) = \mathbf{x}_{\mathrm{T}}(\bar{t}_f)$, and at the same time minimizes some cost function J.

The discrete fly-by problem ($\overline{FB}$) and the discrete orbit transfer problem ($\overline{OT}$) may be stated likewise. The resulting state function $\mathbf{x}^{\star}_{\mathrm{SC}}[t]$ is the optimal trajectory for the given problem. So, all three trajectory optimization problems are actually problems of finding the optimal control vector history $\mathbf{u}^{\star}[\bar{t}]$. The cost functions for the propellant mass optimization problem and the transfer time optimization problem may be defined as before.

3.2 Spacecraft Steering Using Local Steering Laws

3.2.1 Lagrange's Planetary Equations

A **local steering law** (LSL) is an equation (or algorithm) that gives the locally optimal thrust direction that changes some specific osculating orbital element of spacecraft with a maximum rate. To obtainin LSLs, Lagrange's planetary equations in Gauss' form for the orbital reference frame $\mathcal{O}$ (see appendix A.3) may be used, since these equations describe the rate of change of a body's osculating orbital elements (see appendix A.4) due to some (propulsive and/or disturbing) acceleration (or force). As it is shown in [13], this set of equations can be written as[11]

$$\dot{a} = \frac{2a^2}{h}\left(e \sin f a_r + \frac{p}{r} a_t\right) \tag{3.3a}$$

$$\dot{e} = \frac{1}{h}\left\{p \sin f a_r + \left[(p + r)\cos f + re\right] a_t\right\} \tag{3.3b}$$

$$\dot{\imath} = \frac{r\cos(\omega + f)}{h} a_h \tag{3.3c}$$

$$\dot{\Omega} = \frac{r\sin(\omega + f)}{h \sin \imath} a_h \tag{3.3d}$$

$$\dot{\omega} = \frac{1}{eh}\left[-p \cos f a_r + (p + r)\sin f a_t\right] - \frac{r \sin(\omega + f)\cos \imath}{h \sin \imath} a_h \tag{3.3e}$$

$$\dot{f} = \frac{h}{r^2} + \frac{1}{eh}\left[p \cos f a_r - (p + r)\sin f a_t\right] \tag{3.3f}$$

where a_r, a_t, and a_h are the acceleration components along the $\mathcal{O}$-frame unit vectors, $h = \sqrt{\mu a(1 - e^2)}$ is the orbital angular momentum per spacecraft unit mass, and $p = h^2/\mu$ is the semilatus rectum of the orbit.

[11] The variational equation for either the mean anomaly M or the eccentric anomaly E may be used in place of equation (3.3f).

3.2.2 Pure Local Steering Laws for Spacecraft

3.2.2.1 Changing the Orbital Elements with a Maximum Rate

Equations (3.3) are now used to determine a set $\mathcal{L}$ of local steering laws, which change the osculating orbital elements of spacecraft with a maximum rate. Let $\mathbf{Z} = (a, e, \iota, \Omega, \omega)$ denote the vector of the orbital elements except the anomaly.[12] Then, the variational equation for some particular orbital element $Z_{i\in\{1,\ldots,5\}}$ has the form

$$\dot{Z}_i = k_{Z_i,r} \cdot a_r + k_{Z_i,t} \cdot a_t + k_{Z_i,h} \cdot a_h = \mathbf{k}_{Z_i} \cdot \mathbf{a}$$

where $\mathbf{k}_{Z_i} = (k_{Z_i,r}, k_{Z_i,t}, k_{Z_i,h})$ is a vector of functions of the spacecraft's actual state, and $\mathbf{a} = (a_r, a_t, a_h)$ is the spacecraft's acceleration vector due to propulsive and/or disturbing forces. It is clear that in order to change Z_i with a maximum rate, the acceleration along $\mathbf{k}_{Z_i}$ or $-\mathbf{k}_{Z_i}$ must be maximized. Thus, the set $\mathcal{L}$ consists of 10 LSLs $L_{j\in\{1,\ldots,10\}}$ $(i \in \{1, \ldots, 5\})$:

$$\begin{array}{lll} L_i & := \text{maximize} \quad \dot{Z}_i & \Leftrightarrow \quad \mathbf{a} \uparrow\uparrow \mathbf{k}_{Z_i} \\ L_{5+i} & := \text{maximize} \quad -\dot{Z}_i & \Leftrightarrow \quad \mathbf{a} \uparrow\downarrow \mathbf{k}_{Z_i} \end{array} \tag{3.4}$$

Example: $\mathsf{L}_6 = \mathsf{L}_{5+1} :=$ maximize $-\dot{\mathsf{Z}}_1 =$ maximize $-\dot{\mathsf{a}}$.

It will become very convenient to introduce for each LSL a **direction unit vector** $\mathbf{d}_{L_j}$ in a way that $\mathbf{d}_{L_i} = \mathbf{k}_{Z_i}/|\mathbf{k}_{Z_i}|$ and $\mathbf{d}_{L_{5+i}} = -\mathbf{k}_{Z_i}/|\mathbf{k}_{Z_i}|$.

3.2.2.2 Adjusting the Orbital Elements with a Maximum Rate

In the same way, a set $\mathcal{Q}$ of local steering laws $Q_{i\in\{1,\ldots,5\}}$ is defined, so that each Q_i gives a direction, along which the thrust force has to be maximized in order to minimize the difference between some specific orbital element Z_i and its desired value $\tilde{Z}_i$ (which can be smaller or larger, e.g. the respective orbital element of the target body) with a maximum rate:

$$Q_i := \begin{cases} L_i \ , \text{ if } \tilde{Z}_i \geq Z_i \\ L_{5+i}, \text{ if } \tilde{Z}_i < Z_i \end{cases} \tag{3.5}$$

and

$$\mathbf{d}_{Q_i} := \begin{cases} \mathbf{d}_{L_i} \ , \text{ if } \tilde{Z}_i \geq Z_i \\ \mathbf{d}_{L_{5+i}} = -\mathbf{d}_{L_i}, \text{ if } \tilde{Z}_i < Z_i \end{cases} \tag{3.6}$$

3.2.3 Blended Local Steering Laws for Spacecraft

A close look at equations (3.3) reveals some interesting features for spacecraft steering:

- only in-plane forces change the semi-major axis a, the eccentricity e, and the true anomaly f of the orbit

[12] The reason for this exclusion is given in appendix C.

- only out-of-plane forces change the inclination ι and the ascending node Ω of the orbit
- any force changes the orbit's longitude of perihelion ω
- any force changes at least three orbital elements at the same time
- any force that is neither within the orbit plane nor perpendicular to it changes all orbital elements at the same time

Thus, if a pure LSL is used to change some specific orbital element, the others may change in an unwanted way. Consequently, it is not possible to adapt subsequently one orbital element after the other until some target body or orbit is reached. To change more than one orbital element at the same time in the desired direction (i.e. to a larger or smaller value), it is apparent to "blend" the pure local steering laws in $\mathcal{L}$ or $\mathcal{Q}$ so that they constitute a set $\mathcal{M}_\mathcal{L}$ or $\mathcal{M}_\mathcal{Q}$ of **blended steering laws**. For that reason, a vector $\mathbf{c}_L \in \mathbb{R}^{10}$ or $\mathbf{c}_Q \in \mathbb{R}^5$ of weight factors (called **steering law weight vector**) may be defined in a way that each steering law weight vector defines a blended local steering law $M_L(\mathbf{c}_L) \in \mathcal{M}_\mathcal{L}$ or $M_Q(\mathbf{c}_Q) \in \mathcal{M}_\mathcal{Q}$ by giving the locally optimal direction unit vector

$$\mathbf{d}_{M_L(\mathbf{c}_L)} = \frac{\sum_{j=1}^{10} c_{L_j}\mathbf{d}_{L_j}}{\left|\sum_{j=1}^{10} c_{L_j}\mathbf{d}_{L_j}\right|} \qquad \text{or} \qquad \mathbf{d}_{M_Q(\mathbf{c}_Q)} = \frac{\sum_{i=1}^{5} c_{Q_i}\mathbf{d}_{Q_i}}{\left|\sum_{i=1}^{5} c_{Q_i}\mathbf{d}_{Q_i}\right|} \tag{3.7}$$

Since it is often not relevant, whether the pure LSLs are taken from $\mathcal{L}$ or from $\mathcal{Q}$, one can write less formally that a blended LSL $M(\mathbf{c}) \in \mathcal{M}$ is defined by a steering law weight vector $\mathbf{c}$ that gives the direction unit vector $\mathbf{d}$ from some set of pure LSLs.

3.2.4 Locally Optimal Spacecraft Steering

3.2.4.1 Locally Optimal Steering for EP Spacecraft

For EP spacecraft, where the thrust vector can be directed into any desired direction, the thrust clock angle γ^* and thrust cone angle δ^* of the locally optimal thrust unit vector $\mathbf{f}^*$ are the clock angle $\tilde{\gamma}$ and the cone angle $\tilde{\delta}$ of the direction unit vector $\mathbf{d}$:

$$\gamma^* = \tilde{\gamma} = \arctan(d_h, d_t) \tag{3.8a}$$

$$\delta^* = \tilde{\delta} = \operatorname{arccot}\left(\frac{d_r}{\sqrt{d_t^2 + d_h^2}}\right) \tag{3.8b}$$

where $\arctan(y, x)$ is an extended arcustangens, which gives the angle ϕ such that $y = \sin\phi$ and $x = \cos\phi$.

3.2.4.2 Locally Optimal Steering for Solar Sailcraft

It was observed in section 2.2.5 that the thrust vector of solar sailcraft can not be directed into any desired direction but is constrained to lie on the surface of a "bubble" that is directed

away from the sun. Therefore, the locally optimal sail clock angle α^* and sail cone angle β^* are not identical to the locally optimal thrust clock angle γ^* and thrust cone angle δ^*. For the standard SRP force model (see section 2.2.3.4), α^* and β^* can not be calculated analytically from γ^* and δ^*, since the $Q(\beta)$-expression can not be resolved for β . However, at least for the simplified SRP force model (see section 2.2.3.5) they can be expressed analytically. Resolving $\mathbf{f}$ and $\mathbf{d}$ along the $\mathcal{O}$-frame unit vectors, one obtains

$$\mathbf{f} = \cos\delta\mathbf{e}_r + \cos\gamma\sin\delta\mathbf{e}_t + \sin\gamma\sin\delta\mathbf{e}_h \tag{3.9}$$

$$\mathbf{d} = \cos\tilde{\delta}\mathbf{e}_r + \cos\tilde{\gamma}\sin\tilde{\delta}\mathbf{e}_t + \sin\tilde{\gamma}\sin\tilde{\delta}\mathbf{e}_h \tag{3.10}$$

and for the simplified SRP force model

$$\mathbf{f} = \mathbf{n} = \cos\beta\mathbf{e}_r + \cos\alpha\sin\beta\mathbf{e}_t + \sin\alpha\sin\beta\mathbf{e}_h \tag{3.11}$$

so that the SRP force component along $\mathbf{d}$ is

$$\begin{aligned} F_{\mathbf{d}} &= \lambda\frac{\mu m}{r^2}\cos^2\beta(\mathbf{n}\cdot\mathbf{d}) = \\ &= \lambda\frac{\mu m}{r^2}\cos^2\beta\left[\cos\beta\cos\tilde{\delta} + \cos\alpha\sin\beta\cos\tilde{\gamma}\sin\tilde{\delta} + \sin\alpha\sin\beta\sin\tilde{\gamma}\sin\tilde{\delta}\right] \end{aligned} \tag{3.12}$$

Setting the derivative of $F_{\mathbf{d}}$ with respect to α to zero yields the locally optimal sail clock angle α^*

$$\begin{aligned} &\frac{\partial F_{\mathbf{d}}}{\partial\alpha} \stackrel{!}{=} 0 = \lambda\frac{\mu m}{r^2}\sin\beta\cos^2\beta\sin\tilde{\delta}(-\sin\alpha\cos\tilde{\gamma} + \cos\alpha\sin\tilde{\gamma}) \\ \Rightarrow\quad &-\sin\alpha^*\cos\tilde{\gamma} + \cos\alpha^*\sin\tilde{\gamma} = 0 \\ \Leftrightarrow\quad &\sin(\tilde{\gamma} - \alpha^*) = 0 \\ \Rightarrow\quad &\alpha^* = \tilde{\gamma} \end{aligned} \tag{3.13}$$

Calculating the second order derivative $\frac{\partial^2 F_{\mathbf{d}}}{\partial\alpha^2}$, it can be shown that this is truly a maximum. Inserting this result into equation (3.12) yields

$$\begin{aligned} F_{\mathbf{d}} &= \lambda\frac{\mu m}{r^2}\cos^2\beta\left[\cos\beta\cos\tilde{\delta} + \cos^2\tilde{\gamma}\sin\beta\sin\tilde{\delta} + \sin^2\tilde{\gamma}\sin\beta\sin\tilde{\delta}\right] \\ &= \lambda\frac{\mu m}{r^2}\cos^2\beta\left[\cos\beta\cos\tilde{\delta} + \sin\beta\sin\tilde{\delta}\right] \end{aligned} \tag{3.14}$$

Setting now the derivative of $F_{\mathbf{d}}$ with respect to β to zero yields the locally optimal sail cone angle β^*

$$\begin{aligned} &\frac{\partial F_{\mathbf{d}}}{\partial\beta} \stackrel{!}{=} 0 = \lambda\frac{\mu m}{r^2}\left[-3\cos^2\beta\sin\beta\cos\tilde{\delta} + (\cos^3\beta - 2\sin^2\beta\cos\beta)\sin\tilde{\delta}\right] \\ \Rightarrow\quad &-3\cos^2\beta^*\sin\beta^*\cos\tilde{\delta} + \cos^3\beta^*\sin\tilde{\delta} - 2\sin^2\beta^*\cos\beta^*\sin\tilde{\delta} = 0 \\ \Rightarrow\quad &\cot^2\beta^* - 3\cot\tilde{\delta}\cot\beta^* - 2 = 0 \\ \Rightarrow\quad &\beta^* = \operatorname{arccot}\left(\frac{3}{2}\cot\tilde{\delta} \pm \sqrt{\frac{9}{4}\cot^2\tilde{\delta} + 2}\right) \end{aligned}$$

under the condition that $\tilde{\delta} \notin \{0, \pi\}$ ($\Rightarrow \beta^* \notin \{0, \frac{\pi}{2}\}$). Calculating the second order derivative $\frac{\partial^2 F_\mathrm{d}}{\partial \beta^2}$, it can be shown that

$$\beta^* = \operatorname{arccot}\left(\frac{3}{2}\cot\tilde{\delta} + \sqrt{\frac{9}{4}\cot^2\tilde{\delta} + 2}\right) \tag{3.15}$$

for a maximum, where $0 < \tilde{\delta} < \pi$. For the two limiting cases, one gets $\lim_{\tilde{\delta}\to 0}\beta^* = 0$ and $\lim_{\tilde{\delta}\to\pi}\beta^* = \frac{\pi}{2}$. Inserting equations (3.8) into equation (3.13) and (3.15), one gets

$$\alpha^* = \arctan(d_h, d_t) \tag{3.16a}$$

$$\beta^* = \operatorname{arccot}\left(\frac{3}{2}\frac{d_r}{\sqrt{d_t^2 + d_h^2}} + \sqrt{\frac{9}{4}\frac{d_r^2}{d_t^2 + d_h^2} + 2}\right) \tag{3.16b}$$

for the locally optimal sail clock and cone angle.

3.3 Traditional Trajectory Optimization Methods

Traditionally, low-thrust trajectories are optimized by the application of numerical optimal control methods that are based on the calculus of variations. These methods can be divided into **direct methods** such as nonlinear programming (NLP) methods and **indirect methods** such as neighboring extremal methods and gradient methods. Since the theoretical basis of those methods is mathematically extensive and neither necessary to appraise their drawbacks nor to understand the trajectory optimization method that is elaborated within this work, the reader is referred to [15], [77], and [88] for a comprehensive survey of both direct and indirect trajectory optimization methods. The decisive point is that all those approaches can be generally classified as **local trajectory optimization methods** (LTOMs), where the term optimization does not mean "finding *the best* solution" but rather "finding *a* solution" [92]. Prior to optimization, the NLP methods and the gradient methods require an initial guess for the control vector history $\mathbf{u}[\bar{t}]$, whereas the neighboring extremal methods require an initial guess for the starting adjoint vector of LAGRANGE multipliers $\boldsymbol{\lambda}(\bar{t}_0)$ (costate vector) [88]. Figure 3.1 illustrates how trajectory optimization that is based on LTOMs is usually performed.

First, the initial body, the target body, and the initial conditions (launch date t_0, velocity vector $\mathbf{v}_\infty = \dot{\mathbf{r}}_\mathrm{SC}(t_0) - \dot{\mathbf{r}}_\mathrm{Earth}(t_0)$, etc.) are chosen according to the mission objectives and the launcher restrictions. Although those parameters are crucial for mission performance, they are typically chosen according to an expert's judgment and are not part of the actual optimization process. After that, the initial guess for the control vector history $\mathbf{u}[\bar{t}]$ is generated: for this reason, a trajectory simulation is carried out, where different (pure or blended) local steering laws are subsequently applied. The switching between the LSLs is done according to the expert's judgment. The objective is to come as close as possible to the target body, so that in the next step a LTOM is able to converge. If the generated trajectory might not be used as an initial guess, the switching between the LSLs has to be refined and – if several trial-and-error cycles yield no acceptable result – the initial conditions have to be modified (e.g. different launch date or higher hyperbolic excess velocity). All steps require frequent

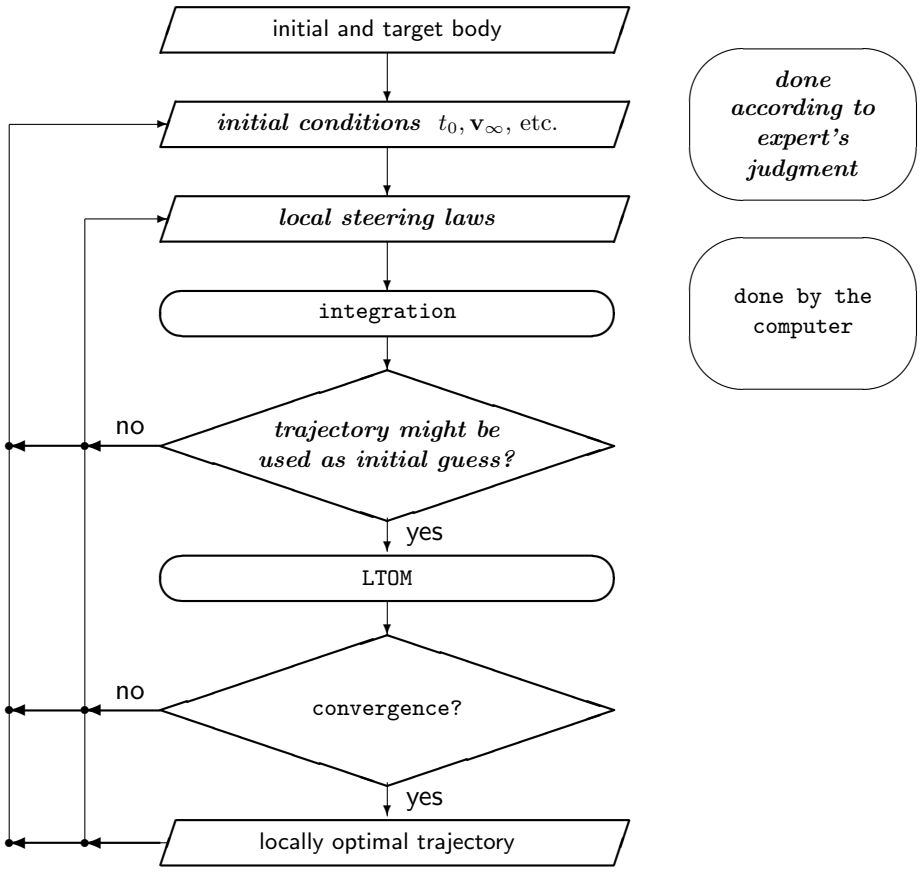


Figure 3.1 Traditional low-thrust trajectory optimization using local trajectory optimization methods

manual interactions. If the trajectory comes finally close enough to the target body, it is taken as the initial guess for the LTOM. If convergence could not be achieved, a new initial guess has to be conceived and the above steps have to be repeated using again a different LSL switching and/or different launch conditions. If the LTOM converges, a locally optimal trajectory is found, which is typically close to the initial guess that is rarely close to the global optimum. The convergence behavior of LTOMs (especially of the indirect methods) is very sensitive to the initial guess. Similar initial guesses often produce very dissimilar optimization results [51, 42], so that trajectory optimization becomes sometimes "more art than science" [42]. This way, the search for a good trajectory usually turns into a time-consuming task.

The drawbacks of LTOMs can be summarized as follows:

1. LTOMs can only be applied by an expert in astrodynamics and optimal control theory.
2. LTOMs require an adequate initial guess prior to optimization, which is often hard to find.
3. The convergence behavior of LTOMs is often very sensitive to the initial guess. Similar

initial guesses often yield dissimilar optimization results, so that the initial guess can not be improved iteratively.

4. LTOMs run often into a local optimum that is close to the initial guess (which is typically far from the global optimum).

5. Trajectory optimization using LTOMs is often very time consuming, since it requires frequent manual interactions and thus practically permanent attendance.

3.4 Concept of a Smart Global Trajectory Optimization Method

Emanating from the drawbacks of LTOMs, a *smart* **global trajectory optimization method** (GTOM) is sought that runs without an initial guess and without the permanent attendance of an expert in astrodynamics and optimal control theory, as it is sketched in figure 3.2.

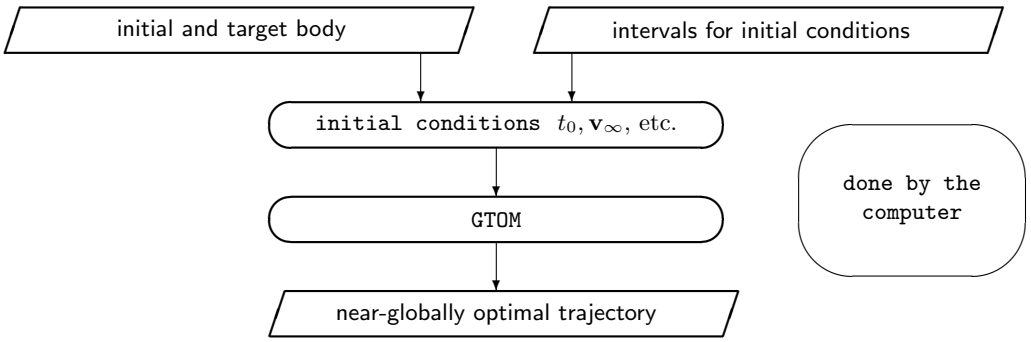


Figure 3.2 Smart low-thrust trajectory optimization using a global trajectory optimization method

4 Trajectory Optimization Using Evolutionary Neurocontrol

It was shown in the last chapter that the problem of searching an optimal spacecraft trajectory $\mathbf{x}^{\star}_{\mathrm{SC}}[t]$ is equivalent to the problem of searching an optimal spacecraft control function $\mathsf{U}^{\star}$. Usually, optimal control methods that are based on the calculus of variations are employed to solve this kind of problems. Prior to optimization, they require an initial guess of the solution, which is typically generated through the (subsequent or parallel) application of local spacecraft steering laws.

Within this chapter, spacecraft trajectory optimization is attacked from a perspective different to that of optimal control: the perspective of artificial intelligence and machine learning. Within this context, a trajectory can be regarded as the result of an explicitly not time-dependent spacecraft steering strategy S that maps the problem relevant variables (e.g. the spacecraft state $\mathbf{x}_{\mathrm{SC}}$ and the target body state $\mathbf{x}_{\mathrm{T}}$) onto some spacecraft control vector, $\mathsf{S} : \{\mathbf{x}_{\mathrm{SC}}, \mathbf{x}_{\mathrm{T}}\} \subset \mathbb{R}^{12} \mapsto \{\mathbf{u}\} \subset \mathbb{R}^{n_u}$ (section 4.1). This way, the problem of searching the optimal spacecraft trajectory is equivalent to the problem of searching (or "learning") the optimal spacecraft steering strategy $\mathsf{S}^{\star}$. An artificial neural network (ANN) may be used as a so-called neurocontroller (NC) to implement such spacecraft steering strategies. It can be regarded as a parameterized function N (the network function) that is – for a given network topology – completely defined by the internal parameter vector $\boldsymbol{\pi} \in \mathbb{R}^{n_\pi}$ of the network. Therefore, each $\boldsymbol{\pi}$ defines a steering strategy $\mathsf{S}_{\boldsymbol{\pi}}$. The problem of searching the optimal spacecraft trajectory is thus equivalent to the problem of searching the optimal parameter vector $\boldsymbol{\pi}^{\star}$ for a given neurocontroller. Evolutionary algorithms (EAs) that work on a population of strings can be used for finding the optimal network parameters, since the parameters can be mapped onto a string $\boldsymbol{\xi}$ (also called chromosome or individual, section 4.3). The trajectory optimization problem is solved, when the optimal chromosome $\boldsymbol{\xi}^{\star}$ is found. Figure 4.1 sketches the subsequent transformations of the optimal chromosome into the optimal trajectory.

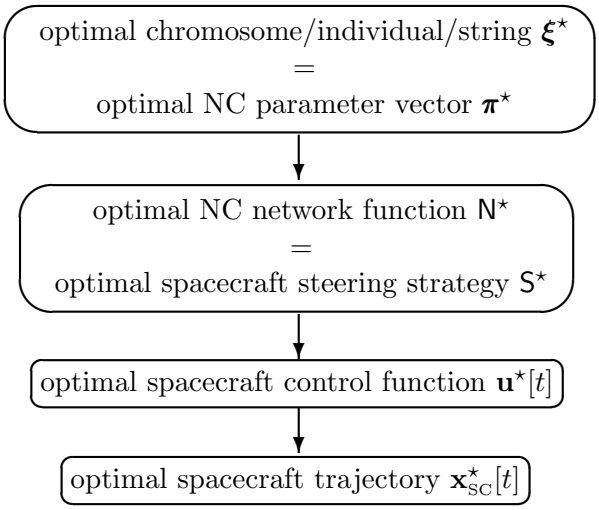


Figure 4.1 From the optimal chromosome to the optimal trajectory

A neurocontroller that employs an evolutionary algorithm for "learning" (or "breeding") the optimal control strategy might be called evolutionary neurocontroller (ENC, section 4.4 and 4.5).

4.1 Spacecraft Steering Using Steering Strategies

4.1.1 Machine Learning

Within the field of artificial intelligence, an immense variety and complexity of learning problems, learning methods, and learning systems has evolved, overlapping and poorly systematized. This makes the formulation of a universally accepted definition of **machine learning** difficult. According to De Jong [30], a common denominator of most learning systems is their capability for making structural changes to themselves over time with the intent of improving their performance on tasks (the performance being defined by their environment), discovering and subsequently exploiting interesting concepts, or improving the consistency and generality of internal knowledge structures.

One important and difficult class of learning problems are **reinforcement learning** (RL) problems, where the optimal behavior of the learning system (called **agent**), as it is defined by an **associative mapping** from **situations** to **actions** $\mathsf{S} : \mathcal{X} \mapsto \mathcal{A}$,[1] has to be learned solely through interaction with the **environment**, which gives an immediate or delayed evaluation[2] (**reward** or **reinforcement**) J of the agent's behavior [48, 89]. Within this work, the associative mapping – that is typically called **policy** in RL-related literature – is termed **strategy**. The optimal strategy $\mathsf{S}^\star$ of the agent is defined as the one that maximizes the sum of positive reinforcements and minimizes the sum of negative reinforcements over time. If, given a situation $X \in \mathcal{X}$, the agent tries an action $A \in \mathcal{A}$ and the environment *immediately* returns a scalar evaluation $J(X, A)$ of the (X, A) pair, one is faced with an **immediate reinforcement learning** problem. A more difficult class of learning problems are **delayed reinforcement learning** problems, where the environment gives only a single scalar evaluation, collectively for $(X, A)[t]$, a sequence of (X, A) pairs occurring in time during the agent's operation. Delayed reinforcement learning problems arise commonly in the optimal control of dynamical systems [48].

4.1.2 Low-Thrust Trajectory Optimization from the Perspective of Machine Learning

From the perspective of machine learning, a **spacecraft steering strategy** may be defined as an associative mapping S that gives the actual spacecraft control vector $\mathbf{u}$ from some input vector $\mathbf{X} \in \mathcal{X}$ that comprises the variables that are important for the optimal steering of the spacecraft (the state of the relevant environment, as expressed for example by the spacecraft state $\mathbf{x}_{\mathrm{SC}}$ and the target body state $\mathbf{x}_{\mathrm{T}}$), $\mathsf{S} : \mathcal{X} = \{\mathbf{x}_{\mathrm{SC}}, \mathbf{x}_{\mathrm{T}}\} \mapsto \mathcal{U} = \{\mathbf{u}\}$.[3] The trajectory can

[1] $\mathcal{X}$ is called **state space** and $\mathcal{A}$ is called **action space**

[2] This evaluation is analogous to the cost function in optimal control theory. To emphasize this fact, it will be denoted by the same letter, J. However, one must have in mind that the cost function is subject to minimization, whereas this evaluation is subject to maximization.

[3] where the action space $\mathcal{A}$ is now the spacecraft control vector space $\mathcal{U}$

then be regarded as the result of the spacecraft steering strategy. The search for the optimal strategy is a delayed reinforcement problem, since such a strategy can only be evaluated ex post, when the trajectory is realized and a reward can be given according to the fulfillment of the optimization objective(s). The rendezvous, the fly-by, and the orbit transfer problem may now be reformulated:

Rendezvous Problem (RV) from the perspective of machine learning:
Find a spacecraft steering strategy S, which forces the state $\mathbf{x}_{\mathrm{SC}}(t) = (\mathbf{r}_{\mathrm{SC}}(t), \dot{\mathbf{r}}_{\mathrm{SC}}(t))$ of the spacecraft from its initial value $\mathbf{x}_{\mathrm{SC}}(t_0)$ to the state $\mathbf{x}_{\mathrm{T}}(t)$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\mathrm{SC}}(t) = \mathsf{G}(\mathbf{x}_{\mathrm{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{x}_{\mathrm{SC}}(t_f) = \mathbf{x}_{\mathrm{T}}(t_f)$, and at the same time maximizes some reward J.

Fly-By Problem (FB) from the perspective of machine learning:
Find a spacecraft steering strategy S, which forces the position $\mathbf{r}_{\mathrm{SC}}(t)$ of the spacecraft from its initial value $\mathbf{r}_{\mathrm{SC}}(t_0)$ to the position $\mathbf{r}_{\mathrm{T}}(t)$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\mathrm{SC}}(t) = \mathsf{G}(\mathbf{x}_{\mathrm{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{r}_{\mathrm{SC}}(t_f) = \mathbf{r}_{\mathrm{T}}(t_f)$, and at the same time maximizes some reward J.

Orbit Transfer Problem (OT) from the perspective of machine learning:
Find a spacecraft steering strategy S, which forces the spacecraft's set of orbital elements $\mathbf{Z}_{\mathrm{SC}}(t)$ that defines the size, shape, and orientation of the orbit (e.g. $\mathbf{Z}_{\mathrm{SC}}(t) = (a(t), e(t), \iota(t), \Omega(t), \omega(t))$) from its initial value $\mathbf{Z}_{\mathrm{SC}}(t_0)$ to the respective set of orbital elements $\mathbf{Z}_{\mathrm{T}}(t)$ of the target body along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\mathrm{SC}}(t) = \mathsf{G}(\mathbf{x}_{\mathrm{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{Z}_{\mathrm{SC}}(t_f) = \mathbf{Z}_{\mathrm{T}}(t_f)$, and at the same time maximizes some reward J.

The resulting steering strategy $\mathsf{S}^\star$ is the optimal spacecraft steering strategy for the given problem. So, all the trajectory optimization problems stated above are actually problems of finding the optimal spacecraft steering strategy $\mathsf{S}^\star$. A very obvious way to implement spacecraft steering strategies is to use artificial neural networks, as they have been successfully applied to "learn" the desired associative mapping for a wide range of problems

4.2 Artificial Neural Networks

4.2.1 What are Artificial Neural Networks?

Being inspired by the processing of information in animal nervous systems, ANNs are a computability paradigm that is alternative to conventional serial digital computers. ANNs are massively parallel, analog, fault tolerant, and adaptive [18, 75, 76]. They are composed of processing elements (called **neurons**) that model the most elementary functions of the biological neuron. Linked together, those elements show some characteristics of the brain like learning from experience, generalizing from previous examples to new ones, and extracting essential characteristics from inputs that contain noisy and/or irrelevant data, so that they are relatively insensitive to minor variations in its input to produce consistent output. That is why professionals from many diverse disciplines (e.g. engineering, biology, economics, psychology) are intrigued by their possibilities and apply them within their fields of research.

Since the neurons of an ANN can be modelled and connected in many ways, ANNs exist in a wide variety. According to the connectivity of the neurons, ANNs can be divided into **feedforward** ones and into **recurrent** ones. An ANN is a feedforward one, if there exists

a numbering method, which numbers all neurons in a way that there is no connection from a neuron with a number i to a neuron with a number $j < i$. An ANN is a recurrent one, if such a numbering method does not exist. Within this work, only feedforward ANNs have been considered.

4.2.2 How do Artificial Neural Networks Work?

Typically, feedforward ANNs have a layered topology, where the set $\mathcal{N}$ of neurons is divided into ℓ subsets $\mathcal{N}_1, \ldots, \mathcal{N}_\ell$ (called **neuron layers**) in a way that only connections from $\mathcal{N}_{k-1}$ go to $\mathcal{N}_k$ for all $k \in \{2, \ldots, \ell\}$. $\mathcal{N}_1$ is called **input layer** and has n_1 **input neurons**, which receive the network's input $\mathbf{X} \in \mathcal{X} \subseteq \mathbb{R}^{n_1}$. $\mathcal{N}_\ell$ is called **output layer** and has n_ℓ **output neurons**, which provide the network's output $\mathbf{Y} \in \mathcal{Y} \subset \mathbb{R}^{n_\ell}$. All other layers/neurons are called **hidden layers/neurons** (if $\ell > 2$).

Example: Figure 4.2 shows an ANN with $\ell = 3$ layers, $n_1 = 3$ input neurons, one hidden layer with $n_2 = 2$ hidden neurons and $n_3 = 1$ output neuron (3-2-1-network).

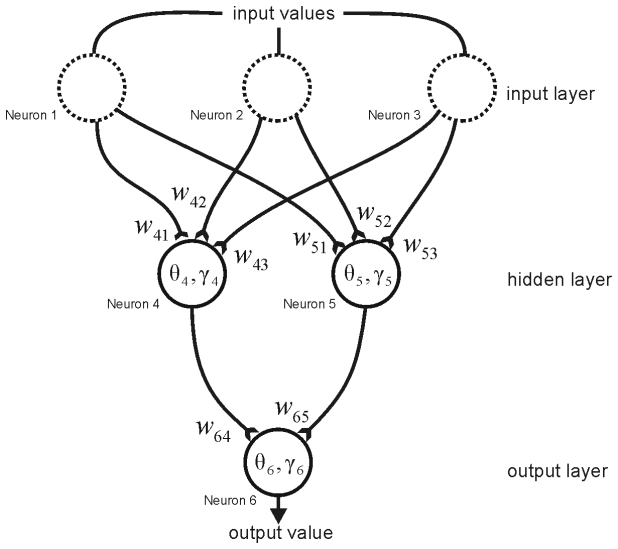


Figure 4.2 Layered feedforward artificial neural network

Each neuron $i \in \mathcal{N}_{k \in \{2,\ldots,\ell\}}$ has a so-called **activation function** that maps from the neuron's weighted input values onto a single output value. The most commonly used activation function for feedforward networks is the **sigmoid** $\mathsf{s}_\gamma : \mathbb{R} \mapsto (0, 1)$, defined by

$$\mathsf{s}_\gamma(x) = \frac{1}{1 + \mathrm{e}^{-x/\gamma}}, \tag{4.1}$$

where the constant γ, called **temperature parameter**, defines the slope of the function (see figure 4.3). Using the sigmoid activation function s_γ, a layered feedforward ANN can be described as a directed graph in which each node (neuron) i in a layer $\mathcal{N}_{k \in \{2,\ldots,\ell\}}$ performs

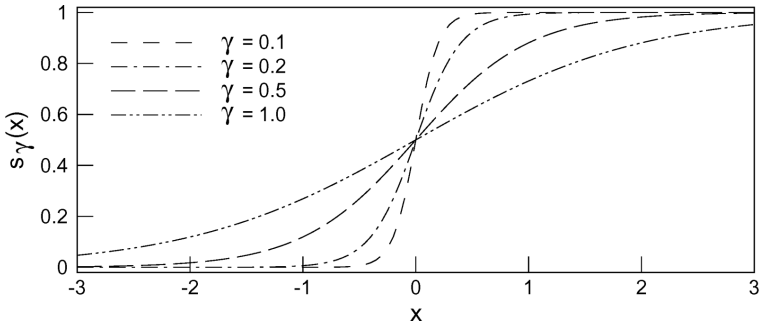


Figure 4.3 The sigmoid $\mathsf{s}_\gamma(x)$ for different values of γ

the function

$$y_i = \frac{1}{1 + \mathrm{e}^{-(\sum_j w_{ij} y_j - \theta_i)/\gamma_i}}, \tag{4.2}$$

where $y_i \in (0,1)$ is the output of neuron i, the $y_j \in (0,1)$ are the output values of the neurons j in the previous layer $\mathcal{N}_{k-1}$, the $w_{ij} \in \mathbb{R}$ are the **connection weights** between the neurons j and neuron i, and $\theta_i \in \mathbb{R}$ is the so-called **bias** (or **threshold**) of neuron i. In the input layer $\mathcal{N}_1$, each neuron i gives directly one component of the network's input values $X_i \in \mathbb{R}$:

$$y_i = X_i \tag{4.3}$$

Layered feedforward ANNs with a sigmoid activation function can be regarded as a continuous parameterized function, called the **network function**

$$\mathsf{N}_{\boldsymbol{\pi}} : \mathcal{X} \subseteq \mathbb{R}^{n_1} \mapsto \mathcal{Y} \subseteq (0,1)^{n_\ell} \tag{4.4}$$

that maps from a set of inputs $\mathcal{X}$ onto a set of outputs $\mathcal{Y}$. The parameter vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{n_\pi})$ of the network function comprises the n_π internal parameters of the ANN (the connection weights w_{ij}, the biases θ_i, and the temperature parameters γ_i of the neurons).

Example: For the ANN in figure 4.2, $\mathsf{n}_\pi = 14$, $\pi_1 = \mathsf{w}_{41}$, $\pi_2 = \mathsf{w}_{42}$, $\pi_3 = \mathsf{w}_{43}$, $\pi_4 = \theta_4$, $\pi_5 = \gamma_4$, ..., $\pi_{14} = \gamma_6$.

4.2.3 Learning in Artificial Neural Networks

Using KOLMOGOROV's theorem, it can be proved that any continuous function A can be represented *exactly* by a finite network of computing units, though the **general learning problem** of determining the values for a given network's parameters is NP-complete [76]. In simple terms, this means that it is very improbable that an algorithm exists that is able to solve the problem in finite time, if the number of unknown variables gets large, though a guessed solution can be checked in finite time. However, in most practical cases no *exact* function representation is required but a finite approximation error is accepted for the network function. Using an adequate **learning algorithm**, an approximate solution for the problem

can be found in reasonable time. The **learning problem** consists of finding the optimal ANN parameter vector $\boldsymbol{\pi}$ so that the network function $\mathsf{N}_{\boldsymbol{\pi}}$ approximates the given function A as closely as possible. A is usually not given explicitly but only implicitly through some exemplary input-output pairs. However, using this **training set**, the **network error** (i.e. the difference between the actual output and the correct output) can be measured and utilized to learn the optimal network function $\mathsf{N}^\star := \mathsf{N}_{\boldsymbol{\pi}^\star}$ by adapting the network parameter vector $\boldsymbol{\pi}$ in a way that minimizes the network error. For this kind of learning problems, a variety of learning algorithms have been developed, the **backpropagation** algorithm – a gradient-based method – being the most widely known [76]. If the ANN has learned the function A, it may not only map the training set correctly but also input-output pairs that have not been a part of the training set (**generalization**).

Learning algorithms for ANNs that rely on a training set fail for delayed reinforcement problems, where the correct output for a given input is *not* known. A very obvious way to determine $\mathsf{N}^\star$ in such cases are evolutionary algorithms.

4.3 Evolutionary Algorithms

4.3.1 What are Evolutionary Algorithms?

The term **evolutionary algorithm** is an umbrella term for computer-based probabilistic search procedures that use computational models of genetics and natural evolution as key elements in their design and implementation. The best known algorithms in this class include **genetic algorithms** (GAs, see Holland [43] and Goldberg [40]), evolution strategies (see Rechenberg [74] and Schwefel [83]), evolutionary programming (see Fogel [35]), and genetic programming (see Koza [49]). Sometimes also the term **evolutionary computation** (EC) is used for the above mentioned algorithms. There are also many hybrid systems, which incorporate various features of the above mentioned algorithms, resulting in a confusing terminology in this field of research. However, all EAs share a common conceptual basis for simulating the evolution of individual structures via inheritance and natural selection, which depends on their perceived performance defined with respect to an environment (survival of the fittest).

EAs use a vocabulary borrowed from biology. The key element of an EA is a **population** Ξ^t that comprises numerous **individuals** $\boldsymbol{\xi}^t_{k\in\{1,\ldots,q\}}$ – also called **chromosomes** or **strings** – which are potential solutions to the given optimization problem. Here, the superscript t denotes the time step (or generation) within the simulated evolution. In each generation, the relatively good solutions reproduce, while the relatively poor solutions die. To give good individuals a better chance to create offspring, a performance measure for their suitability to solve the problem has to be defined. This is done using a **fitness function**[4] J. Without loss of generality, it can be assumed that Ξ^t is always sorted with respect to the fitness of the individuals, so that $\boldsymbol{\xi}^t_1$ is the best individual and $\boldsymbol{\xi}^t_q$ is the worst individual. A **selection scheme** (the simulated environment) selects individuals (**parents**) with a probability

[4] This function, also called objective function or evaluation function, is analogous to the cost function in optimal control theory. To emphasize this fact, it will be denoted by the same letter, J. However, one must have in mind that the cost function is subject to minimization, whereas the fitness function is subject to maximization.

according to their fitness value $J(\boldsymbol{\xi}_k^t)$ to reproduce and to create **offspring** into a newly created population Ξ^{t+1}. The selected parents undergo a series of **"genetic" transformations** (**recombination**, **mutation**), so that the offspring consists of a mixture of the parents "genetic material". Under the **selection pressure** of the environment, the individuals strive for survival. After some reproduction cycles, the population converges against a single solution, which is in the best case the globally optimal solution $\boldsymbol{\xi}^\star$ to the given problem. This parallel processing of potential solutions is in contrast to traditional optimization methods, which process only a single point of the search space. EAs have been applied with success for finding near-global optima in very high-dimensional multimodal search spaces, the type of which most real-world problems are.

4.3.2 How do Evolutionary Algorithms Work?

Within this section, a short introduction into the basic features of EAs is given, and the notations are set up, which are used later on. This is done for a simple two-dimensional parameter optimization problem, using the **Simple Genetic Algorithm** (SGA) according to Goldberg [40]. Although the SGA is not employed within this work, it can illustrate all important characteristics of EAs. Since the SGA allows a relative straightforward analytical treatment, its theoretical properties are well understood.

As an example, let the following function be subject to maximization using the SGA:

$$f(\mathbf{s}) = -\sum_{i=1}^{5} i \cdot \cos\left((i-1)s_1 + i\right) \cdot \sum_{j=1}^{5} j \cdot \cos\left((j+1)s_2 + j\right) \tag{4.5}$$

with $\mathbf{s} = (s_1, s_2) \in \mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 = [\underline{s}_1, \overline{s}_1] \times [\underline{s}_2, \overline{s}_2]$. $\mathcal{S}$ is called **domain**, **search space** or **solution space** of the optimization problem. The optimization objective is to find the $\mathbf{s}^\star$ that maximizes $f(\mathbf{s})$, i.e.

$$f(\mathbf{s}^\star) \geq f(\mathbf{s}) \quad \forall\, \mathbf{s} \in \mathcal{S}$$

Figure 4.4 sketches $f(\mathbf{s})$ for $\mathcal{S} = [-6, 4]^2$. It can be easily seen that $f(\mathbf{s})$ has many **local optima** in $\mathcal{S}$ (over 100), but only one of them is the **global optimum**.[5] Gradient-based optimization methods have problems in locating the global optimum of such **multimodal** functions, especially if the optimization function becomes very high-dimensional.

To solve this optimization problem with the SGA, the optimization parameters $s_{i\in\{1,\ldots,n=2\}}$ have to be coded as a **binary string**, $(s_1, \ldots, s_n) \leftrightarrow \langle b_1, b_2, \ldots, b_\ell \rangle$. In analogy to the biological terms, the positions on the string are termed **loci** (singular: locus) and the value of the string at a specific locus is called **allele** (plural: alleles). The length ℓ of the binary string depends on the required precision of the solution. If d_i decimal places are required for s_i, $\mathcal{S}_i$ must be cut into $2^{\ell_i} - 1$ ranges of equal size, where ℓ_i is the smallest integer such that

[5] $f^\star(s_1^\star \doteq -1.306708, s_2^\star \doteq -1.425128) \doteq 176.541793$

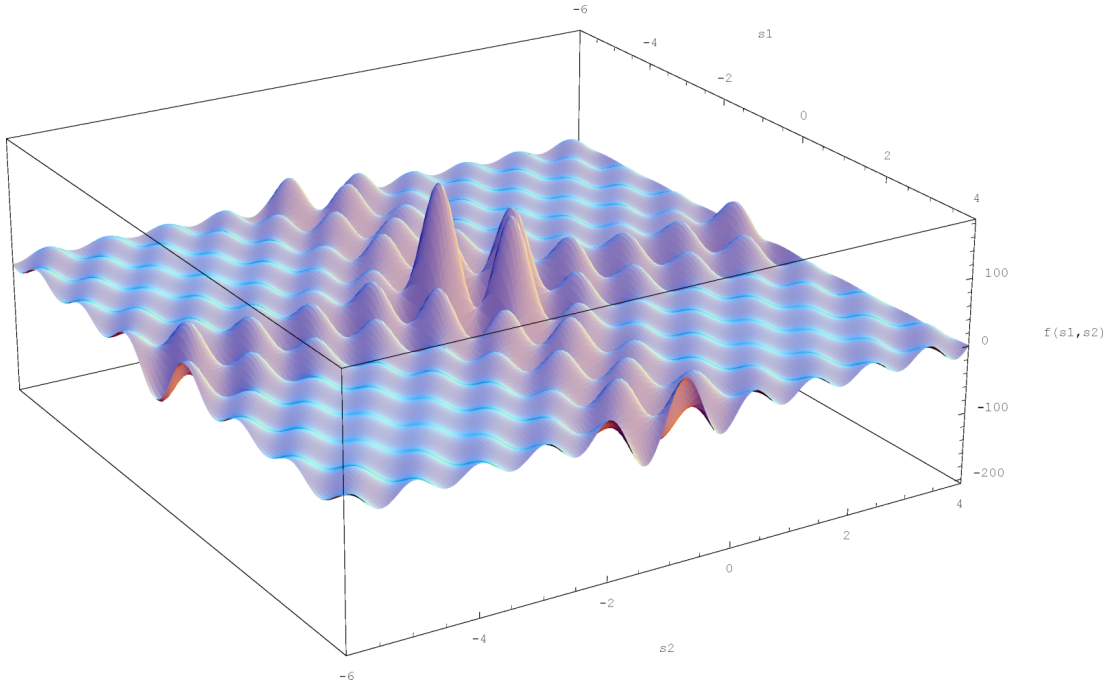


Figure 4.4 Multimodal function $f(\mathbf{s})$ for $\mathbf{s} \in [-6, 4]^2$

$(\overline{s}_i - \underline{s_i}) \cdot 10^{d_i} \leq 2^{\ell_i} - 1$; then $\ell = \sum_{i=1}^{n} \ell_i$.

> **Example:** If a precision of at least four places after the decimal point is required, $\ell_1 = \ell_2 = 17$ (since $2^{16} - 1 = 65535 < 10 \cdot 10^4 < 2^{17} - 1 = 131071$) and thus $\ell = 34$. The optimal chromosome is
>
> $$\boldsymbol{\xi}^\star = \langle 01111000001001011011101010001110 11 \rangle$$
>
> with $f(-1.306742, -1.425151) \doteq 176.541791$ that is due to the binary discretization slightly less than the optimal function value $f^\star = 176.541793$.

A string's corresponding function value might sometimes be taken as its fitness, i.e.

$$J(\boldsymbol{\xi}) = f(\mathbf{s}), \tag{4.6}$$

where the chromosome $\boldsymbol{\xi}$ represents the real vector $\mathbf{s}$. However, as it is shown later, a re-scaling of the fitness function is often necessary.

The initialization of the first population Ξ^0 is very simple. q binary strings are created with random alleles at all loci. After that, reproduction starts. The SGA employs a **reproduction scheme** that is called **generational reproduction**. Using generational reproduction, q individuals are selected from the actual population Ξ^t and copied into the new population Ξ^{t+1}, using a so-called **selection scheme** or **selection method**. The selection method must guarantee that fitter individuals have a better chance to be selected. Thereafter, the "genetic" operators (bit) **mutation** and (one-point) **crossover** are applied to generate new individuals from the old ones. After that, the new population is taken as the actual population and reproduction starts again. During reproduction, the SGA generates new chromosomes in

three subsequent steps:

Selection (step 1): Two parent individuals are selected from Ξ^t according to each individual's fitness $J(\boldsymbol{\xi}_k^t)$ with probability p_k^t. The SGA uses a selection scheme that is called **fitness proportional selection**, since every individual has a selection probability that is proportional to its fitness:[6]

$$p_k^t = \frac{J(\boldsymbol{\xi}_k^t)}{q \cdot J_{\text{avg}}^t} \quad \text{with} \quad J_{\text{avg}}^t = \frac{1}{q}\sum_{i=1}^{q} J(\boldsymbol{\xi}_i^t) \tag{4.7}$$

This selection scheme guarantees that good individuals are selected more often than poor ones. However, this selection method fails in the case of negative fitness values, so that proportional selection is typically used in combination with some **fitness scaling** technique $\mathsf{F}^+ : \mathbb{R} \mapsto \mathbb{R}^+$ that assures $J(\boldsymbol{\xi}) \geq 0 \; \forall \boldsymbol{\xi} \in \{0,1\}^\ell$.

Example: $\mathsf{J}(\boldsymbol{\xi}) = \mathsf{f}(\mathbf{s}) + 200$

$\boldsymbol{\xi}_1^{\mathsf{t}} = \langle 10101010000010111100101010010101 \rangle$
$\boldsymbol{\xi}_2^{\mathsf{t}} = \langle 11001010001111011010010101111110001 \rangle$

$\mathsf{J}(\boldsymbol{\xi}_1^{\mathsf{t}}) = \mathsf{f}(0.642430, -0.168275) + 200 \doteq 222.675911$ and
$\mathsf{J}(\boldsymbol{\xi}_2^{\mathsf{t}}) = \mathsf{f}(1.900069, -3.071435) + 200 \doteq 190.588078$.

Crossover (step 2): After selection, the crossover operator is applied with probability p_c. This operator combines the features of two parent chromosomes into two offspring chromosomes by swapping corresponding segments of the chromosomes. The intuition behind crossover is to exchange information between different potential solutions [61].

Example: Assuming that the crossover point was (randomly) selected after the 10^{th} locus:

$\boldsymbol{\xi}'^{\mathsf{t}}_1 = \langle 1010101000|11110110100101011110001 \rangle$
$\boldsymbol{\xi}'^{\mathsf{t}}_2 = \langle 1100101000|00101111001010101010010101 \rangle$

$\mathsf{J}(\boldsymbol{\xi}'^{\mathsf{t}}_1) = \mathsf{f}(0.650060, -3.071435) + 200 \doteq 191.362376$ and
$\mathsf{J}(\boldsymbol{\xi}'^{\mathsf{t}}_2) = \mathsf{f}(1.892440, -0.168275) + 200 \doteq 223.882069$.
Note that the second offspring has a better fitness than both of its parents.

Mutation (step 3): After crossover, the mutation operator is applied to both offspring. The mutation operator alters one or more alleles of the chromosome by flipping every bit with probability $p_m \ll 1$. The intuition behind this operator is to introduce new alleles and to re-introduce extinct alleles into the population, if all individuals throughout the population

[6] Since this selection method can be imagined as the turn of a biased roulette wheel, where each individual in the population has a slot sized proportional to its fitness, it is also called **roulette wheel selection**.

have the same allele at some specific locus.

Example: Assuming that the 25^{th} locus of the first chromosome was selected for mutation:

$$\boldsymbol{\xi}_1^{t+1} = \langle 10101010001111011010010\underline{1}1111110001\rangle$$
$$\boldsymbol{\xi}_2^{t+1} = \langle 11001010000010111100101010010010101\rangle$$

$J(\boldsymbol{\xi}_1^{t+1}) = f(0.650060, -3.032372) + 200 \doteq 190.715404$ and $J(\boldsymbol{\xi}_2^{t+1}) = J(\boldsymbol{\xi'}_2^{t})$. Thus, this particular mutation resulted in a fitness decrease. However, if the 24^{th} locus was selected for mutation:

$$\boldsymbol{\xi}_1^{t+1} = \langle 10101010001111011010010\underline{0}0111110001\rangle$$

so that $J(\boldsymbol{\xi}_1^{t+1}) = f(0.650060, -3.149560) + 200 \doteq 193.975464$, which is an improvement over the original value $J(\boldsymbol{\xi'}_1^{t}) \doteq 191.362376$.

4.3.3 Why do Evolutionary Algorithms Work?

Within this section, the theoretical foundations of the SGA are sketched, as far as they are vital to appraise the performance characteristics of EAs in general and to understand the problems that might arise during the application of EAs. A more comprehensive introduction into EA theory can be found in [40], [43], and [61].

Although recombination redistributes existing information and mutation introduces new information into the search process, they are completely undirected and correspond to a random walk in search space. This might lead to the impression that EAs work purely randomly and can therefore not be superior to other random search procedures. This is de facto not the case, since the undirected search of recombination and mutation becomes directed when a selection mechanism is added that prefers better individuals and thus provides an active driving force for improvement [9]. As they combine elements of directed and stochastic search, EAs are very different from random search algorithms, although they belong to the class of probabilistic search methods [61]. Another pivotal point is that the evolutionary operators do actually not work on full strings but on **substrings**, which are subspaces of the search space. And, as HOLLAND [43] shows, the number of substrings that is successfully processed in parallel within a single generation is not proportional to q but to q^3, the *cube* of the population size. Future populations are biased to explore regularities in the environment by exploiting above average substrings. To understand how this is done, a measure for the similarity between different chromosomes has to be defined. This is done by introducing the notion of a **schema**, a template that allows the exploration of similarities between chromosomes by introducing a **don't care symbol** "$\star$" into the genetic alphabet. A schema S represents all strings, which match it on all string positions other than "$\star$".

Example: To keep things simple, a string of length $\ell = 10$ is assumed. Then, the schema $S = (\star 1 \star 1100101)$ matches the four strings $\{\langle 0101100101\rangle, \langle 0111100101\rangle, \langle 1101100101\rangle, \langle 1111100101\rangle\}$

The number of fixed (i.e. non-"$\star$"-positions) of a schema is called **order** of the schema and

denoted as $o(S)$.

Example: $o(\star 1 \star 1100101) = 8$, $o(\star 1 \star\star\star\star\star\star\star) = 1$

Consequently, a schema has $r = \ell - o(S)$ don't care symbols. Thus it is an r-dimensional subspace of the ℓ-dimensional search space, as it is visualized in figure 4.5 for $\ell = 3$. It is

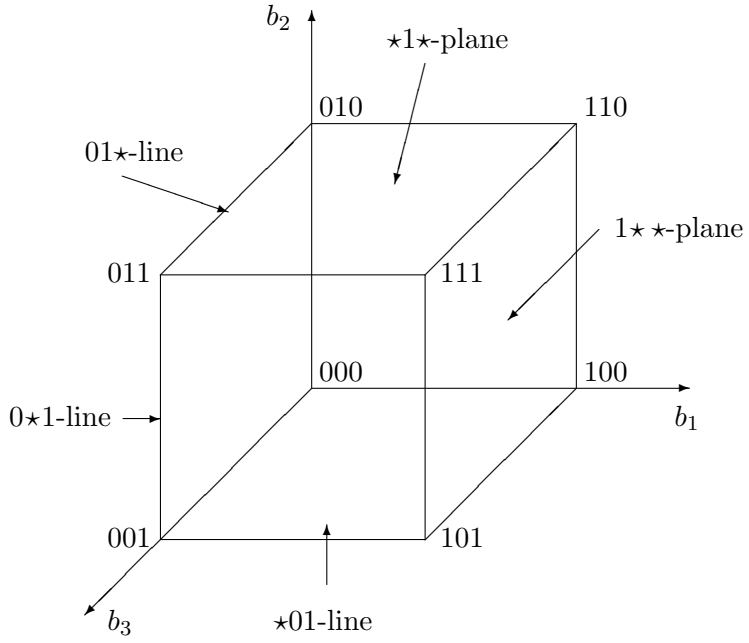


Figure 4.5 Visualization of schemata as subspaces in three-dimensional space

clear that every schema matches exactly 2^r strings and that each string is matched by 2^ℓ schemata. Between 2^ℓ and $q \cdot 2^\ell$ schemata of the 3^ℓ possible schemata may be represented in the population, depending on its "genetic diversity" [61]. The order of a schema defines its speciality. It is useful for calculating its survival probability for mutation, which is

$$p_{sm}(S) = (1 - p_m)^{o(S)} \approx 1 - p_m \cdot o(S) \quad \text{for} \quad p_m \ll 1 \tag{4.8}$$

Another important notion is the **defining length** $\delta(S)$ of a schema, which is the distance between the first and the last fixed string position.

Example: $\delta(\star 1 \star 1100101) = 8$, $\delta(\star 1 \star\star\star\star\star\star\star) = 0$

It defines the compactness of the information contained in the schema. The defining length of a schema is useful for calculating its survival probability for crossover, which is[7]

$$p_{sc}(S) \geq 1 - p_c \cdot \frac{\delta(S)}{\ell - 1} \tag{4.9}$$

[7] The $\geq$-sign is due to the chance for the schema to survive crossover even if the crossover site is selected between fixed positions.

Its fitness at time t, $J^t(S)$, is another important attribute of a schema. It is defined as the average fitness of all strings in the population matched by the schema S. Since under fitness proportional selection each individual $\boldsymbol{\xi}_k^t$ is selected with probability $p_k^t = J(\boldsymbol{\xi}_k^t)/qJ_{\text{avg}}^t$, the expected number of strings matching the schema S in the next generation Ξ^{t+1} is

$$N^{t+1}(S) = N^t(S) \cdot \frac{J^t(S)}{J_{\text{avg}}^t} \tag{4.10}$$

Thus – in the absence of crossover and mutation – "above average" schemata receive an increasing number of strings in the following generations, whereas "below average" schemata receive a decreasing number of strings. If schema S remains above average by $\epsilon\%$, then

$$N^{t+1}(S) = N^0(S) \cdot (1+\epsilon)^{t+1} \tag{4.11}$$

and the schema receives an exponentially increasing number of strings in subsequent generations. However, taking additionally the disruptive effects of crossover and mutation into account, the combined effect is

$$N^{t+1}(S) \geq N^t(S) \cdot \frac{J^t(S)}{J_{\text{avg}}^t} \left[1 - p_c \cdot \frac{\delta(S)}{\ell - 1} - p_m \cdot o(S)\right] \tag{4.12}$$

This equation is called **schema growth equation**. It gives the expected number of strings matching a schema S in the next generation as a function of the actual number of strings matching the schema, the relative fitness of the schema, and its defining length and order. The implications of the schema growth equation are typically stated as the **schema theorem** and the **building block hypothesis**, which is an immediate result of this theorem [61]:

Schema Theorem: Short, low-order, above average schemata receive exponentially increasing trials in subsequent generations of a genetic algorithm.

Building Block Hypothesis: A GA seeks near-optimal performance through the juxtaposition of short, low-order, high-performance schemata, called **building blocks**.

This hypothesis suggests that the problem of coding is critical for the performance of the SGA, since the coding should satisfy the idea of short building blocks. Holland showed in [43] that at least q^3 schemata are successfully passed from generation to generation, a feature for which he coined the term **implicit parallelism**. This implicit parallelism is the fundament for the power of evolutionary algorithms.

4.3.4 Advantages and Disadvantages of Evolutionary Algorithms

Within this section, the advantages and disadvantages of EAs with respect to gradient-based optimization methods are appraised, since they are – providing the largest differences to EAs – representative for traditional optimization methods. Other optimization methods like simulated annealing or dynamic programming share the advantages of EAs in some respect.

The most important advantages of EAs over gradient-based optimization methods are their good **global search behavior**, their **blindness**, their **problem independence**, and their

robustness. In contrast to gradient methods, EAs explore the search space using a *population* of search points. This results in a good global search behavior within multimodal environments, where the final result is relatively independent of the randomly chosen initial population. To improve the actual search point, gradient methods require auxiliary information like derivatives. EAs do not need such auxiliary information, they are blind. This makes EAs especially suited for optimization problems, for which auxiliary information is difficult to obtain or not available like for delayed reinforcement learning problems. EAs are also problem independent, since every problem, whose problem parameters can be coded on a string, can be optimized using an EA. Gradient techniques are often very problem specific. The backpropagation algorithm (see section 4.2.3) provides a good example for this proposition. Backpropagation requires a very specific ANN topology and a differentiable transfer function within the neurons. For all more general ANNs, backpropagation can not be used. Since EAs can be applied for a very wide range (all kinds?) of optimization problems, they are robust optimization methods.

However, if simple evolutionary algorithms like the SGA would be the best optimization method for all kinds of optimization problems, EA and optimization research would have ceased quickly. That this is not the case indicates that the application of EAs still yields some problems, so that up to now, they can not be considered as all-in-one every-purpose hands-off optimization tools. The major problems of (especially binary coded) EAs are that – for some problems – they fail to locate the global optimum with the required precision, and that the inclusion of nontrivial constraints and auxiliary information about the problem (and the solution) is not trivial. Consequently, the major part of present EA research is focused on the following questions:

(1) How can EAs be prohibited from premature convergence against a local optimum?

(2) How can EAs be a applied to problems that require a high precision of the solution?

(3) How can available auxiliary information be introduced into EAs?

(4) How can nontrivial constraints be introduced into EAs?

The following chapter 5 covers items (1) to (3) within the context of this work, and presents the methods that have been chosen to avoid the associated problems. For a discussion of item (4), the reader is referred to the standard EA literature, since this issue is not relevant within this work.

4.4 Evolutionary Neurocontrol

Operating within so-called **neurocontrollers**, ANNs have been successfully applied to RL problems [34]. Neurocontrol approaches to solve RL control problems can be divided into two categories, *indirect* ones and *direct* ones [57, 95]. The **direct neurocontrol** approach, which is used here, employs a single ANN, which is called **action model**. The action model controls the dynamical system by providing a control vector $\mathbf{u} \in \mathcal{U}$ from some input vector $\mathbf{X} \in \mathcal{X}$ that contains the information that is relevant to perform the control task (system state, environmental state, etc.). The more commonly used **indirect neurocontrol** approach,

which is dispensable within this work[8], employs additionally a **system model** and a second ANN, which is called **evaluation model**. Based on the system model, the evaluation model provides a prediction of the evaluation of the action that is considered by the action model [12, 34, 57]. Henceforth, to keep the terminology simple, the term "neurocontroller" is used for the ANN that is precisely speaking "the action model of a direct neurocontroller".

The application of an EA to search for the NC's optimal network function $\mathsf{N}^\star$ makes use of the fact that an NC parameter vector $\boldsymbol{\pi}$ can be mapped onto a string, which provides an equivalent description of the NC's network function N (figure 4.6). By searching for the fittest string $\boldsymbol{\xi}^\star$, the EA searches for the NC's optimal network function $\mathsf{N}^\star$. Such NCs might be called **evolutionary neurocontrollers**.

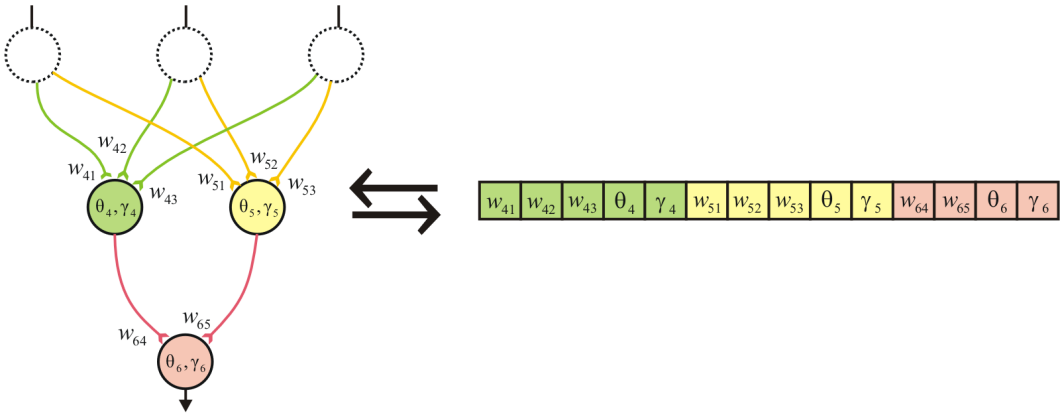


Figure 4.6 Mapping of an ANN onto a (real-valued) string

4.5 Spacecraft Steering Using Evolutionary Neurocontrol

For the implementation of spacecraft steering strategies, as defined in section 4.1.2, an NC may be used. In this case, each NC parameter vector $\boldsymbol{\pi}$ defines a steering strategy $\mathsf{S}_{\boldsymbol{\pi}} : \mathcal{X} \mapsto \mathcal{U}$. The EA is used to determine the optimal NC parameter vector $\boldsymbol{\pi}^\star$ that yields, after some transformations (see figure 4.1, the optimal spacecraft trajectory $\mathbf{x}^\star_{\mathrm{SC}}[t]$.

Now, the two following sections have to address the two fundamental questions concerning the utilization of an NC for spacecraft steering:

1. "What input should the NC get?" (or "What should the NC know to steer the spacecraft?", section 4.5.1) and
2. "What output should the NC give?" (or "What should the NC do to steer the spacecraft?", section 4.5.2)

[8] the indirect approach has been developed to make the optimization of the action model amenable to gradient-based methods like backpropagation

4.5.1 Neurocontroller Input

To be robust, a steering strategy should not depend *explicitly* on time. Everything that the strategy should "have to know" to determine the actual optimal spacecraft control vector $\mathbf{u}(t)$ is – at *any* time – the actual spacecraft state $\mathbf{x}_{\mathrm{SC}}(t)$ and target body state $\mathbf{x}_{\mathrm{T}}(t)$, hence

$$\mathsf{S} : \{(\mathbf{x}_{\mathrm{SC}}, \mathbf{x}_{\mathrm{T}})\} \mapsto \mathcal{U} \tag{4.13}$$

If the spacecraft employs a propulsion system other than a solar sail, the actual propellant mass $m_{\mathrm{P}}(t)$ might be considered as additional input:

$$\mathsf{S} : \{(\mathbf{x}_{\mathrm{SC}}, \mathbf{x}_{\mathrm{T}}, m_{\mathrm{P}})\} \mapsto \mathcal{U} \tag{4.14}$$

However, the number of potential NC input sets is still large: $\mathbf{x}_{\mathrm{SC}}$ and $\mathbf{x}_{\mathrm{T}}$ may be given in coordinates of any reference frame and in combinations of them. Also the difference $\mathbf{x}_{\mathrm{T}} - \mathbf{x}_{\mathrm{SC}}$ may be used, again in coordinates of any reference frame and in combinations of them. If only cartesian coordinates, polar coordinates, orbital elements, and their combinations and differences are considered, there are 14 possible input sets[9]. Some possible input sets are depicted in figures 4.7 to 4.10. The optimal input set is expected to be problem-dependent.

> **Example:** If an orbit transfer from Earth to the orbit of Mercury has to be performed, it is not reasonable to give the spacecraft and the target body state in orbital elements, $\mathbf{E}_{\mathrm{SC}}$ and $\mathbf{E}_{\mathrm{T}}$, since the orbital elements of the target body do not change (except the anomaly, which is irrelevant for an orbit transfer). Thus, better results are expected, if $\mathbf{E}_{\mathrm{T}} - \mathbf{E}_{\mathrm{SC}}$ is used as input, since the NC network and thus the dimension of the solution space is smaller in this case.

4.5.2 Neurocontroller Output

4.5.2.1 Indirect Steering Strategies

One way to obtain spacecraft steering strategies is to use blended LSLs (see section 3.2.3), so that for solar sailcraft

$$\mathsf{S} : \mathcal{X} \mapsto \{\mathbf{c}\} \tag{4.15}$$

and for other spacecraft

$$\mathsf{S} : \mathcal{X} \mapsto \{\mathbf{c}, \chi\} \tag{4.16}$$

Since such steering strategies have implicit knowledge about how to change the orbital elements in an optimal way, they can be considered as **indirect steering strategies**.

Figure 4.7 shows an example for an indirect solar sailcraft steering strategy. The NC receives the sailcraft state $\mathbf{x}_{\mathrm{SC}}$ and the target body state $\mathbf{x}_{\mathrm{T}}$ in polar coordinates and gives the components of the five-dimensional steering law weight vector $\mathbf{c}_Q$ (for matching the orbital elements with a maximum rate). The direction unit vector $\mathbf{d}$ can be calculated from $\mathbf{c}_Q$

[9] if mixed combinations like $\mathbf{x}_{\mathrm{SC}}$ in cartesian coordinates and $\mathbf{x}_{\mathrm{T}}$ in polar coordinates are omitted from the set of potential inputs

using equation (3.7). Figure 4.8 shows another example for an indirect solar sailcraft steering strategy. The NC receives $\mathbf{x}_{\mathrm{SC}}$ and $\mathbf{x}_{\mathrm{T}}$ in cartesian coordinates and gives the components of the ten-dimensional steering law weight vector $\mathbf{c}_L$ (for increasing / decreasing the orbital elements with a maximum rate). Again, the direction unit vector $\mathbf{d}$ can be calculated from $\mathbf{c}_L$ using equation (3.7).

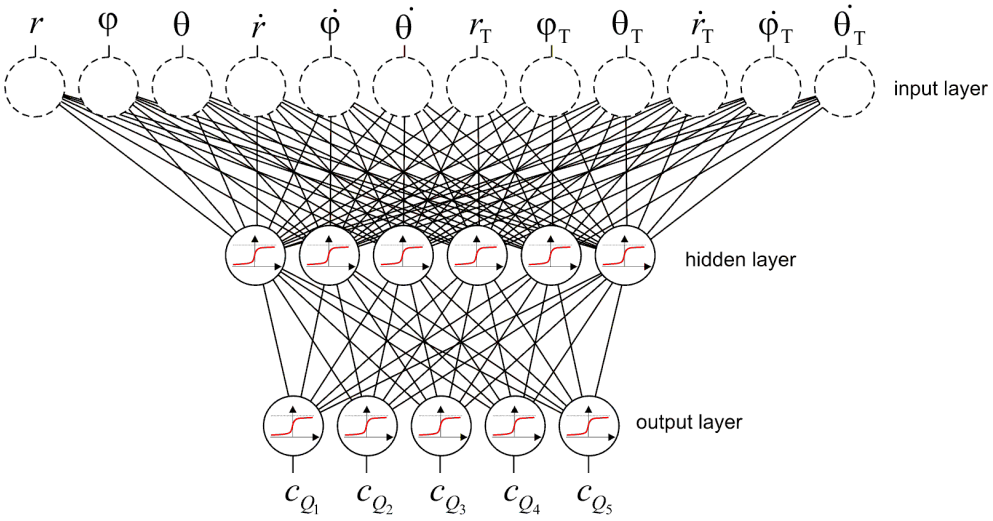


Figure 4.7 NC for indirect steering of solar sailcraft (matching the orbital elements)

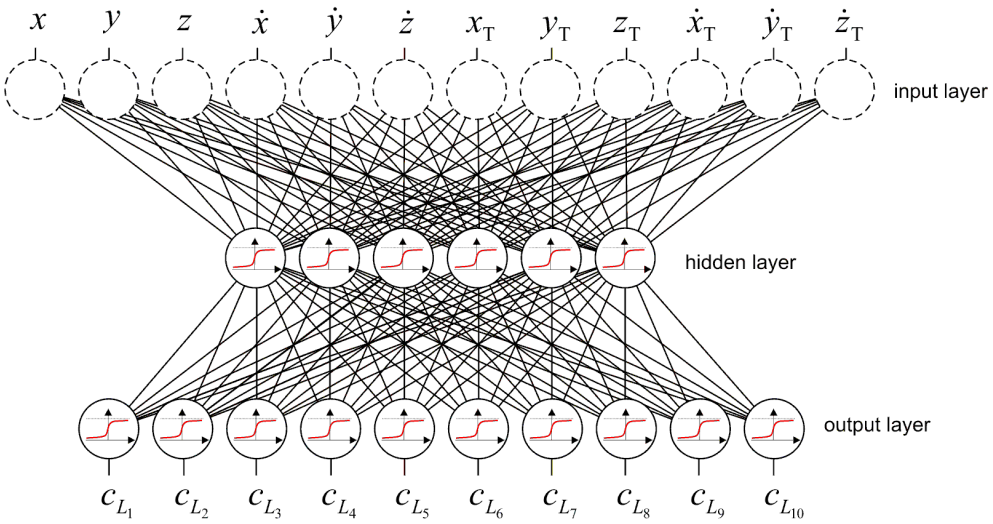


Figure 4.8 NC for indirect steering of solar sailcraft (for increasing/decreasing the orbital elements)

4.5.2.2 Direct Steering Strategies

Using an NC, the implementation of steering strategies is also possible without the use of LSLs, e.g. if the NC provides a three-dimensional output vector $\mathbf{d}'' \in (0,1)^3$, from which the direction unit vector $\mathbf{d}$ can be directly calculated via[10]

$$\mathbf{d}' = 2 \cdot \begin{pmatrix} d_1'' \\ d_2'' \\ d_3'' \end{pmatrix}_{\mathcal{O}} - \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}_{\mathcal{O}} \tag{4.17a}$$

$$\mathbf{d} = \mathbf{d}'/|\mathbf{d}'| \tag{4.17b}$$

so that for solar sailcraft

$$\mathsf{S} : \mathcal{X} \mapsto \{\mathbf{d}\} \tag{4.18}$$

and for EP spacecraft

$$\mathsf{S} : \mathcal{X} \mapsto \{\mathbf{d}, \chi\} \tag{4.19}$$

Since such steering strategies do *not* have implicit knowledge about locally optimal spacecraft steering, they can be considered as **direct steering strategies**.

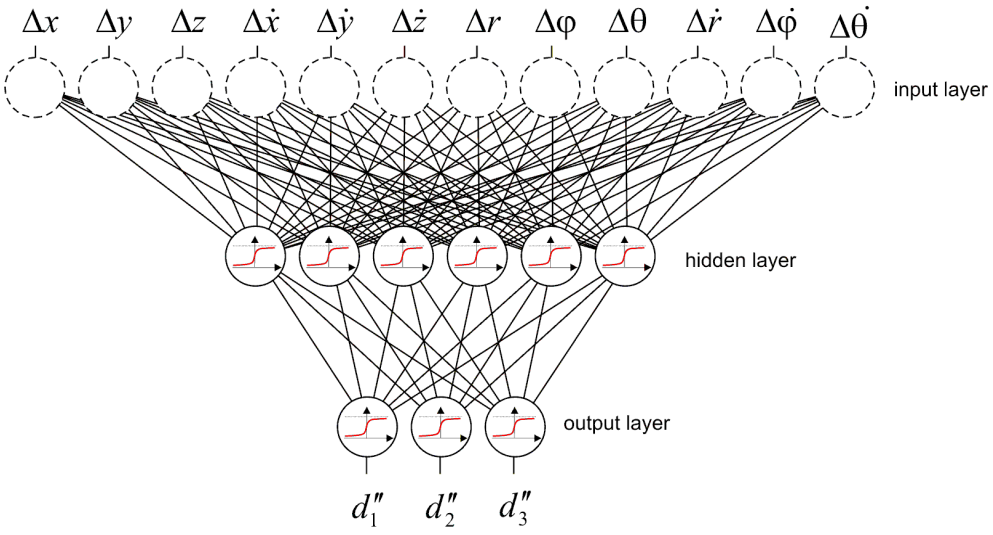


Figure 4.9 NC for direct steering of solar sailcraft

Figure 4.9 shows an example for a direct solar sailcraft steering strategy. The NC receives $\mathbf{x}_{\mathrm{T}} - \mathbf{x}_{\mathrm{SC}}$ in cartesian and polar coordinates and gives a three-dimensional output vector $\mathbf{d}''$,

[10] Of course $\mathbf{d}$ could also be calculated from a two-dimensional output vector, if the two output values are interpreted as two direction angles. However, this is not recommended, since in this case two similar angular values like $\alpha_1 = \pi - \epsilon$ and $\alpha_2 = -(\pi - \epsilon)$ ($\Delta\alpha_{12} = 2\epsilon \approx 0$) can be far apart, when mapped onto a linear scale ($\Delta\alpha_{12} = 2(\pi - \epsilon) \approx 2\pi$). If the optimal value of α oscillates around $\pm\pi$, the corresponding output neuron would have to give alternatingly completely different outputs for subsequent time steps (≈ 0 and ≈ 1), which is impossible for smooth NC inputs.

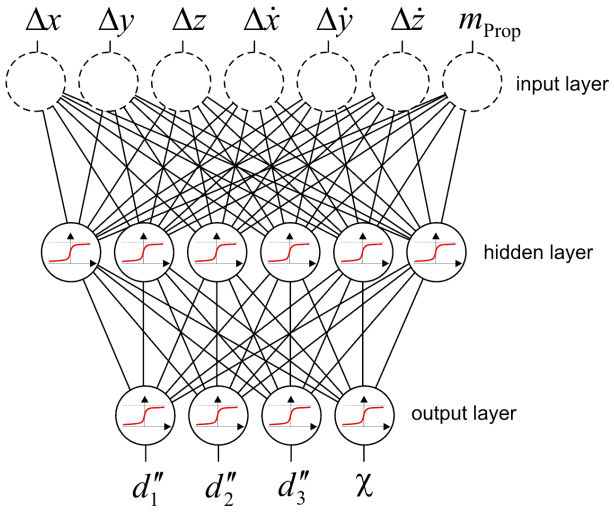


Figure 4.10 NC for direct steering of EP spacecraft

from which the direction unit vector $\mathbf{d}$ can be calculated using equations (4.17). Figure 4.9 shows an example for a direct EP spacecraft steering strategy. The NC receives $\mathbf{x}_{\text{T}} - \mathbf{x}_{\text{SC}}$ in cartesian coordinates and the actual propellant mass m_{P} and gives the three components of $\mathbf{d}''$ and the throttle χ .

Figure 4.11 shows a comparison of direct and indirect steering strategies.

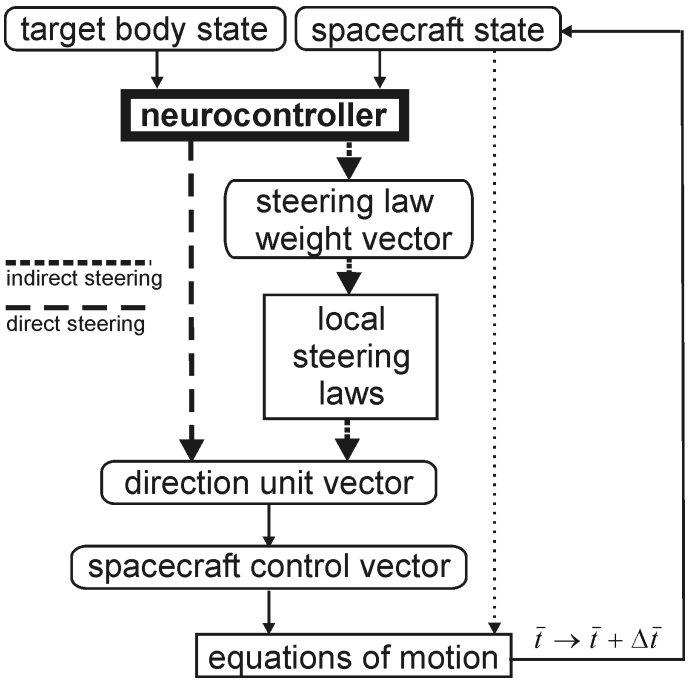


Figure 4.11 Comparison of direct and indirect steering strategies

4.5.3 Neurocontroller Fitness Assignment

As it is discussed in section 3.1.1, the optimality of a trajectory might be defined with respect to various (primary) objectives (e.g. transfer time or propellant consumption). When using an ENC for trajectory optimization, the accuracy of the trajectory with respect to the terminal constraints must also be stated as secondary optimization objectives, since the terminal constraints are not explicitly stated otherwise and need not to be satisfied throughout the search process.

If for example the transfer time for a rendezvous is to be minimized, the fitness function must include the transfer time $T = \bar{t}_f - \bar{t}_0$ as well as the final distance to the target body $\Delta r_f = |\mathbf{r}_{\mathrm{T}}(\bar{t}_f) - \mathbf{r}_{\mathrm{SC}}(\bar{t}_f)|$ and the final velocity relative to the target body $\Delta v_f = |\dot{\mathbf{r}}_{\mathrm{T}}(\bar{t}_f) - \dot{\mathbf{r}}_{\mathrm{SC}}(\bar{t}_f)|$, thus $J(T, \Delta r_f, \Delta v_f)$. Accordingly, if the propellant mass for a fly-by problem is to be minimized, Δv_f is not relevant but the consumed propellant $\Delta m_{\mathrm{P}} = m_{\mathrm{P}}(\bar{t}_f) - m_{\mathrm{P}}(\bar{t}_0)$ must be part of the fitness function, thus $J(\Delta m_{\mathrm{P}}, \Delta r_f)$.

Since the ENC can not be expected to satisfy the terminal constraints exactly ($\Delta r_f = 0\,\mathrm{m}$, $\Delta v_f = 0\,\mathrm{m/s}$), a maximum allowed distance $\Delta r_{f,\max}$ and a maximum allowed relative velocity $\Delta v_{f,\max}$ of the spacecraft at the target body have to be defined. Although those values demand careful consideration, one needs not to be an expert in astrodynamics to find an appropriate setting. For a planetary rendezvous, for example, $\Delta r_{f,\max}$ should lie somewhere between the planetary radius and the planetary gravitational sphere of influence. Ideally, $\Delta v_{f,\max}$ should be set as to minimize the combined transfer and capture time to the planetary target orbit (or as to minimize the total propellant mass for the transfer and the capture phase). This is not possible within the one-body simulation model used within this work, but would necessitates a combined optimization of both phases. For practical purposes, $\Delta v_{f,\max}$ might be set within the range $0.1 - 0.5\,\mathrm{km/s}$. However, if an atmospheric entry, aerocapture or aerobraking is envisaged at the target, $\Delta v_{f,\max}$ could be larger. Using $\Delta r_{f,\max}$ and $\Delta v_{f,\max}$, the final distance and relative velocity at the target body can be normalized:

$$\Delta R = \frac{\Delta r}{\Delta r_{f,\max}} \qquad \Delta R_f = \frac{\Delta r_f}{\Delta r_{f,\max}} \tag{4.20}$$

$$\Delta V = \frac{\Delta v}{\Delta v_{f,\max}} \qquad \Delta V_f = \frac{\Delta v_f}{\Delta v_{f,\max}} \tag{4.21}$$

Furthermore, it will become necessary to define a measure for the accuracy of the trajectory, i.e. for the fulfillment of the terminal constraint, e.g.

$$\Delta X = \sqrt{\frac{1}{2}\left(\Delta R^2 + \Delta V^2\right)} \qquad \Delta X_f = \sqrt{\frac{1}{2}\left(\Delta R_f^2 + \Delta V_f^2\right)} \tag{4.22}$$

Since most individuals never achieve the required accuracy during the search process, a maximum transfer time $T_{\max}$ must be given for the numerical integration of the trajectory. It is important to note that t_f is defined with respect to the best accuracy. Thus, Δr_f and Δv_f are *not* the distance and relative velocity at the end of integration, but at the time when ΔX is minimal.

Now, **sub-fitness functions** may be defined with respect to the primary and the different secondary optimization objectives. During many preliminary tests, it has been found that the performance of the ENC strongly depends on an adequate choice of the sub-fitness functions

and on their composition to an (overall) fitness function. This is reasonable, since the fitness function has not only to decide autonomously which trajectories are good and which are not, but also which trajectories are promising for future "cultivation" and which are not. The primary sub-fitness functions

$$J_T = 1000 \cdot \left(1 - \frac{T}{T_{\max}}\right) \tag{4.23}$$

$$J_{m_\mathrm{P}} = \frac{m_\mathrm{P}(t_0)}{2m_\mathrm{P}(t_0) - m_\mathrm{P}(t_f)} - \frac{1}{3} \tag{4.24}$$

and the secondary sub-fitness functions

$$J_r = \log\left(\frac{1}{\Delta R_f}\right) \tag{4.25}$$

$$J_v = \log\left(\frac{1}{\Delta V_f}\right) \tag{4.26}$$

have empirically found to produce acceptable results. They have been used for all trajectory optimization runs within this work. J_T is constrained to be in the interval $0 \leq J_T \leq 1000$ and J_{m_P} is constrained to $1/6 \leq J_{m_\mathrm{P}} \leq 2/3$. J_r and J_v take positive values if the respective accuracy requirement is fulfilled and negative values, if it is not.

Another empirical finding is that the search process should first concentrate on the accuracy of the trajectories and then on the primary optimization objective.[11] Therefore, the sub-fitnesses for the primary optimization objective are modified to

$$J'_T = \begin{cases} 0 & \text{if } J_r < 0 \vee J_v < 0 \\ J_T & \text{if } J_r \geq 0 \wedge J_v \geq 0 \end{cases} \tag{4.27}$$

and

$$J'_{m_\mathrm{P}} = \begin{cases} 0 & \text{if } J_r < 0 \vee J_v < 0 \\ J_{m_\mathrm{P}} & \text{if } J_r \geq 0 \wedge J_v \geq 0 \end{cases} \tag{4.28}$$

To guide the search process, sub-fitness functions $J_\square$ for other trajectory parameters $\square$ (like the eccentricity e or the orientation of the orbital plane $\mathbf{e}_h$) might be introduced in the same way. They might be used as long as $J_r < 0 \vee J_v < 0$ and then be discarded if $J_r \geq 0 \wedge J_v \geq 0$. However, this guidance is at the expense of the simplicity of the fitness function.

Transfer Time Minimization for Rendezvous: Two fitness functions might be conceived:

$$J_1(T, \Delta r_f, \Delta v_f) = J'_T + \frac{1}{\sqrt{\Delta R_f^2 + \Delta V_f^2}} \tag{4.29}$$

$$J_2(T, \Delta r_f, \Delta v_f) = J'_T + \frac{1}{\sqrt{2 \cdot \max(\Delta R_f, \Delta V_f)^2}} \tag{4.30}$$

[11] However, the primary objective can be used as a selection criterion for the selection method that has been applied within this work (tournament selection, see section 5.2)

Using J_1, a poor final distance can be compensated with a good relative velocity and vice versa. This is not possible for J_2. Empirical results indicate that J_1 is superior to J_2 for most problems. However, for some problems J_1 might run into a local optimum, where it yields very good final distances but fails to match the final velocity.

Propellant Mass Minimization for Rendezvous: The fitness functions for this trajectory optimization problem are similar to those in the last paragraph, but J'_T is replaced with J'_{m_P}.

$$J_1(\Delta m_\mathrm{P}, \Delta r_f, \Delta v_f) = J'_{m_\mathrm{P}} + \frac{1}{\sqrt{\Delta R_f^2 + \Delta V_f^2}} \tag{4.31}$$

$$J_2(\Delta m_\mathrm{P}, \Delta r_f, \Delta v_f) = J'_{m_\mathrm{P}} + \frac{1}{\sqrt{2 \cdot \max(\Delta R_f, \Delta V_f)^2}} \tag{4.32}$$

Transfer Time Minimization for Fly-By: For a fly-by at the target body, only the positions must match and therefore

$$J(T, \Delta r_f) = J'_T + \frac{1}{\Delta R_f} \tag{4.33}$$

Propellant Mass Minimization for Fly-By: Here, J'_T is again replaced with J'_{m_P}, and thus

$$J(\Delta m_\mathrm{P}, \Delta r_f) = J'_{m_\mathrm{P}} + \frac{1}{\Delta R_f} \tag{4.34}$$

4.5.4 Evolutionary Neurocontroller Design

This section summarizes how an ENC may be applied for low-thrust trajectory optimization. To find the optimal spacecraft trajectory, the ENC method is running in two loops (figure 4.12). Within the (inner) trajectory integration loop, an NC steers the spacecraft according to its network function $\mathsf{N}_{\boldsymbol{\pi}}$, that is completely defined by the NC's parameter vector $\boldsymbol{\pi}$, which is set and evaluated by the EA in the (outer) NC optimization loop. The EA holds a population of NC parameter vectors, $\Xi = \{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_q\}$. Within the trajectory integration loop the EA evaluates all NC parameter vectors $\boldsymbol{\pi}_{j \in \{1,\ldots,q\}}$ for their suitability to generate an optimal trajectory. Within the trajectory optimization loop, the NC takes the actual spacecraft state $\mathbf{x}_\mathrm{SC}(\bar{t}_{i \in \{0,\ldots,\tau\}})$ and that of the target body $\mathbf{x}_\mathrm{T}(\bar{t}_i)$ as input values, and maps from them onto some output values. For EP spacecraft, the input values include the actual propellant mass $m_\mathrm{P}(\bar{t}_i)$ and the output includes the throttle $\chi(\bar{t}_i)$. If the NC implements an indirect steering strategy, its remaining output values are interpreted as the required steering law weight vector $\mathbf{c}(\bar{t}_i)$. Using this steering law weight vector, the direction unit vector $\mathbf{d}(\bar{t}_i)$ can be calculated from LAGRANGE's planetary equations. If the NC implements a direct steering strategy, its first three output values are interpreted as the components of the required direction vector $\mathbf{d}'(\bar{t}_i)$. So the required direction unit vector $\mathbf{d}(\bar{t}_i)$ (or $\tilde{\gamma}(\bar{t}_i)$ and $\tilde{\delta}(\bar{t}_i)$) can be calculated in both cases. From this, the spacecraft control vector $\mathbf{u}(\bar{t}_i)$ can be calculated. Then, $\mathbf{x}_\mathrm{SC}(\bar{t}_i)$ and $\mathbf{u}(\bar{t}_i)$ are inserted into the equations of motion, which are (numerically)

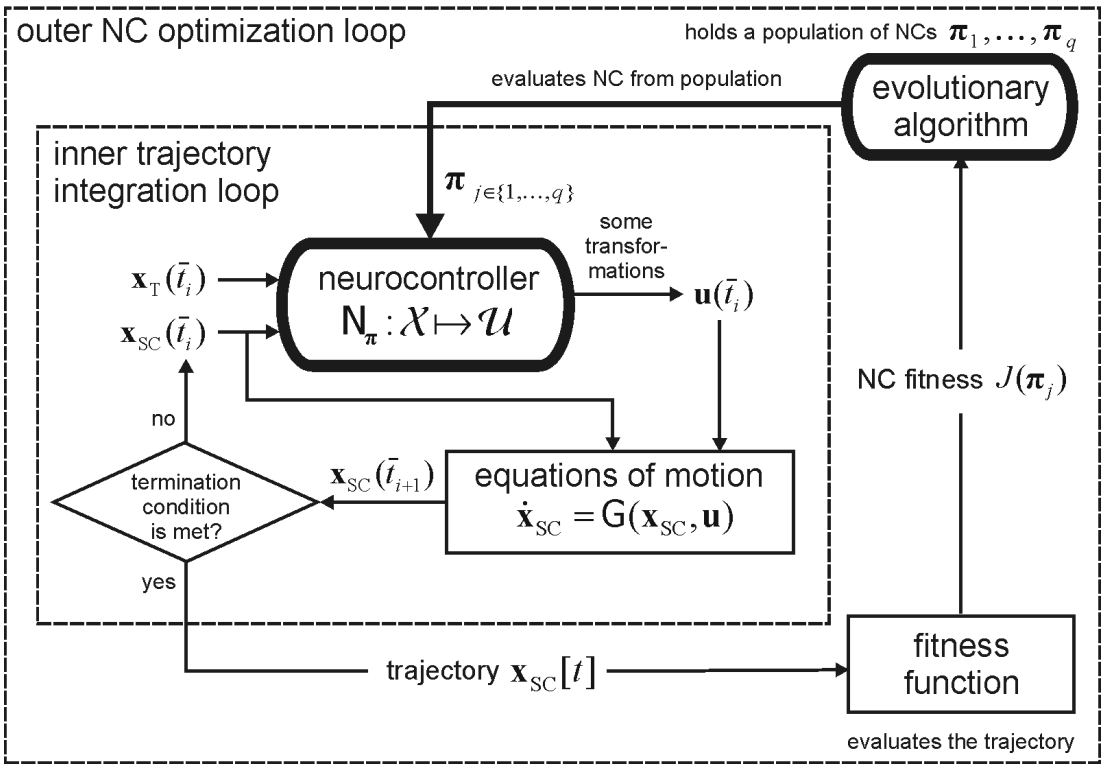


Figure 4.12 Trajectory optimization using evolutionary neurocontrol

integrated over one time step $\Delta\bar{t} = \bar{t}_{i+1} - \bar{t}_i$ to yield $\mathbf{x}_{\mathrm{SC}}(\bar{t}_{i+1})$. This state is fed back into the NC. The trajectory integration loop stops when the accuracy of the trajectory is sufficient ($\Delta X \leq 1 \Rightarrow \bar{t}_{i+1} = \bar{t}_f$) or when the time limit is reached ($\bar{t}_{i+1} = \bar{t}_\tau$). Then, back in the NC optimization loop, the NC's parameter vector (i.e. its trajectory) is rated by the EA's fitness function $J(\boldsymbol{\pi}_j)$. This fitness value (the rating of the simulated environment) is crucial for the probability of $\boldsymbol{\pi}_j$ to reproduce and to create offspring. It has been found that the performance of the ENC depends strongly on an adequate choice of the fitness function. This is reasonable, since the fitness function has not only to decide autonomously which trajectories are good and which are not, but also which trajectories are promising for future "cultivation" and which are not. Under the selection pressure of the environment, the EA breeds NCs that generate more and more suitable steering strategies that in turn generate better and better trajectories. The EA finally converges against a single steering strategy, which gives in the best case a near-globally[12] optimal trajectory $\mathbf{x}^\star_{\mathrm{SC}}[t]$.

[12] *near*-globally, since global optimality can rarely be proved except by complete enumeration, which is not feasible

5 InTrance Implementation

InTrance has been developed within this work to solve low-thrust trajectory optimization problems, fusing artificial neural networks and evolutionary algorithms to evolutionary neurocontrollers. Within this chapter, the implementation of the evolutionary algorithm is described, which is employed within InTrance to find the neurocontroller that resembles the optimal spacecraft steering strategy. This implementation is largely based on approaches that have been proposed to avoid – or at least to mitigate – the problems (see section 4.3.4) that are associated with standard genetic algorithms like GOLDBERG's SGA. It includes:

- real-valued parameter encoding (section 5.1)
- binary multi-objective tournament selection (section 5.2)
- one-at-a-time reproduction (section 5.2)
- real delta coding (section 5.3)
- non-standard evolutionary operators (section 5.4)

5.1 Precision and Real-valued Coding

It is a long-standing debate in the field of EAs, whether a binary parameter representation or a real-valued parameter representation is superior over the other. For a long time, the GA community has been largely focused on the binary parameter representation, since this representation decomposes the optimization problem into the largest number of smallest building blocks. This viewpoint is now considered to be controversial [94]. A major problem that EAs often encounter when using the binary representation is that they are not able to provide high-precision solutions, and that they are not able to operate in the presence of nontrivial constraints [61]. These problems result from the drawbacks of the binary representation when being applied to high-dimensional high-precision numerical problems, as for example to the optimization of the internal parameters of an ANN.

> **Example:** A 24-30-3 feedforward ANN, as frequently used within this work, has 908 internal parameters π_i. If all of them are bounded to be in the range $[-2, 2]$ and a precision of 8 digits after the decimal point is required, the length of the binary vector is $\ell = 908 \cdot 29 = 26332$. For such problem dimensions, GAs perform poorly [61]. Using a real-coded string, $\ell = 908$ and the precision is the machine precision of the computer.

Experiments like those performed by MICHALEWICZ [61] indicate that the real-valued parameter representation is faster, more consistent from run to run, and provides a higher precision (especially in large search spaces). The real-valued representation is also closer to the problem space, which facilitates the development of problem-specific evolutionary operators and the handling of nontrivial constraints. Therefore, the real-valued parameter representation has been used within this work. Using real-valued coding, a chromosome corresponds to a vector of real numbers and an allele corresponds to a real number value, i.e. $\boldsymbol{\xi} = \langle r_1, r_2, \ldots, r_\ell \rangle$, where ℓ is the dimension of the problem.

5.2 Premature Convergence and Selection Methods

The two most important issues in evolutionary search are **selective pressure** and **population diversity**, both being strongly related: an increase in selective pressure decreases population diversity and vice versa. The **selective pressure** (SP) of a selection method is defined as the expected number of copies of the best individual $\boldsymbol{\xi}_1^t$ in the next generation Ξ^{t+1}. For fitness proportional selection, it is the product of its selection probability and the population size:

$$\mathrm{SP}^t = p_1^t \cdot q = \frac{J(\boldsymbol{\xi}_1^t)}{\sum_{i=1}^{q} J(\boldsymbol{\xi}_i^t)} \cdot q = \frac{J(\boldsymbol{\xi}_1^t)}{J_{\mathrm{avg}}^t} \tag{5.1}$$

The problem with fitness proportional selection is that a selective pressure that is too strong supports **premature convergence** to a local optimum, whereas a selective pressure that is too weak makes the search ineffective. Therefore, one of the most important issues for EAs is to sustain an adequate selective pressure throughout the search. Premature convergence is often a result of so-called **superindividuals**, which have a fitness that is much better than the average fitness of the population. According to equation (5.1), such superindividuals have a large number of offspring, thus preventing other individuals from contributing offspring into the next generation. The result is a fast loss of genetic diversity. One way to circumvent this problem is to choose the fitness function with very careful consideration, or to use a fitness scaling method. However, both approaches are not recommendable, since they are problem dependent and put the responsibility on the user to decide when and how to use them.

Some selection methods have been proposed that make fitness scaling unnecessary through assigning the selection probability based on the individual's *relative* fitness – its **rank** – in the population (see [10] or [16] for a comprehensive survey of selection methods). However, all ranking methods have the drawback of being computationally expensive.[1] Another selection method that is based on ranking but is computationally more efficient is **tournament selection**, where a single individual is selected by choosing some number $\mu \geq 1$ of individuals at random from the population Ξ^t and copying the best individual from this group into the next population Ξ^{t+1}. This process is repeated q times to fill the population. Tournament selection does not require a fitness scaling method, since the selection probability is independent of absolute fitness. For this reason, and due to its simplicity, this selection method has also been employed within this work, with $\mu = 2$ (**binary tournament**), which is the most common tournament size [10]. It can be proved[2] that the selection probabilities are then given by

$$p_i = \frac{(q-i+1)^\mu - (q-i)^\mu}{q^\mu} \tag{5.2}$$

and for $\mu = 2$

$$p_1 = \frac{2q-1}{q^2} \tag{5.3}$$

so that

$$\mathrm{SP} = p_1 \cdot q = \frac{2q-1}{q} = \mathrm{const} \quad \text{with} \quad \lim_{q\to\infty} \mathrm{SP} = 2 \tag{5.4}$$

[1] since the population must always be sorted with respect to the fitness of the individuals

[2] see [10]

so that the selective pressure remains constant throughout the search process. The best individual receives on average about two copies in the next generation, only depending on its rank within the population and independent of its absolute fitness. Another advantage of tournament selection is that each tournament can be performed (e.g. randomly) with respect to a different optimization objective. Such a selection mechanism prefers individuals that perform reasonably well with respect to *all* objectives[3], allowing multi-objective optimization without explicitly weighting the objectives, since the individual objectives need not to be part of the fitness function.

Another approach that has been chosen due to its computational efficiency is to let only one reproduction take place at each time step. This so-called **one-at-a-time reproduction** or **steady-state reproduction** is in contrast to the **generational reproduction** of the SGA, but can be combined easily with tournament selection. Generational reproduction generates an entirely new population Ξ^{t+1} from Ξ^t, thus being computationally expensive, since it must be guaranteed that the good individuals in Ξ^t are also present in Ξ^{t+1}. Consequently, the crossover probability must be significantly less than one, and most of the algorithm's runtime is spent for copying strings without progressing the search. One-at-a-time reproduction in combination with tournament selection is conceptually very simple (see figure 5.1): two tournaments are performed to determine the two parent chromosomes, which are the winners of the tournaments. They stay in the population, while the two tournament losers are replaced by the two offspring chromosomes.

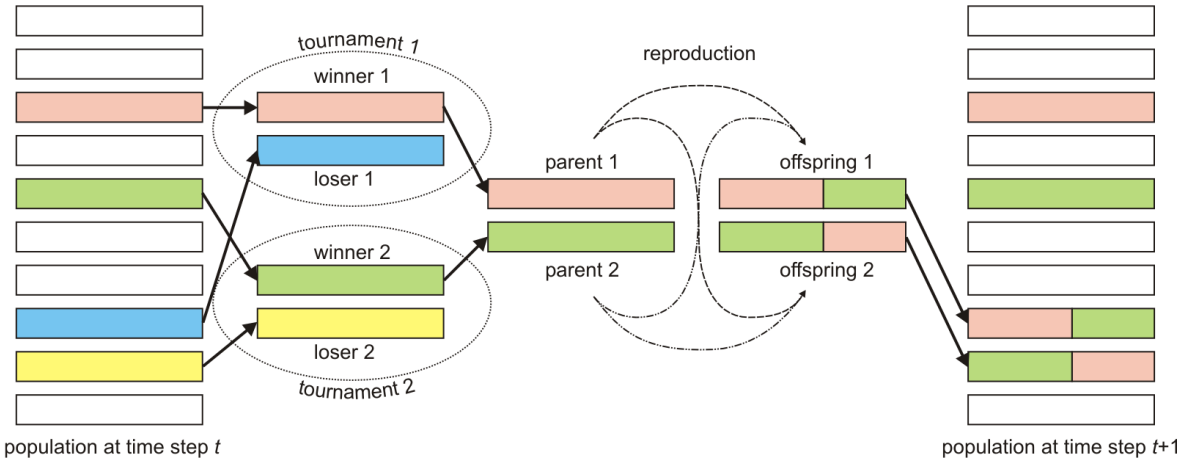


Figure 5.1 One-at-a-time reproduction with tournament selection

[3] It's like a duel between two cowboy gunslingers. To survive a duel, one must draw fast *and* aim accurately.

5.3 Delta Coding and Real Delta Coding

Delta coding (DC) was proposed by WHITLEY et al. [96] to enhance the precision and convergence behavior of genetic search (for binary strings). DC is an iterative search strategy that is based on the idea that a string can also express a distance $\boldsymbol{\delta}$ to some previous solution $\mathbf{h}$, called **interim solution** or **partial solution**, which is the best known solution so far. Using DC, each individual $\boldsymbol{\xi}_k$ consists of two parts, the partial solution and the $\boldsymbol{\delta}$-chromosome:

$$\boldsymbol{\xi}_k = \mathbf{h} + \boldsymbol{\delta}_k$$

or

$$\langle b_1, \ldots, b_\ell \rangle_k = \langle h_1, \ldots, h_\ell \rangle + \langle \delta_1, \ldots, \delta_\ell \rangle_k$$

Consequently, only a dynamically selected subspace of the total search space is explored at any time. This subspace is constructed around the most recent partial solution. DC starts with the initial run of a GA. After the population has converged, the best $\boldsymbol{\delta}$-chromosome is added to the old partial solution to form the new partial solution. After that, a new population is created within an new (reduced or enlarged) search subspace that is centered around the new partial solution. By periodically re-initializing the population, DC avoids premature convergence. DC also provides a mechanism that reduces or enlarges the size of the binary hypercube being currently searched. The **reduction mechanism** allows the algorithm to focus the search on search subspaces that appear promising, whereas the **expansion mechanism** allows the algorithm to explore previously overlooked portions of the search space in later search [59]. To decide whether the population has converged, WHITLEY et al. used the HAMMING-distance between the best and the worst individual, $\sum_{i=1}^{\ell} |b_{i,1} - b_{i,q}|$. The population is re-initialized if the HAMMING-distance is > 1. If the new partial solution is identical to the old one, i.e. if the best $\boldsymbol{\delta}$-chromosome is a zero-string, the hypercube is expanded by one bit. Otherwise the hypercube is contracted by one bit. This continues until a solution is found that meets some user-defined criterion, or until the user-defined maximum number of trials is exhausted [96]. In [59], DC was compared against a standard GA and a mutation-driven stochastic hill-climbing algorithm on a suite of standard EA test functions, and was found to be superior to both algorithms. DC was the only algorithm that consistently found the global optimum for all test functions, using fewer trials than the other algorithms for all but one test function.

DACHWALD et al. extended the idea of delta coding to real-valued strings (floating point delta coding, FPDC) [23, 91], so that

$$\langle r_1, \ldots, r_\ell \rangle_k = \langle h_1, \ldots, h_\ell \rangle + \langle \delta_1, \ldots, \delta_\ell \rangle_k$$

The algorithm that is used within this work is a revised version of FPDC and should be termed **real delta coding** (RDC). Like FPDC, RDC is an extension of WHITLEY's delta coding algorithm to real-valued strings. Figure 5.2 sketches the RDC algorithm. RDC runs in cycles, called **epochs**. Within each epoch, a dynamically selected subspace of the total search space around the most recent partial solution is explored. For the first epoch e_0, the search subspace

$$H_0 = [-\delta_{\max}(e_0), \delta_{\max}(e_0)]^\ell \subset \mathbb{R}^\ell$$

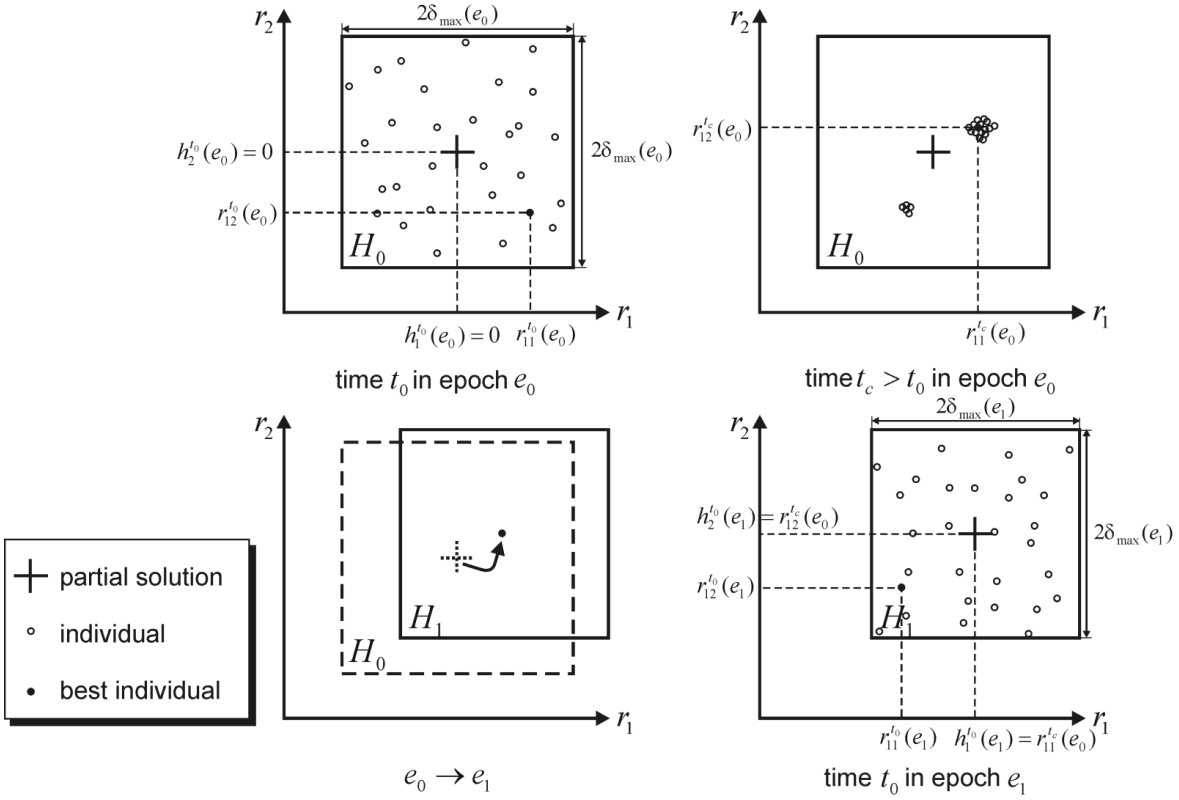


Figure 5.2 Real delta coding

is constructed around the "partial solution" $\mathbf{h}(e_0) = \mathbf{0}$ and the population for the first time step t_0 of epoch e_0 is initialized at random, $\Xi^{t_0}(e_0) = \{\boldsymbol{\delta}_1^{t_0}(e_0), \ldots, \boldsymbol{\delta}_q^{t_0}(e_0)\}$. Then, the EA – as described above – runs until some **epochal convergence criterion** is met. The convergence criterion that has been used within this work does – unlike for DC – not depend on the population diversity but more pragmatically on the relative improvement within the last ν time steps: if at some time step t, the relative improvement within the last ν time steps, $J(\boldsymbol{\xi}_1^t) - J(\boldsymbol{\xi}_1^{t-\nu})$, is less than $\varepsilon\%$, the population is said to be converged and $t = t_c$ (c for convergence). After convergence, $\boldsymbol{\xi}_1^{t_c}(e_0)$, the best found solution in epoch e_0, is taken as the partial solution $\mathbf{h}(e_1)$ for the next epoch. To guarantee the convergence of the algorithm, RDC uses no search space expansion mechanism. The search space reduction mechanism is very simple:

$$\delta_{\max}(e_1) = \kappa \cdot \delta_{\max}(e_0),$$

where $0 < \kappa < 1$ is a user-defined parameter. Within the new search subspace

$$H_1 = [h_1(e_1) - \delta_{\max}(e_1), h_1(e_1) + \delta_{\max}(e_1)] \times \ldots \times [h_\ell(e_1) - \delta_{\max}(e_1), h_\ell(e_1) + \delta_{\max}(e_1)] \subset \mathbb{R}^\ell$$

$\Xi^{t_0}(e_1)$ is again initialized at random, and so on. This is done until some **RDC convergence criterion** is met. This one is similar to the epochal convergence criterion: if at some epoch e_i, the relative improvement to the last epoch, $J(\mathbf{h}(e_{i+1})) - J(\mathbf{h}(e_i))$, is less than $\varepsilon\%$, RDC is

said to be converged. For an ideal RDC performance, the search within each search subspace must be very extensive (ν and κ large, ε small). The optimal values of ν and ε must be set according to the chosen fitness function and a trade-off between search effort and search duration. To put the decision which values to take not on the user, a robust setting for ν and ε is "hard-wired" in the InTrance program. Due to practical search time limitations, two modifications have been made that yield empirical improvements in search behavior within restricted calculation time:

1. The first epoch is performed several (user-defined) times.[4] After those so-called **search space scan epochs**, the best found solution is taken as $\mathbf{h}(e_1)$.

2. At each epoch e_i, the partial solution is defined with respect to the best solution of the last epoch $\boldsymbol{\xi}_1^{t_c}(e_{i-1})$ and *not* with respect to the best solution found so far $\boldsymbol{\xi}^\star$ [96]. RDC takes $\boldsymbol{\xi}_1^{t_c}(e_{i-1})$ as the center of the new search subspace, but saves $\boldsymbol{\xi}^\star$. If $\boldsymbol{\xi}_1^{t_c}(e_{i-1})$ is worse than $\boldsymbol{\xi}^\star$, the search subspace is shifted but not contracted. However, after having done this for two subsequent epochs without finding a better $\boldsymbol{\xi}^\star$, the old $\boldsymbol{\xi}^\star$ is taken again as partial solution and the search subspace is again contracted.

5.4 Evolutionary Operators

Standard genetic operators that work on binary chromosomes can not be applied directly to real-valued chromosomes. Therefore, new genetic operators, which are tailored to work on real-valued strings, have to be designed. Besides the analogues to the one-point crossover operator and the bit mutation operator that is used in the SGA, many other evolutionary operators can be conceived and – consequently – many evolutionary operators have been proposed so far. This section describes the ones that are employed within InTrance.

5.4.1 Crossover

InTrance implements three crossover operators (see figure 5.3):

1. The implemented **one-point crossover operator** works analogous to its binary pendant.

2. If **uniform crossover** is applied, it is decided for each locus in the first offspring (by the simulated toss of a coin) which parent contributes its allele in that position. The second offspring receives the allele from the other parent. Since uniform crossover exchanges single alleles and not segments, it can combine features regardless of their relative location on the string. It has been shown that this ability outweighs the disadvantage of destroying building blocks for some binary-coded problems [61, 90].

3. If **crossover nodes** is applied, it is decided for each coded neuron in the first offspring (by the simulated toss of a coin) which parent contributes its coded parameters for that neuron. The second offspring receives the coded neuron from the other parent. This prevents the logical subgroups of the string – the parameters of a single neuron –

[4] using different randomly generated initial populations Ξ^0

from being torn apart. Crossover nodes was proposed and studied in [63] as a tailored operator for ANN parameter optimization, where it was implemented within a GA that outperformed the backpropagation algorithm on a complex sonar image classification problem.

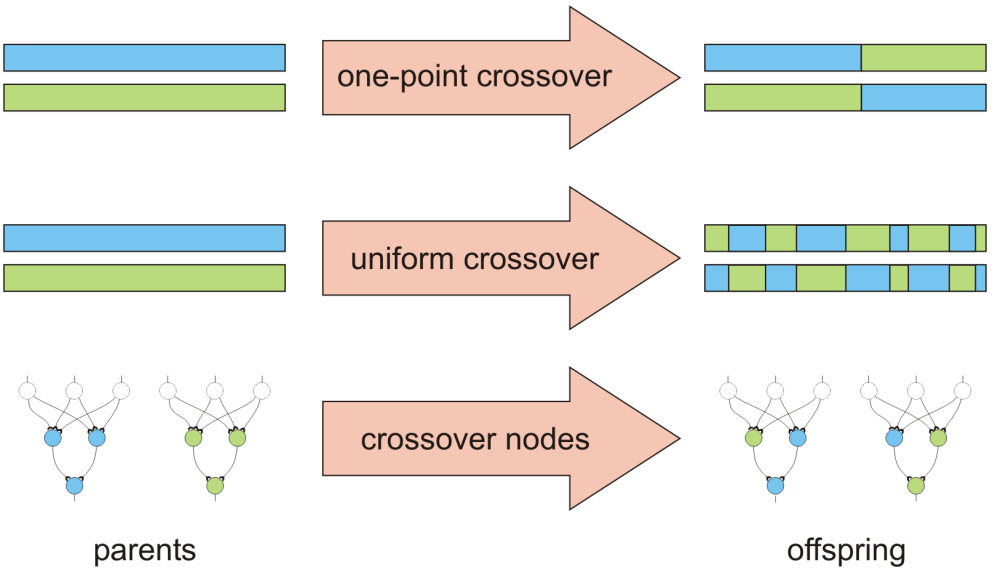


Figure 5.3 Implemented evolutionary operators

The one-point crossover operator and the uniform crossover operator are applied with probability 1/4 respectively, whereas the crossover nodes operator is applied with probability 1/2, so that in total $p_c = 1$. It is to note that all three crossover operators do only exchange real numbers between the chromosomes but do not change the numbers themselves. This can only be done by the mutation operator.

5.4.2 Mutation

WHITLEY pointed out that for DC no mutation operator is necessary, since the population is re-initialized at regular intervals [96]. Preliminary InTrance tests, however, have revealed that this might be different for real-valued strings. In this case, the absence of a mutation operator leads for small population sizes to premature convergence within the epochs. For this reason, InTrance implements a mutation operator, which should be termed **fast uniform mutation**. Fast uniform mutation is based on **uniform mutation** [90]. However, if uniform mutation was applied to a string, it would be decided for each locus (with probability $p_m \approx 10^{-3}$), whether or not the respective allele is to be mutated. This process is very time consuming.[5] Therefore, a fast uniform mutation operator is implemented in InTrance. If fast uniform mutation is applied, it is decided for the entire chromosome (with probability $0 \leq p_m \leq 1$), whether or not a single allele of the chromosome is to be mutated. If this is the case, the locus that is to be mutated is randomly selected. If for example the i^{th} locus of chromosome $\boldsymbol{\delta}_j$ is to be mutated, its allele δ_{ji} is replaced with a new one, $\delta'_{ji} \in [-\delta_{\max}, +\delta_{\max}]$.

[5] since about 1000 random numbers have to be generated for a single mutation

5.5 Additionally Encoded Problem Parameters

If an EA is already employed for the evolution of the ENC, it is manifest to employ this EA also for the parallel optimization of additional problem parameters, which can be done without major additional effort. This way, the following parameters are additionally encoded on the chromosome, making them an explicit part of the optimization problem:

- the launch date t_0
- the launch velocity $|\mathbf{v}_\infty|$ (hyperbolic excess velocity)
- the launch azimuth, defined as the angle between $\mathbf{e}_\varphi$ and the projection of $\mathbf{v}_\infty$ into the $\mathbf{e}_r$-$\mathbf{e}_\varphi$-plane
- the launch elevation, defined as the angle between the projection of $\mathbf{v}_\infty$ into the $\mathbf{e}_r$-$\mathbf{e}_\varphi$-plane and $\mathbf{v}_\infty$
- the initial propellant mass[6] $m_\mathrm{P}(t_0)$

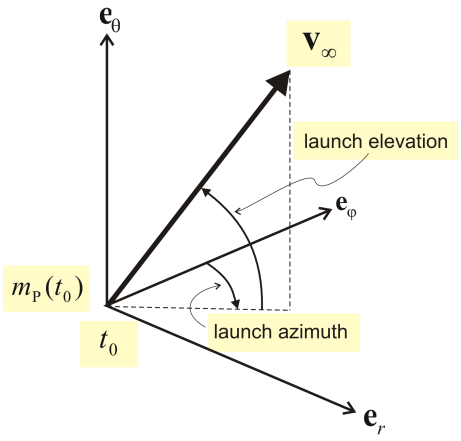


Figure 5.4 Additionally EA encoded problem parameters

For this purpose, an interval is defined for each additional optimization parameter, together with a mapping function that maps the respective allele onto this interval.

[6] except for solar sailcraft

6 InTrance Evaluation

Before InTrance can be considered as a viable low-thrust trajectory optimization tool, its convergence behavior has to be evaluated and the quality of the obtained solutions has to be assessed. Within this chapter, InTrance is used to re-calculate low-thrust problems, for which trajectories can be found in the literature (henceforth called reference problems/trajectories). Since those trajectories are the best ones available so far, they must provide the benchmark to assess the performance of InTrance for low-thrust trajectory optimization. All reference problems employ a one-body simulation model, where the gravitational influence of the planets and other disturbing forces are neglected.

Several studies for interplanetary solar sailcraft missions have been carried out at DLR [47, 51, 52, 54, 55, 86] and elsewhere [80, 98], so that a reasonable sample of solar sailcraft reference trajectories is available. All reference problems employ an ideal solar sailcraft model. Three of them are used within this chapter to assess the suitability of InTrance for solar sailcraft trajectory optimization and to investigate the influence of various factors (accuracy requirements, EA parameters, NC topologies, NC input noise, disturbing forces):[1] a Mercury rendezvous for an advanced solar sailcraft ($a_c = 0.55\,\mathrm{mm/s^2}$, section 6.1), a near-Earth asteroid (NEA) rendezvous for a moderate-performance solar sailcraft ($a_c = 0.14\,\mathrm{mm/s^2}$, section 6.2), and a fast Pluto fly-by for a very advanced solar sailcraft ($a_c = 1.0\,\mathrm{mm/s^2}$, section 6.3). Within section 6.4, InTrance is used to calculate minimal orbit transfer times to various solar system bodies for ideal solar sailcraft with characteristic accelerations in the range $0.1\,\mathrm{mm/s^2} \leq a_c \leq 10.0\,\mathrm{mm/s^2}$, extending the currently available data to moderate-performance solar sailcraft ($a_c \lesssim 0.5\,\mathrm{mm/s^2}$) and to solar sailcraft of extremely high performance ($a_c \gtrsim 2.5\,\mathrm{mm/s^2}$).

For EP spacecraft, it is more difficult to find suitable reference trajectories, since the complete EP system models are typically not given in the literature. However, trajectories for a multiple NEA rendezvous and sample return mission, called "Hera", using an SEP spacecraft with a cluster of three NSTAR thrusters, were found for reference (section 6.5) [4].

6.1 Mercury Rendezvous Mission

To assess the performance of InTrance for solar sailcraft trajectory optimization, a rendezvous trajectory to Mercury for an ideal solar sailcraft with a characteristic acceleration of $a_c = 0.55\,\mathrm{mm/s^2}$, calculated by Leipold et al. [51, 54, 55] using a LTOM, has been taken for reference. This reference trajectory launches at Earth at 15 Jan 03 (MJD 52654.5) with an hyperbolic excess energy of $C_3 = \mathbf{v}_\infty^2 = (\dot{\mathbf{r}}_{\mathrm{SC}}(t_0) - \dot{\mathbf{r}}_{\mathrm{Earth}}(t_0))^2 = 0\,\mathrm{km^2/s^2}$ and takes 665 days to rendezvous Mercury.

[1] For this purpose it is not a restriction that the launch date of some reference trajectories lies by now in the past.

6.1.1 Convergence Behavior and Stability

To evaluate its convergence behavior and stability, InTrance was run five times – using different seed values for the C++ (pseudo-)random number generator[2] – for the same launch date as in [51, 54, 55] (reference launch date). A 12–30–3 neurocontroller[3] has been used, where the input neurons receive the actual solar sailcraft state $\mathbf{x}_{\mathrm{SC}}$ and the actual target body state $\mathbf{x}_{\mathrm{T}}$ in cartesian coordinates, and the output neurons define – according to equation (4.17) – directly the thrust direction unit vector $\mathbf{d}$. On the basis of preliminary InTrance-runs, which indicated that the optimal transfer takes less than 600 days, the maximum transfer time was set to $T_{\max} = 600\,\mathrm{days}$. For discretization, this time interval was cut into $\tau = 600$ finite elements of equal length, so that the solar sailcraft is "allowed" to change its attitude once every day. The final accuracy limit (convergence criterion) was set to $\Delta r_{f,\max} = 100\,000\,\mathrm{km}$ and $\Delta v_{f,\max} = 100\,\mathrm{m/s}$.[4] The population size was set to $q = 50$. As for all InTrance-calculations within this work, a RUNGE-KUTTA-FEHLBERG method of order 4(5) has been used for the numerical integration of the trajectories.[5]

Figures 6.1 and 6.2 show the results of the five InTrance-runs. Although the variation of the sail steering angles differs considerably for the five steering strategies (see figure 6.2.2), the respective trajectories differ only little in transfer time (figure 6.1), which may indicate some robustness concerning the required sail steering accuracy for a real mission. The worst found trajectory ($T = 589\,\mathrm{days}$) takes only 2.6% longer to rendezvous Mercury than the best one, so that there can be some confidence that the best found trajectory (figure 6.2.1) is not far from to the global optimum. It is 91 days ($\Delta T\% = 16\%$)[6] faster than the reference trajectory, revealing that the latter is far from the global optimum. The final distance to Mercury is

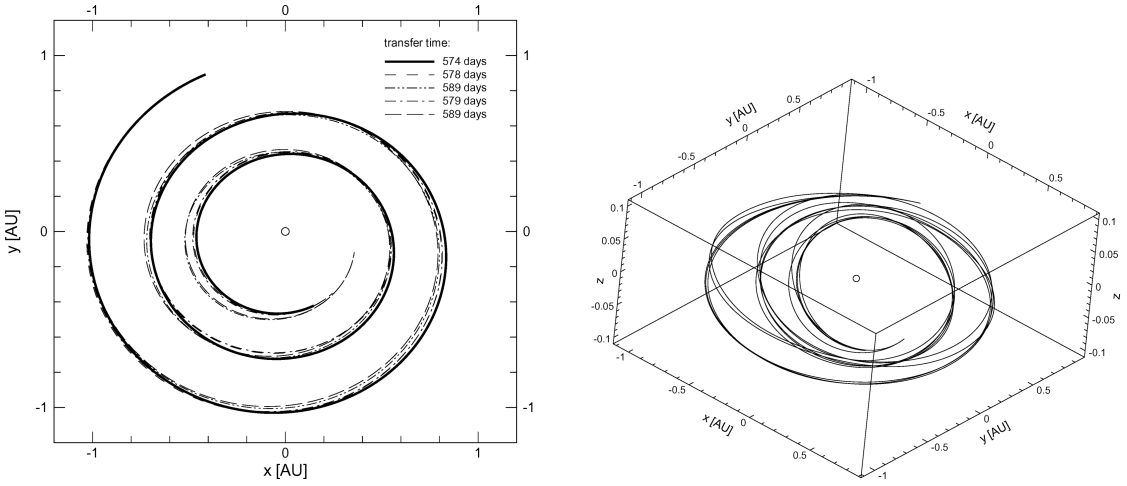


Figure 6.1 Mercury rendezvous trajectories for five different initial NC population (reference launch date)

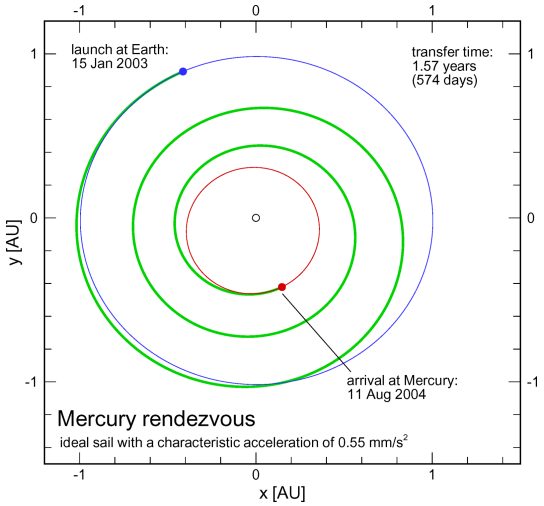
[2] leading to different initial NC populations

[3] 12 input neurons, 1 hidden layer with 30 hidden neurons, 3 output neurons

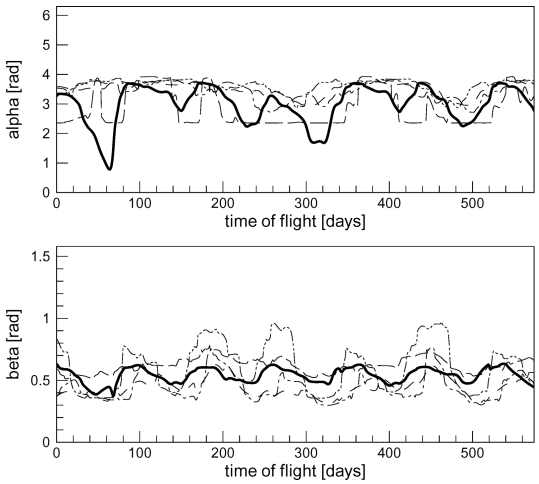
[4] $\Delta v_{f,\max} = 100\,\mathrm{m/s}$ was also used by LEIPOLD et al. [51], whereas $\Delta r_{f,\max}$ is not given in [51]

[5] The RUNGE-KUTTA-FEHLBERG (RKF) method uses a RUNGE-KUTTA method of order 5 to estimate the local error in a RUNGE-KUTTA method of order 4.

[6] $\Delta T\% = \frac{T_{\mathrm{InTrance}} - T_{\mathrm{R}}}{\min(T_{\mathrm{R}}, T_{\mathrm{InTrance}})} \cdot 100\%$, where T_{R} is the transfer time of the reference trajectory



6.2.1: Best InTrance-trajectory



6.2.2: Steering angles (the bold line denotes the steering angles for the trajectory shown on the left side)

Figure 6.2 Mercury rendezvous (reference launch date)

$\Delta r_f \approx 57\,000\,\text{km}$ and the final relative velocity to Mercury is $\Delta v_f \approx 57\,\text{m/s}$, both being well better than the required accuracy limits.

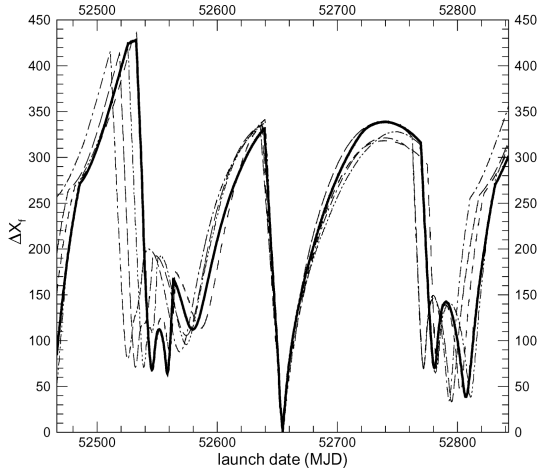


Figure 6.3 Launch date fingerprints for the five different steering strategies shown in figure 6.1 and 6.2

To assess the similarity of the five InTrance-steering strategies, so-called "**launch date fingerprints**" (LDFs) have been calculated (figure 6.3). These LDFs show the accuracy of the steering strategy according to equation (4.22) for different launch dates within a 1 year-interval around the nominal launch date (MJD 52654.5). From the LDFs, one can see that all five steering strategies achieve a good accuracy only for the launch date for which they have been "bred" by the evolutionary algorithm, so that they are not *universally* valid. A **universal steering strategy** would achieve a good accuracy for *all* launch dates. For each

specific launch date, the final trajectory deviation is approximately the same for all five ENCs. This similarity of the LDFs indicates that – generating similar outputs from similar[7] inputs – the underlying steering strategies can also be expected to be similar.[8]

6.1.2 Different Population Sizes and Accuracy Requirements

To assess the influence of search duration (which is due to the RDC epochal convergence criterion (see section 5.3) mainly determined by the population size of the EA) and required accuracy ($\Delta r_{f,\max}$ and $\Delta v_{f,\max}$) on the results, InTrance was run for three different population sizes and two final accuracy limits (FAL1: $\Delta r_{f,\max} = 1\,000\,000\,\mathrm{km}$, $\Delta v_{f,\max} = 500\,\mathrm{m/s}$; FAL2: $\Delta r_{f,\max} = 100\,000\,\mathrm{km}$, $\Delta v_{f,\max} = 100\,\mathrm{m/s}$), using the same NC as in the last section. The results are given in tables 6.1 and 6.2. Table 6.3 shows the average runtime for the calculations.

Population size	Transfer time T					Average	Std. dev.
q	run 1	run 2	run 3	run 4	run 5	T_{avg}	σ_T
25	566	573	577	570	572	571.6	4.04
50	569	568	573	571	568	569.8	2.17
100	570	**564**	**564**	567	569	**566.8**	2.77

Table 6.1 Transfer times to Mercury for different population sizes (FAL1, five different initial NC populations)

Population size	Transfer time T					Average	Std. dev.
q	run 1	run 2	run 3	run 4	run 5	T_{avg}	σ_T
25	585	592	580	579	588	584.8	5.45
50	**574**	578	589	579	589	**581.8**	6.83
100	584	590	576	583	585	583.6	5.03

Table 6.2 Transfer times to Mercury for different population sizes (FAL2, five different initial NC populations)

For the less demanding FAL1, the trajectories are faster and the standard deviation of the results is lower, as expected. However, unlike it may be expected, the quality of the solutions does not depend considerably on the population size and thus on the search duration. However, as table 6.3 shows, the search duration depends considerably on the population size and on the required accuracy. The dependence on population size is straightforward. A larger population takes longer to converge against a single point of the actual search hypercube (or better: a small subspace within the actual search hypercube), where no further improvement is probable. Therefore, the population size had been introduced into the RDC epochal convergence criterion, so that the EA is allowed to search longer when the population size is large. The dependence of search duration on the final accuracy limit can be attributed to the employed fitness function, which allots little value to further improvements in Δr_f and Δv_f, if the required accuracy is already achieved ($\Delta X_f \leq 1$). Consequently, a more demanding

[7] the inputs are only equal for $\bar{t}_0$ and diverge gradually due to the differences in the steering strategies

[8] as it will be shown later, this similarity vanishes, if different NC output sets are used (see figure 6.5)

Accuracy limit	Average runtime [hours] $q = 25$	$q = 50$	$q = 100$
FAL1	3.0	5.8	7.9
FAL2	5.0	6.8	11.6

Table 6.3 Average runtime on a personal computer with a 1.3 GHz AMD Athlon processor

accuracy requirement leaves more room for improvements, thus delaying the convergence of the RDC algorithm. Typically, about 3 individuals (trajectories!) are tested per second, so that – depending on the above mentioned factors – between about 25 000 and 250 000 individuals have been tested per InTrance-run. For the calculations that follow, a population size of $q = 50$ and FAL2 has been used, if it is not stated otherwise.

6.1.3 Different Neurocontrollers

For the calculations above, a 12–30–3 neurocontroller had been used, where the input neurons receive the actual solar sailcraft state and the actual target state in cartesian coordinates, and the output neurons define directly the direction unit vector. Since this NC is only one of many possible NCs that may be used for this trajectory optimization problem, other NCs should be tested as well.

Within this section, different NCs (i.e. different NC input/output sets and different numbers of hidden neurons/layers) are tested. Various different input sets $\mathcal{X}$ (table 6.4) and three different output sets $\mathcal{U}$ (table 6.5) have been considered for this purpose.

Notation	Input set $\mathcal{X}$
(c) 12–□	$\mathbf{x}_{SC}$ and $\mathbf{x}_T$ in cartesian coordinates
(c) 6–□	$\mathbf{x}_T - \mathbf{x}_{SC}$ in cartesian coordinates
(p) 12–□	$\mathbf{x}_{SC}$ and $\mathbf{x}_T$ in polar coordinates
(e) 6–□	$\mathbf{x}_T - \mathbf{x}_{SC}$ as orbital element differences ($a_T - a_{SC}$, etc.)
(cp) 24–□	$\mathbf{x}_{SC}$ and $\mathbf{x}_T$ in cartesian and polar coordinates
(cp) 12–□	$\mathbf{x}_T - \mathbf{x}_{SC}$ in cartesian and polar coordinates
(ce) 12–□	$\mathbf{x}_T - \mathbf{x}_{SC}$ in cartesian coordinates and as orbital element differences
(pe) 12–□	$\mathbf{x}_T - \mathbf{x}_{SC}$ in polar coordinates and as orbital element differences
(cpe) 36–□	$\mathbf{x}_{SC}$ and $\mathbf{x}_T$ in cartesian and polar coordinates and as orbital elements
(cpe) 18–□	$\mathbf{x}_T - \mathbf{x}_{SC}$ in cartesian and polar coordinates and as orbital element differences

(□ = wildcard for hidden layer(s) and output layer)

Table 6.4 Tested NC input sets

Table 6.6 shows the results for the NCs that achieved the required final accuracy limit (FAL2) in all of five InTrance-runs. It is to note that all NCs in table 6.6 represent direct steering strategies. Surprisingly, all tested indirect steering strategies failed (at least in one run) to achieve the required accuracy. On average, the best results (best solution, best average, low standard deviation) have been obtained using the (cp) 24–30–3 (d) neurocontroller. It is also

Notation	Output set $\mathcal{U}$	Strategy type
□–5 (a)	the NC provides the steering law weight vector $\mathbf{c}_Q$ (adapting the orbital elements to that of the target)	indirect
□–10 (m)	the NC provides the steering law weight vector $\mathbf{c}_L$ (minimizing/maximizing the orbital elements)	indirect
□–3 (d)	the NC provides the direction unit vector $\mathbf{d}$, along which the thrust force has to be maximized	direct

(□ = wildcard for input layer and hidden layer(s))

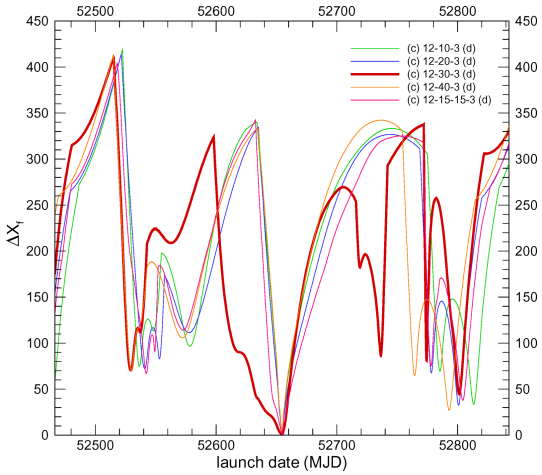
Table 6.5 Tested NC output sets

NC topology	Transfer time T					Average	Std. dev.
	run 1	run 2	run 3	run 4	run 5	T_{avg}	σ_T
(c) 12–10–3 (d)	592	582	587	593	577	586.2	6.76
(c) 12–20–3 (d)	579	583	579	577	580	579.6	2.19
(c) 12–30–3 (d)	574	578	589	579	589	581.8	6.83
(c) 12–40–3 (d)	583	579	575	578	578	578.6	2.88
(c) 12–15–15–3 (d)	577	578	577	581	592	581.0	6.36
(cp) 24–20–3 (d)	580	584	575	**573**	591	580.6	7.23
(cp) 24–30–3 (d)	579	576	**573**	578	575	**576.2**	2.39
(cp) 24–40–3 (d)	591	575	580	581	586	582.6	6.11
(cp) 24–15–15–3 (d)	575	579	578	580	583	579.0	2.92
(c) 6–30–3 (d)	591	592	583	579	581	585.2	5.93
(cpe) 36–30–3 (d)	583	589	579	584	579	582.8	4.15
(cpe) 18–30–3 (d)	581	591	585	577	587	584.2	5.40

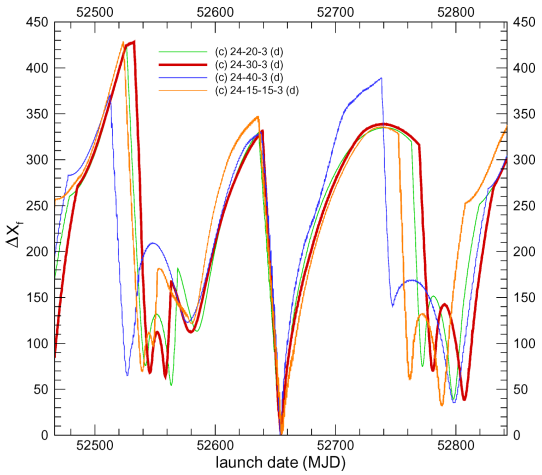
Table 6.6 Transfer times to Mercury for different steering strategy sets and different NC topologies (FAL2, reference launch date, five different initial NC populations)

quite counter-intuitive that only the NCs with cartesian inputs achieved the required accuracy in all cases, although the motion of spacecraft in interplanetary space is better described in polar coordinates or using orbital elements. This might be due to the fact that the variation of the states is much larger in cartesian coordinates than in polar coordinates or in orbital elements, providing thus a more substantial input to the NC.

The LDFs in figure 6.4 indicate that, independent of the number of hidden neurons, the optimal steering strategies for the (c) 12–□–3 (d) neurocontrollers and the (c) 24–□–3 (d) neurocontrollers are quite similar. Consequently, the number of hidden neurons has – at least for this trajectory optimization problem – little effect on the steering strategy that the NC represents. Even NCs with a very small number of hidden neurons provide acceptable results for this problem. In contrast to the results for different NC input sets and different hidden neurons/layers, the LDFs in figure 6.5 show that the resulting optimal steering strategies differ very much for different NC *output* sets. The LDFs for the indirect steering strategies oscillate remarkably within small time scales, leading to different solutions for neighboring launch dates, which is an unfavorable characteristic concerning the universality of the steering strategy. Apart from those results, direct steering strategies are more elegant and have



6.4.1: (c) 12–□–3 (d) neurocontrollers



6.4.2: (c) 24–□–3 (d) neurocontrollers

Figure 6.4 Launch date fingerprints for different neurocontroller input sets and topologies

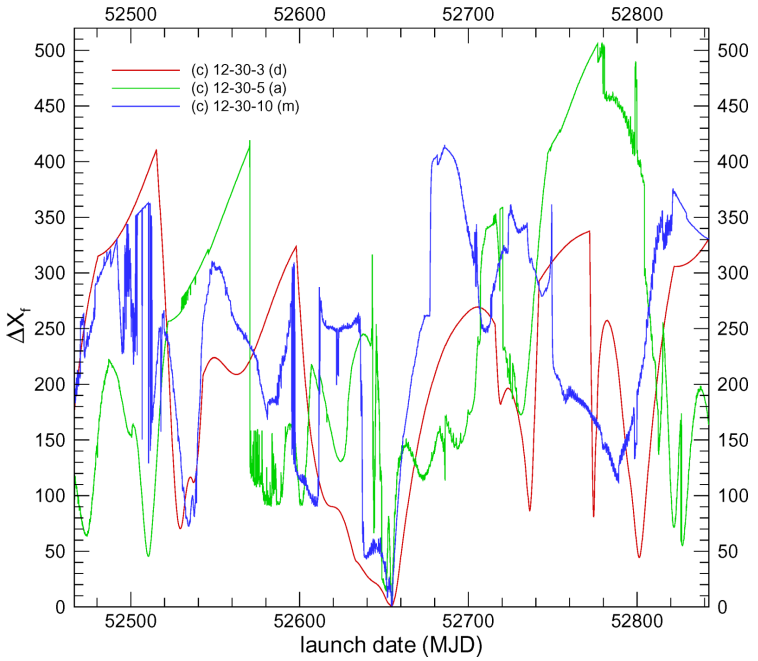


Figure 6.5 Launch date fingerprints for different neurocontroller output sets ((c) 12–30–□ neurocontrollers)

a broader applicability, since indirect steering strategies can not be used for trajectories that turn hyperbolic[9]. Therefore, direct steering strategies have been used for the remainder of this work.

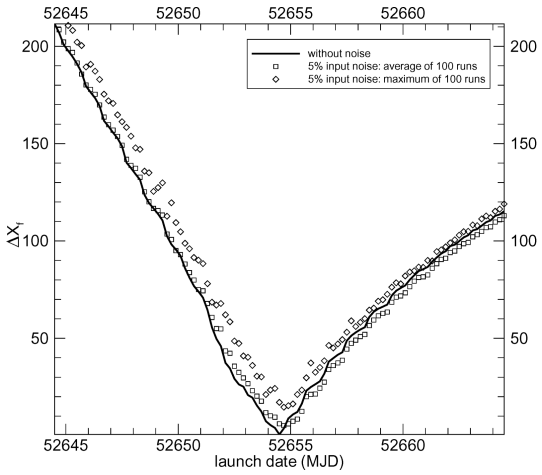
[9] note that Lagrange's planetary equations in Gauss' form are only valid for elliptical orbits

6.1.4 Noise and Disturbing Forces

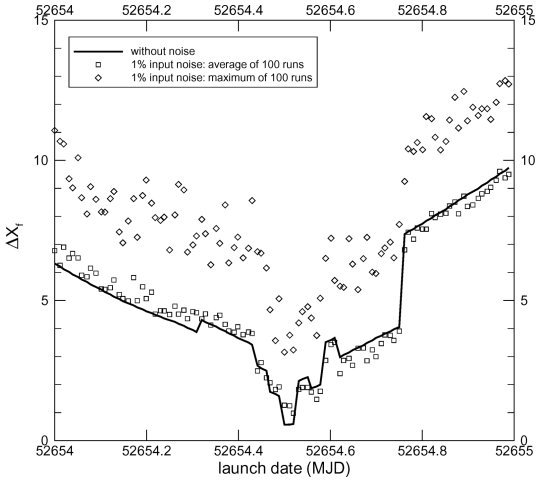
To assess the robustness of InTrance-generated steering strategies further, the best found steering strategy so far (run 3 for the (cp) 24–30–3 (d) neurocontroller in section 6.1.2, see table 6.6) has been evaluated under simulated NC input noise and under simulated random disturbing forces acting on the spacecraft.

6.1.4.1 Neurocontroller Input Noise

The simulated NC input noise is intended to model random errors in the spacecraft's *sensed* position and velocity (not the position and the velocity itself!), as they occur due to errors in the spacecraft's autonomous sensors and/or errors in the measurement of the spacecraft's position and velocity from Earth. An n% NC input noise is simply simulated by multiplying at each time step each true NC input value with a different normally distributed random number, where the mean of the normal distribution is 1 and the standard deviation is n/100.



6.6.1: 5% NC input noise



6.6.2: 1% NC input noise

Figure 6.6 Accuracy of the NC steering strategy under NC input noise

Figure 6.6 shows the final accuracy of the NC steering strategy under an NC input noise of 5% and 1% respectively. For a real space mission, the precision of the spacecraft's sensed state is certainly far below 1%, so that the input noise has little effect on the accuracy of the generated trajectory. The insufficient accuracy ($\Delta X_f > 1$) for launch dates other than the nominal launch date is more due to the non-universality of the NC steering strategy itself than due to the NC input noise, as figure 6.6 shows.

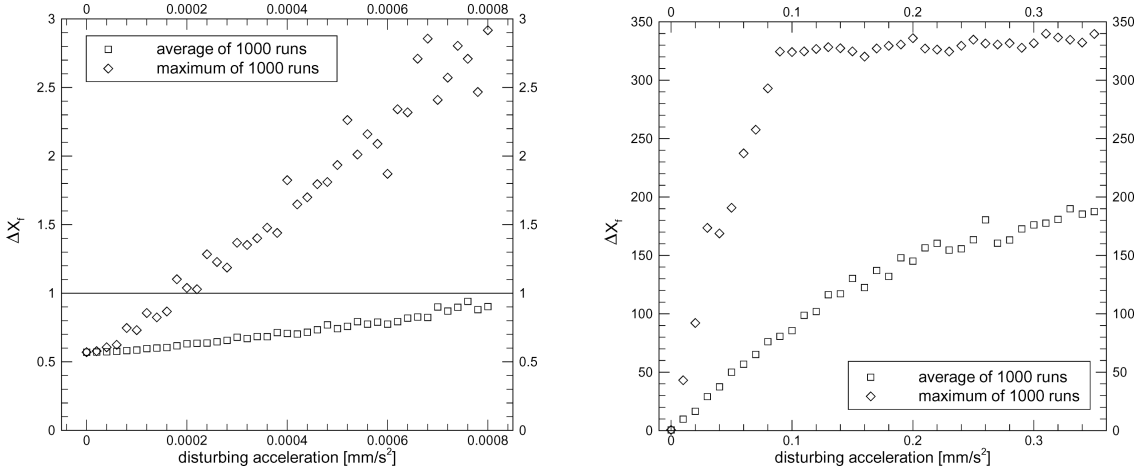
6.1.4.2 Disturbing Forces

The simulated disturbing acceleration acting on the spacecraft is intended to model the stochastic effects due to the sailcraft itself, which have been explicitly excluded from the simulation model (e.g. non-instantaneous attitude control maneuvers, non-flatness of the sail film, etc.). It is *not* intended as a dedicated simulation of the gravitational disturbing forces

of the planets, which are obviously not stochastic. A dedicated simulation of these forces has not yet been undertaken. In any case, they are expected to be much smaller than the sails "disturbing" forces, unless very close planetary fly-bys are involved, although they may have significant secular effects.

> **Example:** Given a mass of about $4.869 \cdot 10^{24}$ kg, the gravitational acceleration of Venus at a distance of 0.05 AU is about $5.8 \cdot 10^{-3}$ mm/s^2 and thus only about 1% of the sail acceleration.

The disturbing acceleration is simulated by adding at each integration step an additional acceleration term to the equations of motion. The magnitude of this term is a normally distributed random number, where the mean of the normal distribution is 0 and the standard deviation is $0 < a_d < a_c$. Although this disturbing force model is quite crude and should for a thorough mission analysis be replaced with a more sophisticated model, it is sufficient for a rough estimate of the effects on the neurocontroller. Figure 6.7 shows the final accuracy of the NC steering strategy under a random disturbing acceleration.



6.7.1: $0 \leq a_d \leq 0.8 \cdot 10^{-3}\,\mathrm{mm/s^2}$ **6.7.2:** $0 \leq a_d \leq 0.35\,\mathrm{mm/s^2}$

Figure 6.7 Accuracy of the NC steering strategy under random disturbing accelerations

Figure 6.7 shows that the NC steering strategy is quite sensitive with respect to stochastic disturbing accelerations. Within this very simple model, the required accuracy ($\Delta X_f < 1$) can be met in the worst case up to only $a_d \approx 0.18 \cdot 10^{-3}\,\mathrm{mm/s^2}$, and in the average case up to $a_d \approx 0.8 \cdot 10^{-3}\,\mathrm{mm/s^2}$. Consequently, for a thorough trajectory design, as it is required for a real mission, a more realistic spacecraft model must be used. Since the translational motion is highly interrelated with the rotational motion, this model should consider all six degrees of freedom. Also the billowing, wrinkling and aging of the sail must be studied through FEM simulations and in-space tests. In this case, also a complete gravitational solar system model should be used. However, even those models can not include the random variation of the solar constant[10]. Nonetheless, this does not prevent using InTrance for the trajectory optimization of a real mission, since at any time a new optimal trajectory can be calculated in real-time (on Earth or on-board).

[10] approx. 0.1% over days [11]

6.1.5 Optimization of the Launch Date

To find the optimal launch date for the Mercury rendezvous, InTrance was used to determine the shortest orbit transfer. Since no rendezvous with the target body but only with the target orbit is required, the time span for launch was set to the orbital period of the initial body[11], i.e. to 1 year. Having run InTrance five times with different initial NC populations, the shortest found orbit transfer takes $T = 510$ days to reach the orbit of Mercury. For this orbit transfer, the solar sailcraft's angular position at launch is $\varphi_{\mathrm{SC}}(\bar{t}_0) \doteq -2.90$ and at arrival $\varphi_{\mathrm{SC}}(\bar{t}_0 + T) \doteq -0.83$. By scanning the planetary positions, it can be found that within a 1 year-interval around the reference launch date (MJD 52654.5), the constellation of Earth and Mercury is most similar to that of the optimal orbit transfer solution for a launch at 27 Mar 03 (where $\varphi_{\mathrm{Earth}}(\mathrm{MJD}\ 52725.8) \doteq -3.02$ and $\varphi_{\mathrm{Mercury}}(\mathrm{MJD}\ 53235.8) \doteq -0.83$).

To find the optimal Mercury rendezvous, InTrance was run (five times) for the launch date that was expected to be optimal (MJD 52725.8). However, to allow the steering strategy to compensate for the small difference in Earth's angular position at launch, the launch date was not prescribed exactly, but the EA was allowed to choose the launch date from the interval $[\mathrm{MJD}\ 52724.0, \mathrm{MJD}\ 52729.0]$. The maximum transfer time was set to $T_{\max} = 600$ days (with $\tau = 600$).

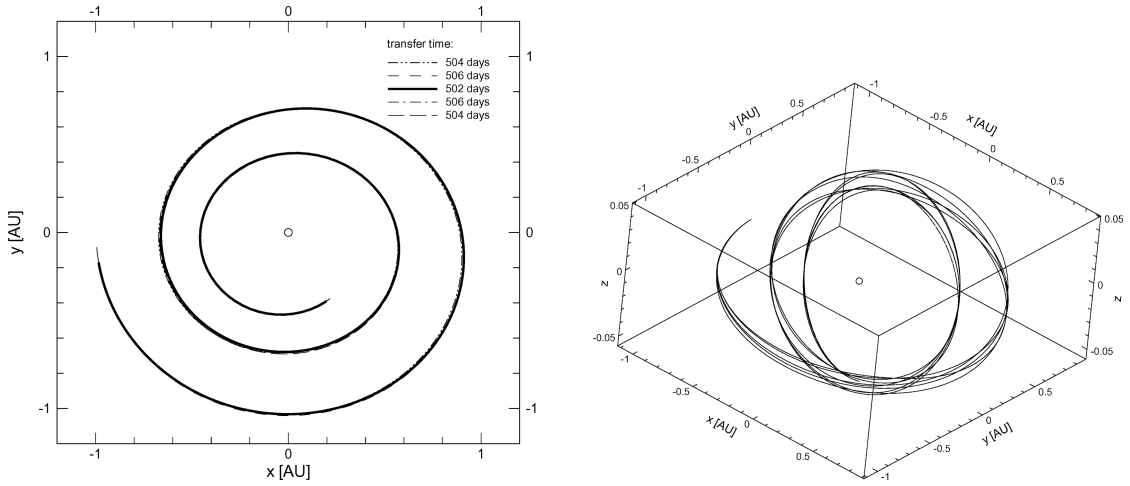
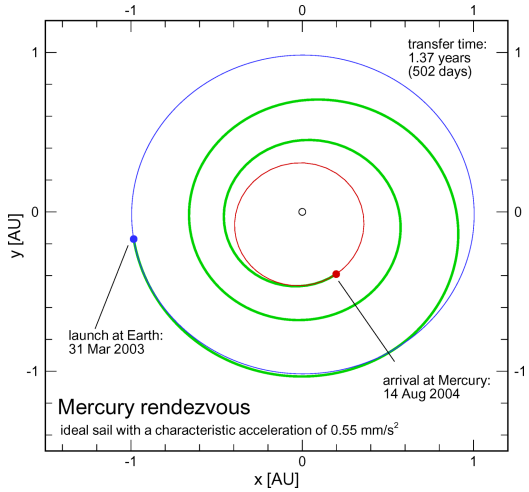


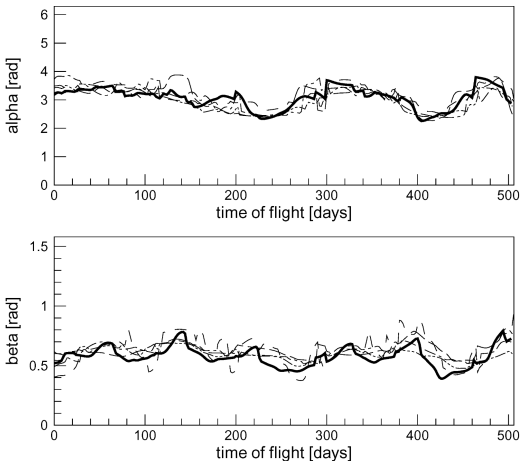
Figure 6.8 Mercury rendezvous trajectories for five different initial NC populations (optimized launch date)

Figure 6.8 shows that all five InTrance-trajectories differ little, so that there can be some confidence that the best found trajectory is close to the global optimum. Taking 502 days to rendezvous Mercury, the best InTrance-trajectory (figure 6.9.1) is now 163 days (32%) faster than the reference trajectory. The final distance to Mercury is $\Delta r_f \approx 20\,000$ km and the final relative velocity to Mercury is $\Delta v_f \approx 20$ m/s, both being well better than the required values (FAL2). The optimal launch date was found to be 31 Mar 03, 75 days later than the reference launch date. It is to note that the optimal transfer time to rendezvous Mercury is even better than the previously found "optimal" orbit transfer time of 510 days, which is obviously not optimal ex post. This might be attributed to the larger time span for launch (1 year instead

[11] it is to note that the orbital period of the target body is not relevant for this problem



6.9.1: Best InTrance-trajectory



6.9.2: Steering angles

Figure 6.9 Mercury rendezvous (optimized launch date)

of 5 days), which makes the search more difficult for the EA, since different launch dates within a 1 year-time span require significantly different steering strategies, whereas the good steering strategies within a 5 day-time span are more similar.

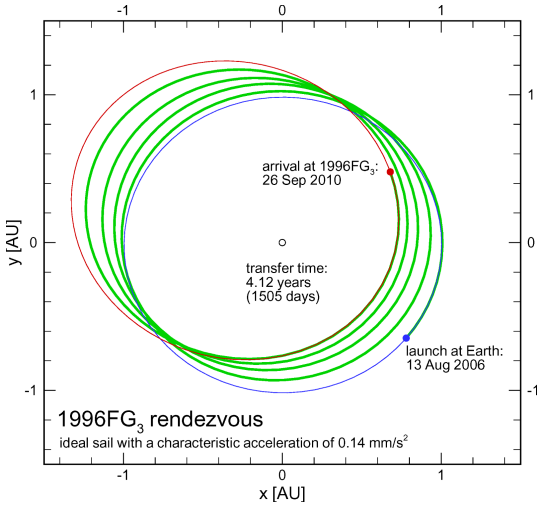
6.2 Near-Earth Asteroid Rendezvous Mission

As another test case to assess the performance of InTrance, a rendezvous trajectory to near-Earth asteroid 1996FG$_3$ for an ideal solar sailcraft with a characteristic acceleration of $a_c = 0.14\,\mathrm{mm/s^2}$, calculated by Leipold et al. [47, 86] using a LTOM, has been chosen. This reference trajectory launches at Earth at 13 Aug 06 (MJD 53960.5) and takes 1640 days to rendezvous 1996FG$_3$, if the solar sailcraft is inserted directly into an interplanetary trajectory with an hyperbolic excess energy of $C_3 = 4\,\mathrm{km^2/s^2}$.

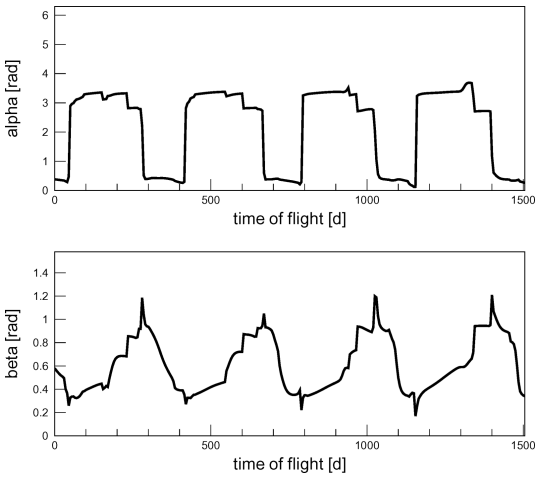
6.2.1 Convergence Behavior and Stability

To evaluate its convergence behavior and stability, InTrance was run five times – using different initial NC populations – for the same launch date as in [47, 86]. For the calculations within this section, a (cp) 24–30–3 (d) neurocontroller has been used. The maximum transfer time was set to $T_{\max} = 1800$ days. For discretization, this time interval was cut into $\tau = 360$ finite elements, so that the solar sailcraft is "allowed" to change its attitude once every five days. The hyperbolic excess energy of the reference trajectory has been removed, since it was considered to be counter-productive for transfer time minimization (see also section 6.2.3). The final accuracy limit was set to $\Delta r_{f,\max} = 300\,000\,\mathrm{km}$ and $\Delta v_{f,\max} = 100\,\mathrm{m/s}$.

The best found InTrance-trajectory (figure 6.10.1) is 135 days faster (9.0%) than the reference trajectory, reducing at the same time the C_3-requirement from $4\,\mathrm{km^2/s^2}$ to $0\,\mathrm{km^2/s^2}$ and thus permitting a reduction of the launcher requirements and eventually of launch costs. The final distance to 1996FG$_3$ is $\Delta r_f \approx 200\,000\,\mathrm{km}$ and the final relative velocity to 1996FG$_3$ is



6.10.1: Best InTrance-trajectory



6.10.2: Steering angles

Figure 6.10 1996FG$_3$ rendezvous (reference launch date)

$\Delta v_f \approx 65\,\mathrm{m/s}$, both being well better than the required values.

6.2.2 Optimization of the Launch Date

To find the optimal launch date for the 1996FG$_3$ rendezvous, InTrance was used to determine the shortest orbit transfer. Having run InTrance (five times), the shortest found orbit transfer takes $T = 1415\,\mathrm{days}$ to reach the orbit of 1996FG$_3$. For this transfer, the solar sailcraft's angular position at launch is $\varphi_{\mathrm{SC}}(\bar{t}_0) \doteq 0.38$ and at arrival $\varphi_{\mathrm{SC}}(\bar{t}_0+T) \doteq 1.00$. By scanning the planetary positions, it can be found that within a 1 year-interval around the reference launch date (MJD 53960.5), the constellation of Earth and 1996FG$_3$ is most similar to that of the optimal orbit transfer solution for a launch at 29 Oct 05 (where $\varphi_{\mathrm{Earth}}(\mathrm{MJD}\ 53672.5) = 0.63$ and $\varphi_{1996\mathrm{FG}_3}(\mathrm{MJD}\ 55087.5) = 0.98$).

To find the optimal 1996FG$_3$ rendezvous, InTrance was run (five times) for the launch date that was expected to be optimal (MJD 53672.5). However, to allow the steering strategy to compensate for the small difference in Earth's and 1996FG$_3$'s angular position, the launch date was not prescribed exactly, but the EA was allowed to choose the launch date from the interval [MJD 53658.5, MJD 53672.5]. The maximum transfer time was set to $T_{\max} = 1600\,\mathrm{days}$ (with $\tau = 320$).

Figures 6.11 and 6.12 show the results of the five InTrance-runs. Although the variation of the sail steering angles differs considerably for the five steering strategies (see figure 6.12.2), the respective trajectories differ only little in transfer time (figure 6.11), which may indicate some robustness concerning the required sail steering accuracy for a real mission. The worst found trajectory ($T = 1460\,\mathrm{days}$) takes only 1.7% longer to rendezvous 1996FG$_3$ than the best one, so that there can be some confidence that the best found trajectory (figure 6.12.1) is close to the global optimum. Taking 1435 days to rendezvous 1996FG$_3$, it is now 205 days (14%) faster than the reference trajectory. The final distance to 1996FG$_3$ is $\Delta r_f \approx 267\,000\,\mathrm{km}$ and the final relative velocity to Mercury is $\Delta v_f \approx 89\,\mathrm{m/s}$, both being better than the required

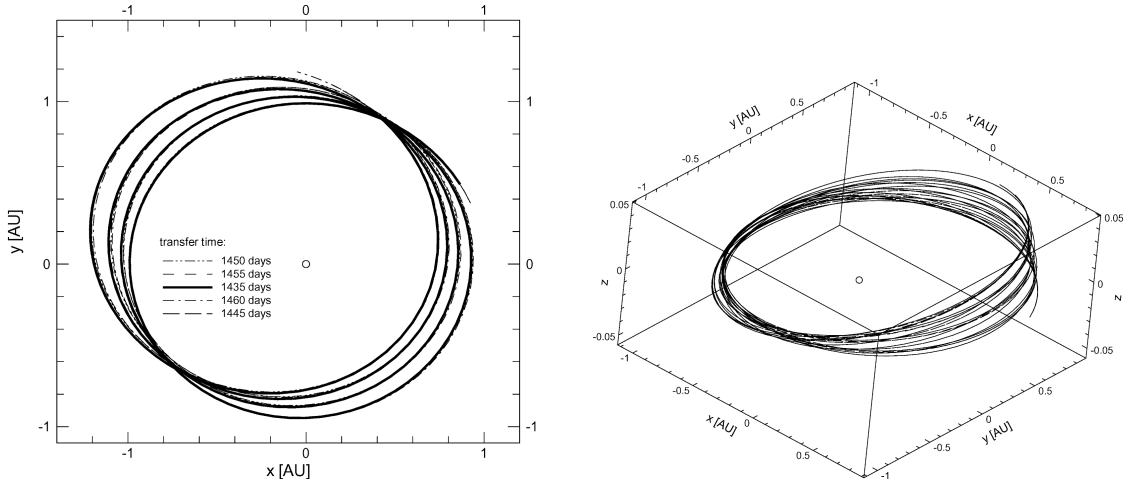
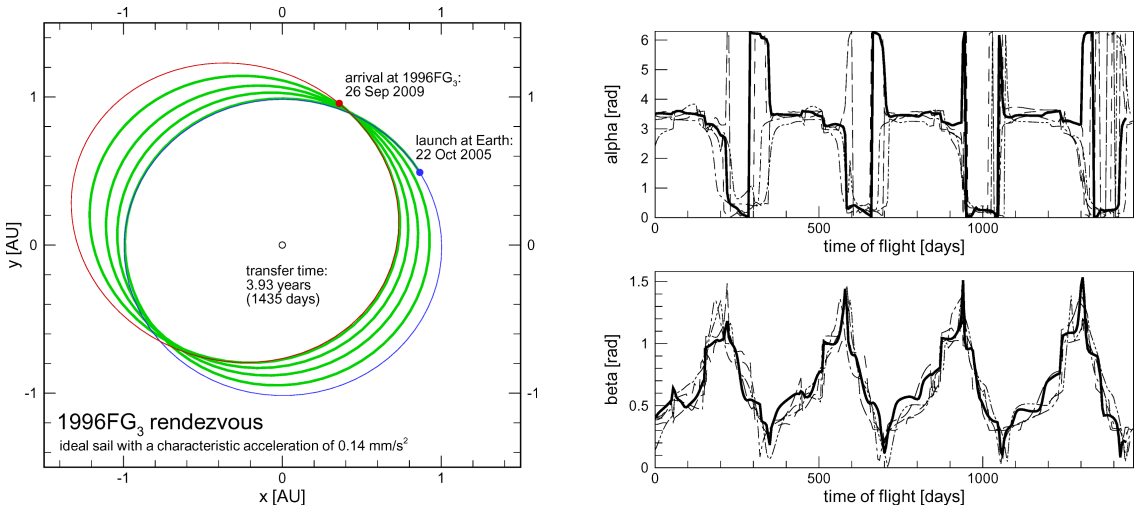


Figure 6.11 1996FG$_3$ rendezvous trajectories for five different initial NC populations (optimized launch date)



6.12.1: Best InTrance-trajectory

6.12.2: Steering angles

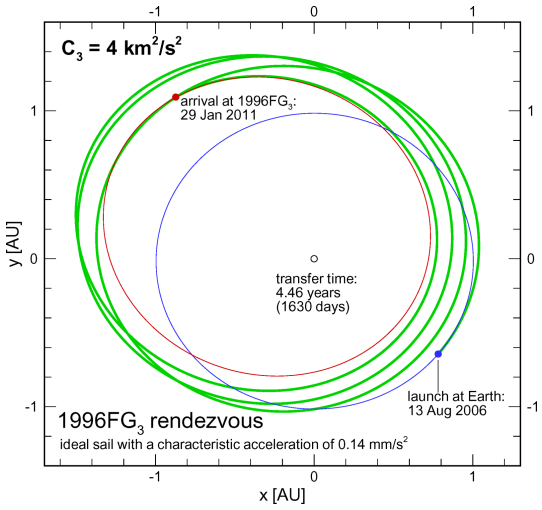
Figure 6.12 1996FG$_3$ rendezvous (optimized launch date)

values. The optimal launch date was found to be 22 Oct 05, 295 days earlier than the reference launch date. It is to note that the optimal transfer time to rendezvous 1996FG$_3$ is slightly worse than the previously found "optimal" orbit transfer time of 1415 days. This might be attributed to the slightly different constellation of the bodies at launch and arrival.

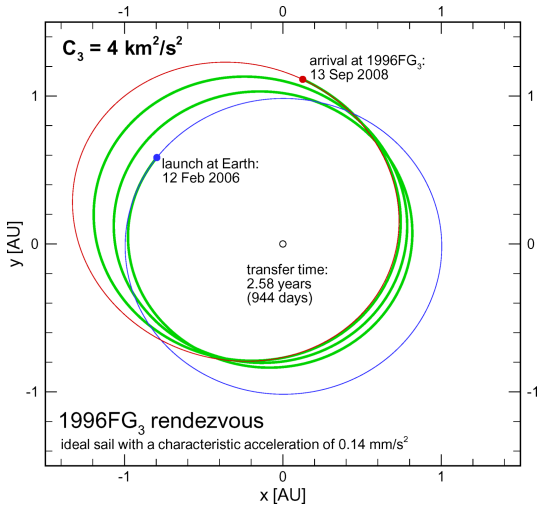
6.2.3 Interplanetary Insertion with Hyperbolic Excess Energy

For the hyperbolic excess energy of the reference trajectory ($C_3 = 4\,\mathrm{km}^2/\mathrm{s}^2$), the best found InTrance-trajectory yields a minimum transfer time of 1630 days for the reference launch date (figure 6.13.1), which is only 10 days faster than the reference trajectory. Since the

optimal direction of the hyperbolic excess velocity vector $\mathbf{v}_\infty$ is not evident, it was encoded additionally on the chromosome, leaving it to the EA to co-evolve it with the NC.



6.13.1: Reference launch date



6.13.2: Optimized launch date

Figure 6.13 1996FG$_3$ rendezvous with $C_3 = 4\,\mathrm{km}^2/\mathrm{s}^2$

Figure 6.13 shows that it is really counter-productive (although optimal for this launch date) to raise the aphelion of the trajectory immediately at launch, because the solar sailcraft is "thrown" far away from the sun, where it is not able to produce reasonable thrust.

To find out whether an hyperbolic excess energy of $4\,\mathrm{km}^2/\mathrm{s}^2$ could be spent more efficiently, InTrance was run to find the optimal orbit transfer. A constellation, which is similar to the constellation of the optimal orbit transfer was found for a launch at 12 Feb 06, a half year earlier than the reference launch date. Figure 6.13.2 shows the best found trajectory for this launch date. Applying also $C_3 = 4\,\mathrm{km}^2/\mathrm{s}^2$, this trajectory takes only 944 days to rendezvous 1996FG$_3$, being 696 days (74%) faster than the reference trajectory. Here, the solar sailcraft is inserted into an interplanetary trajectory with an hyperbolic excess velocity vector in the reverse direction of Earth's velocity vector. This "throws" the solar sailcraft to the sun, where it is able to produce reasonable thrust.

6.3 Fast Pluto Fly-By Mission

A fast Pluto fly-by trajectory for an advanced ideal solar sailcraft with a characteristic acceleration of $a_c = 1.0\,\mathrm{mm/s}^2$, calculated by Leipold [51, 52] using a LTOM, has been chosen as the third test case to assess the performance of InTrance. The reference trajectory launches at Earth at 05 Feb 03 (MJD 52675.5) and takes 13.15 years (4805 days) to reach Pluto, if the solar sailcraft is inserted directly into an interplanetary trajectory with zero hyperbolic excess energy. Since for trajectories that employ a solar photonic assist maneuver, the transfer time does not only depend on the characteristic acceleration of the solar sailcraft but also on the minimal solar distance $r_{\min}$ (see section 6.3.2), the respective value of the reference trajectory ($r_{\min} = 0.49\,\mathrm{AU}$) has been chosen to allow a fair comparison.

6.3.1 Convergence Behavior and Stability

InTrance was run five times – using different initial NC populations – for the same launch date as in [51, 52]. For the calculations within this section, a (cp) 12–30–3 (d) neurocontroller has been used. The maximum transfer time was set to $T_{\max} = 5000$ days. For discretization, this time interval was cut into $\tau = 2000$ finite elements, so that the solar sailcraft is "allowed" to change its attitude once every 2.5 days.

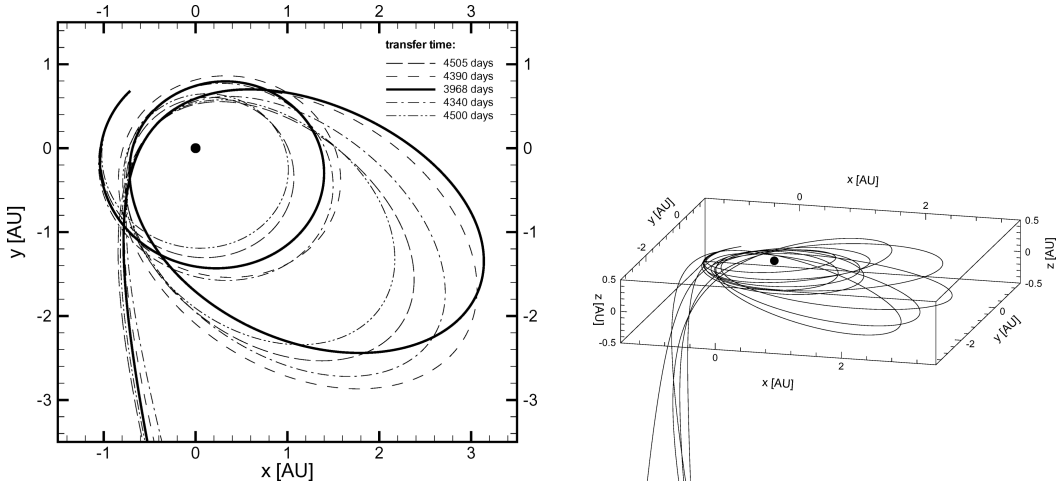
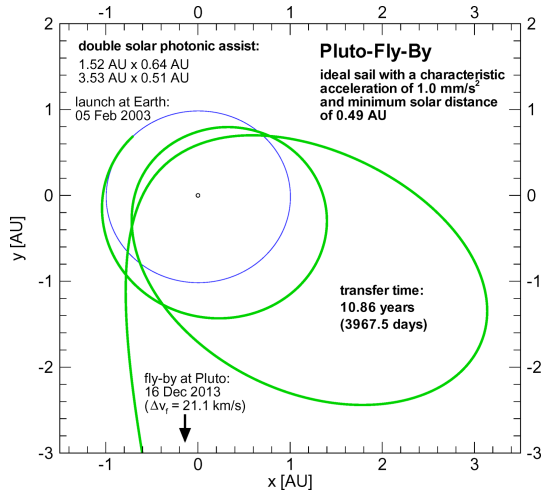
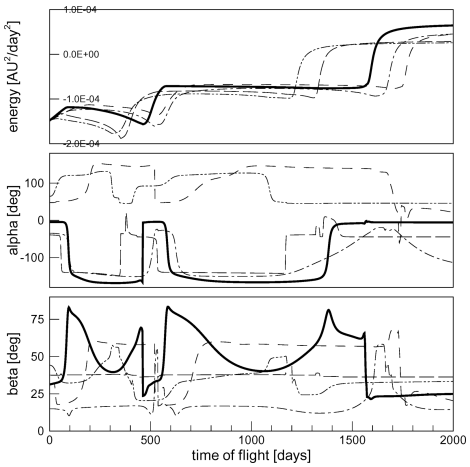


Figure 6.14 Fast Pluto fly-by trajectories for five different initial NC populations ($r_{\min} = 0.49$ AU)



6.15.1: Best InTrance-trajectory



6.15.2: Total (kinetic+potential) energy and steering angles

Figure 6.15 Fast Pluto fly-by using a double solar photonic assist maneuver ($r_{\min} = 0.49$ AU)

Figures 6.14 and 6.15 show that the variation of the steering angles and the trajectories differ very much for the five InTrance-runs, so that one can not be confident that the best found trajectory is very close to the global optimum. However, all five trajectories have much shorter transfer times than the reference trajectory, since they employ a double solar photonic assist,

whereas the reference trajectory performs only a single one. The best InTrance-trajectory (figure 6.15.1) is 837.5 days (21%) faster than the reference trajectory, revealing that the latter is again far from the global optimum. The final distance from Pluto is $\Delta r_f \approx 52\,000\,\mathrm{km}$, being well better than the required value of $\Delta r_{f,\max} = 1\,000\,000\,\mathrm{km}$, and the fly-by velocity at Pluto is $\Delta v_f = 21.1\,\mathrm{km/s}$.

6.3.2 Close Solar Approach

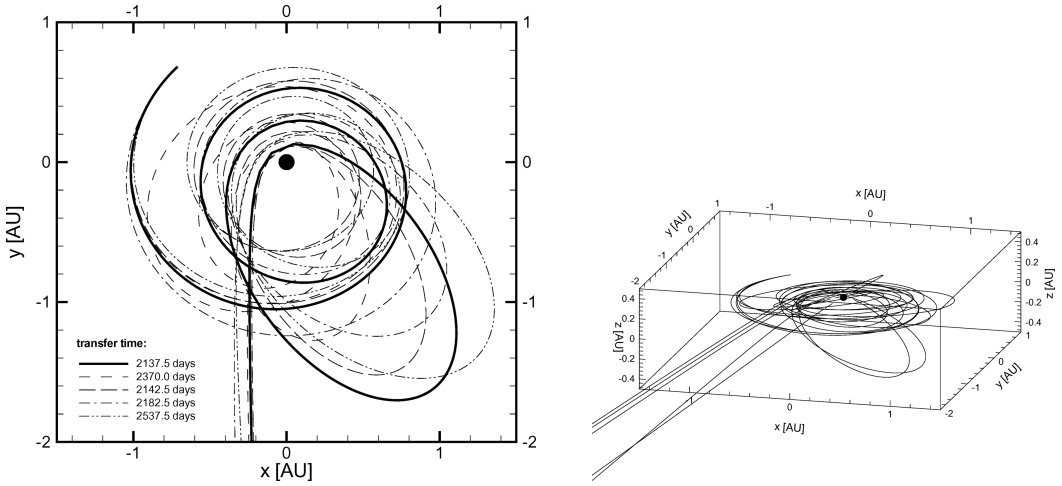
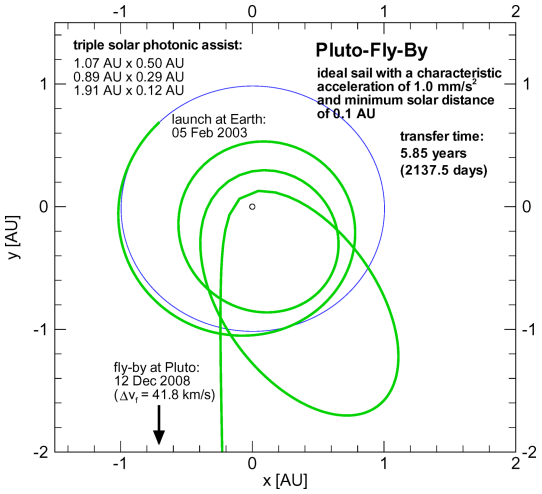
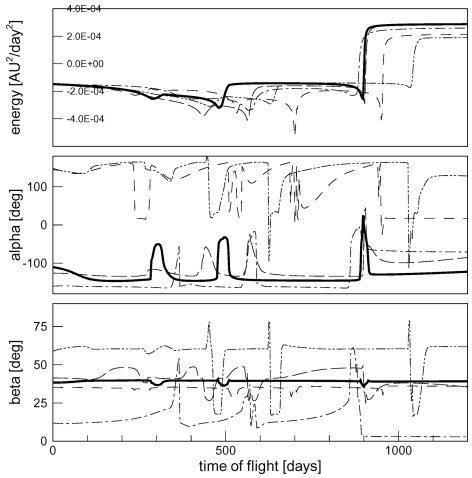


Figure 6.16 Fast Pluto fly-by trajectories for five different initial NC populations ($r_{\min} = 0.10\,\mathrm{AU}$)



6.17.1: Best InTrance-trajectory



6.17.2: Total (kinetic+potential) energy and steering angles

Figure 6.17 Fast Pluto fly-by using a triple solar photonic assist maneuver ($r_{\min} = 0.10\,\mathrm{AU}$)

If the solar sailcraft is allowed to approach the sun as close as $r_{\min} = 0.1\,\mathrm{AU}$, InTrance even finds a triple solar photonic assist trajectory, which approximately doubles the solar sailcraft's solar system escape velocity, so that the fly-by velocity at Pluto is $\Delta v_f = 41.8\,\mathrm{km/s}$

($\Delta r_f \approx 450\,\mathrm{km}$). The transfer time is considerably reduced to 5.85 years (2137.5 days)[12]. Again, the steering angles (figure 6.17.1) and the trajectories (figure 6.16) differ very much, so that one can not be not confident that the best found trajectory is very close to the global optimum.

6.4 Minimal Orbit Transfer Times for Ideal Solar Sailcraft

As the last benchmark for the evaluation of InTrance performance for solar sailcraft trajectory optimization, minimal transfer times to various solar system bodies[13] have been calculated. Sauer gives minimal transfer times to Mercury, Venus, and Mars for ideal high-performance sailcraft with $0.5\,\mathrm{mm/s^2} \lesssim a_c \lesssim 2.5\,\mathrm{mm/s^2}$ (figure 6.18) [80]. To verify the transfer times that

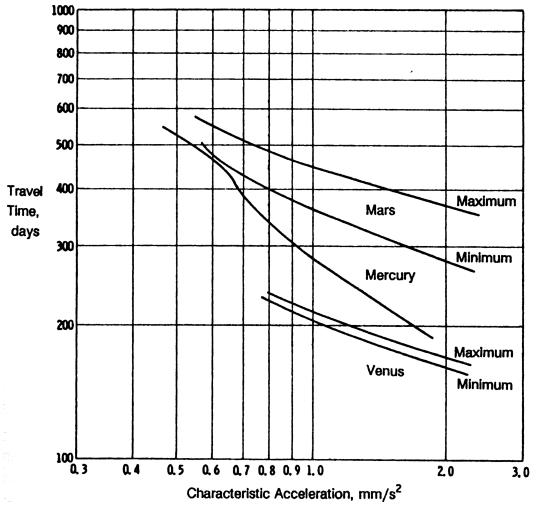


Figure 6.18 Upper and lower bound for minimal transfer times to Mercury, Venus, and Mars that are given by Sauer [80] (only lower bound for Mercury, diagram taken from Wright [98])

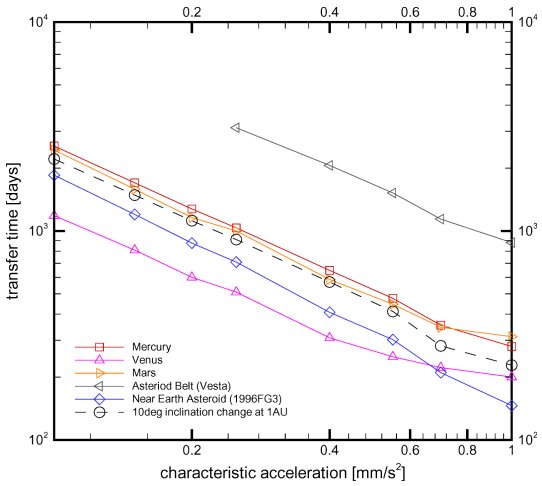
are given by Sauer and to extend them to solar sailcraft of moderate[14] ($a_c \lesssim 0.5\,\mathrm{mm/s^2}$) and extremely high ($a_c \gtrsim 2.5\,\mathrm{mm/s^2}$) performance, minimal orbit transfer times to various targets have been calculated for characteristic accelerations in the range $0.1\,\mathrm{mm/s^2} \leq a_c \leq 10.0\,\mathrm{mm/s^2}$. The InTrance-results are presented in figure 6.19. They are consistent with those in figure 6.18.[15] The results reveal that the transfer times can be described for low

[12] However, for such a close solar approach the thermal load on the sail according to equation (2.9) would be significantly higher ($T_{\max}(r = 0.1\,\mathrm{AU}) \approx 730\,\mathrm{K}$ for an Al|Cr-coated sail), exceeding the temperature limit of conventional polyimide substrates like Kapton®

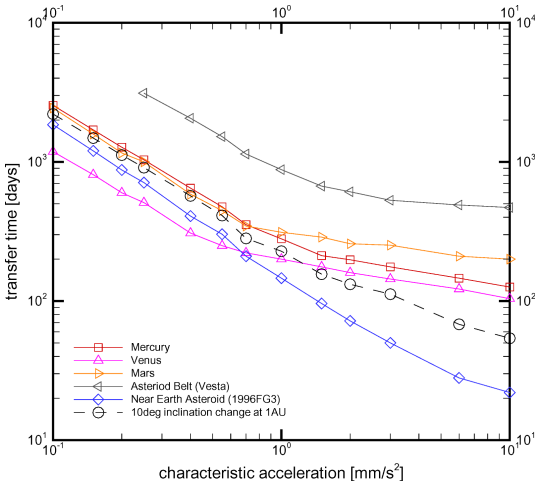
[13] and for a 10° inclination change at 1 AU as an additional test case

[14] Figure 6.18 gives evidence that solar sailcraft trajectories are typically presented for high-performance sailcraft. This limitation is mainly caused by the drawbacks of traditional local trajectory optimizers: trajectories for moderate-performance sailcraft require multiple revolutions around the sun to reach the target body/orbit. For such trajectories, an adequate initial guess is very hard to find.

[15] It is to note that the the results are neither valid for a given launch date nor are they average values. Since no rendezvous with the target body but only with its orbit was required, they represent the lower bound for the transfer time. However, they are valid rendezvous trajectories for a single constellation of the initial and the target body.



6.19.1: $0.1\,\mathrm{mm/s^2} \leq a_c \leq 1.0\,\mathrm{mm/s^2}$



6.19.2: $0.1\,\mathrm{mm/s^2} \leq a_c \leq 10.0\,\mathrm{mm/s^2}$

Figure 6.19 Minimum orbit transfer times for ideal solar sailcraft

characteristic accelerations with a very simple approximation function of the form

$$\frac{T_{\min}}{1\,\mathrm{day}} = \frac{c_1}{(\frac{a_c}{1\,\mathrm{mm/s^2}})^{c_2}}$$

e.g.

$$\frac{T_{\min,\mathrm{Mercury}}}{1\,\mathrm{day}} = \frac{255}{\frac{a_c}{1\,\mathrm{mm/s^2}}}$$

This approximation gives for $0.1\,\mathrm{mm/s^2} \leq a_c \leq 0.75\,\mathrm{mm/s^2}$ a maximum error of 4.2%. For $1996\mathrm{FG}_3$,

$$\frac{T_{\min,1996\mathrm{FG}_3}}{1\,\mathrm{day}} = \frac{146}{(\frac{a_c}{1\,\mathrm{mm/s^2}})^{1.127}}$$

gives for $0.1\,\mathrm{mm/s^2} \leq a_c \leq 1.0\,\mathrm{mm/s^2}$ a maximum error of 5.5%. The coefficients c_1 and c_2 depend on the target body. It can be speculated that they are a function of the initial and the target body's orbital elements (quod esset demonstrandum).

6.5 Multiple Near-Earth Asteroid Rendezvous Using SEP

The assessment of InTrance trajectory optimization performance for EP spacecraft is more difficult than for solar sailcraft, since suitable reference trajectories are hard to find in the literature. This is due to the fact that the complete EP system models are typically not given. The most suitable set of trajectories that was found for reference is for a proposed multiple NEA sample return mission[16], called "Hera", which employs a spacecraft with a cluster of

[16] mission proposal to NASA under the lead of the Arkansas-Oklahoma Center for Space and Planetary Science

three NSTAR thrusters [4] to return samples from three different NEAs to Earth within a mission duration of 4.83 years. However, only a few technical data (table 6.7) as well as the launch and arrival dates at the different target bodies, and the consumed propellant for the four transfers could be obtained for this mission (table 6.8).

Main propulsion system	3 NSTAR thrusters
Power generation by GaAs solar panels (at 1 AU)	6.0 kW
Maximum input power per PPU	2.5 kW
Exponent for variation of power with solar distance ($\varkappa$)	~ 1.7
Dry mass	676.3 kg
Propellant mass (Xenon)	365.8 kg
Hydrazine propellant mass for proximity operations	3×20.0 kg
Launch vehicle	Delta 7925-10

Table 6.7 Technical data for the Hera mission (Baseline-V1.02) [4]

Transfer time	Consumed propellant	Arrival date	Body	Stay time	Launch date
			Earth		15 Jan 06
200 days	52.6 kg	03 Aug 06	$1999AO_{10}$	99.0 days	10 Nov 06
750 days	209.1 kg	29 Nov 08	$2000AG_6$	98.2 days	07 Mar 09
230 days	80.8 kg	23 Oct 09	1989UQ	205.5 days	17 May 10
180 days	23.3 kg	13 Nov 10	Earth		

Table 6.8 Hera mission parameters [4]

Nevertheless, a re-calculation of the mission using InTrance is possible. The hyperbolic excess energy for interplanetary injection was not given for the first leg of the reference trajectory. However, the launch vehicle is the same as for the Mars Global Surveyor (MGS) mission and the launch mass is also similar (1102 kg instead of 1060 kg). For the interplanetary orbit insertion of the MGS spacecraft, the launcher provided an hyperbolic excess energy of about $10\,\mathrm{km^2/s^2}$ [72]. Therefore, the same value was taken as the maximum C_3 for the InTrance-calculations. However, InTrance, which was allowed to choose C_3 from the interval $[0\,\mathrm{km^2/s^2}, 10\,\mathrm{km^2/s^2}]$, found $C_3 = 4.1\,\mathrm{km^2/s^2}$ to be optimal. The primary goal of the InTrance-calculations was to minimize the consumed propellant.[17] For this purpose, the original Hera transfer times have been taken as $T_{\max}$, and the trajectories have been optimized with respect to minimal propellant consumption. For all calculations, a (cp) 25–30–4 (d) neurocontroller has been used.

Figure 6.20 shows the best found InTrance-trajectories. The thin trajectory segments denote coast arcs, where the thrust is turned off. Table 6.9 shows that the InTrance-trajectories consume about 10% less propellant for all legs.[18]

[17] However, trajectories with shorter transfer times could have been found by InTrance, if they had been also optimal with respect to propellant consumption (this was not the case).

[18] The results are very accurate with respect to the terminal constraint. The accuracy of the reference

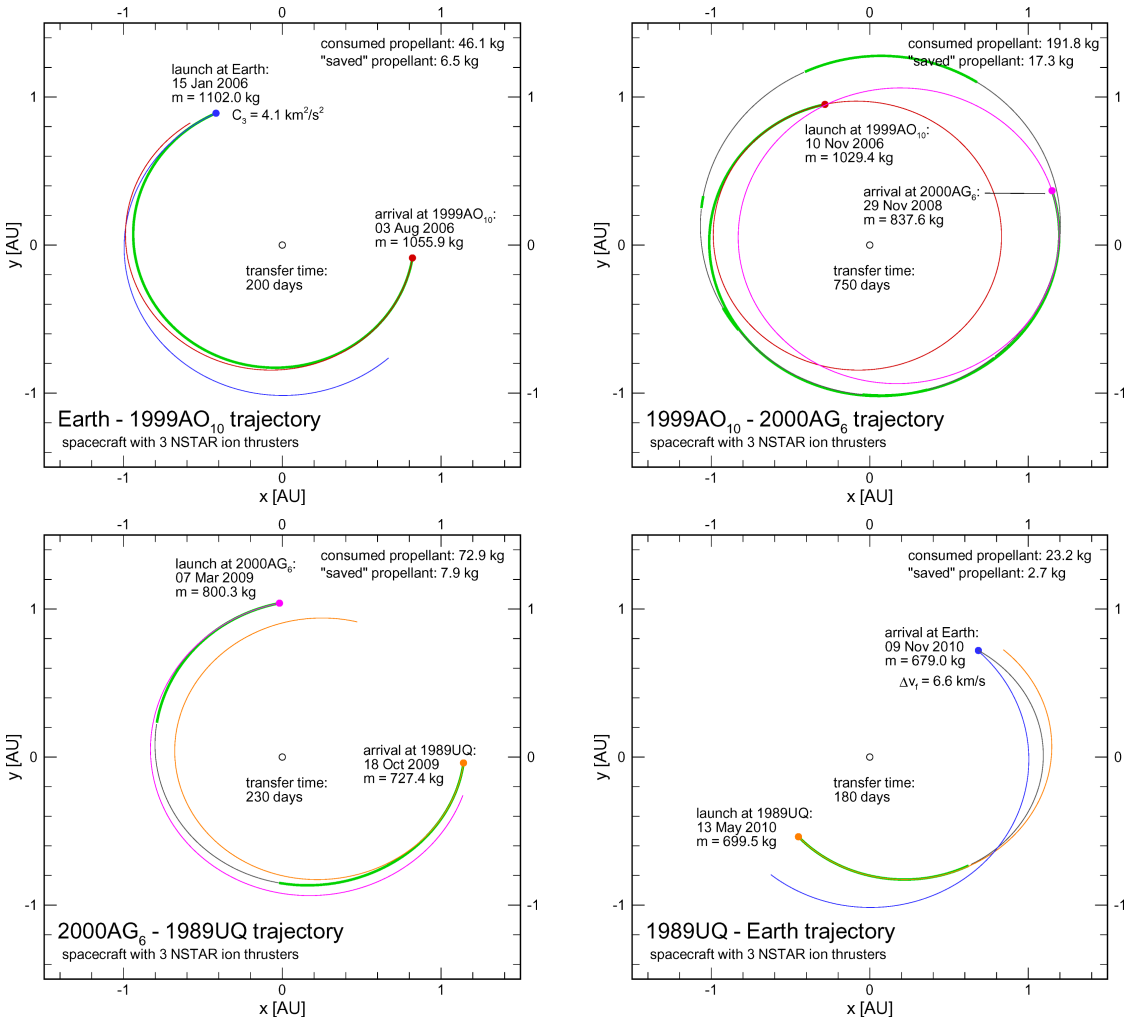


Figure 6.20 Re-calculated InTrance-trajectories for the Hera mission

Trajectory leg	Consumed propellant Reference [kg]	Consumed propellant InTrance [kg]	Saved propellant [kg]	"Saved ΔV" [m/s]	Trajectory accuracy Δr_f $[10^3\text{km}]$	Trajectory accuracy Δv_f [m/s]
Earth-1999AO$_{10}$	52.6	46.1	6.5	182	10	10
1999AO$_{10}$-2000AG$_6$	209.1	191.8	17.3	615	45	45
2000AG$_6$-1989UQ	80.8	72.9	7.9	320	4	4
1989UQ-Earth	23.2	20.5	2.7	118		

Table 6.9 Comparison of the Hera reference trajectories with the best found InTrance-trajectories

trajectories is not given in [4]. Nevertheless, the propellant savings can not be attributed to a poorer accuracy of the InTrance-trajectories, since the "saved ΔV" (the ΔV that the saved propellant may provide, assuming a conservative $I_{sp} = 3000\,s$) is much larger than Δv_f.

7 Mission Analysis Using InTrance

Trajectory analysis and design is a crucial part of the feasibility analysis of any space mission, since it confines the mission objectives and sets the performance requirements for the propulsion system. Those requirements determine, whether or not – using a given propulsion system – the mission is feasible. InTrance is intended as a tool to support mission analysis.

Within this chapter, the suitability of InTrance as a mission analysis tool is demonstrated by performing trajectory analyses for some relevant interplanetary space missions. Those analyses have three major goals:

1. to assess the performance of solar sail propulsion with respect to chemical and solar electric propulsion for interplanetary missions. This is done exemplary for missions to near-Earth asteroids (NEAs, section 7.2) and to Mercury (section 7.3).

2. to demonstrate that for solar sailcraft mission analysis the non-perfect reflectivity of the solar sail must be considered through an appropriate SRP force model like the standard SRP force model according to section 2.2.3.4 (section 7.1). The common simplification that the non-perfect reflectivity of the sail can be taken into account by using an overall efficiency factor should only be made for *very* preliminary mission feasibility analysis. Consequently, the standard SRP force model has been used for all calculations within this chapter.

3. to demonstrate that InTrance can also be used to support mission analysis for low-thrust propulsion systems other than solar sails. This is done exemplary for a piloted Mars mission (section 7.4).

7.1 Minimal Orbit Transfer Times for Non-Ideal Solar Sailcraft

Solar sailcraft trajectory/mission analyses usually employ the ideal or the simplified SRP force model (see section 2.2.3.3 and 2.2.3.5).[1] This is mainly due to the fact that, using the standard SRP force model (see section 2.2.3.4), it is difficult to conceive an adequate initial guess for the variation of the sail steering angles, since the LSLs give only the locally optimal thrust direction, from which the corresponding sail attitude can not be obtained analytically. From the trajectory analysts point of view, the ideal and the simplified SRP force models are equivalent, since the shape of both SRP force "bubbles" is identical (see section 2.2.3.6). Therefore, a lower sail efficiency η can always be offset with a proportionally larger sail area, $A_{\text{simplified}} = A_{\text{ideal}}/\eta$, so that $a_{c,\text{simplified}} = a_{c,\text{ideal}}$. Thus, from the perspective of trajectory analysis, both SRP force models describe a perfectly reflecting or ideal solar sailcraft. This equivalency is not the case for the standard SRP force model, even if $a_{c,\text{standard}} = a_{c,\text{ideal}}$ is chosen, since the shape of the SRP force "bubbles" is different.

[1] The only known calculations for the standard SRP force model have been done in [19], where a simple Earth-Venus-transfer and a simple Earth-Mars-transfer were calculated using a local trajectory optimization method (direct collocation method).

Within this section, the minimal orbit transfer times for solar sailcraft are calculated using the more realistic standard SRP force model (non-perfectly reflecting or non-ideal solar sailcraft). The results are compared to the transfer times that have been obtained within section 6.4 for perfectly reflecting (ideal) solar sailcraft, using the ideal/simplified SRP force model. For all calculations, the *size* of the "bubbles" was the same ($a_{c,\text{standard}} = a_{c,\text{ideal}}$).

Figure 7.1 shows a comparison of the minimal transfer times to various solar system bodies for ideal and for non-ideal solar sailcraft.

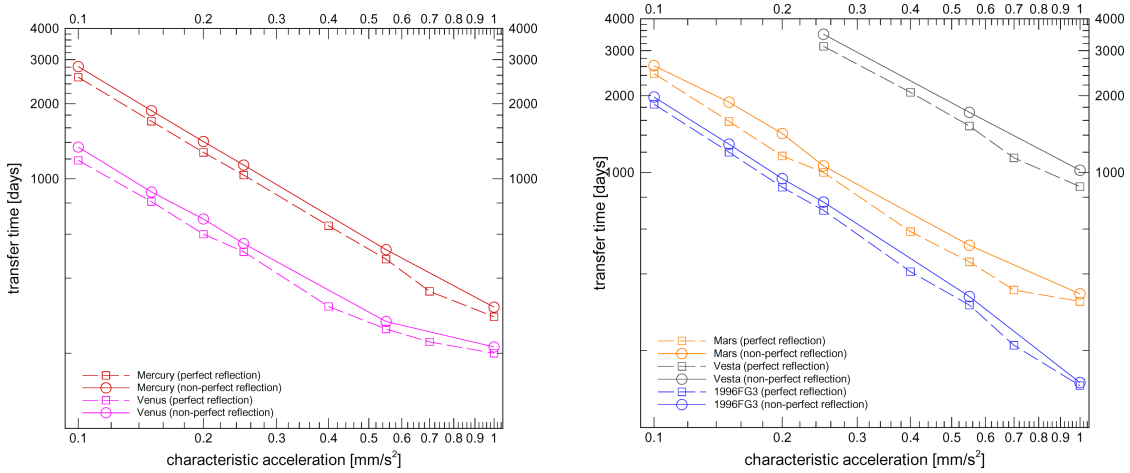


Figure 7.1 Comparison of minimum orbit transfer times for ideal and non-ideal solar sailcraft

As the results show, there is a considerable increase (approx. $5 - 15\%$[2]) in minimal orbit transfer time, if the standard SRP force model is used. One possible explanation for these results is that InTrance has more difficulties in finding trajectories for non-ideal solar sails, and thus fails to find globally optimal solutions. Although a part of the differences might be attributed to this cause, this explanation is unlikely to explain the entire differences. The results are in accordance with the result in [19], where an increase of 7.8% in transfer time had been obtained for a simple Earth-Mars-transfer ($a_c \doteq 1.5\,\text{mm/s}^2$). For a simple Earth-Venus-transfer, an increase of even 24% in transfer time (306 days for $a_c \doteq 0.55\,\text{mm/s}^2$) had been obtained in [19], which suggests that the trajectory is far from the global optimum (InTrance yields a minimum transfer time of 268 days for exactly the same problem).

The above results demonstrate that for a thorough mission analysis the non-perfect reflectivity of the solar sail must be considered through an appropriate SRP force model. The simplification, that the non-perfect reflectivity of the sail can be taken into account by using an overall efficiency factor η, should only be made for *very* preliminary mission feasibility analyses. Consequently, the standard SRP force model has been used for all calculations within this chapter.

[2] being larger for trajectories that require large sail cone angles, where the difference between the ideal and the non-ideal bubble is larger (see figure 2.10)

7.2 Near-Earth Asteroid Rendezvous and Sample Return Missions

NEAs are a very promising category of target bodies for the first solar sail mission(s), since they are of great scientific relevance and can be accessed relatively easily. Within this section, some missions to NEAs (rendezvous, multiple rendezvous, and sample return) are investigated, comparing solar sail propulsion with solar electric propulsion.

7.2.1 Mission Objectives

Comets and most of the asteroids are in some sense the fossils of the solar system [21]. They vary highly in size, surface properties, composition, and probably origin. Especially the undifferentiated primitive carbonaceous (C type) asteroids are expected to hold key information for understanding the origin of the solar system and the formation of the planets, since they are – unlike the planets – primitive bodies that have undergone little physical and chemical alteration, and thus represent most closely the properties of the primordial solar nebula. Therefore, they have a very high exploration priority [47]. C type asteroids are the largest taxonomic class ($\sim 40\%$), but most of them are located in the outer asteroid belt beyond 2.7 AU, where they are not easily accessible [31].

Although a large amount of asteroid samples exists on Earth as meteorites, many questions are yet unresolved. One of the most important unresolved questions is the linkage between asteroid classes and meteorite classes. Many asteroid classes have reflectance spectra that are similar to those of meteorites. For example, C type asteroids are supposed to be linked to chondritic meteorites due to their similar spectra, indicating carbon-rich material. However, some asteroid types have no meteoritic analogue in their spectra, like the P and D type asteroids, which are supposed to be even more primitive than the C type asteroids and to carry organic compounds [31]. It is also curious that no large asteroid class seems to match the meteorite spectra of ordinary chondrites, the most common meteoritic samples [31].

Near-Earth objects (NEOs) are asteroids (NEAs) and short-period comets with orbits that intersect or pass near the orbit of Earth. About 650 NEOs with diameters $\gtrsim 1$ km are currently (May 03) known [5], but the entire population contains perhaps more than 1 000 objects of this size [21]. They pose a significant hazard to human civilization and to life on Earth. It is today widely accepted that NEO impacts have caused at least one mass extinction (65 million years ago at the Cretaceous/Tertiary boundary), and they are suspected to have caused several global catastrophes before [93]. Even NEOs that do not intersect the orbit of Earth may evolve into Earth-crossers, since their orbits are chaotic, having a relatively short dynamical lifetime ($\sim 10^7$ years) [39][31]. One day, it might become necessary to prevent a specific object from impacting the Earth by nudging it out of its orbit. To be able to do this, the bulk properties of NEOs (material strengths, composition, structure, moments of inertia, etc.) should be determined as soon as possible [44].

Since NEAs are probably fragments of main belt asteroids, they are believed to be a representative sample of them [47]. Unlike their parent bodies, some NEOs are the most readily accessible extraterrestrial bodies. The energy requirement to rendezvous with some of them is less than to land on the Moon's surface [21]. Since relatively short transfer times to such bodies are expected even for solar sails with very moderate performance, they are ideal targets

for a first technical demonstration of solar sail propulsion, although the true potential and advantage of solar sails over other propulsion systems is expected to become evident only for high-ΔV-missions.

Within this section, the suitability of solar sail propulsion for missions to selected NEAs is assessed and compared to SEP. To allow a better comparison, the target objects of the ENEAS mission [47] and the Hera mission [4] have been adopted (see table 7.1). For a real mission, of course, the target object(s) have to be chosen as to maximize the scientific return on investment.

Asteroid	a [AU]	e	ι [°]	Spectral type
1989UQ	0.915	0.265	1.291	?
$1996FG_3$	1.054	0.350	1.991	C
$1999AO_{10}$	0.912	0.110	2.628	?
$2000AG_6$	1.015	0.190	2.474	?

Table 7.1 Orbital and physical parameters of selected near-Earth asteroids [5, 6]

7.2.2 Near-Earth Asteroid Rendezvous Mission

7.2.2.1 Mission Analysis for Solar Sailcraft

Based on the successful deployment experiment at DLR in December 1999 (see section 2.2.2), a dedicated mission for the exploration of NEAs with solar sailcraft (ENEAS) had been proposed in August 2000 by DLR in cooperation with the Westfälische Wilhelms-Universität at Münster (Germany) as a candidate within the German small satellite program for space sciences [47, 86]. ENEAS is intended to feature a deployable $(50\,\mathrm{m})^2$ solar sail that is capable to transport a micro-satellite with a mass of 75 kg (with a scientific payload of 5 kg, CCD camera + IR spectrometer + magnetometer) to a NEA within less than five years. Table 7.2 summarizes the ENEAS parameters.

Sail area	A	$(50\,\mathrm{m})^2$
Sail assembly mass	m_{SA}	73 kg
Sail assembly loading	σ_{SA}	$29.2\,\mathrm{g/m^2}$
Payload mass (incl. spacecraft bus)	m_{PL}	75 kg
Total sailcraft mass	m	148 kg
Sailcraft loading	σ	$59.2\,\mathrm{g/m^2}$
Characteristic acceleration	a_c	$0.140\,\mathrm{mm/s^2}$
Characteristic SRP force	F_c	20.7 mN

Table 7.2 Parameters for the ENEAS solar sailcraft

$1996FG_3$ had been chosen as the target body for the ENEAS mission, since it has orbital elements not too different from that of Earth and since it is an object of exceptional scientific relevance. Observations [67, 71] indicate that $1996FG_3$ is a binary C type asteroid, consisting of a primary body with a rotation period of $P_1 \approx 3.6$ hours and a secondary body with an

orbital period of $P \approx 16$ hours. Figure 7.2 sketches the physical parameters of the 1996FG$_3$ system. Based on the typical C type albedo of 0.06, the primary body has an estimated diameter of $d_1 \approx 1.4$ km and the secondary body an estimated diameter of $d_2 \approx 0.4$ km. The separation of the two bodies is approx. 1.7 times the diameter of the primary body. The determined average bulk density is $\bar{\rho} = 1.4 \pm 0.3\,\mathrm{g/cm^3}$, which is highly suggestive of a "rubble pile" structure. ENEAS is intended to determine the morphological properties, the compositional properties, and the evolution of the 1996FG$_3$ system [47].

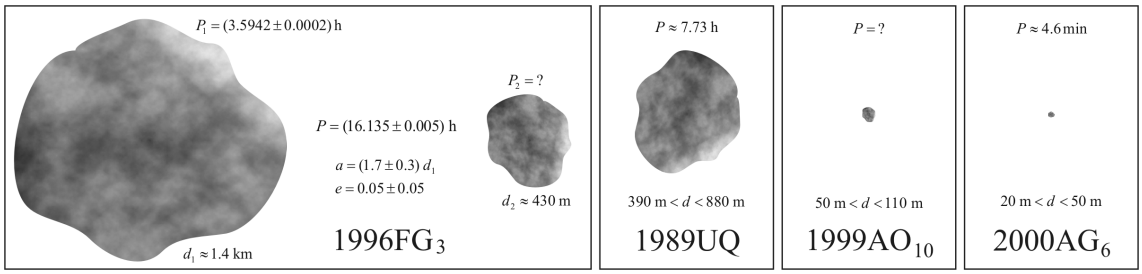
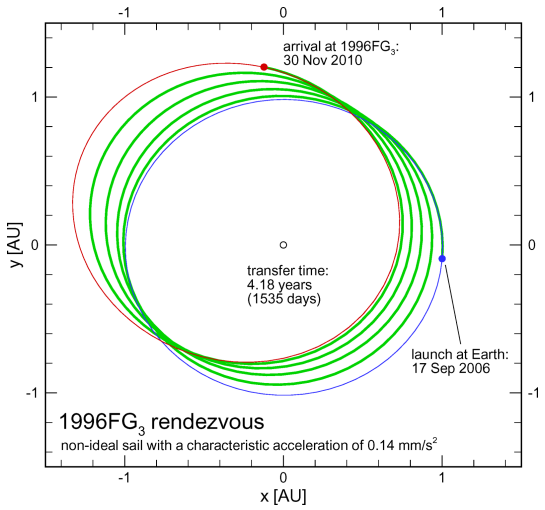


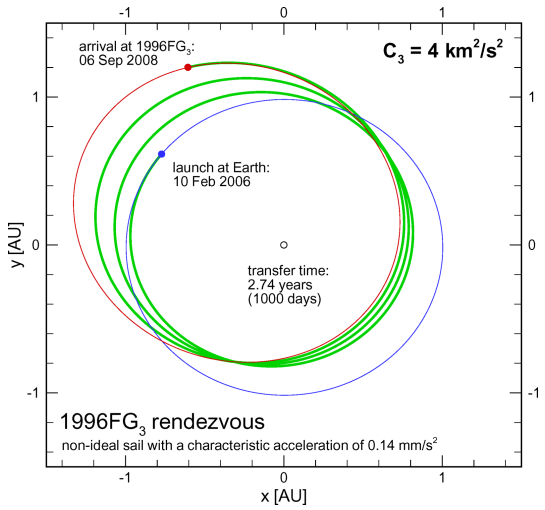
Figure 7.2 Sketch of some physical properties for selected NEAs [5, 6, 31, 71]

Since for solar sailcraft of moderate performance it is difficult and time consuming to gain orbital energy in the Earth's gravitational field, the launcher inserts the ENEAS sailcraft directly into an interplanetary trajectory. After injection, the sail and the attitude control mast are deployed in a 3-axis stabilized mode. Then, the sail is oriented to follow the pre-calculated attitude profile that leads to an optimal interplanetary transfer trajectory. During the transfer, the ENEAS sailcraft runs almost autonomously, so that ground monitoring is carried out on a weekly basis only [86]. The trajectory optimization results in section 6.2 show that, assuming a perfectly reflecting solar sail, the ENEAS sailcraft can reach 1996FG$_3$ within 3.93 years (1435 days), if it is inserted with zero hyperbolic excess energy (see figure 6.12.1) or even within 2.58 years, if it is inserted with an hyperbolic excess energy of $C_3 = 4\,\mathrm{km^2/s^2}$ (figure 6.13.2). InTrance trajectory optimization using the standard SRP force model, which takes into account the non-perfect reflectivity of the ENEAS sail, yields an optimal transfer time of 4.18 years (1535 days) for $C_3 = 0\,\mathrm{km^2/s^2}$ (figure 7.3.1) and 2.74 years (1000 days) for $C_3 = 4\,\mathrm{km^2/s^2}$ (figure 7.3.2), being considerably larger (7% and 6% respectively) than for an ideal solar sail. Thus, it would be very beneficial, if the launcher could provide some hyperbolic excess energy for interplanetary injection, but it is not absolutely necessary to accomplish the mission. For hyperbolic excess velocities larger than about $4\,\mathrm{km^2/s^2}$ the transfer time further decreases, however just slightly (figure 7.4).

For a first solar sail technology demonstration mission in deep space, a target object might be required that is still easier accessible than 1996FG$_3$, to keep operation costs low and to provide as quickly as possible feedback for the tested technologies. Despite its small dimension, the NEA 1999AO$_{10}$ might be a potential target object for such a technology demonstration mission. With $C_3 = 0\,\mathrm{km^2/s^2}$ and $a_c = 0.14\,\mathrm{mm/s^2}$, as InTrance-optimization shows, it could be reached with the ENEAS sailcraft within nearly half of the time (2.14 years / 780 days) that is required for 1996FG$_3$. During the transfer and at the asteroid, critical technologies (attitude control, autonomy, sail aging due to the erosive effects of the space environment, close proximity operations at the asteroid, etc.) could be tested for relatively low cost.



7.3.1: $C_3 = 0\,\mathrm{km}^2/\mathrm{s}^2$



7.3.2: $C_3 = 4\,\mathrm{km}^2/\mathrm{s}^2$

Figure 7.3 ENEAS trajectory options

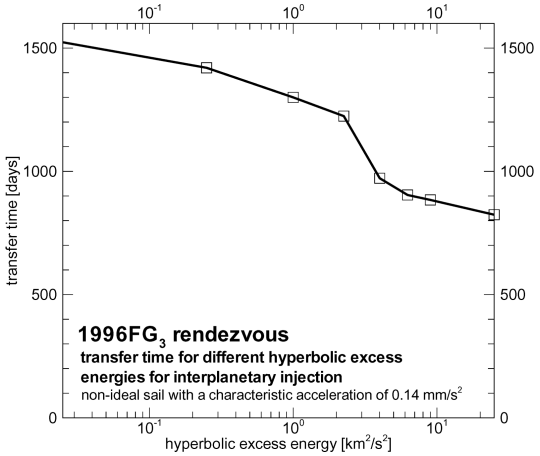


Figure 7.4 Influence of C_3 on ENEAS transfer time

7.2.2.2 Mission Analysis for SEP Spacecraft

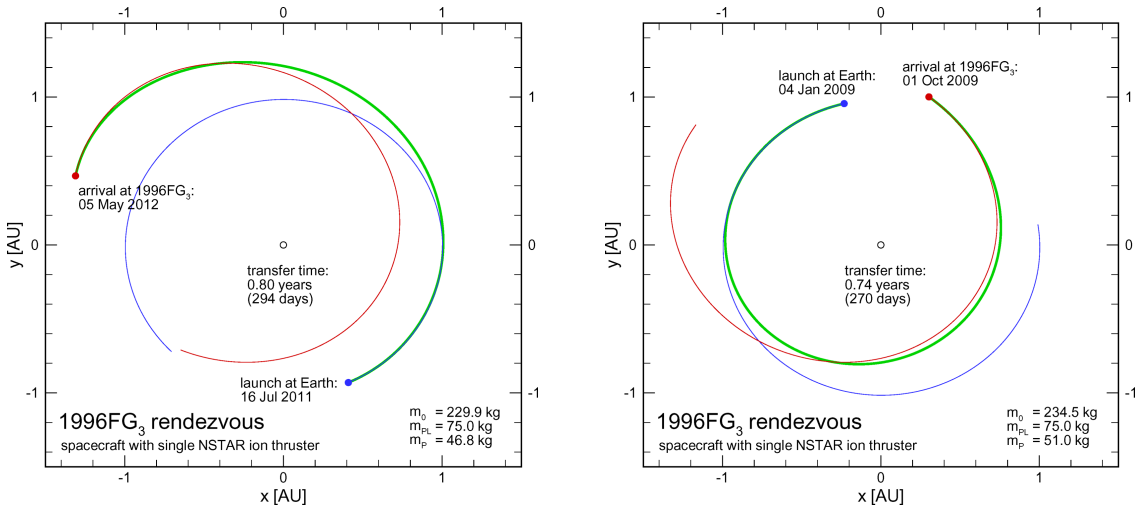
To assess the capability of solar sail propulsion for this NEA rendezvous mission, the relevant mission and system parameters have to be compared with other propulsion systems like SEP systems. Therefore, InTrance has been applied for the calculation of the trajectories for an SEP spacecraft that accomplishes the same mission objective (i.e. deliver a 75 kg payload to 1996FG$_3$). The spacecraft model that is used for comparison, called ENEASEP, employs a single NSTAR thruster for propulsion and two SCARLET solar arrays for power generation. Its parameters are given in table 7.3.

As mentioned in section 3.1.1, SEP spacecraft trajectory optimization is a multi-objective problem, since trajectories may not only be optimized with respect to transfer time but also with respect to the required propellant mass, and usually a trade-off between both op-

Payload mass (incl. spacecraft bus)	m_{PL}	75.0 kg
Ion propulsion system mass	m_{Thr}	48.0 kg
Solar array mass	$2m_{\mathrm{SAW}}$	55.4 kg
Dry mass (without propellant and tank)	m_{dry}	178.4 kg
Maximum thrust	F_{max}	71.3 mN
Total launch mass ($m_{\mathrm{P}} = 51\,\mathrm{kg}$, $m_{\mathrm{Tank}} = 5.1\,\mathrm{kg}$)	m_0	234.5 kg
Maximum acceleration with full tank	a_0	0.304 mm/s^2
Maximum acceleration with (near-)empty tank	a_f	0.400 mm/s^2

Table 7.3 Parameters for the ENEASEP spacecraft

timization objectives has to be made, so that the "optimal" solution is only one of many PARETO-optimal solutions (see section 3.1.1). Figure 7.5 exemplifies two InTrance-solutions for this problem on (or at least close to) the PARETO-optimal front. Thus, using spacecraft with a single NSTAR thruster, the same payload could be delivered to 1996FG$_3$ within 294 days (if $m_{\mathrm{P}} = 46.8\,\mathrm{kg}$) or even within 270 days, if slightly more propellant is consumed ($m_{\mathrm{P}} = 51.0\,\mathrm{kg}$).



7.5.1: $m_{\mathrm{P}} = 46.8\,\mathrm{kg}$ **7.5.2:** $m_{\mathrm{P}} = 51.0\,\mathrm{kg}$

Figure 7.5 ENEASEP trajectory options

7.2.2.3 Comparison of the Solar Sail Option and the SEP Option

The results demonstrate that for this mission, solar sailcraft is clearly outperformed by the SEP option, if only the transfer time is considered. This is not surprising, since the required velocity increment for the orbit transfer is moderate ($\Delta V(m_{\mathrm{P}} = 46.8\,\mathrm{kg}) = 6.7\,\mathrm{km/s}$ and $\Delta V(m_{\mathrm{P}} = 51.0\,\mathrm{kg}) = 7.0\,\mathrm{km/s}$). However, the launch mass of ENEASEP is larger ($m_0 = 229.9\,\mathrm{kg}$ and $m_0 = 234.5\,\mathrm{kg}$ respectively) than for the solar sail option ($m = 148.0\,\mathrm{kg}$), requiring eventually a heavier and thus more expensive launch vehicle. If ground operation costs can be kept low (e.g. due to a high on-board autonomy during transfer), and if the

transfer time plays a subordinate role with respect to cost, the solar sail might be the favorable option for such a NEA rendezvous mission.

7.2.3 Near-Earth Asteroid Sample Return Mission

7.2.3.1 Mission Analysis for Solar Sailcraft

The ENEAS sailcraft is intended to rendezvous 1996FG$_3$ for remote sensing with a minimum scientific payload of 5 kg (CCD camera + IR spectrometer + magnetometer). To study the 1996FG$_3$ system in more detail, it would be necessary to place a lander on the surface of the asteroid (e.g. for mass spectrometry and/or alpha-proton spectrometry). Some investigations (e.g. micro-structure and isotope analysis) to determine the age and the evolution of 1996FG$_3$ could probably only be accomplished by taking samples of the asteroid back to Earth. Due to their high ΔV-capability, solar sailcraft are supposed to be especially capable to perform such sample return missions. However, compared to the ENEAS rendezvous mission, the payload has to be extended considerably, including a lander and a sample return capsule. The key questions for the design of such an ENEAS-SR (sample return) mission are:

Q1: What is the maximum acceptable mission duration $T_{\max}$?

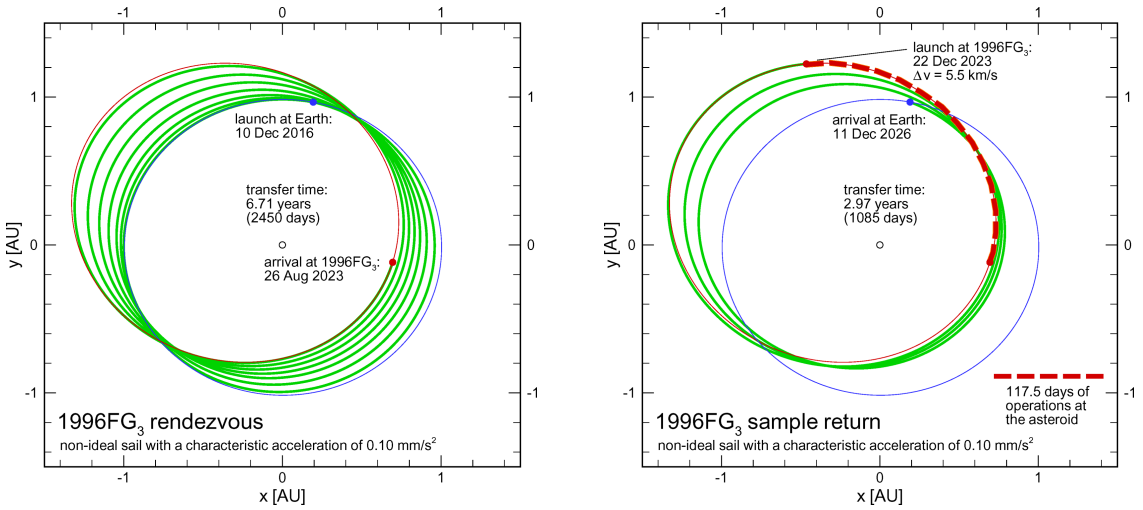
Q2: What is the minimum characteristic acceleration $a_{c,\min}$ to perform the mission in $T_{\max}$?

Q3: What is the expected sail assembly loading σ_{SA} and sail dimension s for the solar sailcraft?

Q4: What is the maximum payload mass to get $a_{c,\min}$ for the specified σ_{SA} and s?

Answer to Q1: At present, the maximum acceptable mission duration seems to be determined by the trip time that is required with chemical propulsion, including (eventually multiple) gravity assist maneuvers. Due to the relatively large ΔV-requirement for a 1996FG$_3$ sample return mission with a chemical propulsion system, which was calculated to be about 8.7 km/s, such a mission would require either heavy spacecraft and thus an expensive launch vehicle, resulting in a short trip time of 1-2 years, or several gravity assists, resulting in a long trip time, similar to the Rosetta mission, which was originally planned to rendezvous comet 46P/Wirtanen with three intermediate gravity assist maneuvers (Mars-Earth-Earth) and a trip time of approx. 9 years. Since the rationale for using solar sailcraft aims at low-cost missions, only the gravity assist option is a reasonable chemical alternative. Thus, for the ENEAS-SR mission, a total mission duration of ten years or less seems to be acceptable.

Answer to Q2: Trajectory calculations using InTrance show that the ENEAS-SR mission to 1996FG$_3$ can be achieved even with a low characteristic acceleration of 0.10 mm/s^2 in exactly 10 years, including a rendezvous trajectory of 6.7 years (2450 days, figure 7.6.1), 117.5 days of operations at the asteroid, and an Earth return trajectory of 3.0 years (1085 days, figure 7.6.2). The Earth return leg is much shorter than the outward leg, since no rendezvous is required at Earth.



7.6.1: Outward trajectory

7.6.2: Return trajectory

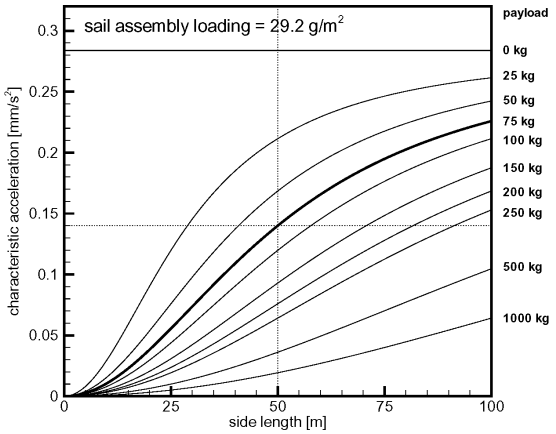
Figure 7.6 ENEAS-SR trajectory option

Answer to Q3: Looking at equation (2.21), the characteristic acceleration of solar sailcraft with a square sail is

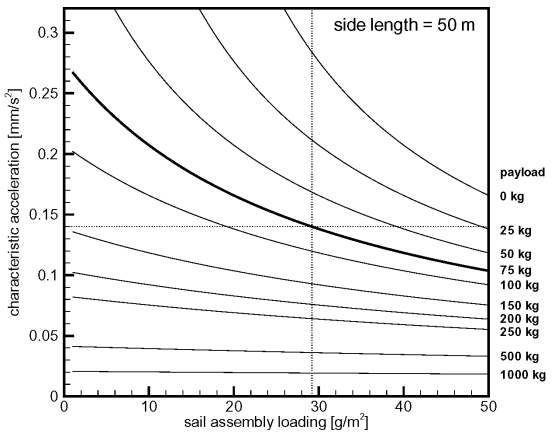
$$a_c = \frac{P_{\text{eff},0}}{\sigma_{\text{SA}} + \frac{m_{\text{PL}}}{s^2}} \tag{7.1}$$

One can see that the performance depends on three design parameters, the sail assembly loading σ_{SA}, the payload mass m_{PL}, and the side length s (or area s^2) of the solar sail, defining a three-dimensional solar sail design space. Diagrams 7.7.1 and 7.7.2 show parametric sections of this design space for a fixed $\sigma_{\text{SA}} = 29.2\,\text{g/m}^2$ and a fixed $s = 50\,\text{m}$ respectively (as for the ENEAS sailcraft). Diagram 7.7.2 shows that for $\sigma_{\text{SA}} = 29.2\,\text{g/m}^2$ a characteristic acceleration of up to $0.284\,\text{mm/s}^2$ can be achieved without any payload. For a smaller characteristic acceleration, a positive payload mass can be accommodated, depending on the sail size. To achieve a characteristic acceleration beyond $0.284\,\text{mm/s}^2$, the sail size has to be enlarged and/or the sail assembly loading has to be reduced. Diagram 7.7.3 shows the required sail size for different sail assembly loadings and payload masses, to obtain a characteristic acceleration of $0.10\,\text{mm/s}^2$. Based on the experiences with the ground-based solar sail technology demonstration, a maximum sail size of $(70\,\text{m})^2$ with a sail assembly mass of 111 kg ($\sigma_{\text{SA}} = 22.7\,\text{g/m}^2$, sail film + booms + deployment module) is considered as a realistic – however still challenging – baseline for the ENEAS-SR mission [84].

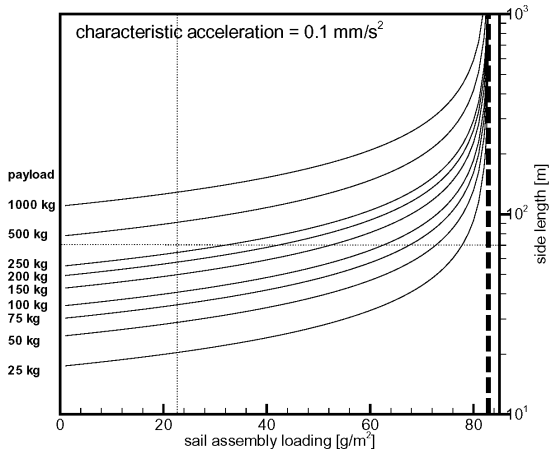
Answer to Q4: The specified σ_{SA} and s yield a payload mass of 295 kg to get a characteristic acceleration of $0.10\,\text{mm/s}^2$. The realization of such a mission within the specified mass budget, including a lander of about 150 kg and a sample return capsule of about 50 kg, appears to be feasible. Table 7.4 summarizes the ENEAS-SR parameters.



7.7.1: The characteristic acceleration a_c as a function of s and m_{PL} for $\sigma_{\mathrm{SA}} = 29.2\,\mathrm{g/m^2}$



7.7.2: The characteristic acceleration a_c as a function of σ_{SA} and m_{PL} for $s = 50\,\mathrm{m}$



7.7.3: The side length s of the solar sail that is required to achieve a characteristic acceleration of $0.10\,\mathrm{mm/s^2}$ as a function of σ_{SA} and m_{PL}

Figure 7.7

Sail area	A	$(70\,\mathrm{m})^2$
Sail assembly mass	m_{SA}	$111\,\mathrm{kg}$
Sail assembly loading	σ_{SA}	$22.7\,\mathrm{g/m^2}$
Payload mass (incl. spacecraft bus)	m_{PL}	$295\,\mathrm{kg}$
Total sailcraft mass	m	$406\,\mathrm{kg}$
Sailcraft loading	σ	$82.9\,\mathrm{g/m^2}$
Characteristic acceleration	a_c	$0.100\,\mathrm{mm/s^2}$
Characteristic SRP force	F_c	$40.6\,\mathrm{mN}$

Table 7.4 Parameters for the ENEAS-SR solar sailcraft

The ENEAS-SR transfer trajectory to 1996FG$_3$ is analogue to the ENEAS trajectory, as described in section 7.2.2.1. At the end of the transfer trajectory, the solar sailcraft will make a rendezvous with 1996FG$_3$ within its gravitational sphere of influence (HILL-sphere) of between 70 km radius (at perihelion) and 150 km radius (at aphelion). Even in the near-field of the asteroid, the SRP acceleration of between 0.05 mm/s^2 (at the aphelion of 1996FG$_3$) and 0.21 mm/s^2 (at perihelion) is larger than the asteroid's gravitational acceleration (0.01 to 0.00005 mm/s^2 in a distance ranging from 5 to 50 km), so that the sailcraft is able to hover on an artificial equilibrium surface in the hemisphere that is opposite to the sun (figure 7.8).

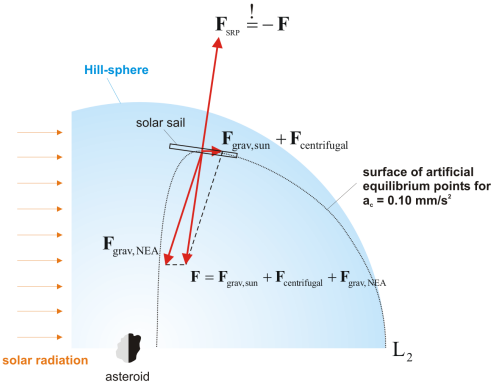


Figure 7.8 Hovering at the asteroid

Those quasi-stationary hovering positions are unstable but can be stabilized using a feedback control loop [66]. Hovering near the asteroid, the (likely complex) gravitational field of the target body is studied, so that a coarse gravitational field model can be determined. Thereafter, the lander with an integrated Earth return capsule is separated from the solar sail to go into closer orbit about the asteroid. While measuring the asteroid's gravitational field with increasing accuracy, the orbit of the lander is continuously lowered until a safe landing trajectory can be computed (some or all of those extensive computations may be performed on Earth). Once landed, the sample is fed directly into the Earth return capsule and brought back by the lander to the hovering sailcraft. In this mission phase, the sailcraft is waiting edge-on (so that no SRP force is acting on the sail) at the L2 Lagrange point for the lander to assist the rendezvous. The lander design, the sample extraction mechanisms and the subsystems that are required to rendezvous the waiting sailcraft require further studies and are beyond the scope of this analysis. Since 1996FG$_3$ is a binary system, it would be interesting to land and extract samples from both bodies, to investigate the origin and the collisional evolution of the 1996FG$_3$ system. Since the gravitational acceleration is very low near the asteroid and the required ΔV for the lander less than 10 m/s, a cold gas system with a propellant mass of less than 4 kg will suffice to perform all operations. After rendezvous with the hovering sailcraft, the re-docked ENEAS-SR solar sailcraft returns the sample to Earth. Finally, some hours before the arrival of ENEAS-SR at Earth, the return capsule is separated from the lander, spun-up to maintain the required entry attitude, and injected into an Earth reentry trajectory, where it is decelerated by atmospheric friction and breaking parachutes. The return trajectory is much faster than the transfer trajectory to 1996FG$_3$ since no rendezvous is required at Earth. Thus, the sailcraft may arrive with a relatively large hyperbolic excess velocity of about 5.5 km/s. The gravitational acceleration of Earth adds another 11.2 km/s,

so that the Earth entry velocity may reach about $\sqrt{5.5^2 + 11.2^2}\,\mathrm{km/s} = 12.5\,\mathrm{km/s}$. This is slightly less than the entry velocity of NASA's Stardust capsule, which has the highest entry velocity (12.9 km/s) of any Earth-returning mission up to date [32, 33].

7.2.3.2 Mission Analysis for SEP Spacecraft

To assess the capability of solar sail propulsion for this NEA sample return mission, the relevant mission and system parameters are again compared with two SEP systems, using a single NSTAR thruster (called ENEASEP1-SR, see table 7.5), and using a cluster of three NSTAR thrusters (called ENEASEP3-SR, see table 7.6) respectively, that accomplish the same mission objective. Again, InTrance has been applied for the optimization of the SEP trajectories.

Payload mass (incl. spacecraft bus)	m_{PL}	295.0 kg
Ion propulsion system mass	m_{Thr}	48.0 kg
Solar array mass	$2 \cdot m_{\mathrm{SAW}}$	55.4 kg
Dry mass (without propellant and tank)	m_{dry}	398.4 kg
Maximum thrust	F_{max}	71.3 mN
Total launch mass with 180 kg propellant mass and tank	m_0	596.4 kg
Maximum acceleration with full tank	a_0	$0.120\,\mathrm{mm/s^2}$
Maximum acceleration with (near-)empty tank	a_f	$0.171\,\mathrm{mm/s^2}$

Table 7.5 Parameters for the ENEASEP1-SR spacecraft with a single NSTAR thruster

Payload mass (incl. spacecraft bus)	m_{PL}	295.0 kg
Ion propulsion system mass	$3 \cdot m_{\mathrm{Thr}}$	144.0 kg
Solar array mass	$4 \cdot m_{\mathrm{SAW}}$	110.8 kg
Dry mass (without propellant and tank)	m_{dry}	549.8 kg
Maximum thrust	F_{max}	213.9 mN
Total launch mass with 215 kg propellant mass and tank	m_0	788.5 kg
Maximum acceleration with full tank	a_0	$0.270\,\mathrm{mm/s^2}$
Maximum acceleration with (near-)empty tank	a_f	$0.389\,\mathrm{mm/s^2}$

Table 7.6 Parameters for the ENEASEP3-SR spacecraft with a cluster of three NSTAR thrusters

The InTrance-results for ENEASEP1-SR (figure 7.9) show that with a single NSTAR thruster the same payload could be delivered to 1996FG$_3$ within 690 days (if $C_3 = 0\,\mathrm{km^2/s^2}$). However, the launch mass of ENEASEP1-SR is 47% larger than for the solar sail option, requiring a heavier and thus more expensive launch vehicle. The time of operations at the asteroid is in this case approx. 118 days and the Earth return leg takes 465 days, so that the total mission duration is 3.48 years (1237 days), which is approximately a third of the mission duration for ENEAS-SR. Approx. 180 kg of Xenon are required as propellant. The hyperbolic excess velocity at arrival is about 5.5 km/s, which is the same value as for ENEAS-SR, so that the Earth reentry velocity may reach also about 12.5 km/s.

Using ENEASEP3-SR, the total mission duration can be even further reduced to 1.91 years

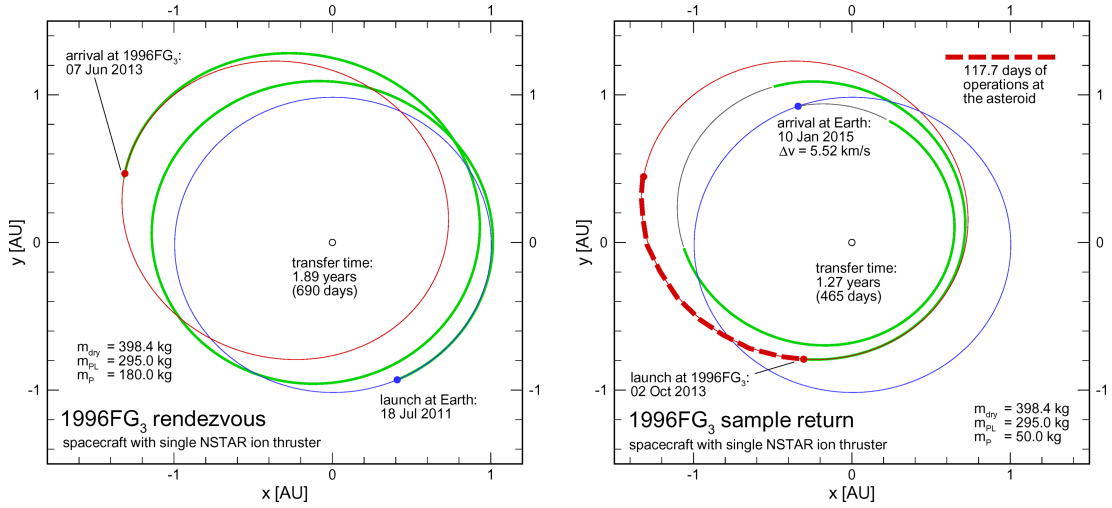


Figure 7.9 ENEASEP1-SR outward and return trajectory

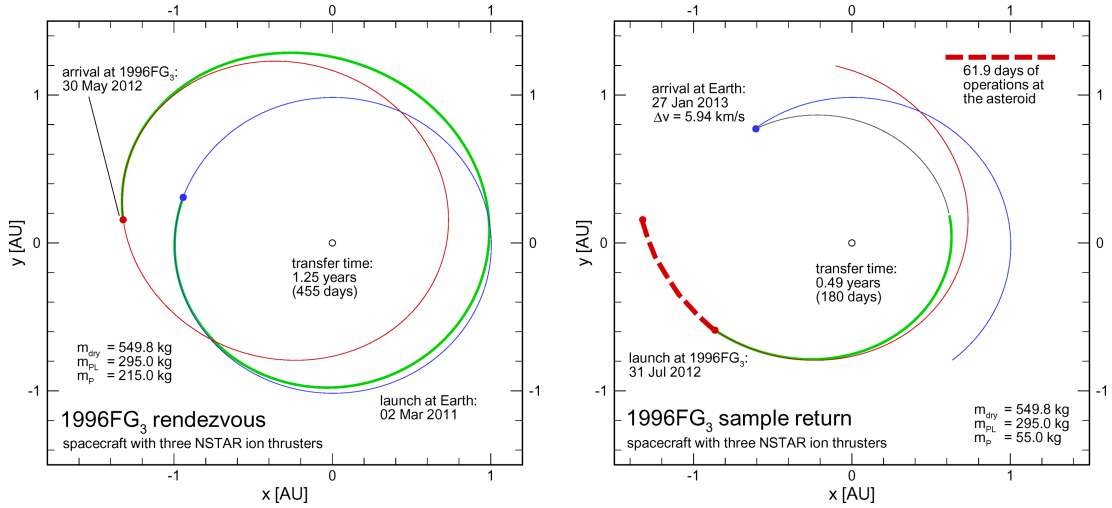


Figure 7.10 ENEASEP3-SR outward and return trajectory

(697 days) for a different optimal launch date, as the InTrance-results in figure 7.10 show ($C_3 = 0\,\mathrm{km}^2/\mathrm{s}^2$). The faster transfer requires not only a larger thrust but also approx. 20% more propellant (215 kg Xenon). In this case, the launch mass of ENEASEP3-SR is 32% larger than for ENEASEP1-SR and 94% larger than for the solar sail option. However, the hyperbolic excess velocity at arrival is larger (5.9 km/s, leading to a slightly larger reentry velocity of about 12.7 km/s) and the time for operations at the asteroid is only 62 days, which might be too short to accomplish the mission objectives. For a longer stay time of up to one year, no reasonable return trajectories have been found due to the unfavorable constellation of Earth and 1996FG$_3$. However, there are return options for stay times longer than one year.

7.2.3.3 Comparison of the Solar Sail Option and the SEP Option

The results demonstrate that also for this mission, solar sailcraft is clearly outperformed by the SEP option, if only the transfer time is considered. This is despite the fact that the required velocity increment for the mission is approx. 50% larger ($\Delta V = 11.0\,\mathrm{km/s}$ for ENEASEP1-SR and $\Delta V = 9.4\,\mathrm{km/s}$ for ENEASEP3-SR) than for the ENEASEP mission. However, the required thruster operation time (24 500 hours for ENEASEP1-SR and 13 300 hours for ENEASEP3-SR) exceeds by far the currently demonstrated NSTAR thruster lifetime of 8 192 hours on DS1 [17]. Consequently, a higher thruster lifetime must be achieved to perform such a mission. Of course, lifetime and aging is also an issue for solar sails, for which adequate tests have also to be performed. Using a solar sail for propulsion, ENEAS-SR could be launched on a smaller and thus less expensive launcher than ENEASEP1-SR and ENEASEP3-SR. If ground operation costs can be kept low, and if the transfer time plays a subordinate role with respect to cost, the solar sail might the best option for such a mission.

7.2.4 Multiple Near-Earth Asteroid Rendezvous and Sample Return Mission

Within the last section, a $(70\,\mathrm{m})^2$ solar sail with a mass of 111 kg was presumed for the ENEAS-SR mission, to accomplish a characteristic acceleration of $0.1\,\mathrm{mm/s^2}$ for a 295 kg payload. Within this section, it is investigated, whether – using the same solar sail – a multiple NEA rendezvous mission can be performed with the small ENEAS payload of 75 kg. Consistently, this multiple NEA rendezvous mission is termed ENEAS+. Table 7.7 summarizes the ENEAS+ parameters, together with the parameters for a mission termed ENEAS+SR, which will be described later.

		ENEAS+	ENEAS+SR
Sail area	A	$(70\,\mathrm{m})^2$	$(139\,\mathrm{m})^2$
Sail assembly mass	m_{SA}	111 kg	437 kg
Sail assembly loading	σ_{SA}	$22.7\,\mathrm{g/m^2}$	$22.7\,\mathrm{g/m^2}$
Payload mass (incl. spacecraft bus)	m_{PL}	75 kg	295 kg
Total sailcraft mass	m	186 kg	732 kg
Sailcraft loading	σ	$38.0\,\mathrm{g/m^2}$	$38.0\,\mathrm{g/m^2}$
Characteristic acceleration	a_c	$0.218\,\mathrm{mm/s^2}$	$0.218\,\mathrm{mm/s^2}$
Characteristic SRP force	F_c	40.6 mN	160 mN

Table 7.7 Parameters for the ENEAS+ and the ENEAS+SR solar sailcraft

For the given solar sail and the given payload, the characteristic acceleration of the ENEAS+ solar sailcraft is $0.218\,\mathrm{mm/s^2}$. To compare solar sail propulsion with the SEP option, the target objects of the Hera-mission have been adopted (see section 6.5). InTrance has been used to calculate the transfer times between the targets for various launch dates to find the optimal mission sequence. The best found mission sequence is illustrated in figure 7.11 and in table 7.8. The trajectory from $1999\mathrm{AO}_{10}$ to Earth has been included, although it is not part of the actual ENEAS+ mission. However, the trajectories are valid for any solar sailcraft that accomplishes a characteristic acceleration of $0.218\,\mathrm{mm/s^2}$. Using a larger solar sail (see

right column of table 7.7), this mission might be also performed with the ENEAS-SR payload. With such a solar sailcraft, all three visited NEAs could be sampled, and the samples could be returned to Earth (this mission might be termed ENEAS+SR).

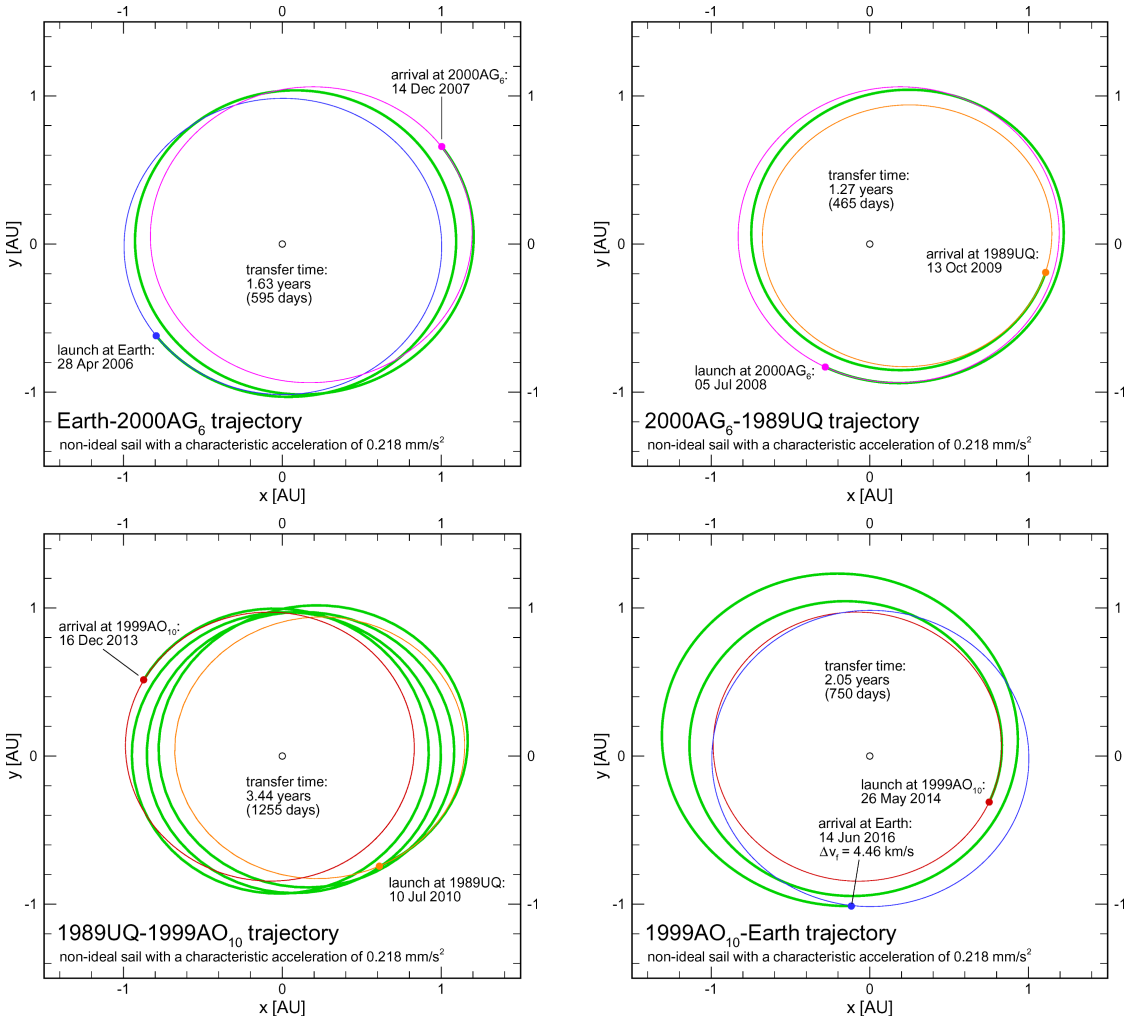


Figure 7.11 ENEAS+/ENEAS+SR trajectory options

Transfer time	Arrival date	Body	Stay time	Launch date
		Earth		28 Apr 06
595 days	14 Dec 07	$2000AG_6$	203.5 days	05 Jul 08
465 days	13 Oct 09	1989UQ	270.0 days	10 Jul 10
1255 days	16 Dec 13	$1999AO_{10}$	161.4 days	26 May 14
750 days	14 Jun 16	Earth		

Table 7.8 ENEAS+/ENEAS+SR mission data

As table 7.8 shows, the total mission durations of ENEAS+ and ENEAS+SR (7.63 years and 10.13 years respectively) are considerably longer (59% and 110% respectively) than for the Hera-mission. Regarding the question, whether solar sail or solar electric propulsion is superior for such a mission, the same conclusions as in the previous sections 7.2.2.3 and 7.2.3.3 can be drawn.

7.3 Mercury Rendezvous Mission

Within this section, two solar sailcraft rendezvous missions to Mercury are assessed, one being comparable to the American MESSENGER mission and the other one to ESA's cornerstone mission BepiColombo (same scientific payload, same transfer time), the former with a chemical propulsion system and the latter with a SEP system.

7.3.1 Mission Objectives

Although Mercury holds answers to many important questions regarding the formation and evolution of the solar system, it is – due to its difficult accessibility ($\Delta V \geq 13.12\,\mathrm{km/s}$ [73]) – the least characterized and understood terrestrial planet. Up to now, information about Mercury is limited to Earth-based observations and the data set gained by Mariner 10 during its three subsequent fly-bys (1974-75), imaging (only[3]) 46% of Mercury's surface and discovering its magnetic field. Important questions about Mercury include the origin of its anomalously high metal to silicate ratio and its implications for planetary accretion processes, its geological evolution, the mechanisms of the magnetic field generation, the processes that control the volatile species that are eventually present in polar deposits, and the nature of its exosphere and magnetosphere [87]. Radio science experiments could provide a detailed mapping of Mercury's gravity field (including its temporal tidal variations) and information about its internal structure [46, 62]. Another import scientific objective at Mercury could be the detection of asteroids with aphelia inside the Earth's orbit (called Inner Earth Objects, IEOs) that would be very difficult to detect from Earth-based or near-Earth-based telescopes[4].

7.3.2 Propulsion Options

Two dedicated missions are scheduled to rendezvous Mercury within the next ten years: MESSENGER (USA) and BepiColombo (Europe/Japan). Both missions address the key questions that have been identified since Mariner 10 [41]. MESSENGER is a relatively lightweight Mercury orbiter that employs chemical propulsion and a Venus-Venus-Mercury-Mercury gravity assist trajectory to reach the planet within more than five years, whereas BepiColombo is a relatively heavyweight spacecraft[5] that employs SEP and also multiple gravity assists at Venus and Mercury to reach Mercury within less than three years.[6] Table 7.9 gives the most important mission parameters for MESSENGER and BepiColombo [8, 50, 79]. A comparison

[3] since the fly-bys were in 3:2 resonance

[4] in fact, no IEO has been detected so far [41]

[5] consisting of three distinct sub-spacecrafts: the Mercury Planetary Orbiter (MPO), the Mercury Magnetospheric Orbiter (MMO), and the Mercury Surface Element (MSE)

[6] Single launch option for Jan 09 with Ariane-5. However, this is only *one* launch option among many; however, the present baseline is a double launch of the sub-spacecrafts on two Soyuz-Fregat launchers.

	MESSENGER	BepiColombo (single launch option with Ariane-5)
Scientific payload	380 kg	757 kg
Launch	Mar 04	Jan 09
Launch mass	1093 kg	2495 kg
Launcher	Delta 2925H-9.5	Ariane-5
C_3	$15.2\,\mathrm{km^2/s^2}$	$7.8\,\mathrm{km^2/s^2}$
Gravity assists	(2×)Venus-(2×)Mercury	(2×)Venus-(2×)Mercury
Transfer Time	5.07 years	2.64 years
Mercury orbit insertion	Apr 09	Aug 11

Table 7.9 Comparison of the significant (tentative) mission parameters for MESSENGER and BepiColombo

of the values that are given in table 7.9 shows clearly the advantage of SEP over chemical propulsion.

Hughes [45] investigated the use of solar sail propulsion for a BepiColombo-like mission, using the simplified SRP force model ($\eta = 0.9$). According to his results, Mercury can be reached within 2.4 years, if the solar sail has a characteristic acceleration of $a_c = 0.3\,\mathrm{mm/s^2}$. For an assumed payload (MPO+MMO+MSE) of 553 kg (which is in contrast to the 757 kg that is given in [8]), he derived a required sail size of $(178\,\mathrm{m})^2$, assuming a sail assembly loading of $\sigma_{\mathrm{SA}} = 10\,\mathrm{g/m^2}$. The mass of the solar sailcraft is in this case 872 kg, which is significantly less than the launch mass for the SEP option[7], and – at the same time – the hyperbolic excess energy requirement is lowered to $C_3 = 0\,\mathrm{km^2/s^2}$.

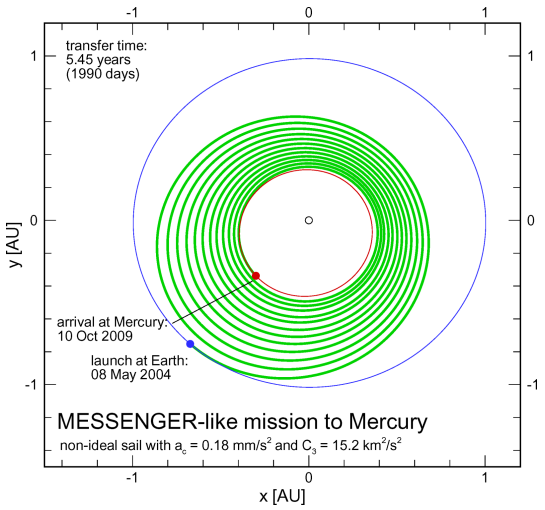
7.3.3 Mission Analysis for Solar Sailcraft

Scientific payload	380 kg	757 kg
Launch	May 04	Mar 08
characteristic acceleration	$0.18\,\mathrm{mm/s^2}$	$0.35\,\mathrm{mm/s^2}$
C_3	$15.2\,\mathrm{km^2/s^2}$	$7.8\,\mathrm{km^2/s^2}$
Gravity assists	–	–
Transfer Time	5.45 years	2.64 years
Mercury orbit insertion	Oct 09	Nov 10

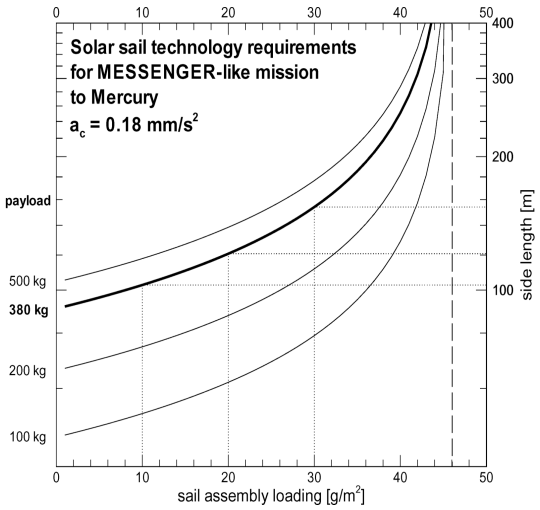
Table 7.10 Comparison of the significant mission parameters for a MESSENGER- and a BepiColombo-like Mercury mission using a solar sail

Within this section, InTrance is used to derive the performance requirements for solar sails that would be able to transport the payload of MESSENGER and BepiColombo respectively to Mercury. In contrast to [45], the more realistic standard SRP force model is employed and the relevant mission parameters are taken from [8] and [79] (table 7.10). For a fair comparison, the respective C_3-values according to table 7.9 are used also for the solar sail option.

[7] in [45], a launch mass of 2272 kg is assumed for the SEP option



7.12.1: Trajectory



7.12.2: Performance requirements

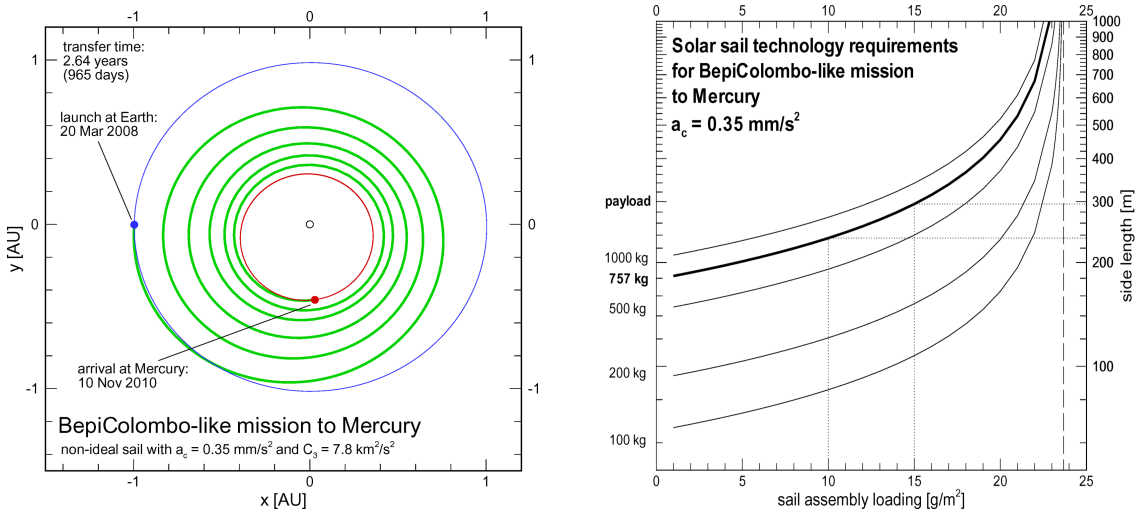
Figure 7.12 MESSENGER-like mission to Mercury using a solar sail

sail assembly loading σ_{SA} [g/m^2]	sail size s [m]	sail assembly mass m_{SA} [kg]	launch mass m [kg]
10	102.7	105	485
20	120.8	292	672
30	153.9	711	1091

Table 7.11 Required sail size s for various sail assembly loadings σ_{SA} to achieve a characteristic acceleration of $a_c = 0.18\,\mathrm{mm/s^2}$

As figure 7.12.1 shows, a solar sail with a characteristic acceleration of $0.18\,\mathrm{mm/s^2}$ is able to reach Mercury within approximately the baselined MESSENGER transfer time, if it is injected with the same hyperbolic excess energy of $15.2\,\mathrm{km^2/s^2}$. In contrast to the chemical baseline mission scenario, no gravity assist maneuver is necessary to achieve the required ΔV, which results in a more flexible mission profile. Figure 7.12.2 gives the sail size that is required to achieve $a_c = 0.18\,\mathrm{mm/s^2}$ for different sail assembly loadings and payload masses. The bold line denotes the σ_{SA}-s-relation for the MESSENGER payload mass of 380 kg. It can be seen that the required sail size increases drastically for $\sigma_{SA} \gtrsim 30\,\mathrm{g/m^2}$, and that for $\sigma_{SA} \approx 46\,\mathrm{g/m^2}$ the required sail size approaches infinity. Table 7.11 gives values for three points on the MESSENGER-payload-curve. It can be seen that the sail assembly loading must be below $30\,\mathrm{g/m^2}$ to yield a benefit in launch mass with respect to the MESSENGER baseline mission scenario. For an advanced solar sail with a low sail assembly loading of approx. $10\,\mathrm{g/m^2}$ the launch mass is less than half of the actual MESSENGER launch mass. Such a solar sail could considerably reduce the launcher requirements and thus the launch costs for such a MESSENGER-like mission.

To reach Mercury with a solar sail within the BepiColombo reference transfer time (figure 7.13.1), a characteristic acceleration of $0.35\,\mathrm{mm/s^2}$ is necessary, if the solar sailcraft is injected with the same hyperbolic excess energy of $7.8\,\mathrm{km^2/s^2}$. In contrast to the SEP mission



7.13.1: Trajectory

7.13.2: Performance requirements

Figure 7.13 BepiColombo-like mission to Mercury using a solar sail

sail assembly loading σ_{SA} [g/m^2]	sail size s [m]	sail assembly mass m_{SA} [kg]	launch mass m [kg]
10	235.2	553	1310
15	295.3	1308	2065

Table 7.12 Required sail size s for various sail assembly loadings σ_{SA} to achieve a characteristic acceleration of $a_c = 0.35\,\mathrm{mm/s^2}$

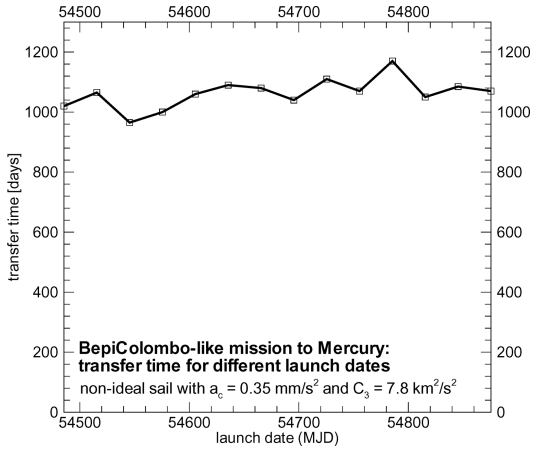


Figure 7.14 "Launch window" for a BepiColombo-like mission to Mercury using a solar sail

scenario, no gravity assist maneuver is necessary to achieve the required ΔV. This leads to a practically ever-existent launch window (see figure 7.14), allowing a very flexible mission profile. For the BepiColombo reference scenario, a one-month launch window opens up only every 1.6 years [8]. Figure 7.13.2 gives the sail size that is required to achieve $a_c = 0.35\,\mathrm{mm/s^2}$ for

different sail assembly loadings and payload masses. The bold line denotes the σ_{SA}-s-relation for the BepiColombo payload mass of 757 kg. It can be seen that in this case the required sail size increases drastically for $\sigma_{\mathrm{SA}} \gtrsim 15\,\mathrm{g/m^2}$, approaching infinity for $\sigma_{\mathrm{SA}} \approx 24\,\mathrm{g/m^2}$. Table 7.12 gives values for two points on the BepiColombo-payload-curve. To yield a considerable benefit in launch mass with respect to the BepiColombo baseline mission scenario, the sail assembly loading must be below approx. $15\,\mathrm{g/m^2}$. For an advanced solar sail with a low sail assembly loading of approx. $10\,\mathrm{g/m^2}$ the launch mass is nearly half of actual BepiColombo launch mass. Again, such a solar sail could considerably reduce the launcher requirements and thus launch costs for such a BepiColombo-like mission. However, the required sail dimensions are impressive and far from existent or near-term technology. However, they would be lower, if additional gravity assist maneuvers were performed, which can not be calculated within the current one-body simulation model of InTrance. This is why the optimization of low-thrust trajectories that include gravity assist maneuvers is a desirable feature for future InTrance versions.

7.4 Piloted Mars Mission

To provide a final innovational example of how InTrance can be applied to support space mission analysis, it is used to analyze the feasibility of a piloted Mars mission for spacecraft using a nuclear electric propulsion system.

7.4.1 Mission Objectives and Propulsion Options

Beyond the ISS and the Moon, Mars is the logical next step towards the manned exploration and conquest of space. Differing from "ordinary" robotic missions due to large payloads and restricted flight times, the feasibility of piloted Mars missions depends crucially on an adequate propulsion system. To reduce the risk for the crew, a short mission duration (of less than approximately two years) and a short stay time (of less than approximately three months) is desirable (fast mission). Such a requirement precludes the application of chemical propulsion systems, which necessitate in this case an immense effort (several thousand tons in LEO for an Earth return payload of about 75 t), since at least one trajectory leg requires a large ΔV. Due to their larger specific impulse, low-thrust propulsion systems are expected to enable relatively short missions with reasonable effort. Within this section, InTrance is employed to analyze mission opportunities for an exemplary spacecraft with a NEP system (300 N maximum thrust, 6000 s specific impulse, 160 t launch mass at Earth, 75 t Earth return payload), providing an illustrative example of how InTrance is recently used at DLR to analyze the capability of various low-thrust propulsion systems to enable fast piloted Mars missions [81, 82].

7.4.2 Mission Analysis for NEP Spacecraft

Using InTrance, time-optimal trajectories have been found to have three different topologies (A, B, and C, figure 7.15), depending on the constellation of Earth and Mars at the respective departure, on the closest tolerable solar distance (r_{min}), and on the maximum relative velocity at the target body ($\Delta v_{f,\mathrm{max}}$). Within this categorization, trajectories of type A neither cross

the orbit of Earth nor that of Mars. They have short transfer times and require a moderate to high ΔV. However, Type A trajectories are only possible for favorable constellations of the two planets (type A phase). Trajectories of type B cross the orbit of Earth, moving thereby closer to the sun. They have longer transfer times and require a high to very high ΔV. Trajectories of type C move farther away from the sun than Mars, having moderate to very long transfer times and a moderate to high ΔV-requirement. Type B and C phases are defined accordingly as the time intervals, in which type B and C trajectories are time-optimal due to the constellation of Earth and Mars at departure. The phases alternate (A→C→B→A→... for the Earth-Mars transfer and A→B→C→A→... for the Mars-Earth transfer) as similar constellations recur.

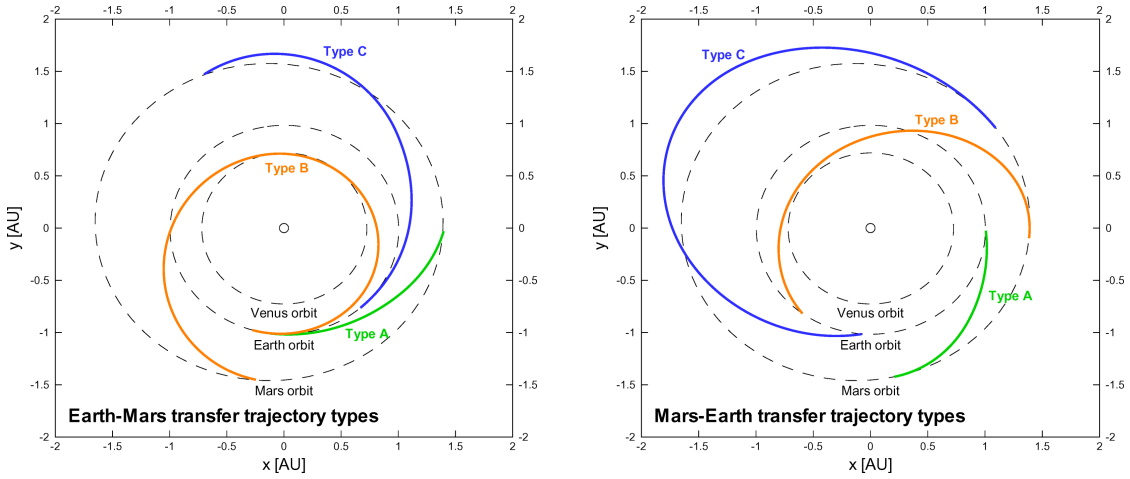


Figure 7.15 Trajectory types (A, B, and C, see text) for Earth-Mars and Mars-Earth transfers

Figure 7.16 shows for an Earth return trajectory, how the transfer time varies within one A→B→C-cycle. The left part of the downward slope is associated with type C trajectories, the right part of the downward slope is associated with type A trajectories, and the upward slope is associated with type B trajectories. Thus, type C trajectories evolve gradually into type A trajectories, whereas there is a tremendous increase in flight time, when type B trajectories become non-optimal and type C trajectories provide the time-optimal option to return to Earth. As it can be seen, type B trajectories (with reasonable transfer times) can also be flown later in time, if a closer solar approach is tolerated. In this case, a trade-off has to be made concerning the medical risk for the crew (long transfer-time vs. close solar fly-by). A similar diagram can be drawn for the Earth-Mars leg of the mission. However, what is more meaningful, is to plot the transfer time for this leg against the *arrival* date at Mars, together with a plot of the Earth return transfer time against the *departure* date at Mars, as it done in figure 7.17.1 for $r_{\min} = 0.7\,\mathrm{AU}$. Looking at the displacement of both curves, one can see that for a short stay at Mars, the combination of a short Earth-Mars leg with a short Mars-Earth leg (a type A-A trajectory pair) is not possible with the given propulsion system. On the basis of this diagram, different options for a piloted mission can be discussed. The horizontal bar of the "H" gives the stay time and the two vertical bars give the combined transfer time, so that the size of the "H" defines the total mission duration. For each stay time-value a minimal flight time exists, which can be plotted against the stay

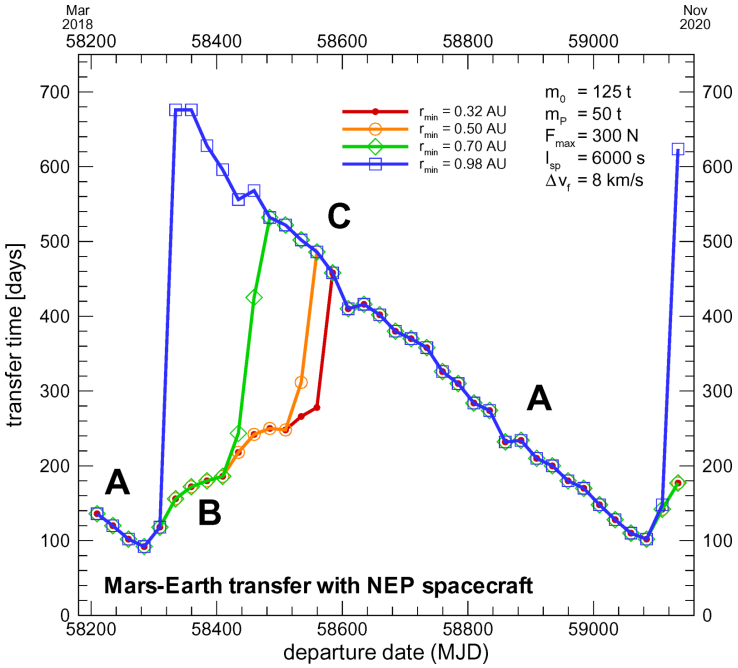
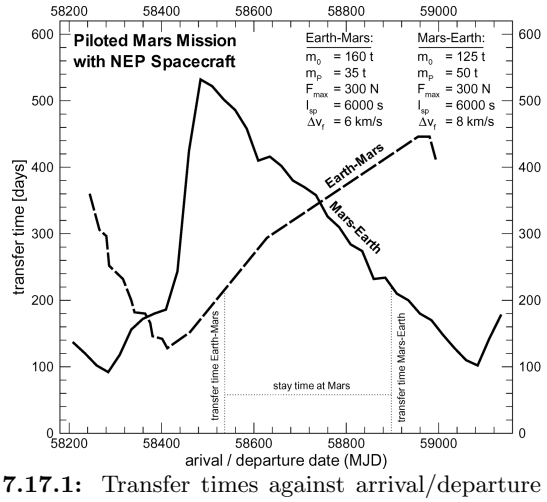
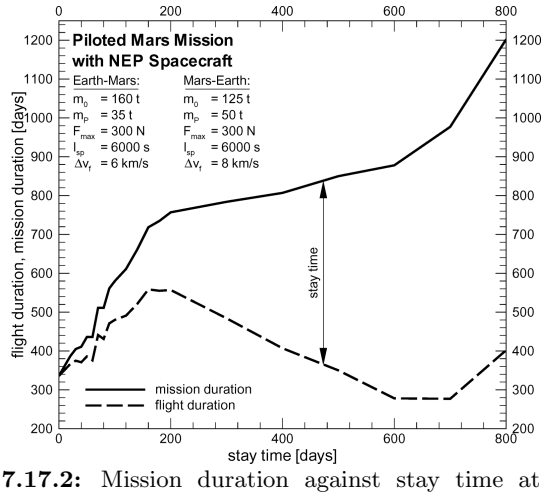


Figure 7.16 Transfer time for Earth return in dependence of the departure date at Mars and the minimum tolerable solar distance $r_{\min}$



7.17.1: Transfer times against arrival/departure date at Mars



7.17.2: Mission duration against stay time at Mars

Figure 7.17

time, as it is done in figure 7.17.2. As this diagram shows – using the given spacecraft and propulsion system parameters – a stay time of three month can be realized within a total mission duration of 561 days (1.54 years). For a total mission duration of 2 years, the stay time at Mars can be extended to about 140 days (4.7 months). The diagram shows also that – using the given propulsion system – type A-A transfers with short total flight durations are only possible for long stay times at Mars of about 600 days.

8 Summary and Conclusions

Low-thrust propulsion systems provide a small thrust force over an extended period of time to modify the orbit of spacecraft and to generate the required velocity increment. Unlike for high-thrust propulsion systems – like chemical rockets – the trajectory can not be approximated by ballistic flight arcs combined with singular thrusting events that change the spacecraft's velocity instantaneously while its position remains fixed. Low-thrust trajectory optimization is equivalent to the search for the optimal thrust vector variation along the entire trajectory from a given initial state A to a desired final state B. The optimality of the trajectory can be defined with respect to several (possibly competing) objectives, e.g. transfer time and propellant consumption. Traditionally, low-thrust trajectory optimization problems are solved using various numerical optimal control methods that are based on the calculus of variations. All these methods can be generally classified as *local* trajectory optimization methods. Their convergence behavior depends on an adequate initial guess of the solution, which is often hard to find, so that the search for a good trajectory can become very time-consuming. Even if convergence is achieved, a local optimum is typically found, which is close to the initial guess that is rarely close to the global optimum.

Within this work, low-thrust trajectory optimization problems have been attacked from a different perspective: the perspective of artificial intelligence and machine learning. Inspired by natural archetypes, a smart global method for spacecraft trajectory optimization was developed that fuses artificial neural networks and evolutionary algorithms to evolutionary neurocontrollers (this method was termed InTrance, which stands for "Intelligent Trajectory optimization using neurocontroller evolution"). From the perspective of machine learning, a trajectory is regarded as the result of an explicitly not time-dependent steering strategy that manipulates the spacecraft's thrust vector according to the actually perceived state of the environment. A standard feedforward artificial neural network is used as a so-called neurocontroller to implement such a spacecraft steering strategy. This way, the trajectory is defined by the internal parameters of the neurocontroller. An evolutionary algorithm is used for finding the optimal network parameters. The trajectory optimization problem is solved, if the optimal parameter vector is found, which defines the optimal spacecraft steering strategy, which in turn generates the optimal trajectory. Using an evolutionary algorithm for the optimization of the neurocontroller, this algorithm may be additionally used for finding good initial conditions.

InTrance has been applied to a variety of interplanetary low-thrust trajectory optimization problems, for which reference trajectories have been found in the literature. In accordance with those reference problems, a one-body simulation model without disturbing forces was used. The re-calculation of those problems has revealed that many of the trajectories, which have been generated using traditional local trajectory optimization methods, are quite far from the global optimum. Using InTrance, improvements in transfer time of up to 74% have been achieved. For mission feasibility analysis, the obtained InTrance-trajectories are usually sufficiently accurate with respect to the terminal constraint. However, especially for very difficult problems, demanding accuracy requirements, and indirect steering strategies, InTrance does not in all instances converge against the near-globally optimal solution, or does

not achieve the required accuracy, so that more than one InTrance-run[1] should be performed to assess the quality of the generated solution(s) and to raise the likelihood for obtaining a near-globally optimal solution.[2] If a more accurate solution than the InTrance-generated one is required, the InTrance-solution might be used as an initial guess for some numerical optimal control method. The obtained results indicate that – using an evolutionary neurocontroller for spacecraft steering – InTrance is a suitable method for finding near-globally optimal low-thrust trajectories. Since evolutionary neurocontrollers explore the trajectory search space more exhaustively than a human expert can do by using traditional optimal control methods, they are able to find spacecraft steering strategies that generate better trajectories, which are closer to the global optimum. Unlike the traditional methods, InTrance runs without an initial guess and without the permanent attendance of an expert in astrodynamics and optimal control theory.

The influence of various InTrance-parameters – like neurocontroller type, population size, and accuracy requirements – on the convergence behavior of the algorithm and on the quality of the obtained solutions has been investigated. Both, quality and convergence behavior have been found to depend considerably only on the chosen neurocontroller input and output set, and on an adequate choice of the fitness function. Thereby, direct steering strategies have been found to perform better than indirect steering strategies. Concerning the other varied parameters, the convergence behavior of InTrance has found to be quite robust. Many InTrance-parameters have *not* been varied, since reasonably robust settings had been found during the development of InTrance, and since the primary objective of this work was *not* to find the optimal neurocontroller and evolutionary algorithm parameters for each particular problem, but to demonstrate that evolutionary neurocontrol can be successfully applied to the near-globally optimal steering of low-thrust spacecraft. The optimization of the neurocontroller and the evolutionary algorithm as well as investigations about the problem-dependency of the parameters remain a wide field of research. The InTrance-generated steering strategies have been found to be quite insensitive to neurocontroller input noise, as it is expected for a real mission due to errors in the spacecraft's autonomous sensors and/or errors in the measurement of the spacecraft's position and velocity from Earth. However, the InTrance-generated steering strategies have been found to be sensitive to disturbing accelerations acting on the spacecraft, so that a thorough mission design should employ a more sophisticated simulation and spacecraft model than the one that has been used within this work.

Many promising directions concerning the application of evolutionary neurocontrol for spacecraft trajectory optimization are proposed for future research:

- Evolutionary neurocontrol might be applied to a wide variety of trajectory optimization problems, including different propulsion systems and planetocentric problems. To solve combined low-thrust planetary escape → interplanetary transfer → planetary capture problems, three neurocontrollers might be encoded on a single chromosome, so that they can be co-evolved by the evolutionary algorithm. To find still faster transfer trajectories, the inclusion of gravity-assist maneuvers would also be desirable. The number of intermediate fly-bys – e.g. at near-Earth objects – along a trajectory would be another interesting objective for optimization, if longer transfer times can be tolerated. Evolutionary neurocontrol may also be successfully applied to autonomous multiple

[1] using different randomly generated initial populations

[2] This could of course be implemented into the software, however at the cost of runtime performance.

rendezvous trajectory optimization (e.g. rendezvous three arbitrary near-Earth objects from a set of some hundred objects in minimal time, or low-thrust Earth-Mars-cycler trajectories, etc.)

- The found neurocontroller steering strategies are not universal steering strategies, but achieve acceptable results only for a launch within a small interval around the launch date, for which they are optimized. For a given spacecraft, a *universal* steering strategy would achieve the required accuracy for an entire class of transfer problems (e.g. rendezvous with any arbitrary solar system body for all launch dates). Such universal steering strategies have not been found.[3] It remains questionable, whether universal steering strategies exist and whether they can be represented using a feedforward artificial neural network. Universal steering strategies would be very interesting, since they would suggest that low-thrust trajectory optimization problems might have a (not analytically obtainable and yet unknown) closed-form solution. It may also be beneficial to utilize more general artificial neural networks, where the neural transfer function and the network topology are also optimized by the evolutionary algorithm. (Truncated) FOURIER series provide another option to represent the variation of control variables. They might be used instead of artificial neural networks to represent a (non-universal) steering law. In this case, the evolutionary algorithm could be used to determine the optimal coefficients of the series. For a universal steering law, the coefficients can not be fixed but are expected to be a function of the problem parameters.

- To perform more thorough mission analyses, the implementation of a more sophisticated simulation model, that includes the gravitational influence of all major solar system bodies is desirable. Additionally, more realistic spacecraft models must be used in this case. Especially for solar sailcraft, the translational motion is highly interrelated with the rotational motion, so that a model for all six degrees of freedom should be developed. Also the billowing, wrinkling and aging of the sail must be studied through FEM simulations and in-space tests.

The secondary objective of this work was to assess the performance of solar sail propulsion with respect to chemical and solar electric propulsion (SEP) for interplanetary missions. This has been done exemplary for missions to near-Earth asteroids and to Mercury. For the near-Earth asteroid missions, a near-term sail engineering technology was assumed. The obtained InTrance-results indicate that for this kind of missions (moderate velocity increments, relatively far from the sun), near-term solar sailcraft is clearly outperformed by the SEP option, if only transfer time is considered. However, the launch mass for the solar sail option is considerably smaller in all cases, thus allowing smaller and cheaper launch vehicles. If ground operation costs can be kept low and if the transfer time plays a subordinate role with respect to cost, the solar sail might be the favorable option for such near-Earth asteroid rendezvous and sample return missions. In any case, the development of near-term sails is an indispensable first stepping stone on the way to more advanced solar sails. For the Mercury rendezvous, it has been found that medium-term solar sails are able to deliver the same payload to Mercury than chemical propulsion systems and SEP systems within the same transfer time, allowing at the same time smaller launch vehicles and direct trajectories with simpler

[3] However, they have not been deliberately searched; this would require that a *single* neurocontroller is trained to perform well for a diverse set of problems (different launch dates, different target bodies, etc.).

mission profiles and practically ever-existent launch windows. Another important result, that has been obtained using InTrance, is the fact, that for a thorough mission analysis the non-perfect reflectivity of the solar sail must be considered through an appropriate solar radiation pressure force model. The widely used simplification, that the non-perfect reflectivity of the sail can be taken into account by using an overall efficiency factor, should only be made for *very* preliminary mission feasibility analyses. Finally, to provide another innovative example of how InTrance can be applied to support space mission analysis for low-thrust propulsion systems, it has been used to analyze the feasibility of a piloted Mars mission for spacecraft using a nuclear electric propulsion system.

In conclusion, it is astonishing that small and primitive artificial neural networks, bred by a crudely simulated biological process, are able to perform tasks that – after some million years of intellectual and cultural evolution and some thousand years of scientific evolution – we are not able to solve with our bare mind. Luckily, although we do not *understand* their solution method, we are able to replicate their solutions. The citation by Emerson Pugh at the beginning of this work states that we might probably never be able to understand the human brain, since it is as complex as we are. The results that have been obtained within this work support this statement, although we might probably be more simple than Pugh had ever imagined.

Bibliography

[1] JPL Solar System Dynamics Group Website. http://ssd.jpl.nasa.gov.

[2] I. Physikalisches Institut der JLU-Gießen, Abt. Plasma- und Atomstoßphysik Website. http://www.uni-giessen.de/1.physik/schartner/homepage.html.

[3] NASA Glenn Research Center Website. http://www.grc.nasa.gov.

[4] The Hera Mission Website. http://www.uark.edu/misc/hera.

[5] European Asteroid Research Node Website. http://earn.dlr.de.

[6] Near Earth Objects Dynamic Site Website. http://newton.dm.unipi.it/cgi-bin/neodys/ neoibo.

[7] F. Angrilli and S. Bortolami. Attitude and Orbital Modelling of Solar-Sail Spacecraft. ESA Journal, 14:431–446, 1990.

[8] A. Anselmi and G. E. N. Scoon. BepiColombo, ESA's Mercury Cornerstone mission. Planetary and Space Science, 49:1409–1420, 2001.

[9] Th. Bäck. Evolutionary Algorithms. ACM SIGBIO Newsletter, pages 26–31, 1992.

[10] Th. Bäck. Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. Orlando, USA, 1994. First IEEE Conference on Evolutionary Computation.

[11] P. L. Barry and T. Phillips. The Inconstant Sun. Science@NASA. http://science.nasa.gov/headlines/y2003/17jan_solcon.htm.

[12] A. G. Barto. Connectionist Learning for Control. In: W. Miller, R. S. Sutton, and P. J. Werbos, editors, Neural Networks for Control, pages 5–58. MIT Press, Cambridge, London, 1990.

[13] R. H. Battin. An Introduction to the Mathematics and Methods of Astrodynamics. AIAA Education Series. American Institute of Aeronautics and Astronautics, Reston, Revised Edition, 1999.

[14] J. K. Beatty, C. C. Petersen, and A. Chaikin, editors. The New Solar System. Cambridge University Press, Cambridge, New York, Melbourne, Fourth Edition, 1999.

[15] J. T. Betts. Survey of Numerical Methods for Trajectory Optimization. Journal of Guidance, Control, and Dynamics, 21(2):193–207, 1998.

[16] T. Blickle and L. Thiele. A Comparison of Selection Schemes used in Genetic Algorithms. TIK Report Nr. 11 Version 2, Computer Engineering and Communication Networks Lab (TIK), Swiss Ferderal Institute of Technology (ETH), Zurich, Switzerland, 1995.

[17] J. R. Brophy, R. Y. Kakuda, J. E. Polk, J. R. Anderson, M. G. Marcucci, D. Brinza, M. D. Henry, K. K. Fujii, K. R. Mantha, J. F. Stocky, J. Sovey, M. Patterson, V. Rawlin, J. Hamley, T. Bond, J. Christensen, H. Cardwell, G. Benson, J. Gallagher, M. Matranga, and D. Bushway. Ion Propulsion System (NSTAR). DS1 Technology Validation Report. JPL Publication 00-10, 10/2000, NASA/JPL, 2000.

[18] M. Caudill and C. Butler. Naturally Intelligent Systems. MIT Press, Cambridge, London, 1990.

[19] T. Cichan and R. Melton. Optimal Trajectories for Non-Ideal Solar Sails. Quebec, Canada, August 2001. AIAA/AAS Astrodynamics Specialist Conference.

[20] M. F. Closs. Numerical Modelling and Optimisation of Radio-Frequency Ion Thrusters. Doctoral Thesis, Universität der Bundeswehr München; Fakultät für Luft- und Raumfahrttechnik; Institut für Raumfahrttechnik, 2001.

[21] Commitee on Planetary and Lunar Exploration, Space Studies Board, and Commission on Physical Sciences, Mathematics, and Applications. The Exploration of Near-Earth Objects. Technical report, National Research Council, 1998.

[22] B. Dachwald. Optimization of Interplanetary Solar Sailcraft Trajectories Using Evolutionary Neurocontrol. Journal of Guidance, Control, and Dynamics, 27(1):66–72.

[23] B. Dachwald. Optimierung des Lernverhaltens neuronaler Netze mit Hilfe genetischer Algorithmen. Diploma Thesis, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, Institut für Meßtechnik, July 1993. (in German).

[24] B. Dachwald. Verwendung eines neuronalen Reglers und evolutionärer Algorithmen zur Berechnung optimaler interplanetarer Sonnenseglerbahnen. Stuttgart, Germany, September 2002. Deutscher Luft- und Raumfahrtkongress 2002. DGLR-JT2002-089 (in German).

[25] B. Dachwald. Interplanetary Mission Analysis for Non-Perfectly Reflecting Solar Sailcraft Using Evolutionary Neurocontrol. Big Sky, USA, August 2003. AAS/AIAA Astrodynamics Specialist Conference. AAS 03-579.

[26] B. Dachwald and W. Seboldt. Optimization of Interplanetary Rendezvous Trajectories for Solar Sailcraft using a Neurocontroller. Monterey, USA, August 2002. AIAA/AAS Astrodynamics Specialist Conference. AIAA-2002-4989.

[27] B. Dachwald and W. Seboldt. Solar Sailcraft of the First Generation: Mission Applications to Near-Earth Asteroids. Bremen, Germany, September/October 2003. 54$^{\text{th}}$ International Astronautical Congress. IAC-03-Q.5.06.

[28] B. Dachwald, W. Seboldt, and B. Häusler. Performance Requirements for Near-Term Interplanetary Solar Sailcraft Missions. Versailles, France, May 2002. 6$^{\text{th}}$ International Symposium on Propulsion for Space Transportation of the XXI$^{\text{st}}$ Century.

[29] B. Dachwald, W. Seboldt, and L. Richter. Multiple Rendezvous and Sample Return Missions to Near-Earth Asteroids Using Solar Sailcraft. Noordwijk, the Netherlands, September 2003. Proceedings of 5$^{\text{th}}$ IAA International Conference on Low-Cost Planetary Missions.

[30] K. De Jong. Learning with Genetic Algorithms: An Overview. Machine Learning, 3:121–138, 1988.

[31] I. de Pater and J. J. Lissauer. Planetary Sciences. Cambridge University Press, Cambridge, New York, Melbourne, 2001.

[32] P. N. Desai, R. A. Mitcheltree, and F. McNeil Cheatwood. Entry Trajectory Issues for the Stardust Sample Return Capsule. Arcachon, France, March 1999. International Symposium on Atmospheric Reentry Vehicles and Systems.

[33] P. N. Desai, R. A. Mitcheltree, and F. McNeil Cheatwood. Sample Returns Missions in the Coming Decade. Rio de Janeiro, Brazil, 2000. 51$^{\text{st}}$ International Astronautical Congress. IAF-00-Q.2.04.

[34] D. C. Dracopoulos. Evolutionary Learning Algorithms for Neural Adaptive Control. Perspectives in Neural Computing. Springer, Berlin, Heidelberg, New York, 1997.

[35] L. J. Fogel, A. J. Owens, and M. J. Walsh. Artificial Intelligence Through Simulated Evolution. John Wiley & Sons, Chichester, 1966.

[36] P. Fortescue and J. Stark. Spacecraft Systems Engineering. John Wiley & Sons, Chichester, New York, Brisbane, Second Edition, 1995.

[37] R. L. Forward. Roundtrip Interstellar Travel Using Laser-Pushed Lightsails. Journal of Spacecraft and Rockets, 21(2):187–195, 1984.

[38] R. L. Forward. Statite: A Spacecraft That Does Not Orbit. Journal of Spacecraft and Rockets, 28(5):606–611, 1991.

[39] B. Gladman, P. Michel, and C. Froeschle. The Near-Earth Object Population. Icarus, 146:176–189, 2000.

[40] D. E. Goldberg. Genetic Algorithms in Search, Optimization & Machine Learning. Addison–Wesley, Reading, Menlo Park, Sydney, 1989.

[41] R. Grard and A. Balogh. Returns to Mercury: Science and mission objectives. Planetary and Space Science, 49:1395–1407, 2001.

[42] J. W. Hartmann. Low-Thrust Trajectory Optimization using Stochastic Optimization Techniques. Master Thesis, University of Illinois at Urbana-Champaign, 1996.

[43] J. H. Holland. Adaption in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, Cambridge, London, First MIT Press Edition, 1992.

[44] W. F. Huebner and J. M. Greenberg. Needs for determining material strengths and bulk properties of NEOs. Planetary and Space Science, 48:797–799, 2000.

[45] G. W. Hughes. Low Cost Mercury Orbiter and Sample Return Missions Using Solar Sail Propulsion. Presentation at the RAS Discussion Meeting on Solar Sail Mission Applications, May 2002.

[46] L. Iess and G. Boscagli. Advanced radio science instrumentation for the mission BepiColombo to Mercury. Planetary and Space Science, 49:1597–1608, 2001.

[47] E. K. Jessberger, W. Seboldt, K.-H. Glassmeier, G. Neukum, M. Pätzold, G. Arnold, H.-U. Auster, D. deNiem, F. Guckenbiehl, B. Häusler, G. Hahn, N. Hanowski, A. Harris, H. Hirsch, E. Kührt, M. Leipold, E. Lorenz, H. Michaelis, D. Möhlmann, S. Mottola, D. Neuhaus, H. Palme, H. Rauer, M. Rezazad, L. Richter, D. Stöffler, R. Willnecker, J. Brückner, G. Klingelhöfer, and T. Spohn. ENEAS – Exploration of Near-Earth Asteroids with a Sailcraft. Technical report, August 2000. Proposal for a Small Satellite Mission within the Space Sciences Program of DLR.

[48] S. S. Keerthi and B. Ravindran. A Tutorial Survey of Reinforcement Learning. Technical report, Department of Computer Science and Automation, Indian Institute of Science, Bangalore, 1995.

[49] J. R. Koza. Genetic Programming. MIT Press, Cambridge, London, 1992.

[50] Y. Langevin. Chemical and Solar Electric Propulsion Options for a Cornerstone Mission to Mercury. Acta Astronautica, 47(2-9):443–452, 2000.

[51] M. Leipold. Solar Sail Mission Design. Doctoral Thesis, Lehrstuhl für Flugmechanik und Flugregelung; Technische Universität München, 1999. DLR-FB-2000-22.

[52] M. Leipold. To the Sun and Pluto with Solar Sails and Micro-Sciencecraft. Acta Astronautica, 45(4-9):549–555, 1999.

[53] M. Leipold, M. Eiden, C. E. Garner, L. Herbeck, D. Kassing, T. Niederstadt, T. Krüger, G. Pagel, M. Rezazad, H. Rozemeijer, W. Seboldt, C. Schöppinger, C. Sickinger, and W. Unkenbold. Solar Sail Technology Development and Demonstration. Laurel, USA, 2000. 4^{th} IAA International Conference on Low-Cost Planetary Missions. IAA-L-0707.

[54] M. Leipold, E. Pfeiffer, P. Groepper, M. Eiden, W. Seboldt, L. Herbeck, and W. Unkenbold. Solar Sail Technology for Advanced Space Science Missions. Toulouse, France, 2001. 52^{nd} International Astronautical Congress. IAF-01-S.6.10.

[55] M. Leipold, W. Seboldt, S. Lingner, E. Borg, A. Herrmann, A. Pabsch, O. Wagner, and J. Brückner. Mercury Sun-Synchronous Polar Orbiter with a Solar Sail. Acta Astronautica, 39(1-4):143–151, 1996.

[56] H. J. Leiter, R. Killinger, H. Bassner, J. Müller, R. Kukies, and Th. Fröhlich. RITA – A commercial Radio Frequency Ion Thruster for Satellites and Interplanetary Probes. Versailles, France, May 2002. 6^{th} International Symposium on Propulsion for Space Transportation of the XXI^{st} Century.

[57] A. Likas and I. Lagaris. Training Reinforcement Neurocontrollers using the Polytope Algorithm. Neural Processing Letters, (9):119–127, 1999.

[58] H. W. Loeb, H. Bassner, W. Berry, M. Hechler, M. Leipold, H. Meusemann, E. Noack, G. Schwehm, and W. Seboldt. Electric Propulsion – Ready to Enhance Interplanetary Missions. Paris, France, November 1996. Symposium on Scientific Satellites and Prospects in Europe.

[59] K. E. Mathias and D. Whitley. Initial Performance Comparisons for the Delta Coding Algorithm. Orlando, USA, 1994. First IEEE Conference on Evolutionary Computation.

[60] C. R. McInnes. Solar Sailing. Technology, Dynamics and Mission Applications. Springer–Praxis Series in Space Science and Technology. Springer–Praxis, Berlin, Heidelberg, New York, Chicester, 1999.

[61] Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. Springer, Berlin, Heidelberg, New York, Third, Revised and Extended Edition, 1999.

[62] A. Milani, A. Rossi, D. Vokrouhlicky, D. Villani, and C. Bonanno. Gravity field and rotation state of Mercury from the BepiColombo Radio Science Experiments. Planetary and Space Science, 49:1579–1596, 2001.

[63] D. J. Montana and L. Davies. Training Feedforward Neural Networks Using Genetic Algorithms. Los Altos, USA, 1989. 1989 Joint Conference on Artificial Intelligence.

[64] O. Montenbruck and E. Gill. Satellite Orbits. Models, Methods, and Applications. Springer, Berlin, Heidelberg, New York, 2000.

[65] O. Montenbruck and Th. Pfleger. Astronomie mit dem Personal Computer. Springer, Berlin, Heidelberg, New York, 3. Edition, 1999. (in German).

[66] E. Morrow, D. J. Scheeres, and D. Lubin. Solar Sail Orbit Operations at Asteroids. Journal of Spacecraft and Rockets, 38(2):279–286, 2001.

[67] S. Mottola and F. Lahulla. Mutual Eclipse Events in Asteroidal Binary System 1996FG_3: Observations and a Numerical Model. Icarus, 146:556–567, 2000.

[68] D. M. Murphy. The Scarlet Solar Array: Technology Validation and Flight Results. DS1 Technology Validation Report. JPL Publication 00-10, 10/2000, NASA/JPL, 2000.

[69] D. M. Murphy, T. W. Murphy, and P. A. Gierow. Scalable Solar Sail Subsystem Design Considerations. Denver, USA, April 2002. AIAA 43^{rd} Structures, Structural Dynamics, and Materials Conference. AIAA-2002-1703.

[70] C. D. Murray and S. F. Dermott. Solar System Dynamics. Cambridge University Press, Cambridge, New York, Melbourne, 1999.

[71] P. Pravec, L. Sarounova, D. L. Rabinowitz, M. D. Hicks, M. Wolf, Y. N. Krugly, F. P. Velichko, V. G. Shevchenko, V. G. Chiorny, N. M. Gaftonyuk, and G. Genevier. Two-Period Lightcurves of 1996FG_3, 1998PG and (5407) 1992AX: One Probable and Two Possible Binary Asteroids. Icarus, 146:190–203, 2000.

[72] Mars Global Surveyor Project. Mission Plan. JPL Publication MGS 542-405, NASA/JPL, 1996. Final Version, Rev. B.

[73] G. D. Racca. Capability of solar electric propulsion for planetary missions. Planetary and Space Science, 49:1437–1444, 2001.

[74] I. Rechenberg. Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog Verlag, Stuttgart, 1973. (in German).

[75] R. Rojas. Was können neuronale Netze? In: R.-H. Schulz, editor, Mathematische Aspekte der angewandten Informatik, pages 55–88. Wissenschaftsverlag, Mannheim, 1994. (in German).

[76] R. Rojas. Neural Networks. A Systematic Introduction. Springer, Berlin, Heidelberg, New York, 1996.

[77] I. M. Ross and F. Fahroo. A Perspective on Methods for Trajectory Optimization. Monterey, USA, August 2002. AIAA/AAS Astrodynamics Specialist Conference. AIAA-2002-4727.

[78] R. Sachs. Interplanetare Transferbahnen für Sonnensegler-Missionen. Diploma Thesis, DLR, Forschungsbereich Flugmechanik / Flugführung, Hauptabteilung Systemanalyse Raumfahrt, October 1994. (in German).

[79] A. G. Santo, R. E. Gold, R. L. McNutt Jr., S. C. Solomon, C. J. Ercol, R. W. Farquhar, T. J. Hartka, J. E. Jenkins, J. V. McAdams, L. E. Mosher, D. F. Persons, D. A. Artis, R. S. Bokulic, R. F. Conde, G. Dakermanji, M. E. Goss, D. R. Haley, K. J. Heeres, R. H. Maurer, R. C. Moore, E. H. Rodberg, T. G. Stern, S. R. Wiley, B. G. Williams, C. L. Yen, and M. R. Peterson. The MESSENGER mission to Mercury: Spacecraft and mission design. Planetary and Space Science, 49:1481–1500, 2001.

[80] C. G. Sauer. Optimum Solar-Sail Interplanetary Trajectories. San Diego, USA, August 1976. AIAA/AAS Astrodynamics Conference. 76-792.

[81] T. D. Schmidt, B. Dachwald, W. Seboldt, and M. Auweter-Kurtz. Flight Opportunities from Mars to Earth for Piloted Missions Using Continuous Thrust Propulsion. Huntsville, USA, July 2003. 39$^{\text{th}}$ AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit. AIAA-2003-4573.

[82] T. D. Schmidt, W. Seboldt, and M. Auweter-Kurtz. Propulsion Options for Manned Mars Missions. Versailles, France, May 2002. 6$^{\text{th}}$ International Symposium on Propulsion for Space Transportation of the XXI$^{\text{st}}$ Century.

[83] H.-P. Schwefel. Numerical Optimization for Computer Models. John Wiley & Sons, Chichester, 1981.

[84] W. Seboldt and B. Dachwald. Solar Sail (WP 01-80). Technical report, ESA Propulsion 2000 Program (Phase II), 2003. Final Report Issue 1.

[85] W. Seboldt and B. Dachwald. Solar Sailcraft of the First Generation: Technology Development. Bremen, Germany, September/October 2003. 54$^{\text{th}}$ International Astronautical Congress. IAC-03-S.6.03.

[86] W. Seboldt, M. Leipold, M. Rezazad, L. Herbeck, W. Unkenbold, D. Kassing, and M. Eiden. Ground-based Demonstration of Solar Sail Technology. Rio de Janeiro, Brazil, 2000. 51$^{\text{st}}$ International Astronautical Congress. IAF-00-S.6.11.

[87] S. C. Solomon, R. L. McNutt Jr., R. E. Gold, M. H. Acuna, D. N. Baker, W. V. Boynton, C. R. Chapman, A. F. Cheng, G. Gloeckler, J. W. Head III, S. M. Krimigis, W. E. McKlintock, S. L. Murchie, S. J. Peale, R. J. Phillips, M. S. Robinson, J. A. Slavin, D. E. Smith, R. G. Strom, J. I. Trombka, and M. T. Zuber. The MESSENGER mission to Mercury: Scientific objectives and implementation. Planetary and Space Science, 49:1445–1465, 2001.

[88] R. F. Stengel. Optimal Control and Estimation. Dover Books on Mathematics. Dover Publications, Inc., New York, 1994.

[89] R. S. Sutton and A. G. Barto. Reinforcement Learning. MIT Press, Cambridge, London, 1998.

[90] G. Syswerda. Uniform Crossover in Genetic Algorithms. San Mateo, USA, 1989. Third International Conference on Genetic Algorithms.

[91] L. Tsinas and B. Dachwald. In: . Proceedings of the 1^{s}t IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, 27–29 June 1994, Orlando, USA, volume 2, pages 770–774, Piscataway (NJ), USA, 1994. IEEE.

[92] M. Vasile. A Global Approach to Optimal Space Trajectory Design. Ponce, Puerto Rico, February 2002. AAS/AIAA Space Flight Mechanics Meeting. AAS 03-141.

[93] P. D. Ward and D. Brownlee. Rare Earth. Why Complex Life Is Uncommon in the Universe. Copernicus, New York, 2000.

[94] D. Whitley. An Overview of Evolutionary Algorithms: Practical Issues an Common Pitfalls. Journal of Information and Software Technology, (43):817–831, 2001.

[95] D. Whitley, S. Dominic, R. Das, and C. W. Anderson. Genetic Reinforcement Learning for Neurocontrol Problems. Machine Learning, 13:259–284, 1993.

[96] D. Whitley, K. Mathias, and P. Fitzhorn. Delta coding: An iterative search strategy for genetic algorithms. San Mateo, USA, 1991. Fourth International Conference on Genetic Algorithms.

[97] S. Williams and V. Coverstone-Carroll. Mars missions using Solar Electric Propulsion. Journal of Spacecraft and Rockets, 37(1):71–77, 2000.

[98] J. L. Wright. Space Sailing. Gordon and Breach Science Publishers, Philadelphia, 1992.

Appendices

A Reference Frames

This appendix outlines the three heliocentric reference frames that are used throughout this work for describing the interplanetary translational motion of spacecraft. These reference frames are

- an inertial cartesian reference frame $\mathcal{J}$ (section A.1),
- an ecliptic spherical reference frame $\mathcal{E}$ (section A.2) and
- an orbit polar reference frame $\mathcal{O}$ (section A.3)

The origin of all reference frames is the position of the sun's point mass.[1] Apart from those reference frames, a body's orbital motion in space is commonly described by a set of six so-called orbital elements (section A.4).

A.1 Inertial Cartesian Reference Frame

The reference frame $\mathcal{J} : (\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ is an inertial right-handed cartesian coordinate frame, which is defined with respect to the mean ecliptic and equinox of J2000 at the J2000 epoch [70]. The spacecraft's position, velocity, and acceleration in $\mathcal{J}$-frame components are

$$\mathbf{r} = x\mathbf{e}_x + y\mathbf{e}_y + z\mathbf{e}_z = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{A.1}$$

$$\dot{\mathbf{r}} = \dot{x}\mathbf{e}_x + \dot{y}\mathbf{e}_y + \dot{z}\mathbf{e}_z = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \tag{A.2}$$

$$\ddot{\mathbf{r}} = \ddot{x}\mathbf{e}_x + \ddot{y}\mathbf{e}_y + \ddot{z}\mathbf{e}_z = \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} \tag{A.3}$$

A.2 Ecliptic Reference Frame

Due to the nature of the problem, the translational motion of spacecraft is better described in an ecliptic spherical reference frame $\mathcal{E}$. The reference frame $\mathcal{E} : (\mathbf{e}_r, \mathbf{e}_\varphi, \mathbf{e}_\theta)$ is an orthogonal right-handed spherical coordinate frame, which is defined according to figure A.1. $\mathbf{e}_r$ points always along the sun-spacecraft line, $\mathbf{e}_\theta$ lies in the $\mathbf{e}_r$-$\mathbf{e}_z$-plane and points along the direction of increasing θ, and $\mathbf{e}_\varphi$ completes the right-handed coordinate system ($\mathbf{e}_r \times \mathbf{e}_\varphi = \mathbf{e}_\theta$). The **azimuth angle** φ is the angle between $\mathbf{e}_x$ and the projection of $\mathbf{r}$ into the ecliptic ($\mathbf{e}_x$-$\mathbf{e}_y$-plane). The **elevation angle** θ is the angle between the ecliptic and $\mathbf{r}$. Expressed in $\mathcal{J}$-frame

[1] which is assumed to be in rest, since the sun's motion through the interstellar medium and the gravitational forces by bodies other than the sun are neglected

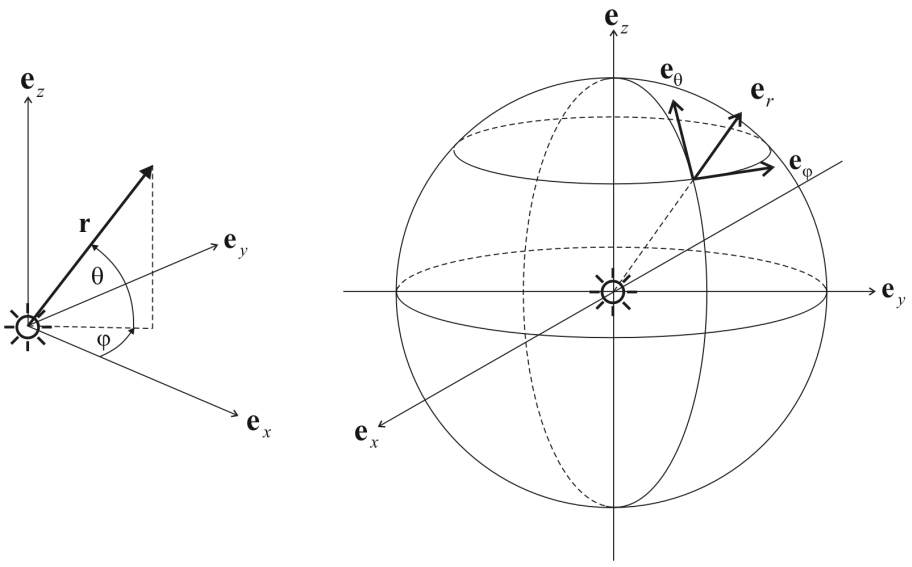


Figure A.1 Ecliptic reference frame

components, the unit vectors are

$$\begin{aligned}
\mathbf{e}_r &= \begin{pmatrix} \cos\varphi\cos\theta \\ \sin\varphi\cos\theta \\ \sin\theta \end{pmatrix} \\
\mathbf{e}_\varphi &= \begin{pmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{pmatrix} \\
\mathbf{e}_\theta &= \begin{pmatrix} -\cos\varphi\sin\theta \\ -\sin\varphi\sin\theta \\ \cos\theta \end{pmatrix}
\end{aligned} \tag{A.4}$$

The spacecraft's position, velocity, and acceleration in $\mathcal{E}$-frame components are

$$\mathbf{r} = r\mathbf{e}_r \tag{A.5}$$

$$\dot{\mathbf{r}} = \dot{r}\mathbf{e}_r + r\dot{\varphi}\cos\theta\mathbf{e}_\varphi + r\dot{\theta}\mathbf{e}_\theta \tag{A.6}$$

$$\begin{aligned}
\ddot{\mathbf{r}} = {} & (\ddot{r} - r\dot{\theta}^2 - r\dot{\varphi}^2\cos^2\theta)\mathbf{e}_r + \\
& + (2\dot{r}\dot{\varphi}\cos\theta + r\ddot{\varphi}\cos\theta - 2r\dot{\varphi}\dot{\theta}\sin\theta)\mathbf{e}_\varphi + \\
& + (2\dot{r}\dot{\theta} + r\ddot{\theta} + r\dot{\varphi}^2\sin\theta\cos\theta)\mathbf{e}_\theta
\end{aligned} \tag{A.7}$$

A.3 Orbit Reference Frame

It is very convenient to describe the translational motion of spacecraft in $\mathcal{E}$-frame coordinates, but the thrust vector of spacecraft is better described in a reference frame that has two unit vectors within the (osculating) orbital plane. For that purpose, the reference frame $\mathcal{O} : (\mathbf{e}_r, \mathbf{e}_t, \mathbf{e}_h)$, an orthogonal right-handed polar coordinate frame, is defined according to figure A.2. $\mathbf{e}_r$ points always along the sun-spacecraft line, $\mathbf{e}_h$ is the orbit plane normal (pointing along the spacecraft's orbital angular momentum vector), and $\mathbf{e}_t$ completes the right-handed coordinate system ($\mathbf{e}_r \times \mathbf{e}_t = \mathbf{e}_h$).

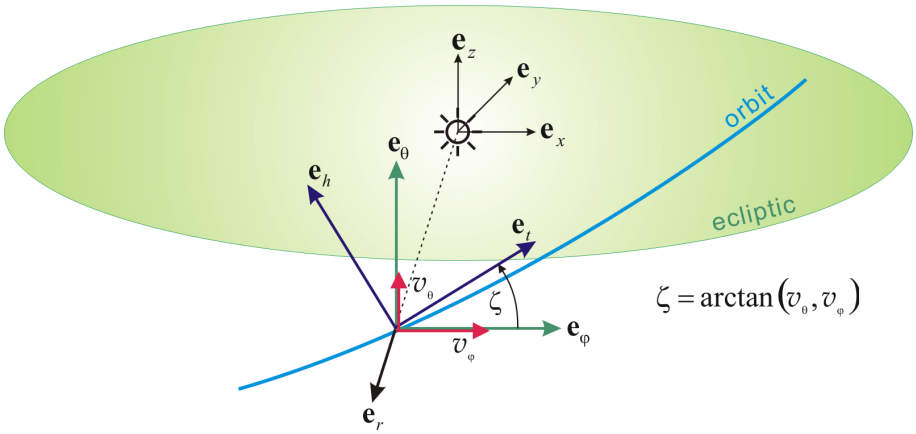


Figure A.2 Orbit reference frame

The $\mathcal{O}$-frame can be created from the $\mathcal{E}$-frame by a rotation of

$$\zeta = \arctan(v_\theta, v_\varphi) = \arctan(\dot{\theta}, \dot{\varphi}\cos\theta) \tag{A.8}$$

about $\mathbf{e}_r$, where $\arctan(y, x)$ is an extended arcustangens, which gives the angle ϕ such that $x = \cos\phi$ and $y = \sin\phi$. Expressed in $\mathcal{J}$-frame components, the unit vectors are

$$\begin{aligned}
\mathbf{e}_r &= \begin{pmatrix} \cos\varphi\cos\theta \\ \sin\varphi\cos\theta \\ \sin\theta \end{pmatrix} \\
\mathbf{e}_t = \cos\zeta\mathbf{e}_\varphi + \sin\zeta\mathbf{e}_\theta &= \begin{pmatrix} -\sin\varphi\cos\zeta - \cos\varphi\sin\theta\sin\zeta \\ \cos\varphi\cos\zeta - \sin\varphi\sin\theta\sin\zeta \\ \cos\theta\sin\zeta \end{pmatrix} \\
\mathbf{e}_h = -\sin\zeta\mathbf{e}_\varphi + \cos\zeta\mathbf{e}_\theta &= \begin{pmatrix} \sin\varphi\sin\zeta - \cos\varphi\sin\theta\cos\zeta \\ -\cos\varphi\sin\zeta - \sin\varphi\sin\theta\cos\zeta \\ \cos\theta\cos\zeta \end{pmatrix}
\end{aligned} \tag{A.9}$$

A.4 Orbital Elements

In astrodynamics a set of six so-called orbital elements is commonly used for describing a body's orbital motion in space (see figure A.3).

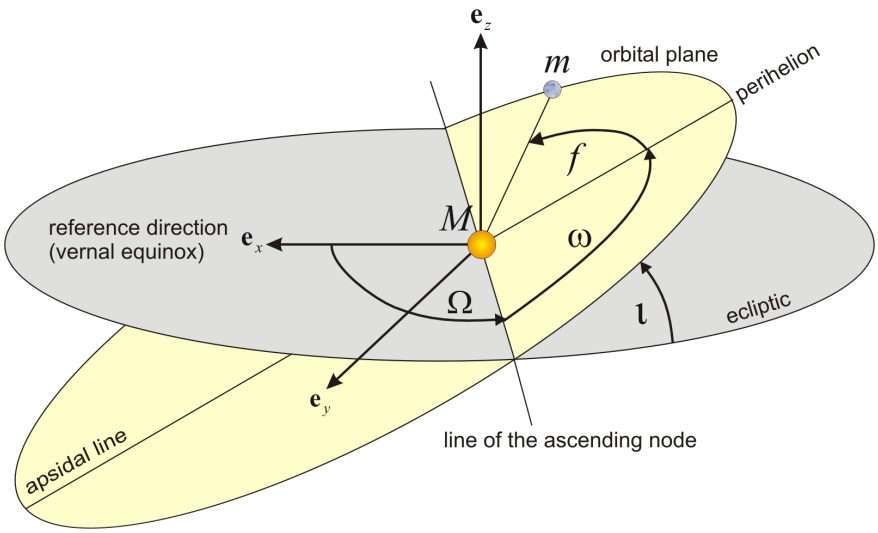


Figure A.3 Orbital motion of two bodies (mass M and $m \ll M$) in three-dimensional space

Three orbital elements describe the body's motion within the orbital plane. Usually the following three elements are used for this purpose:

- The **semi-major axis** a denotes the size of the orbit,
- the **eccentricity** e denotes the shape of the orbit, and
- the **true anomaly** f denotes the position on the orbit (the angle of the apsidal line to the body's position vector).

The position on the orbit can also be expressed by the **true longitude** θ, the **mean longitude** Λ, the **mean anomaly** M, and the **eccentric anomaly** E.[2]

The orientation of the orbital plane with respect to the inertial cartesian reference frame $\mathcal{J}$ is usually defined by the three **Euler**-angles Ω, ι, and ω:

- Ω denotes the **longitude of the ascending node**, as measured from the vernal equinox,
- ι denotes the orbital plane's angle of **inclination** to the ecliptic, and
- ω denotes the angle of the apsidal line to the direction of the ascending node, called the **argument of perihelion**

Instead of ω the **longitude of perihelion** $\varpi = \Omega + \omega$ is widely used.

[2] see e.g. [70] or [13] for their definition and computation

B NSTAR Cluster Control Strategies

If an SEP spacecraft employs more than one electric thruster, the available power has to be shared between the n_{Thr} thrusters. For the NSTAR thruster, three ways to do this – three NSTAR cluster control strategies, A, B, and C – can be easily imagined.

NSTAR Cluster Control Strategy A: If $\chi P_{\mathrm{av}}(r) < n_{\mathrm{Thr}} \cdot P_{\mathrm{min}}$, all thrusters are switched off. If $\chi P_{\mathrm{av}}(r) > n_{\mathrm{Thr}} \cdot P_{\mathrm{max}}$, all thrusters run on full power level $P_{\mathrm{PPU}} = P_{\mathrm{max}}$.
If $n_{\mathrm{Thr}} \cdot P_{\mathrm{min}} \leq \chi P_{\mathrm{av}}(r) \leq n_{\mathrm{Thr}} \cdot P_{\mathrm{max}}$, the power is equally shared between the thrusters.

$$P_{\mathrm{PPU}}(\chi, r) = \frac{\chi P_{\mathrm{av}}(r)}{n_{\mathrm{Thr}}} \tag{B.1}$$

Thus, using equations (2.49),

$$\dot{m}_{\mathrm{P}}(\chi, r) = 0.74343 \cdot n_{\mathrm{Thr}} + 0.20951 \cdot \chi P_{\mathrm{av}}(r) + 0.25205 \frac{\chi^2 P_{\mathrm{av}}^2(r)}{n_{\mathrm{Thr}}} \tag{B.2a}$$

$$F(\chi, r) = -3.4318 \cdot n_{\mathrm{Thr}} + 37.365 \cdot \chi P_{\mathrm{av}}(r) \tag{B.2b}$$

NSTAR Cluster Control Strategy B: If $\chi P_{\mathrm{av}}(r) < P_{\mathrm{min}}$, all thrusters are switched off. If $\chi P_{\mathrm{av}}(r) > n_{\mathrm{Thr}} \cdot P_{\mathrm{max}}$, all thrusters run on full power level $P_{\mathrm{PPU}} = P_{\mathrm{max}}$.
If $P_{\mathrm{min}} \leq \chi P_{\mathrm{av}}(r) \leq n_{\mathrm{Thr}} \cdot P_{\mathrm{max}}$, a number

$$n_{\mathrm{Thr,full}}(\chi, r) = \mathrm{floor}\left(\frac{\chi P_{\mathrm{av}}(r)}{n_{\mathrm{Thr}} \cdot P_{\mathrm{max}}}\right) \tag{B.3}$$

of thrusters[1] run on full power level P_{max} and one thruster runs with the remaining power

$$P_{\mathrm{PPU,rem}}(\chi, r) = \chi P_{\mathrm{av}}(r) - n_{\mathrm{Thr,full}}(\chi, r) \cdot P_{\mathrm{max}} \tag{B.4}$$

as long as $P_{\mathrm{PPU,rem}}(\chi, r) > P_{\mathrm{min}}$. Thus, using equations (2.49),

$$\begin{aligned} \dot{m}_{\mathrm{P}}(\chi, r) &= 2.1707 \cdot n_{\mathrm{Thr,full}}(\chi, r) \\ &\quad + 0.74343 + 0.20951 \cdot P_{\mathrm{PPU,rem}}(\chi, r) + 0.25205 \cdot P_{\mathrm{PPU,rem}}^2(\chi, r) \end{aligned} \tag{B.5a}$$

$$\begin{aligned} F(\chi, r) &= 71.298 \cdot n_{\mathrm{Thr,full}}(\chi, r) \\ &\quad - 3.4318 + 37.365 \cdot P_{\mathrm{PPU,rem}}(\chi, r) \end{aligned} \tag{B.5b}$$

NSTAR Cluster Control Strategy C: If $\chi P_{\mathrm{av}}(r) < P_{\mathrm{min}}$, all thrusters are switched off. If $\chi P_{\mathrm{av}}(r) > n_{\mathrm{Thr}} \cdot P_{\mathrm{max}}$, all thrusters run on full power level $P_{\mathrm{PPU}} = P_{\mathrm{max}}$.
If $P_{\mathrm{min}} \leq \chi P_{\mathrm{av}}(r) \leq n_{\mathrm{Thr}} \cdot P_{\mathrm{max}}$, a number

$$n_{\mathrm{Thr,opt}}(\chi, r) = \mathrm{floor}\left(\frac{\chi P_{\mathrm{av}}(r)}{n_{\mathrm{Thr}} \cdot P_{\mathrm{opt}}}\right) \tag{B.6}$$

[1] floor(x) is the largest integer number that is smaller than x

of thrusters could be run on the optimal power level $P_{\mathrm{opt}} = 1.8337\,\mathrm{kW}$, where $\mathrm{I_{sp}} = \mathrm{I_{sp_{max}}}$. However, if $\chi P_{\mathrm{av}}(r) > n_{\mathrm{Thr,opt}}(\chi, r) \cdot P_{\mathrm{max}}$, one more thruster must be used[2]:

$$n_{\mathrm{Thr,opt}}(\chi, r) = \mathrm{ceil}\left(\frac{\chi P_{\mathrm{av}}(r)}{n_{\mathrm{Thr}} \cdot P_{\mathrm{opt}}}\right) \tag{B.7}$$

Now, $\chi P_{\mathrm{av}}(r)$ is split among the $n_{\mathrm{Thr,opt}}(\chi, r)$ thrusters, i.e.

$$P_{\mathrm{PPU}}(\chi, r) = \frac{\chi P_{\mathrm{av}}(r)}{n_{\mathrm{Thr,opt}}(\chi, r)} \tag{B.8}$$

Thus, using equations (2.49),

$$\dot{m}_{\mathrm{P}}(\chi, r) = 0.74343 \cdot n_{\mathrm{Thr,opt}}(\chi, r) + 0.20951 \cdot \chi P_{\mathrm{av}}(r) + 0.25205 \frac{\chi^2 P_{\mathrm{av}}^2(r)}{n_{\mathrm{Thr,opt}}(\chi, r)} \tag{B.9a}$$

$$F(\chi, r) = -3.4318 \cdot n_{\mathrm{Thr,opt}}(\chi, r) + 37.365 \cdot \chi P_{\mathrm{av}}(r) \tag{B.9b}$$

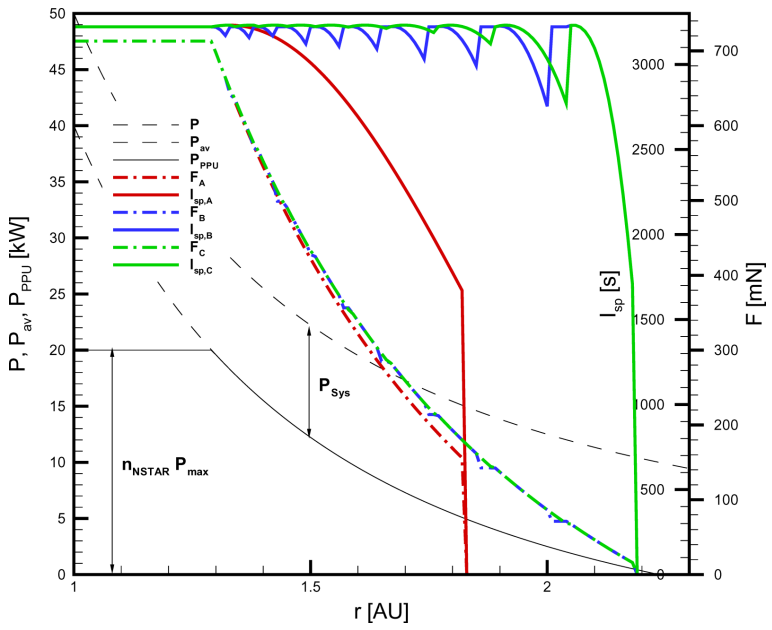


Figure B.1 Dependence of power levels, thrust, and specific impulse on solar distance for a cluster of NSTAR thrusters ($n_{\mathrm{Thr}} = 10$, $\varkappa = 2$, $P_{\mathrm{Sys}} = 10\,\mathrm{kW}$, NSTAR cluster control strategies A, B, and C, see text)

Figure B.1 shows that control strategy A is not reasonable, since the specific impulse decreases rapidly for $P_{\mathrm{av}} < n_{\mathrm{Thr}} \cdot P_{\mathrm{max}}$ and cuts off for $r \gtrsim 1.83\,\mathrm{AU}$. Control strategies B and C give much better results, those of control strategy C being slightly better. There might be slightly better NSTAR cluster control strategies than the ones elaborated herein. However, for the objectives of this work, control strategy C is acceptable and has been used for all calculations involving NSTAR clusters.

[2] if the excessive power should not be discarded; ceil(x) is the smallest integer number that is larger than x

C Locally Optimal Spacecraft Steering

Semi-major Axis: To increase the semi-major axis a with a maximum rate, the thrust vector has to be along the direction defined by

$$\mathbf{k}_a = e \sin f \mathbf{e}_r + \frac{p}{r}\mathbf{e}_t \tag{C.1}$$

To decrease a with a maximum rate, the thrust vector has to be along the direction defined by $-\mathbf{k}_a$.

Eccentricity: To increase the eccentricity e with a maximum rate, the thrust vector has to be along the direction defined by

$$\mathbf{k}_e = p \sin f \mathbf{e}_r + [(p + r)\cos f + re]\mathbf{e}_t \tag{C.2}$$

To decrease e with a maximum rate, the thrust vector has to be along the direction defined by $-\mathbf{k}_e$.

Inclination: To increase the inclination ι with a maximum rate, the thrust vector has to be along the direction defined by

$$\mathbf{k}_\iota = \cos(\omega + f)\mathbf{e}_h \tag{C.3}$$

To decrease ι with a maximum rate, the thrust vector has to be along the direction defined by $-\mathbf{k}_\iota$.

Longitude of the Ascending Node: To increase the longitude of the ascending node Ω with a maximum rate, the thrust vector has to be along the direction defined by

$$\mathbf{k}_\Omega = \sin(\omega + f)\mathbf{e}_h \tag{C.4}$$

To decrease Ω with a maximum rate, the thrust vector has to be along the direction defined by $-\mathbf{k}_\Omega$. Equation (C.4) is only valid for inclined ($\iota \neq 0$) orbits, since $|\dot{\Omega}| \to \infty$ for $\iota \to 0$.

Argument of Perihelion: To increase the argument of perihelion ω with a maximum rate, the thrust vector has to be along the direction defined by

$$\mathbf{k}_\omega = -\frac{p\cos f}{e}\mathbf{e}_r + \frac{(p + r)\sin f}{e}\mathbf{e}_t - \frac{r\sin(\omega + f)\cos\iota}{\sin\iota}\mathbf{e}_h \tag{C.5}$$

To decrease ω with a maximum rate, the thrust vector has to be along the direction defined by $-\mathbf{k}_\omega$. Equation (C.5) is only valid for non-circular ($e \neq 0$) inclined ($\iota \neq 0$) orbits, since $|\dot{\omega}| \to \infty$ for $e \to 0$ and for $\iota \to 0$.

True Anomaly: For a change of the true anomaly with a maximum rate, no suitable local steering law can be derived. This is reasonable, if equation (3.3f) is examined: It can be seen that even in the absence of a disturbing and/or propulsive acceleration acting on the body, the true anomaly is changing with $\dot{f} = h/r^2$. Apart from this term, $\dot{f}$ can only be changed by an in-plane acceleration. However, any in-plane acceleration results in $\dot{\omega} = -\dot{f}$, so that $\dot{\varphi} = \dot{\Omega} + \dot{\omega} + \dot{f}$ remains unchanged. Thus, increasing/decreasing $\dot{\varphi}$ is better done by increasing/descreasing h/r^2, i.e. by spiralling inwards/outwards.

D InTrance Program Overview

To give a detailed program description of the "InTrance" program (Intelligent Trajectory optimization using neurocontroller evolution) can not be within the scope of this work but of a (not yet existing) InTrance user manual. However, this appendix gives a short overview of the implemented program functionalities (section D.1), the input files that are required for trajectory optimization and the output files that are produced by the program (section D.2).

InTrance is written with Microsoft Visual C++ 6.0 to run under the DOS environment of any Microsoft Windows operating system. Nevertheless, only standard C++ and no PC- or Windows-specific functions are used, so that InTrance should also compile without any modification under all operating systems for which a C++ compiler is available. At the bottom of InTrance, extended versions of the astrodynamical and mathematical routines (vectors, matrices, numerical integration methods, etc.) provided by MONTENBRUCK et al. [64, 65] are used.

D.1 Functionalities

InTrance implements the following functionalities that can be selected through the InTrance input file (see section D.2):

- optimize an interplanetary rendezvous, fly-by, or orbit transfer trajectory for a solar sail, SEP, or NEP spacecraft
- evaluate a given chromosome
- evaluate the performance of a given NC for other launch dates
- evaluate the performance of a given NC under input noise, network noise, and disturbing forces that are acting on the spacecraft
- evaluate a control vector history

The command to run the program is simply:

`InTrance` <*InTrance input file*>.

D.2 Input and Output Files

D.2.1 Overview

All program parameters are passed to the program through a set of input files that are specified in the main "InTrance input file". This allows the subsequent execution of different tasks in batch mode. Figure D.1 gives a sketch for the most important functionality: trajectory optimization.

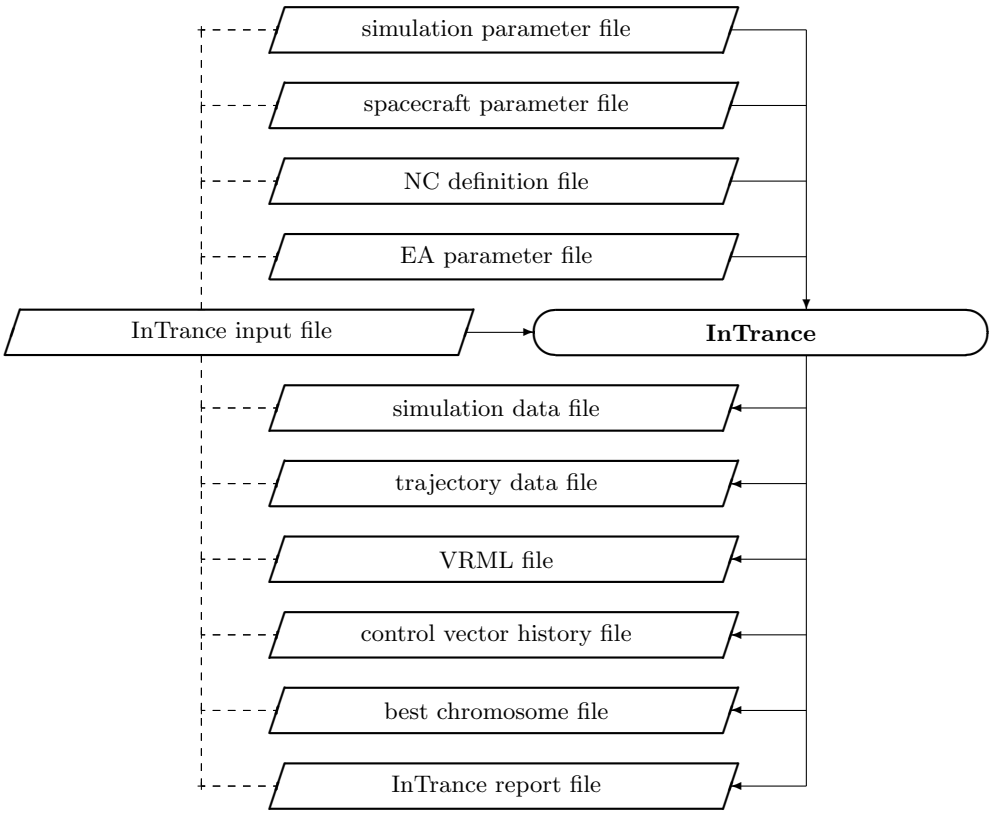


Figure D.1 InTrance input and output files for trajectory optimization

D.2.2 The InTrance Input File

The InTrance input file determines the functionality that has to be executed by the InTrance program and specifies the names of the required input and output files. An example file is printed below. All parameters and options are explained by the comments that are included in the file.

```
// purpose of this file
InTrance 1.1 input file
// seed for C++ (pseudo-)random number generation
1
// input parameters
// command (must be
//    (1) "optimize NC" or
//    (2) "evaluate NC" or
//    (3) "evaluate NC LDF" or
//    (4) "evaluate NC input noise" or
//    (5) "evaluate NC network noise" or
//    (6) "evaluate disturbing acceleration" or
//    (7) "evaluate control vector history")
// ===== in case of "optimize NC" =====
//    <line  1> "optimize NC"
//   input files:
//    <line  2> name of the simulation parameter file
//    <line  3> name of the spacecraft parameter file
//    <line  4> name of the NC definition file
//    <line  5> name of the EA parameter file
//    <line  6> in case of a warm start: name of the chromosome file
//              in case of a cold start: "cold start"
//   output files:
//    <line  7> name of the simulation data file (*.csv)
//    <line  8> name of the trajectory data file (*.dat)
//    <line  9> name of the VRML file (*.wrl)
//    <line 10> name of the control vector history file
//    <line 11> name of the best chromosome file
//    <line 12> name of the InTrance report file
// ===== in case of "evaluate NC" =====
//    <line  1> "evaluate NC"
//   input files:
//    <line  2> name of the simulation parameter file
//    <line  3> name of the spacecraft parameter file
//    <line  4> name of the NC definition file
//    <line  5> name of the EA parameter file
//    <line  6> name of the chromosome file
//   output files:
//    <line  7> name of the simulation data file (*.csv)
//    <line  8> name of the trajectory data file (*.dat)
//    <line  9> name of the VRML file (*.wrl)
//    <line 10> name of the control vector history file
// ===== in case of "evaluate NC temporal" =====
//    <line  1> "evaluate NC LDF"
//    <line  2> number of time steps
//    <line  3> number of evaluations per step
//   input files:
//    <line  4> name of the simulation parameter file
//    <line  5> name of the spacecraft parameter file
//    <line  6> name of the NC definition file
//    <line  7> name of the EA parameter file
//    <line  8> name of the chromosome file
```

```
//   output file:
//    <line  9> name of the evaluation output file (*.csv)
// ===== in case of "evaluate NC input noise" =====
//    <line  1> "evaluate NC input noise"
//    <line  2> minimum NC input noise
//    <line  3> maximum NC input noise
//    <line  4> number of steps
//    <line  5> number of evaluations per step
//   input files:
//    <line  6> name of the simulation parameter file
//    <line  7> name of the spacecraft parameter file
//    <line  8> name of the NC definition file
//    <line  9> name of the EA parameter file
//    <line 10> name of the chromosome file
//   output file:
//    <line 11> name of the evaluation output file (*.csv)
// ===== in case of "evaluate NC network noise" =====
//    <line  1> "evaluate NC network noise"
//    <line  2> minimum NC network noise
//    <line  3> maximum NC network noise
//    <line  4> number of steps
//    <line  5> number of evaluations per step
//   input files:
//    <line  6> name of the simulation parameter file
//    <line  7> name of the spacecraft parameter file
//    <line  8> name of the NC definition file
//    <line  9> name of the EA parameter file
//    <line 10> name of the chromosome file
//   output file:
//    <line 11> name of the evaluation output file (*.csv)
// ===== in case of "evaluate disturbing acceleration" =====
//    <line  1> "evaluate disturbing acceleration"
//    <line  2> minimum disturbing acceleration
//    <line  3> maximum disturbing acceleration
//    <line  4> number of steps
//    <line  5> number of evaluations per step
//   input files:
//    <line  6> name of the simulation parameter file
//    <line  7> name of the spacecraft parameter file
//    <line  8> name of the NC definition file
//    <line  9> name of the EA parameter file
//    <line 10> name of the chromosome file
//   output file:
//    <line 11> name of the evaluation output file (*.csv)
// ===== in case of "evaluate control vector history" =====
//    <line  1> "evaluate control vector history"
//   input files:
//    <line  2> name of the simulation parameter file
//    <line  3> name of the spacecraft parameter file
//    <line  4> name of the control vector history file
//   output files:
//    <line  5> name of the simulation data file (*.csv)
//    <line  6> name of the trajectory data file (*.dat)
//    <line  7> name of the VRML file (*.wrl)
optimize NC
default.sim
default.scp
default.top
default.eap
cold start
default.csv
default.dat
default.wrl
default.ang
default.eac
default.rep
// don't change these comments!
```

D.2.3 The Simulation Parameter File

The simulation parameter file determines the simulation parameters and specifies the optimization problem. An example file is printed below. All parameters and options are explained by the comments that are included in the file.

```
// purpose of this file
InTrance 1.1 simulation parameter file
// minimum simulation start time (earliest launch date) [modified julian date]
52654.5
// maximum simulation start time (latest launch date) [modified julian date]
52654.5
// integration interval [days]
600.0
// number of equidistant output steps [days]
30
// minimum number of output points
600
// initial state (must be
//    (1) "body" or
//    (2) "orbit")
// ===== in case of "body" =====
//    <line  1> "body"
//    <line  2> name of initial body (e.g. "earth")
//    <line  3> minimum launch velocity [km/s]
//    <line  4> maximum launch velocity [km/s]
//    <line  5> minimum azimuth of launch velocity vector [rad]
//    <line  6> maximum azimuth of launch velocity vector [rad]
//    <line  7> minimum elevation of launch velocity vector [rad]
//    <line  8> maximum elevation of launch velocity vector [rad]
// ===== in case of "orbit" =====
//    <line  1> "orbit"
//    <line  2> a (semi-major axis) [AU]
//    <line  3> e (eccentricity)
//    <line  4> i (inclination) [rad]
//    <line  5> Omega (longitude of the ascending node) [rad]
//    <line  6> omega (argument of perihelion) [rad]
//    <line  7> M (mean anomaly) [rad]
//    <line  8> minimum launch velocity [km/s]
//    <line  9> maximum launch velocity [km/s]
//    <line 10> minimum azimuth of launch velocity vector [rad]
//    <line 11> maximum azimuth of launch velocity vector [rad]
//    <line 12> minimum elevation of launch velocity vector [rad]
//    <line 13> maximum elevation of launch velocity vector [rad]
body
earth
0.0
0.0
```

```
0.0
0.0
0.0
0.0
// target state (must be
//    (1) "body rendezvous" or
//    (2) "body fly-by" or
//    (3) "body orbit rendezvous" or
//    (4) "body orbit fly-by" or
//    (5) "free orbit rendezvous" or
//    (6) "free orbit fly-by")
// ===== in case of "body rendezvous" =====
//    <line 1> "body rendezvous"
//    <line 2> name of target body (e.g. "mars")
//    <line 3> maximum final distance to target body [km]
//    <line 4> maximum final relative velocity to target body [km/s]
// ===== in case of "body fly-by" =====
//    <line 1> "body fly-by"
//    <line 2> name of target body (e.g. "mars")
//    <line 3> maximum final distance to target body [km]
// ===== in case of "body orbit rendezvous" =====
//    <line 1> "body orbit rendezvous"
//    <line 2> name of target orbit (e.g. "mars")
//    <line 3> maximum final distance to (virtual) target body in specified orbit [km]
//    <line 4> maximum final relative velocity to (virtual) target body in specified orbit [km/s]
// ===== in case of "body orbit fly-by" =====
//    <line 1> "body orbit fly-by"
//    <line 2> name of target orbit (e.g. "mars")
//    <line 3> maximum final distance to (virtual) target body in specified orbit [km]
// ===== in case of "free orbit rendezvous" =====
//    <line 1> "free orbit rendezvous"
//    <line 2> a (semi-major axis) [AU]
//    <line 3> e (eccentricity)
//    <line 4> i (inclination) [rad]
//    <line 5> Omega (longitude of the ascending node) [rad]
//    <line 6> omega (argument of perihelion) [rad]
//    <line 7> maximum final distance to (virtual) target body in specified orbit [km]
//    <line 8> maximum final relative velocity to (virtual) target body in specified orbit [km/s]
// ===== in case of "free orbit fly-by" =====
//    <line 1> "free orbit fly-by"
//    <line 2> a (semi-major axis) [AU]
//    <line 3> e (eccentricity)
//    <line 4> i (inclination) [rad]
//    <line 5> Omega (longitude of the ascending node) [rad]
//    <line 6> omega (argument of perihelion) [rad]
//    <line 7> maximum final distance to (virtual) target body in specified orbit [km]
body rendezvous
mercury
100000.0
0.1
// optimization goal (must be
//    (1) "minimum transfer time" or
//    (2) "minimum propellant mass")
minimum transfer time
// disturbing acceleration (standard deviation) [mm/s^2]
0.0
// minimum solar distance [AU]
0.1
// DES solver (must be
//    (1) "RK4" or
//    (2) "RKF54" or
//    (3) "DOPRI8" or
//    (4) "DESG" or
//    (5) "GJ4P" (only for solar sailcraft))
RKF54
// maximum relative DES solver error
1e-6
// maximum absolute DES solver error
1e-6
// don't change these comments!
```

D.2.4 The Spacecraft Parameter File

The spacecraft parameter file specifies the spacecraft. An example file for a spacecraft with an ideal solar sail and for a spacecraft with SEP are printed below. All parameters and options are explained by the comments that are included in the file.

```
// purpose of this file
InTrance 1.1 spacecraft parameter file
// type of spacecraft (must be
//    (1) "solar sail" or
//    (2) "NEP" or
//    (3) "SEP")
```

```
// ===== in case of "solar sail" ====
//    <line 1> "solar sail"
//    <line 2> "lightness number" or "characteristic acceleration"
//             ----- in case of "lightness number" -----
//                <line 2> "lightness number"
//                <line 3> lightness number
//             ----- in case of "characteristic acceleration" -----
//                <line 2> "characteristic acceleration"
//                <line 3> characteristic acceleration [mm/s^2]
//    <line 4> optical sail model (must be "ideal sail" or "real sail")
//             ----- in case of "ideal sail" -----
//                <line 4> "ideal sail"
//             ----- in case of "real sail" -----
//                <line 4> "real sail"
//                <line 5> reflection coefficient
//                <line 6> coefficient for specular reflection
//                <line 7> front side emission coefficient
//                <line 8> back side emission coefficient
//                <line 9> front side non-Lambertian coefficient
//                <line 10> back side non-Lambertian coefficient
//                <line 11> reflection coefficient annual rate [-%/yr]
//                <line 12> coefficient for specular reflection annual rate [-%/yr]
//                <line 13> front side emission coefficient annual rate [-%/yr]
//                <line 14> back side emission coefficient annual rate [-%/yr]
//                <line 15> front side non-Lambertian coefficient annual rate [-%/yr]
//                <line 16> back side non-Lambertian coefficient annual rate [-%/yr]
// ===== in case of "NEP" =====
//    <line 1> "NEP"
//    <line 2> spacecraft dry mass [kg]
//    <line 3> minimum propellant mass [kg]
//    <line 4> maximum propellant mass [kg]
//    <line 5> maximum thrust [N]
//    <line 6> specific impulse [s]
//    <line 7> throttle type (must be "throttle on/off" or "throttle variable")
// ===== in case of "SEP" =====
//    <line 1> "SEP"
//    <line 2> spacecraft mass [kg]
//    <line 3> minimum propellant mass [kg]
//    <line 4> maximum propellant mass [kg]
//    <line 5> solar array(s) characteristic power P(1AU) [kW]
//    <line 6> variation (exponent) p of solar array power with solar distance r => P(r)/P(1AU) ~ (1/r)^p
//    <line 7> number of 2.0 kW (NSTAR) thrusters
//    <line 8> throttle type (must be "throttle on/off" or "throttle variable")
solar sail
characteristic acceleration
0.55
ideal sail
// don't change these comments!


// purpose of this file
InTrance 1.1 spacecraft parameter file
...
// ===== in case of "SEP" =====
//    <line 1> "SEP"
//    <line 2> spacecraft mass [kg]
//    <line 3> minimum propellant mass [kg]
//    <line 4> maximum propellant mass [kg]
//    <line 5> solar array(s) characteristic power P(1AU) [kW]
//    <line 6> variation (exponent) p of solar array power with solar distance r => P(r)/P(1AU) ~ (1/r)^p
//    <line 7> number of 2.0 kW (NSTAR) thrusters
//    <line 8> throttle type (must be "throttle on/off" or "throttle variable")
SEP
150.0
20.0
30.0
2.0
2.0
1.0
throttle variable
// don't change these comments!
```

D.2.5 The NC Definition File

The NC definition file specifies the topology of the used NC as well as the NC's input noise and internal network noise. An example file with robust default values is printed below. All options are explained by the comments that are included in the file.

```
// purpose of this file
InTrance 1.1 neurocontroller definition file
// input type (must be
//    (1) "cartesian coordinates " or
```

```
//    (2) "polar coordinates" or
//    (3) "orbital elements" or
//    (4) "cartesian coordinates and polar coordinates" or
//    (5) "cartesian coordinates and orbital elements" or
//    (6) "polar coordinates and orbital elements" or
//    (7) "cartesian coordinates, polar coordinates and orbital elements")
cartesian coordinates
// number of input neurons
//    for input type (1), (2) and (3)  6 or 12 input neurons have to be used for solar sailcraft and
//                                     7 or 13 input neurons for SEP and NEP spacecraft
//    for input type (4), (5) and (6) 12 or 24 input neurons have to be used for solar sailcraft and
//                                    13 or 25 input neurons for SEP and NEP spacecraft
//    for input type (7)              18 or 36 input neurons have to be used for solar sailcraft and
//                                    19 or 37 input neurons for SEP and NEP spacecraft
12
// number of hidden layers
1
// number of neurons in the hidden layers (1 line per layer)
30
// number of output neurons
//    10 (solar sailcraft) or 11 (SEP and NEP spacecraft) if the NC should implement "LSL minimize maximize" steering
//     5 (solar sailcraft) or  6 (SEP and NEP spacecraft) if the NC should implement "LSL optimize" steering
//     3 (solar sailcraft) or  4 (SEP and NEP spacecraft) if the NC should implement direct steering
3
// input noise (standard deviation)
0.0
// network noise (standard deviation)
0.0
// don't change these comments!
```

D.2.6 The EA Parameter File

The EA parameter file specifies the parameters of the EA that is used for the evolution of the NC and the initial state of the spacecraft. An example file with robust default values is printed below.

```
// purpose of this file
InTrance 1.1 evolutionary algorithm parameter file
// size of initial search space (hypercube)
1.0
// search space shrinking factor
0.5
// population size
50
// number of search space scan epochs
10
// mutation probability per chromosome
0.8
// selection pressure on transfer time / mass
0.01
// don't change these comments!
```

D.2.7 The Simulation Data File

The simulation data file contains all relevant physical data (states, velocities, masses, controls, etc.) for each time step of the best trajectory found by InTrance. A detailed description of the simulation data file is beyond the scope of this work and has to be part of an InTrance user manual. The simulation data file is a so-called "comma separated textfile" that can be displayed using Microsoft Excel[1], if it has the extension "`csv`".

D.2.8 The Trajectory Data File

The trajectory data file is a "light" version of the simulation data file that can be displayed using standard data visualization tools like Amtec Tecplot[2]. Therefore, the trajectory data

[1] http://www.microsoft.com/office/excel/default.asp
[2] http://www.amtec.com

file must have the extension "`dat`".

D.2.9 The VRML File

The VRML (Virtual Reality Modeling Language) file contains a dree-dimensional animation of the best trajectory found by InTrance. It can be displayed using a VRML internet browser plug-in[3]. Therefore, the VRML file must have the extension "`wrl`".

D.2.10 The Control Vector History File

The control vector history file contains the control vector history for the best solution found by InTrance. The first row denotes the number of time steps. For solar sailcraft, the first column denotes the time step as MJD. The second column denotes the sail clock angle and the third column denotes the sail cone angle. For EP spacecraft, the first column denotes again the time step as MJD, the second column denotes the thrust clock angle, the third column denotes the thrust cone angle, and the fourth column denotes the throttle.

D.2.11 The (Best) Chromosome File

The chromosome file contains the best solution found by InTrance. The first row denotes the number of optimization parameters (NC parameters and additional parameters). The last five parameters are associated with the launch date, the launch velocity, the launch azimuth, the launch elevation and the initial propellant mass.

D.2.12 The InTrance Report File

The InTrance report file contains a record of the EA progress, mirroring the information that is displayed on the screen. A detailed description of the report file is beyond the scope of this work and has to be part of an InTrance user manual.

[3] The generated files have been tested using the "Cortona VRML Client 4.0", freely available at http://www.parallelgraphics.com