

Multiphase Low-Thrust Trajectory Optimization Using Evolutionary Neurocontrol

Multiphase Low-Thrust Trajectory Optimization Using Evolutionary Neurocontrol

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben;
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
dinsdag 7 juni 2016 om 12:30 uur

door

Andreas OHNDORF
Diplom-Ingenieur Luft- und Raumfahrttechnik (univ.),
Universität der Bundeswehr München, Duitsland
geboren te Finsterwalde, Duitsland

Dit proefschrift is goedgekeurd door de
promotor: Prof.dr. E. K. A. Gill

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr. E. K. A. Gill,	Technische Universiteit Delft

Onafhankelijke leden:

Prof.ir. B. A. C. Ambrosius,	Technische Universiteit Delft
Prof.dr.ir. M. Mulder,	Technische Universiteit Delft
Prof. D. Scheeres,	University of Colorado, U.S.A.
Prof. B. Dachwald,	Aachen University of Applied Sciences (FH), Germany
Prof. M. Vasile,	University of Strathclyde, U.K.
Prof. M. Lavagna,	Politecnico di Milano, Italy

Abstract

To fulfill the objectives of deep space missions, such as in situ measurements at an outer planet's moon or investigations at main belt asteroids, spacecraft must be provided with sufficient energy to get to these distant objects. This energy can be expressed with the so-called ΔV -budget, which is the sum of required velocity changes along a spacecraft's trajectory. As today's and future deep space missions are infeasible using chemical propulsion alone, their trajectories involve one or more close flybys at mass-rich celestial bodies to gain additional orbit energy. These maneuvers are called gravity assists and depend on the relative positions of the assisting body and the respective target. Due to the orbital motion of both bodies, the required constellation may however repeat only every few decades. This constrains both trajectory and mission design and small launch windows can be the result. Any project delay hence threatens an on-time launch and thus potentially puts an entire mission at risk.

Mitigation of that risk is possible through using low-thrust propulsion which can provide the required ΔV of a deep space mission without gravity assists. Contrary to chemical propulsion, having thrust values up to kilo-Newtons at specific impulse (I_{sp}) values of 300-400 s, low-thrust propulsion currently offers only approximately one Newton at maximum. This thrust is achieved either through the ejection of carried-along particles, which are accelerated to very high velocities, or the reflection of sunlight photons. High exhaust gas velocities and very low propellant consumption make the respective I_{sp} of low-thrust propulsion one magnitude higher than for chemical propulsion. The low-thrust propulsion concept of solar sailing even utilizes the solar radiation pressure for the generation of thrust, making it independent on any propellant.

The different characteristics of low-thrust propulsion and chemical propulsion result in different trajectories. Therefore the methods for the optimization of trajectories of chemically propelled spacecraft are of limited use for the optimization of low-thrust trajectories. New methods were developed for this purpose, and one of them is Evolutionary Neurocontrol. This global optimization method combines the two biology-inspired mechanisms artificial neural networks and evolutionary algorithms. Called neurocontrollers, the artificial neural networks are used for spacecraft control. The optimization capability of evolutionary algorithms is used for the training of neurocontrollers. Contrary to other optimization methods, Evolutionary Neurocontrol does not require an initial guess solution to work, which increases its usability for non-experts in optimal control and optimization. Evolutionary Neurocontrol was applied successfully in the past to various low-thrust transfer problems. Each of those problems, however, consisted of only one single heliocentric transfer from one celestial body to another. The problem of global optimization of multiphase low-thrust trajectories remained unsolved.

This thesis describes how Evolutionary Neurocontrol can be extended to multiphase low-thrust transfers. An existing implementation was revised and complemented with new capabilities, concepts, and functionalities. Examples of the new features are a generic multiphase simulation framework, the support of non-heliocentric transfers, and third-body perturbation. The resulting method has been validated on various complex low-thrust transfer problems, which included two-phase transfers, like Earth-Moon-transfers, or heliocentric rendezvous missions with multiple targets or multiple propulsion technologies. If available, the results were compared with published reference solutions. Finally, Evolutionary Neurocontrol was successfully applied to the design of a trajectory for a so-called Interstellar Heliopause Probe mission. Including a close flyby at Jupiter and using two different propulsion technologies, the resulting transfer brought the spacecraft to a heliocentric distance of 200 AU in less than 25 years.

Abstract

Het doeleinde van missies in de verre ruimte, bijvoorbeeld in situ metingen bij een maan van een der buitenplaneten of onderzoeken in de asteroidengordel, vereist een hoog v -budget. Dit is met chemische voortstuwing alleen niet te bereiken en daarom worden overdrachtsbanen gepland met een of meer passagen dicht aan een zwaar hemellichaam voorbij om extra baanenergie te verkrijgen. Zulke manoeuvres worden gravity assists genoemd, waarbij de energiewinst afhangt van de relatieve geometrie. Zowel het uiteindelijke doel als het ondersteunende lichaam onderweg bewegen zich, zodat de vereiste constellatie misschien niet vaker dan enkele malen per eeuw voorkomt. Dat beperkt de keuze van mogelijke trajecten, het tijdvenster van de lancering en de opzet van de gehele missie. Eventuele vertragingen bedreigen derhalve het complete project.

Dit risico kan worden geband door een systeem met geringe stuwkracht te gebruiken ('low-thrust propulsion'), dat de benodigde v voor een deep-space missie kan leveren zonder gravity assists. Zo'n 'low-thrust propulsion' systeem biedt op het ogenblik maximaal n Newton, zulks in tegenstelling tot chemische voortstuwingssystemen die, bij Isp waarden van 300 tot 400 seconden, kilo-Newtons kunnen bereiken. Een hoge snelheid van het uittredende gas en een zeer laag brandstofverbruik maken de Isp tienmaal groter. Het concept om de stralingsdruk van de zon te gebruiken middels zonnezeilen biedt een 'low-thrust voortstuwingssysteem dat van brandstof onafhankelijk is.

De verschillende eigenschappen van systemen met geringe stuwkracht en van chemische voortstuwingssystemen leiden tot verschillende trajecten. Daarom zijn methoden ter optimalisatie van trajecten voor ruimtesondes met chemische voortstuwing van weinig nut bij de optimalisatie van trajecten waarbij een 'low-thrust systeem wordt gebruikt. Daarvoor zijn nieuwe methoden ontwikkeld, waarvan n de zogenaamde 'evolutionary neurocontrol' is. Deze globale optimalisatie methode combineert twee mechanismen, die bekend zijn uit de biologie, n.l. kunstmatige neurale netwerken en evolutie algoritmen. Kunstmatige neurale netwerken worden voor het sturen van ruimtevaartuigen ingezet onder de naam neurocontrole. In tegenstelling tot andere optimalisatie methoden heeft 'evolutionary neurocontrol geen vooronderstellingen nodig om te functioneren, wat de bruikbaarheid voor niet-experten verhoogt waar het optimale controle en optimalisatie betreft. 'Evolutionary neurocontrol werd al met succes toegepast op verschillende transferproblemen met geringe stuwkracht. Elk van deze opgaven bestond echter uit slechts n enkele heliocentrische transfer van een hemellichaam naar een tweede. Een globale optimalisatie van trajecten in meerdere fasen en met geringe voortstuwingkracht werd daarbij niet bereikt.

Dit proefschrift beschijft op welke wijze de evolutionary neurocontrol methode werd uitgebreid om ook te kunnen worden gebruikt voor transferbanen met meer dan een 'low-thrust propulsion fase. Een bestaande aanwending werd herzien en met nieuwe functionaliteiten aangevuld. Voorbeelden van het laatste zijn een generisch kader voor simulaties met meerdere fasen, de mogelijkheid niet-heliocentrische overdrachtsbanen te berekenen en storingsen van derde lichamen. De geldigheid van de nieuwe methode werd aangetoond voor verschillende transferproblemen, o.a. voor een met twee etappes zoals voor een aarde-maan overdrachtsbaan, of voor heliocentrische rendez-vous missies met meer dan een doel. De resultaten werden vergeleken met reeds gepubliceerde referentie oplossingen, indien die ter beschikking stonden. Tenslotte werd de evolutionary neurocontrol methode met succes gebruikt om een traject te ontwerpen voor een sonde in de interstellaire heliopauze. Het resultaat is een overdrachtsbaan met een gravity assist bij Jupiter, die de sonde in minder dan 25 jaar naar een heliocentrische afstand van 200 AE brengt, waarbij twee verschillende voortstuwingstechnieken worden gebruikt.

Dedicated to my parents, Angelika and Dieter, to whom I owe so much.

Acknowledgements

I very much thank:

- Prof. Dr.-Ing. B. Dachwald
- Dr. rer. nat. W. Seboldt

for their support and the many discussions, which raised and stimulated my interest in the presented field of research.

Contents

List of Figures	viii
List of Tables	xi
List of Acronyms	xii
List of Symbols	xv
List of Constants	xxi
1 Introduction	1
1.1 Optimization of Multiphase Low-thrust Trajectories	2
1.2 Motivation and Scientific Rationale	3
1.3 Thesis Outline	4
2 Low-thrust Propulsion	7
2.1 Basics of Propulsion	7
2.2 Electric Propulsion	9
2.2.1 Electric Propulsion Principle	10
2.2.2 Electric Propulsion Technologies	11
2.2.3 Solarelectric Propulsion	13
2.2.4 Nuclear Electric Propulsion	13
2.2.5 Radioisotope Electric Propulsion	14
2.3 Solar Sails	15
2.3.1 Solar Sailing Principle	16
2.3.2 Performance Metrics	17
2.3.3 SRP Force Models	18
3 Trajectory Optimization	23
3.1 Objectives and Optimality	24
3.2 The Multiphase, Low-thrust Optimization Problem	26
3.2.1 Low-thrust and High-thrust Trajectories	26
3.2.2 Low-thrust Trajectories From an Optimal Control Perspective	30

3.2.3	Single-phase Versus Multiphase	32
3.3	Low-thrust Steering Using Local Steering Laws	35
3.4	Trajectory Optimization Methods	36
3.4.1	Local Trajectory Optimization Methods	36
3.4.2	Global Trajectory Optimization Methods	38
3.5	Smart Multiphase, Low-thrust Trajectory Optimization	40
4	Multiphase Trajectory Optimization With Evolutionary Neurocontrol	41
4.1	Artificial Intelligence and Machine Learning	42
4.1.1	Artificial Intelligence	42
4.1.2	Machine Learning	43
4.1.3	Low-thrust Trajectory Optimization From Machine Learning Perspective	43
4.2	Artificial Neural Networks	44
4.2.1	Inspiration by Natural Neural Networks	44
4.2.2	Composition and Internal Structure	45
4.2.3	Application	47
4.2.4	Training of Artificial Neural Networks	48
4.3	Evolutionary Algorithms	49
4.3.1	Natural Evolution	49
4.3.2	Elements and Mechanisms	51
4.3.3	Advantages and Disadvantages Compared to Other Optimizing Tech- niques	54
4.3.4	Applications of Evolutionary Algorithms	55
4.4	Multiphase Evolutionary Neurocontrol for Trajectory Optimization	55
4.4.1	Spacecraft Steering With Neurocontrollers	56
4.4.2	Optimization of Neurocontrollers Through Training	57
4.4.3	Optimization of Multiphase Missions at the Example of an Earth-Mars- Earth Double Rendezvous	59
4.4.4	Phase Transition Conditions	63
5	Implementation	65
5.1	Support of Non-heliocentric Low-thrust Transfers	66
5.2	Multiphase Framework	67
5.2.1	Implications on Initial and Final Conditions	68
5.2.2	Spacecraft Design	69
5.2.3	Potential Applications of the Developed Framework	69
5.3	Simulation	70
5.3.1	Spacecraft Mass Breakdown	70

5.3.2	Equations of Motion	71
5.3.3	Third-body Perturbation	72
5.3.4	Numerical Integration	74
5.3.5	Eclipse Treatment	75
5.3.6	Launch Rocket Excess Energy	77
5.3.7	Target State, Deviation, and Proximity	79
5.3.8	Dynamic Adaption of Accuracy Boundary Constraints	81
5.3.9	Multiphase Constraints	82
5.3.10	Dynamic Control Step Size	83
5.3.11	Dynamic Parameter Encoding	85
5.4	Evolutionary Algorithm	86
5.4.1	Real Delta Coding	86
5.4.2	Hypercube Size Control	88
5.4.3	Search Space Scan	89
5.4.4	Selection	90
5.4.5	Crossover	91
5.4.6	Mutation	92
5.4.7	Fitness in Multiphase Problems	92
5.4.8	Encoding of Variable Initial Conditions on a Chromosome	93
5.5	Artificial Neural Networks	95
5.5.1	Encoding and Scaling of Input Data	95
5.5.2	Decoding of Output Values	96
6	Validation and Mission Analysis	99
6.1	Implementation Validation	100
6.1.1	Validation of Numerical Integration Schemes	100
6.1.2	Genetic Algorithm Validation	101
6.2	Non-heliocentric Single-phase Transfers	102
6.2.1	Test Cases	102
6.2.2	Results	104
6.3	Earth-Moon Transfers	105
6.3.1	Problem Description and Setup	107
6.3.2	Results and Discussion	108
6.4	Multiple Asteroid Rendezvous – Dawn	110
6.4.1	Mission Description, Objectives, and Spacecraft Design	110
6.4.2	Problem Description and Setup	112
6.4.3	Results and Discussion	112
6.5	Solar System Escape to the Heliosphere Bowshock	114

6.5.1	Mission Design and Optimization Setup	117
6.5.2	Results and Discussion	118
7	Conclusions	121
7.1	Summary	121
7.1.1	Problem Description and State-of-the-Art Solutions	121
7.1.2	Approach and Results	123
7.2	Recommendations for Further Work	125
7.2.1	Fundamental Research	125
7.2.2	Application-oriented	126
A	Coordinate Frames	129
A.1	Inertial Cartesian Frame	129
A.2	Polar Reference Frame	131
A.3	Orbit Reference Frame	132
B	Orbital Elements	133
C	Optical SRP Force Model	137
D	InTrance Configuration Files	141
D.1	Earth-Moon Configuration	141
D.1.1	InTrance Configuration	141
D.1.2	Evolutionary Algorithm Configuration	141
D.1.3	Neurocontroller Configuration of Phase 1	142
D.1.4	Neurocontroller Configuration of Phase 2	142
D.1.5	Spacecraft Configuration of Phase 1	142
D.1.6	Spacecraft Configuration of Phase 2	142
D.1.7	Simulation Configuration of Phase 1	142
D.1.8	Simulation Configuration of Phase 2	144
D.2	Dawn Configuration	146
D.2.1	InTrance Configuration	146
D.2.2	Evolutionary Algorithm Configuration	147
D.2.3	Neurocontroller Configuration of Phase 1	147
D.2.4	Neurocontroller Configuration of Phase 2	147
D.2.5	Spacecraft Configuration of Phase 1	147
D.2.6	Spacecraft Configuration of Phase 2	148
D.2.7	Simulation Configuration of Phase 1	148
D.2.8	Simulation Configuration of Phase 2	150
D.3	IHP Configuration	151

D.3.1	InTrance Configuration	151
D.3.2	Evolutionary Algorithm Configuration	152
D.3.3	Neurocontroller Configuration of Phase 1	152
D.3.4	Neurocontroller Configuration of Phase 2	152
D.3.5	Spacecraft Configuration of Phase 1	152
D.3.6	Spacecraft Configuration of Phase 2	153
D.3.7	Simulation Configuration of Phase 1	153
D.3.8	Simulation Configuration of Phase 2	154
	Bibliography	157
	Index	164
	Curriculum Vitae	169
	List of Publications	171

List of Figures

2.1	Dry mass to payload mass ratio over mission-specific impulse to exhaust velocity ratio.	8
2.2	Specific impulse over thrust force of different propulsion technologies.	8
2.3	NSTAR ion engine	12
2.4	RIT-22 ion engine	12
2.5	Square solar sail.	16
2.6	Rotating solar sail (Heliogyro)	16
2.7	Ideal solar radiation pressure force model	18
2.8	Optical solar radiation pressure force model	20
3.1	Pareto-optimality curves of an example EP space mission with three, four, or five thrust units	25
3.2	Example of a multiphase high-thrust trajectory	27
3.3	Example of a multiphase low-thrust trajectory	27
3.4	Local trajectory optimization methods	37
3.5	Smart global optimization of multiphase low-thrust trajectories	40
4.1	From the optimal chromosome to the optimal trajectory	42
4.2	Natural neuron	45
4.3	Natural neural network	45
4.4	Sigmoid neuron activation function	46
4.5	Artificial neuron	46
4.6	Artificial neural network	46
4.7	Supervised learning	48
4.8	Diploid chromosome and its DNA	50
4.9	Natural gene on a chromosome	51
4.10	Evolutionary algorithm elements	52
4.11	Crossover operators	53
4.12	Neurocontroller for spacecraft steering	56
4.13	Mapping of the neurocontroller parameters on a chromosome	58

4.14	Neurocontroller training and optimization of initial conditions	59
4.15	Low-thrust Earth-Mars-Earth double rendezvous	61
4.16	Multiphase-mission parameters on a single chromosome	62
5.1	Chained simulations	68
5.2	Three body environment	73
5.3	Eclipse geometry	75
5.4	Occultation of the Sun disc by a spherical body	76
5.5	Launch system-provided hyperbolic excess velocity v_∞	77
5.6	Ariane V ECB launch performance diagram	78
5.7	Approximation of Ariane V ECB's launch performance, $\delta = 0$	78
5.8	Stay-time constraint	83
5.9	Control step size and Moon distance over flight time	84
5.10	Evolutionary algorithm parameter hypercube control	88
5.11	Fitness evolution during search space scan	89
5.12	One-at-a-time reproduction with tournament selection	91
6.1	Rosenbrock 2D test function	101
6.2	Multimodal 2D test function	101
6.3	Trajectory of non-heliocentric validation case A.	105
6.4	Trajectory of non-heliocentric validation case B.	105
6.5	Trajectory of non-heliocentric validation case C.	106
6.6	Trajectory of non-heliocentric validation case D1.	106
6.7	Trajectory of non-heliocentric validation case D2.	106
6.8	Trajectory of non-heliocentric validation case E.	106
6.9	Trajectory of Earth-Moon case A	109
6.10	Trajectory of Earth-Moon case B	109
6.11	Earth-Moon transfers: First phase from Earth to Moon's SOI	109
6.12	Earth-Moon transfers: Second phase from SOI boundary to a low orbit about the Moon	109
6.13	Reference trajectory of Dawn.	111
6.14	Recalculated Dawn trajectory	113
6.15	Trajectory of alternative A of a Dawn-like transfer	115
6.16	Trajectory of alternative B of a Dawn-like transfer	115
6.17	The heliosphere in the local interstellar medium.	116
6.18	Interstellar heliopause probe with folded solar power generator panels.	117
6.19	Interstellar heliopause probe with solar power generator panels unfolding.	117
6.20	SEP+REP transfer to 200 AU with a swing-by at Jupiter	119

6.21 Close-up of the IHP gravity assist at Jupiter	119
A.1 Cartesian reference frame	130
A.2 Polar reference frame	130
A.3 Orbit reference frame	132
B.1 Elliptical orbit and orbit plane Keplerian elements	134
B.2 Elliptical orbit in space	135
C.1 Components of optical solar radiation pressure force	138

List of Tables

2.1	Characteristics of selected electrical thrusters	11
2.2	Optical parameters of solar sail models and configurations	21
3.1	Examples of high-thrust and low-thrust transfers of single-phase and multi-phase nature	33
4.1	Natural neuron types	45
4.2	Natural and artificial neural networks	47
4.3	Transition condition types for three scenarios	64
5.1	Classification of the mechanisms and techniques in InTrance	66
5.2	Sphere-of-influence radii and Hill sphere radii of selected celestial bodies	81
5.3	InTrance objective functions	93
5.4	Simulation parameter encoding	94
5.5	Neurocontroller input data	96
6.1	Calls to a DES function during integration of Hull’s class D test function	101
6.2	Results of the EA runs for a multimodal, two-dimensional test function.	102
6.3	Test case specification of test cases for non-heliocentric orbit changes	103
6.4	Results of non-heliocentric orbit changes.	104
6.5	Orbit specification of Earth-Moon transfers.	108
6.6	Results of single-phase Earth-Moon transfers	108
6.7	Earth-Moon transfer: Comparison of the number of ENC parameters and node parameters	108
6.8	Physical parameters and orbit elements of asteroids Vesta and Ceres	111
6.9	Results of two options of a Dawn-like transfer to Vesta and Ceres	114
6.10	Data of the fastest IHP transfer, including a gravity assists at Jupiter.	118

List of Acronyms

ACS	A ttitude C ontrol (S ub) s ystem
AI	A rtificial I ntelligence
ANN	A rtificial N eural N etwork
ASRG	A dvanced S tirling R adioisotope G enerator
AU	A stronomical U nit
BOL	B eginning- o f- L ife
BOM	B eginning- o f- M ission
BVP	B oundary V alue P roblem
CalTech	C alifornia Institute of T echnology
CFD	C omputational F luid D ynamics
DC	D elta C oding
DES	D ifferential E quation S ystem
DNA	D eoxyribo N ucleic A cid
DOF	D egree O f F reedom
DPE	D ynamic P arameter E ncoding
DS1	D eep S pace 1
DS4G	D ual S tage F our G rids
DSSC	D ynamic S tep S ize C ontrol
EA	E volutionary A lgorithm
EC	E volutionary C omputing
ECI	E arth- C entered I nertial
EGA	E arth G ravity A ssist
ENC	E volutionary N euro C ontrol
EOM	E nd- O f- M ission
EOM	E quations O f M otion
EP	E lectric P ropulsion
EPGS	E lectric P ower G eneration S ystem
EPS	E lectric P ropulsion S ystem
ES	E volutionary S trategies
ESA	E uropean S pace A gency
FB	F ly B y
FPDC	F loating- P oint D elta C oding
GA	G enetic A lgorithm
GA	G ravity A ssist
GEO	G Eostationary O rbit
GP	G enetic P rogramming
GTO	G eostationary T ransfer O rbit

GTOM	G lobal T rajectory O ptimization M ethod
HCI	H elio C entric I nertial
HET	H all E ffect T hruster
IHP	I nterstellar H eliopause P robe
InTrance	I Ntelligent spacecraft T RAjectory optimization using N eurocontroller E volution
IPS	I on P ropulsion S ystem
ITAR	I nternational T raffic and A rms R egulations
JAXA	J apan A erospace E Xploration A gency
JGA	J upiter G ravity A ssist
JIMO	J upiter I cy M oons O rbiter
JPL	J et P ropulsion L aboratory
LEO	L ow E arth O rbital
LEOP	L aunch and E arly O perations P hase
LH2	L iquid H ydrogen
LISM	L ocal I nterstellar M edium
LOX	L iquid O Xygen
LSL	L ocal S teering L aw
LTOM	L ocal T rajectory O ptimization M ethod
MCI	M oon- C entered I nertial
MCM	M idcourse C orrection M aneuver
MENC	M ultiphase E volutionary N euro C ontrol
MET	M ission- E lapsed- T ime
MGA	M ars G ravity A ssist
MJD	M odified J ulian D ate
ML	M achine L earning
MLP	M ulti L ayer P erceptron
MPDT	M agneto P lasma D ynamic T hruster
NASA	N ational A eronautics and S pace A dministration
NC	N euro C ontroller
NEP	N uclear E lectric P ropulsion
NHGRI	N ational H uman G enome R esearch I nstitute
NLP	N on L inear P rogramming
NPGS	N uclearelectric P ower G eneration S ystem
NSTAR	N ASA S olar T echnology A pplication R eadiness
OCR	O ptical C haracter R ecognition
ODE	O rdinary D ifferential E quation
OOP	O bject- O riented P rogramming
PAEHT	P ower- A ugmented E lectrothermal H ydrazine T hruster
PPT	P ulsed P lasma T hruster
PPU	P ower P rocessor U nit
PSO	P article S warm O ptimizer
PV	P hoto- V oltaic
RDC	R eal D elta C oding
REP	R adioisotope E lectric P ropulsion
RF	R adio- F requency

RIT	R adio-frequency I on T hruster
RK	R unge- K utta
RKF	R unge- K utta- F ehlberg
RL	R einforcement L earning
RPS	R adioisotope P ower S ystem
RTG	R adioisotope T hermoelectric G enerator
RV	R endez V ous
SDC	S tatic/ D ynamic C ontrol
SEP	S olar E lectric P ropulsion
SMART-1	S mall M issions for A dvanced R esearch in T echnology-1
SOI	S phere- O f- I nfluence
SP	S olar P anel
SP	S elective P ressure
SRD	S olar R adiation D ose
SRP	S olar R adiation P ressure
SSE	S um S quare E rrors
SSS	S earch S pace S can
TRL	T echnology R eadiness L evel
VGA	V enus G ravity A ssist
VRML	V irtual R eality M odeling L anguage

List of Symbols

Greek Symbols

α	(azimuth or sail clock) angle absorption coefficient mass-specific electrical power
β	(sail) pitch angle, cone angle
χ	throttle
δ	thrust cone angle delta coding distance deviation
ϵ	emissivity convergence criterion thrust centerline angle
η	efficiency
γ	sigmoid temperature parameter gravitation constant
κ	power law exponent
λ	lightness number
μ	gravitation parameter
ν	epoch convergence buffer size true anomaly
Ω	right ascension of ascending node
ω	argument of pericenter
Φ	flight-path-angle
φ	azimuth angle
π	parameter vector of an artificial neural network
ρ	reflection coefficient range

σ	sail loading
τ	transmission coefficient
θ	neuron activation threshold
ϑ	elevation angle
Ξ	population (of individuals)
ξ	chromosome, individual, solution, string
ζ	orbit angle

Indices

∞	hyperbolic excess
o	initial launch reference
1	burnout (mass)
a	absorption apocenter
b	back side beam
CB	central body
CP	capture
c	characteristic cartesian convergence child
dw	dwel time or stay-time
d	diffuse
ES	escape
ex	exhaust
e	emission electric(al) exponential
FB	flyby
f	front side final

g	gravitation
h	orbit-normal
i	index
j	index
k	kinetic
l	linear
max	maximum
min	minimum
m	mission mutation
o	other
p	penumbra propellant polar parent pericenter
r	radial reflection distance
SA	sail assembly
SP	solar panel
S	Sun
s	spacecraft shadow (solar) sail specular state
th	thermal
T	target transfer
t	thrust transversal
u	umbra control
v	velocity

* optimal

Latin Symbols

a_c	characteristic acceleration
a	acceleration semimajor axis
N	artificial neural network transfer function
A	area
c	constant
ΔV	velocity increment
n_π	artificial neural network parameter vector dimension
e	eccentricity genetic algorithm search epoch
J	fitness function
S	solar radiation energy flux
U	control function
J	inertial cartesian reference frame
I_{sp}	specific impulse
l	length chromosome dimension
m	mass
\mathcal{A}	action domain
\mathcal{X}	situation domain
G	dynamic constraint
x	neuron input value
O	objective function
\mathcal{O}	rotating orbit reference frame
\mathcal{P}	polar reference frame
P_{PPU}	power processor unit input power
p	pressure
p_{SRP}	solar radiation pressure

$p_{\text{SRP},0}$	solar radiation pressure at 1 AU
p	semilatus rectum, orbit parameter
S	strategy
t	time
e	unit vector
u	control vector
x	artificial neural network input data vector
y	artificial neural network output data vector
r	position vector
v	velocity vector
B	Non-Lambertian coefficient
b	binary allele or genotype value
E	(orbit) energy eccentric anomaly
F	force
f	specific force function
f_s	shadow function
H	parameter subspace during an epoch hyperbolic anomaly
h	delta coding partial solution orbit angular momentum control step size
I	linear impulse
M	mean anomaly
n	dimension, count, number of mean motion
$n_{\mathbf{u}}$	control vector dimension
P	power proximity
p	probability
q	population dimension

r	distance radius real-value allele or genotype value
s	training sample
T	temperature, orbit period
u	control variable
V	violation
v	velocity
v_{ex}	effective exhaust (gas) velocity
W	energy flux
w	neuron connection weight factor
y	neuron output value

Other Symbols and Notations

$\ddot{\square}$	second time derivative of \square
Δ	difference
$\dot{\square}$	first time derivative of \square
(\dots)	open interval
$\langle \dots \rangle$	EA chromosome
$[\dots]$	closed interval
$^{(i)}\square$	\square of phase i
$\bar{\square}$	upper bound discrete \square
$\underline{\square}$	lower bound

List of Constants

AU Astronomical unit, $1 \text{ AU} = 1.49597870691 \cdot 10^{11} \text{ m}$ [59]

c Vacuum speed of light, $c = 2.99792458 \cdot 10^8 \text{ m/s}$ [59]

g_0 Earth sea level norm acceleration, $g_0 = 9.80665 \text{ m/s}^2$ [85]

h Planck's constant, $h = 6.6262 \cdot 10^{-34} \text{ Js}$ [61]

k Boltzmann's constant, $k = 1.3806488 \cdot 10^{-23} \text{ J/K}$ [61]

σ Stefan-Boltzman constant, $\frac{2\pi^5 k^4}{15h^3 c^2} = 5.670373 \cdot 10^{-8} \frac{\text{J}}{\text{K}^4 \text{m}^2 \text{s}}$ [61]

S_0 Solar radiation flux density at 1 AU, $S_0 = 1368 \text{ W/m}^2$ [7]

Introduction

Deep space exploration missions have contributed significantly to the understanding of our solar system. Examples are missions to the planets and their moons, polar Sun-orbiter missions, or solar system escape missions. However, they require huge amounts of orbit energy to reach their targets. This energy, typically expressed as the velocity increment or ΔV -budget, is either provided by a launcher or by a propulsion system. Launched onto a direct solar system escape trajectory by the National Aeronautics and Space Administration (NASA) in 2006, the mission New Horizons currently sets the record for the highest achieved Earth-relative velocity with $\Delta V = 16.2 \text{ km/s}$ [36]. This can be seen as a practical limit, as chemical propulsion is limited to the energy that is stored in the on-board propellant.

The ΔV -budget of deep space missions often exceeds the capabilities of chemical propulsion. Including maneuvers called gravity assists (GA), or swing-bys, where spacecraft can gain additional energy, into mission design can provide the required additional orbit energy. A GA is a flyby of a spacecraft at short distance to a celestial body, e.g., a planet or a moon. Depending on the spacecraft's initial velocity and flight direction and on the mass and flight direction of the flyby body, the gravitation of that body alters the spacecraft's inertial flight-path. The spacecraft velocity vector after a GA therefore differs in direction or magnitude or both from the one before the flyby. Such a maneuver can thus accelerate or decelerate a spacecraft, or also only change the direction of its velocity vector. The latter is exploited for orbit inclination changes, for example.

Whether a GA is positive for a particular mission depends on the phasing of potential swing-by bodies during the planned mission time frame. The orbit periods of outer planets are greater than the orbit periods of the inner planets of our solar system. Usable geometries for a swing-by therefore occur only every few decades, which is a severe constraint to the planning of a deep space mission. Including multiple swing-by maneuvers into a mission's trajectory design is even more complicated as the resulting phasing problem is more complex than for missions with only one GA. The higher ΔV , resulting from multiple swing-bys, is therefore linked to the substantial drawback of a reduced flexibility in mission execution. Reducing the number of necessary GA, or omitting them at all, would reduce a mission's complexity. Furthermore, it offers more launch options, e.g., launch opportunities for a longer time or repeating in shorter intervals. Future deep space missions are unlikely to set lower ΔV -requirements and chemical propulsion is inherently limited. The mitigation of these problems consequently calls for new and more capable propulsion technologies.

Such high-performance propulsion systems exist today at various stages of technical development and qualification, expressed with the so-called technology readiness level (TRL). Electric propulsion (EP) basically offers significantly higher ΔV than chemical propulsion, through higher exhaust gas velocities and lower propellant consumptions. The maximum thrust level

of EP is however only a few Newtons at maximum. They are unusable for high-thrust applications, like launching from a planet's surface. However, past and current missions employed EP successfully for primary deep-space propulsion. Examples are Deep Space 1 (DS1) in 1998 and Dawn in 2007, conducted by the Jet Propulsion Laboratory (JPL); Hayabusa in 2003 of the Japan Aerospace and Exploration Agency (JAXA); and Small Missions for Advanced Research in Technology-1 (SMART-1) of the European Space Agency (ESA) in 2003.

The low-thrust propulsion concept of solar sailing even would enable completely new mission types because it does not require any propellant. Solar sails are large, ultra-lightweight structures that exploit the solar radiation pressure for acceleration, i.e., for acceleration and deceleration. The sails direct the force vector resulting from reflected sunlight into the required direction. The TRL of solar sails is however lower than for EP. No space mission using solar sailing for primary propulsion has flown yet. However, once in-space validation has taken place, solar sail-driven propulsion is expected to become a further viable alternative to chemical propulsion for deep space missions. In fact, for very advanced mission concepts it will be the only viable option.

1.1 Optimization of Multiphase Low-thrust Trajectories

The flight path of a body, i.e., its position and velocity vectors over time, is called trajectory. It depends on the initial state (position, velocity) at a time t_0 and the forces that act on that body during $t \geq t_0$. For a spacecraft and neglecting third-body perturbations, these are primarily the gravitational attraction of a central body and the thrust forces generated by its propulsion system. Gravity depends on the masses of the involved bodies and the distance between them and the spacecraft.

The achievable thrust levels from propulsion systems and the time span during which a resulting force acts on a spacecraft depends heavily on the propulsion technology. Chemical propulsion features high thrust forces that act for a short time, rarely more than an hour. The maximum thrust of low-thrust propulsion is in the range of milli-Newtons. The high exhaust velocity and low propellant mass flow rate however allows the application of this small but highly efficient thrust over long times. The measure for a propulsion system's efficiency is the specific impulse I_{sp} . While chemical propulsion is limited to $I_{sp} \leq 500$ s, EP as already successfully demonstrated I_{sp} values of several thousand seconds [35].

The different thrust characteristics of high-thrust and low-thrust propulsion result in high-thrust trajectories and low-thrust trajectories. The long coast arcs of high-thrust trajectories, during which the spacecraft travels primarily under the influence of the gravitation of the central body, depend on the position and velocity vector at the time of burnout of the propulsion subsystem or the launcher. In simple terms, for the computation of the trajectory that transfers a spacecraft from one celestial body to another, the associated vector set of position and velocity and the burnout time must be calculated. This is not the case for low-thrust trajectories because the long-acting low-thrust force changes the flightpath continuously. The design of a low-thrust trajectory that leads to a predefined target or celestial body instead requires the calculation of a thrust vector history. The resulting increase in optimization variables is therefore considerable.

Regardless of the actually used propulsion technology, the design of a particular mission inevitably requires the optimization of the transfer trajectories. This is because the trajectory influences spacecraft design parameters, like required propellant mass, life time, communications, and power and thermal control systems. Due to the described lower number of trajectory-influencing variables, this is easier for missions using chemically propelled spacecraft. Proven methods have been developed for that particular purpose, but they are of

limited use for the optimization of low-thrust trajectories. Optimized low-thrust transfers require an optimized thrust profile, i.e., a thrust vector history over time, which must be followed in order to accomplish the mission. The thrust vector history not only decides whether the target body is met at all but also affects crucial figures of merit, e.g., the flight time Δt and/or the total propellant consumption. Its optimization is therefore essential. New methods have been developed and successfully applied to this optimization problem type. Thereby some of these rely on traditional optimization methods, e.g., gradient-based, deterministic, or local optimization techniques. An example is the collocation or direct transcription algorithm [8, 10]. It transcribes the optimal control problem into a nonlinear programming problem which is solved using quadratic programming. Other optimization methods employ new approaches such as static-dynamic control (SDC) [95], shape-based methods or stochastic global optimization methods, like particle swarm optimizers (PSO) [1]. A promising global optimization method is evolutionary neurocontrol (ENC). Its foundations are control and optimization principles from artificial intelligence and evolutionary computing. It has been applied successfully to the optimization of several heliocentric low-thrust trajectory problems [15]. The focus was thereby on single-phase transfers. Examples are flyby or rendezvous trajectories to planets or asteroids but also solar sail transfers into near-polar Sun orbits [14].

1.2 Motivation and Scientific Rationale

Real-life missions often exhibit multiple flight legs or phases, as the maximization of scientific return of expensive deep space missions often requires to visit more than a single target. Complex flyby or multi-rendezvous missions have therefore become common practice without adequate methods and tools to efficiently treat them. The design and optimization of the required trajectories is by principle challenging, not only for astrodynamics specialists but also for the used software tools, and therefore often requires a team of experts. Moreover, analysis and resulting mission design is done through separate optimization of each mission phase and therefore is, by principle, not globally optimal. Human experts must furthermore assure that the trajectories of adjacent missions phases fit together at their start and end points. These transition points are mostly also selected by experts according to their experience and knowledge. Consequently, today's low-thrust optimization methods are not satisfying and require further development and extension.

Existing low-thrust trajectory optimization methods are often limited to a particular type of problem or unusable without expert knowledge in orbital mechanics and optimization techniques. The optimization of mission designs that contain more than one transfer to a target body is a further problem for most optimization method implementations. In addition, successful use of these tools often requires a user to check obtained solutions for validity. In dependence on the solution's validity and according to his knowledge, the user then possibly alters simulation and optimization parameters and starts a new optimization run. Some optimization schemes require an initial-guess solution from which they derive an optimum from. The generation of such a solution is very difficult for single-phase low-thrust missions. For multiphase low-thrust missions it can become almost impossible.

For a proper nomenclature, this thesis defines a **phase** as a specific part of the mission. It can be a complete trajectory but also only a certain part of it. For a mission comprising two rendezvous-type transfers to different asteroids, for example, each of the trajectories can be treated as one phase and the stay-time at the first asteroid as a separate third phase. A different example is the transfer from an Earth-bound orbit into an orbit around the the Earth's Moon. Here, the transfer from the Earth to the Moon's sphere-of-influence can be treated as one phase. The subsequent flight leg from there to the final, presumably lower, orbit about the Moon would be another phase. A subdivision into multiple phases is also

possible for an interplanetary transfer of a spacecraft having more than one propulsion system or propulsion stage. In this case, the point of jettison of the first stage can mark the separation between the phases. Naturally, for all these examples, the parameters of subsequent phases, such as the propellant mass, influence the optimization of the trajectories of previous phases and vice versa. A generic treatment of such complicated constraints is often impossible with existing low-thrust trajectory optimization methodologies. Certain assumptions are therefore often made, which, by principle, prohibit global optimization of the complete mission.

In order to solve existing methodological shortcomings and to allow true global optimization of multiphase low-thrust missions, this thesis tries to answer the following research questions:

1. **Which methodology enables the global optimization of multiphase low-thrust mission designs?**
2. **How can the methodology be validated and what are the results of these validations?**
3. **How does the methodology apply to preliminary mission analysis, i.e., what problem types are solvable and which are potentially still not?**

The focus of this thesis is on the development and validation of a methodology that answers these questions. As the application of that methodology is preliminary mission design, high-fidelity trajectories are not required. Their computation would require taking into account additional effects, like known disturbance accelerations¹, air-drag at low-altitude orbits, and relativistic effects. Inclusion of these effects, however, disguises the actual multiphase optimization problem and often increases computation times significantly. For detailed mission analysis, high-fidelity trajectory are nevertheless required, but this is achievable through existing low-thrust trajectory optimization methodologies. They can provide the required accuracy easily if provided with a global optimal solution.

1.3 Thesis Outline

Following this introduction, Chapter 2 describes the principles of spaceflight propulsion and describes in detail the differences between chemical propulsion and low-thrust propulsion. The two main low-thrust technologies types of electric propulsion and solar sailing are described with corresponding models. Different technologies of both types are explained and examples of existing implementations and components are presented, such as thrusters, electric power generation systems, and solar sails types.

Chapter 3 provides an introduction to trajectory optimization. The differences between trajectories resulting from chemical propulsion and low-thrust propulsion are explained and also the difference between single-phase and multiphase trajectories. Furthermore, the influence of the optimization criterion on the trajectory is discussed and how the concept of Pareto-optimality is usable to balance contradicting criteria. This is followed by the formulation of the general multiphase, low-thrust trajectory optimization problem. Different methods and techniques for the optimization of low-thrust trajectories are presented. Finally, in this chapter a globally optimal, multiphase, low-thrust trajectory optimization method is developed.

The application of the concepts of machine learning and artificial intelligence to the optimization of multiphase low-thrust trajectories is shown in Chapter 4. It introduces to the basic theory of mechanisms and techniques that enable ENC. These are artificial neural

¹For detailed mission analysis, at least the gravity forces of all planets of the solar system as well as those of the big asteroids must be taken into account.

networks and evolutionary algorithms. The networks, called neurocontrollers, thereby steer the low-thrust propelled spacecraft according to a strategy that is encoded in their internal parameters. Evolutionary algorithms then train the neurocontrollers for optimal steering through optimization of these parameters.

InTrance, a software which implements ENC for heliocentric single-phase low-thrust trajectory optimization, had to be completely revised and extended to allow its application for solving multiphase trajectory optimization problems. The new capabilities divide into three categories: capabilities for multiphase trajectory optimization, the support of non-heliocentric trajectory calculations, and the extension of the trajectory simulation with new features, such as the increase of the robustness to find optimal solutions more reliably. Chapter 5 describes these features in detail.

Chapter 6 describes the validation of newly developed concepts and the application to mission analysis. Validation comprises tests for logical correctness of existing and newly developed components. This was achieved through comparison of published solutions of reference problems with recalculated solutions. The analyzed problems are non-heliocentric single-phase transfers, single-phase and two-phase Earth-Moon transfers, and heliocentric double-rendezvous transfers. The final example shows, how a flight time-minimal, solar-system escape mission that comprises two propulsion stages and a gravity assist at Jupiter was optimized with the developed method.

The final Chapter 7 summarizes the results of this research and discusses potential further research activities in this field. It sketches how generic problems related to ENC for trajectory optimization can be solved, such as providing optimal input data to an artificial neural network (ANN). Furthermore, it describes how ENC can help to search for optimally generalizing ANNs or to find an analytical solution of the general, low-thrust transfer problem.

2

Low-thrust Propulsion

The understanding of the application of ENC to low-thrust trajectory optimization requires knowledge of the underlying propulsion physics. This chapter shall therefore familiarize the reader with spacecraft propulsion systems and provide that required basic knowledge. For this purpose, section 2.1 gives a brief introduction into the basics of spacecraft propulsion, its enabling physics, and existing technologies. For detailed information on spacecraft propulsion technologies, the reader is referred to the numerous standard literature on this topic.

The following section 2.2 and section 2.3 briefly explain two examples of low-thrust propulsion systems. The first is EP, which achieved already very high exhaust velocities of up to $\approx 150\,000\text{ km/s}^1$ [93]. The second is the concept of solar sails, which uses only the sunlight photon's radiation energy and requires no propellant at all. For both technologies, the enabling principles and mathematic models are described and, if possible, also current or past real mission applications of each technology are presented.

2.1 Basics of Propulsion

The space age started with the first man-made satellite that was brought into an Earth-bound orbit. This was achieved by the Soviet Union with the launch of Sputnik 1 on top of a modified intercontinental rocket R-7 on 4th October 1957 [21]. All chemical rocket motors developed since then share the same enabling principle with the Sputnik 1 launcher: the conversion of chemical energy that is bound in the carried-with propellant chemicals into thermal energy via an exothermal reaction in the combustion chamber. The high-temperature, high-pressure reaction products then expand through a nozzle. During the expansion, the exhaust gas pressure interacts with the combustion chamber walls, and the resulting reaction force accelerates the rocket into the direction opposite to the exhaust plume. The range of achievable thrust thereby spans from a few Newtons to more than several thousand kilo-Newtons (kN). The five F-1 engines of the first stage of Apollo's launch system Saturn V, for example, used the propellant kerosene and liquid oxygen (LOX) as oxidizer. Each F-1 had a thrust force of $F = 7\,893\text{ kN}$ with a specific impulse I_{sp} of 304 s [89].

The specific impulse, or characteristic mission impulse, is a measure of a rocket engine's efficiency and can be understood as the momentum given to the rocket per kilogram of propellant expelled from the engine's nozzle. It is measured in seconds and calculated with

¹First successful test of the dual stage four grids (DS4G) laboratory thruster model test in 2005; DS4G thruster developed by ESA and Australian National University

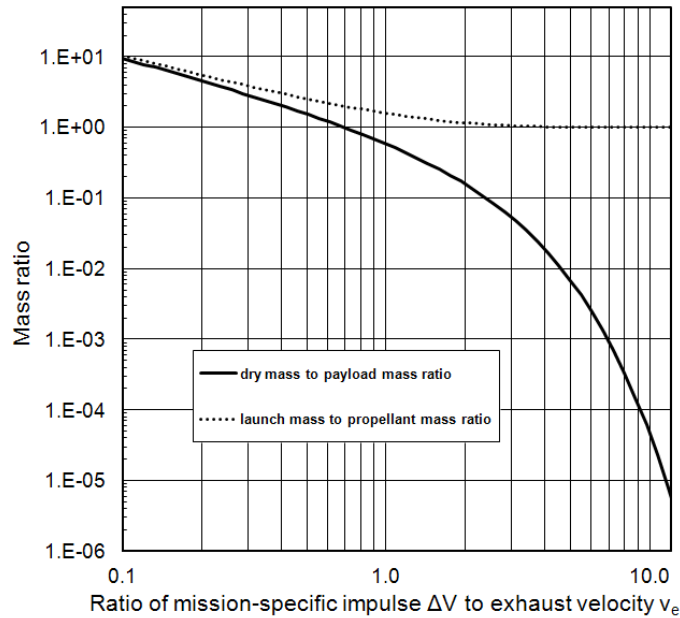


Figure 2.1: Ratio of Dry Mass to Payload Mass and the Ratio of Launch Mass to Propellant Mass Over the Ratio of Mission-specific Impulse to Exhaust Velocity.

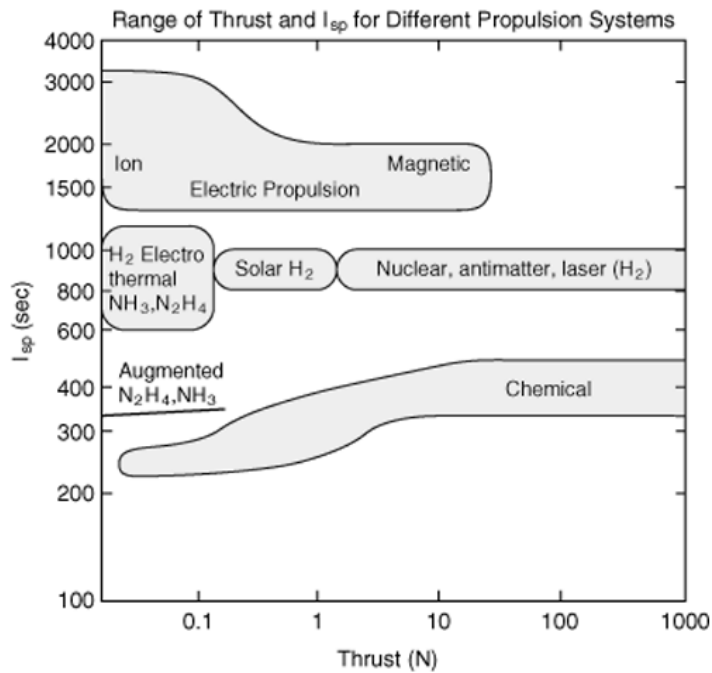


Figure 2.2: Specific Impulse, I_{sp} , of Different Propulsion Technologies, with Electric Propulsion as a Prominent Low-thrust Technology in the Upper Left Part. Courtesy JPL/NASA.

the exhaust gas velocity v_{ex} and the norm acceleration at Earth sea level g_0 with

$$I_{\text{sp}} = \frac{v_{\text{ex}}}{g_0} = \frac{v_{\text{ex}} \dot{m}_{\text{p}}}{g_0 \dot{m}_{\text{p}}} = \frac{F}{g_0 \dot{m}_{\text{p}}}. \quad (2.1)$$

Equation (2.1) provides an alternative expression, using the thrust force F and the propellant mass flow \dot{m}_{p} . The energy that is stored in the chemicals in the rocket's tanks limits the exhaust velocity to maximum 4.6 km/s¹. Current most advanced liquid propellant rocket motors have almost reached that velocity limit. The Vinci cryogenic upper-stage² of the Ariane V ECB, for example, achieves $v_{\text{ex}} = 4.56$ km/s and $I_{\text{sp}} = 465$ s through burning LH2 and LOX.

Tsiolkovsky's rocket equation

$$\Delta V = v_{\text{ex}} \ln \left(\frac{m_0}{m_1} \right) \quad (2.2)$$

shows the influence of v_{ex} and the ratio of initial mass, also called launch mass or wet mass, m_0 to burnout mass m_1 , or dry mass on the resulting velocity increment ΔV . The plot in Fig. 2.1 shows this dependency, which allows several options for preliminary mission design. First, if ΔV and m_1 are fixed, one needs to increase v_{ex} to reduce $m_{\text{p}} = m_0 - m_1$, which can require a more capable propulsion technology or the use of different propellant and oxidizer chemicals. There are however technically and chemically limits to both strategies. Second, if m_{p} is fixed, a higher v_{ex} either allows for a higher m_1 or a higher ΔV . Third, if m_0 and ΔV are fixed, one need to either increase v_{ex} or m_{p} by reduction of m_1 to achieve the required ΔV . To realize high- ΔV missions, therefore one can either increase v_{ex} , or reduce m_1 , or do both, if technically feasible.

Figure 2.2 compares chemical propulsion systems with low-thrust propulsion systems and other, some of them being currently theoretical propulsion technologies. The upper left part of this figures shows EP as an example of low-thrust propulsion. Their upper thrust force limit of around 10 N clearly inhibits their use for a launch from Earth surface. The approximately one magnitude higher I_{sp} makes it however well suited for deep-space propulsion because it allows novel high- ΔV space missions, or higher payload ratios, or a combination of both.

2.2 Electric Propulsion

Although the principles and mechanisms of EP are known since the 1960s [84] [46], it took approximately 30 years until their realization and in-space validation. Interestingly, the high- ΔV -capability of electric propulsion did not found its first application in deep-space propulsion but in station-keeping of geostationary communication satellites [22]. NASA's New Millennium Deep Space 1 (DS 1) mission to the comet 19P/Borrelly [60], ESA's SMART-1 mission to the Moon [29], and JAXA's asteroid sample return mission Hayabusa [48] to the asteroid Itokawa are prominent examples of later deep space missions using EP.

Common to all EP-technology thrusters is their increased electrical power requirement. Depending on how that power is generated, they divide into solar electric propulsion (SEP), nuclear electric propulsion (NEP), and radioisotope electric propulsion (REP). While SEP depends on the Sun-spacecraft distance, NEP and REP do not suffer from this drawback. After an introduction into the general principles of EP and different thruster technologies, three EP-types and their models are presented.

¹ $v_{\text{ex}} \approx 4.6$ km/s for liquid hydrogen (LH2)/LOX at oxidizer-propellant-ratio of 4.83 [89]

²<http://cs.astrium.eads.net/sp/launcher-propulsion/rocket-engines/vinci-rocket-engine.html>

2.2.1 Electric Propulsion Principle

According to (2.1), v_{ex} determines I_{sp} , and electric thrusters allow significantly higher v_{ex} -values because, contrary to chemical rocket motors, they are not constrained to the energy within the propellant and oxidizer chemicals. The energy for the acceleration of the expelled ions instead stems from an external power source and is therefore principally unlimited. The beam power P_{b} of the ions streaming through a nozzle if an EP thruster is

$$P_{\text{b}} = \frac{dE_{\text{k}}}{dt} = \frac{\dot{m}_{\text{p}}}{2} v_{\text{ex}}^2 = \frac{F v_{\text{ex}}}{2}. \quad (2.3)$$

It corresponds to the energy per time that is necessary to accelerate the propellant mass flow \dot{m}_{p} to v_{ex} . The electrical input power P_{e} cannot be completely exploited for ion acceleration. This is expressed with the thruster efficiency η_{t} when relating P_{e} with (2.3)

$$\eta_{\text{t}} P_{\text{e}} = P_{\text{b}} = \frac{\dot{m}_{\text{p}}}{2} v_{\text{ex}}^2. \quad (2.4)$$

Basically, any mean of electrical power generation is suited to supply an EP-thruster as long as its power output allows operating the thrust unit. A system consisting of one or multiple low-thrust engines and an electric power generation subsystem is called electric propulsion system (EPS). If the thrust module is an ion engine, i.e., its enabling principle is the acceleration of ions, the propulsion system is also called ion propulsion system (IPS).

Equation (2.4) shows the dependency of v_{ex} on P_{e} . Theoretically, and for constant \dot{m}_{p} , the higher P_{e} the higher is also v_{ex} . In practice, there are however technical limitations to the allowed P_{e} of a thrust unit, such as increased material degrading through sputtering and the danger of performance degrading sparkovers.

Besides thrusters and an electrical power supply, an EPS also contains at least one power processor unit (PPU). A PPU conditions and transforms the input power and voltage to the different requirements of the respective thruster's components. Due to the imperfection of this conversion, the electrical input power $P_{\text{PPU}} = P_{\text{e}}$ of a PPU must be supplied with an efficiency factor η_{PPU} to obtain the thruster input power P_{T}

$$\eta_{\text{PPU}} = \frac{P_{\text{T}}}{P_{\text{PPU}}}. \quad (2.5)$$

P_{T} again divides into the ion beam power P_{b} and the so-called other power P_{o} , which accounts for the electrical power to ionize the propellant and to operate the thruster components, e.g., heaters, grids, and control units

$$\eta_{\text{PPU}} P_{\text{PPU}} = P_{\text{T}} = P_{\text{o}} + P_{\text{b}} = P_{\text{o}} + \frac{\dot{m}_{\text{p}} v_{\text{ex}}^2}{2}. \quad (2.6)$$

The electrical efficiency η_{e} is defined as the ratio of beam power to thruster input power

$$\eta_{\text{e}} = \frac{P_{\text{b}}}{P_{\text{T}}} = \frac{P_{\text{b}}}{P_{\text{o}} + P_{\text{b}}}. \quad (2.7)$$

Using (2.1) and $\eta_{\text{t}} = \eta_{\text{e}} \eta_{\text{PPU}}$, the thrust force is

$$F = \frac{2\eta_{\text{t}} P_{\text{PPU}}}{g_0 I_{\text{sp}}}. \quad (2.8)$$

Table 2.1: Maximum thrust force F^* , maximum electrical thruster input power P_e^* , and specific impulse I_{sp} of selected electrical thrusters [35]. NEXT-data taken from [65]. RIT-22 data taken from [15].

Thruster	Technology	F^* [mN]	P_e^* [W]	I_{sp} [s]	Manufacturer	Application
NSTAR	Kaufman	93	2 567	3 127	Boeing, U.S.A.	DS 1, DAWN
NEXT	Kaufman	236	6 900	4 190	JPL/NASA	-
SPT-50	Hall effect	20	350	1 100	Fakel, Russia	-
SPT-70	Hall effect	40	700	1 500	Fakel, Russia	-
SPT-100	Hall effect	80	1 350	1 600	Fakel, Russia	Small GEO
PPS1350-G	Hall effect	88	1 500	1 650	Snecma, France	SMART 1
SPT-140	Hall effect	300	5 000	1 750	Fakel, Russia	-
RIT-10	RIT	15	459	3 400	Astrium, Germany	Eureca, Artemis
RIT-22	RIT	175	6 200	4 763	Astrium, Germany	-
T5	Kaufman	18	476	3 200	QinetiQ, U.K.	GOCE
T6	Kaufman	210	4 500	4 700	QinetiQ, U.K.	Bepi Colombo

2.2.2 Electric Propulsion Technologies

Depending on the respective mechanism for the acceleration of the propellant, EP further divides into electrothermal, electrostatic, or electrodynamic propulsion. These types are briefly presented in the following and Tab. 2.1 gives an overview of the capabilities of selected thruster models.

2.2.2.1 Electrothermal Propulsion

The enabling mechanism of electrothermal propulsion is the generation of thrust through expansion of high-temperature, high-pressure gas through a nozzle, which is the same mechanism as for chemical engines. The required energy is however externally provided; an electric arc, a tungsten wire coil through which an electric current flows, or microwaves are viable energy sources. Thrust units of this technology are called resistojets or arcjets. Current space-qualified thruster models achieve $v_{ex} = 10$ km/s with thrust levels less than 0.5 N and $\eta_t > 70\%$. Several propellant types are possible but the most common are hydrogen, nitrogen, ammonia, and hydrazine. Hydrazine resistojets are called power-augmented electrothermal hydrazine thruster (PAEHT) and have an $I_{sp} > 500$ s with $P_e < 2$ kW [37].

2.2.2.2 Electrostatic Propulsion

Electrically charged particles, e.g., electrons or ions, accelerate electrostatically between an electrical potential difference. This principle is exploited for electrostatic propulsion. The expulsion of accelerated ions also gave engines of this technology the name ion engines. Different propellants are feasible, e.g., Mercury, Cesium, or noble gases. However, only the nontoxic noble gas Xenon (Xe) is used today. It is easier handled and stored than metallic propellants and its exhaust plume does not contaminate the spacecraft.

Ionization of propellant gas molecules is achieved through bombardment with accelerated electrons from either electrostatic (Kaufman thrusters) or radio-frequency (RF)-induced origin. In Kaufman thrusters, an ionizer cathode in the discharge chamber emits electrons, and

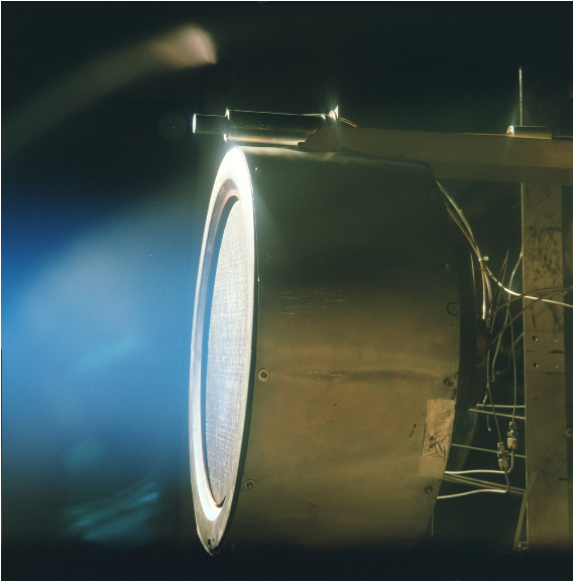


Figure 2.3: NSTAR Ion Engine. Courtesy NASA/JPL-CalTech.



Figure 2.4: RIT-22 Thruster. Courtesy Airbus.

these electrons are then accelerated towards an anode. On their way to the anode, electrons hit Xe-atoms and ionize them. Within the second mechanism, electrons are excited with RF-waves of 0.7-1.0 MHz. Electrons leave their Xe-atoms and potentially collide with electrons of other Xe-atoms, which leaves positively charged Xe-ions and electrons. These so-called radio-frequency ion thruster (RIT) were developed by Giessen University, Germany, in cooperation with Astrium. After ionization, the ions are accelerated by an electrostatic field with a potential difference of several kV. Most designs involve two or three acceleration grids and achieve exhaust velocities of tens of thousands kilometers per second. In order to prevent an electrical field between the expelled ions and the spacecraft, a neutralizer cathode injects electrons into the exhaust beam.

Because of their very high exhaust velocities, ion engines are suited to propel interplanetary missions, during which long-duration thrust periods are acceptable. For example, the ion engine that was used for NASA's DS 1 mission was the Kaufman-type NASA Solar Technology Readiness (NSTAR) thruster.

Another thruster type that utilizes the electrostatic acceleration of ions is the Hall effect thruster (HET). HETs lack acceleration grids and were first developed and used in the Soviet Union for station-keeping on a multitude of satellites. As a result of a technology transfer, HETs were later also further developed in the U.S. and in Europe. Current thruster models, e.g., the T220 of NASA, achieve $F = 500 \text{ mN}$, an $I_{sp} = 2450 \text{ s}$, and $\eta_e = 59\%$ with $P_e = 10 \text{ kW}$. Another flight-proven HET thruster model is the Snecma PPS1350-G, used on ESA's successful mission to the Moon SMART-1 [30].

2.2.2.3 Electrodynamic Propulsion

These thrusters use the interaction of electric and magnetic fields to accelerate a plasma. Except for the pulsed plasma thruster (PPT), which has very small thrust capability, other thrusters have not yet reached a TRL suited for space application. This is also related to the problem of providing the necessary energy on board a spacecraft to operate such thrusters efficiently. Once this problem is solved, a so-called magnetoplasmadynamic thruster

(MPDT) is however an interesting option also for human deep-space missions because of their significantly higher thrust level.

2.2.3 Solarelectric Propulsion

SEP combines electrical engines with a solar-cell-based electric power generation system (EPGS). Exploiting the photo-voltaic (PV) effect and arranged on a solar panel (SP), solar cells convert the sunlight photon's energy directly into electrical energy. The resulting power $P_{e,SP}$ supplies the spacecraft bus and the propulsion subsystem, which contains the thrust units' PPU

$$P_{e,SP} = P_{e,bus} + P_{PPU}. \quad (2.9)$$

Solar cells are in use for in-space electrical power generation since the beginning of spaceflight and have been continuously improved. Current space-qualified solar cell types have conversion efficiencies up to nearly 30 % [25]. An important performance figure is the power-specific mass, which is the ratio of the SP mass m_{SP} to its electric output power $P_{e,SP}$

$$\alpha = \frac{m_{SP}}{P_{e,SP}}. \quad (2.10)$$

Solar panels have power-specific masses of 13 kg/kW for panels with $P_{e,SP} > 10$ kW and 7 kg/kW for $P_{e,SP} = 5-6$ kW [89]. Concentrator lenses and GaAs-cells with a conversion efficiency up to 30 % can theoretically achieve $\alpha = 3-5$ kg/kW, but big-scale in-space validation still remains unachieved.

The intensity of the sunlight radiation is inversely dependent on the square of the distance to the Sun and, theoretically, so is the SP's power output as well. The conversion efficiency however reduces with increasing temperature, which again depends on an SP's sunlight exposure. A power law exponent κ of 1.5 [31] is therefore more realistic. With the power $P_{e,SP,0}$ at the reference distance $r_0 = 1$ AU and the distance from the Sun r_s , $P_{e,SP}$ and m_{SP} are given by

$$P_{e,SP} = P_{e,SP,0} \left(\frac{r_0}{r_s} \right)^\kappa \quad (2.11)$$

and

$$m_{SP} = \alpha P_{e,SP,0}. \quad (2.12)$$

2.2.4 Nuclear Electric Propulsion

Interplanetary missions to the outer solar system, i.e., beyond Mars orbit, cannot rely on a PV-based electrical power generation for propulsion. Supplying an EPS at these distances would require SPs of dimensions almost impossible to assemble today. A nuclearelectric power generation system (NPGS), which converts thermal energy from nuclear fission processes to electrical energy, would be an alternative. Past studies and tests of nuclear reactors mostly dealt with the application to nuclear-thermal rockets. The increasing public concern of potential environment pollution because of launch failures, or nuclear reactors reentering the Earth's atmosphere, let the interest in this technology decline, however. So far, only the Soviet Union launched satellites carrying nuclear reactors for electrical power supply. At this point it is worth mentioning that, contrary to radioisotope thermoelectric generators (see subsection 2.2.5), a uranium fission reactor does not produce any radioactive products prior its activation [89]. This holds also true for the case of launch failures and a potential reentry. The activation of an NPGS must therefore take place only after successful launch and attainment of a safe orbit.

Nuclear reactors are therefore generally safe, and the advantage of a high, sunlight-independent electrical power output¹ would offer new mission possibilities. Operating multiple high- I_{sp} ion engines or MPDTs at the same time and at outer solar system distances would become viable. It is therefore not surprising that this technology regained interest in recent years and was also considered for big-scale space missions. NASA's Jupiter Icy Moons Orbiter (JIMO) proposal, for example, included an NPGS to supply an EPS with eight ion engines [66].

Compared with PV-based EPGSs, an NPGS has considerable advantages:

- independence on Sun distance, spacecraft attitude, and shadowing,
- electrical power output levels between 10–150 kW,
- high mass-specific power.

For a simple nuclear electric power and force model, one can neglect the spacecraft bus power and the decrease of the reactor's nominal power $P_{e,n}^*$ over time. In this case, the power output is constant from reactor activation until end-of-mission (EOM). A NEP-system's thrust force

$$F_{NEP} = \chi f_{P,NEP} P_{e,n}^* \quad (2.13)$$

is calculated with the throttle setting $\chi \in (0, 1)$ and the power-specific force

$$f_{P,NEP} = \frac{F_{NEP}^*}{P_{e,n}^*}, \quad (2.14)$$

whereby F_{NEP}^* is the NEP system's maximum thrust. The propellant mass flow \dot{m}_p calculates with (2.1) and (2.13)

$$\dot{m}_p = \frac{F_{NEP}}{g_0 I_{sp}}, \quad (2.15)$$

and the NEP-acceleration vector is

$$\mathbf{a}_{thr,NEP} = \frac{F_{NEP}}{m_{SC}} \mathbf{e}_f = \frac{\chi f_{P,NEP} P_{e,n}^*}{m_{SC}} \mathbf{e}_f, \quad (2.16)$$

with the thrust acceleration unit vector \mathbf{e}_f and the spacecraft wet mass m_{SC} . The mass of a nuclear reactor m_n is determined with $P_{e,n}^*$ and the power-specific mass α_n ²

$$m_n = \alpha_n P_{e,n}^*. \quad (2.17)$$

2.2.5 Radioisotope Electric Propulsion

While an NPGS exploits the thermomechanical conversion of thermal energy generated through nuclear fission to mechanical energy, a radioisotope power system (RPS) uses the thermal energy of the natural decay of the radioisotope Pu-238 for the same purpose. The half-life of Pu-238 of 87.7 yrs allows for irradiation of thermal energy for long enough to supply space missions lasting for decades. A radioisotope thermoelectric generator (RTG) converts the thermal energy emitted by plutonium oxide pellets glowing at temperatures of 1235 °C. An RTG's thermal efficiency is only $\eta_{th} \approx 6\%$ [89], which necessitates irradiation of excess thermal energy into space via radiators. Thermomechanical generators, e.g., Stirling or Brayton systems, could increase η_{th} to 30–40% [89]. Such an RTG, the advanced Stirling radioisotope generator (ASRG), is being qualified by NASA for a first space application.

¹Russian Topaz reactors for space application achieved 10 kW with an overall mass of 320 kg [89].

² $\alpha_n = 32 \text{ kg/kWe}$ for Topaz nuclear reactor [89]

The ASRG is specified with a mass-specific power output of 8 W/kg [80], corresponding to a power-specific mass of $\alpha_{\text{RTG}} = 125 \text{ kg/kW}$. This is approximately one third more than present general purpose heat source-RTGs with 5.2 W/kg and $P_{\text{e,RTG}} = 140 \text{ W}$.

Because of their independence on sunlight, RTGs have been the preferred EPGS technology for many deep space missions. They powered the Viking Mars probes, the Mariner and Voyager spacecraft, and also the ESA/NASA Cassini-Huygens mission to Saturn. The electrical power was thereby exclusively used to supply the spacecraft bus and the scientific payload but not to power a propulsion system. However, combining multiple RTGs with a low-power electric thruster, e.g., a throttled RIT-10 ion engine, would offer the interesting alternative of REP. For example, REP could shorten the flight time to very large heliocentric distances of 150–200 AU by further long-lasting acceleration beyond the orbits of Jupiter and Saturn [27].

An RTG's electrical power $P_{\text{e,RTG}}$ inevitably follows the natural exponential decay process of radioactive isotopes according to

$$P_{\text{e,RTG}} = P_{\text{e,RTG}}^* \cdot e^{-\lambda t}. \quad (2.18)$$

The power output decreases from the beginning-of-mission (BOM) power $P_{\text{e,RTG}}^*$ over time with a decay constant λ . It is calculated from the half-life $\Delta t_{1/2}$, which is the time span during which the number of radioactive particles of a specimen has halved, as

$$\lambda = \frac{\ln 2}{\Delta t_{1/2}}. \quad (2.19)$$

The mass of an RTG-based EPGS is determined with the BOM power $P_{\text{e,RTG}}^*$ and the power-specific mass α_{RTG}

$$m_{\text{RTG}} = \alpha_{\text{RTG}} P_{\text{e,RTG}}^*. \quad (2.20)$$

To determine the acceleration vector $\mathbf{a}_{\text{thr,REP}}$, one needs the power-specific thrust $f_{\text{P,REP}}$, the throttle factor $\chi \in (0, 1)$, and the spacecraft mass m_{SC}

$$\mathbf{a}_{\text{thr,REP}} = \chi \frac{f_{\text{P,REP}} P_{\text{e,RTG}}}{m_{\text{SC}}} \mathbf{e}_f = \chi \frac{f_{\text{P,REP}} P_{\text{e,RTG}}^* e^{-\lambda t}}{m_{\text{SC}}} \mathbf{e}_f. \quad (2.21)$$

2.3 Solar Sails

Solar sailing is a propulsion technology whose most remarkable feature is the absence of propellant. This gives solar sailing spacecraft, the solar sails, in principle unlimited ΔV -capability. Solar sails could thus enable novel and challenging mission types in interplanetary space and nearby major celestial bodies. A couple of still unsolved technical problems, however, have prohibited their usage until today.

A manifold of different solar sail concepts exists, e.g., the square sail design in Fig. 2.5 or the heliogyro configuration in Fig. 2.6. They do not only differ in shape and performance but also in terms of TRL. It ranges from pure theory over ground qualification tests to first in-space validation tests of smaller scale sails. Vulpetti et al. [92] and McInnes [55] give an overview of different sail designs and mission scenarios based on solar sailing. The next subsection briefly explains the underlying principle of solar sailing, followed by a description of commonly used force models.

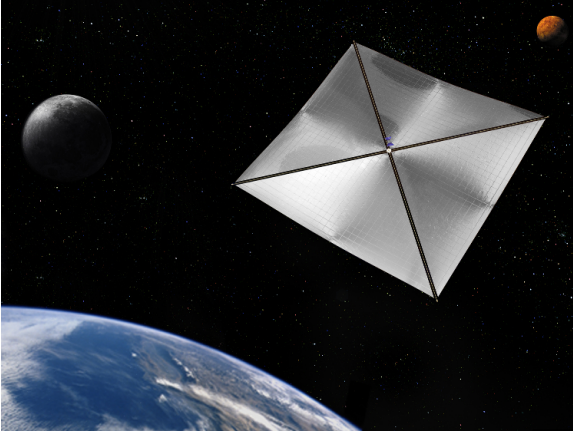


Figure 2.5: Square Solar Sail. Flat square-shaped solar sail with 3-axis control. The triangular sail segments are kept under tension with four extendable booms. Courtesy NASA.

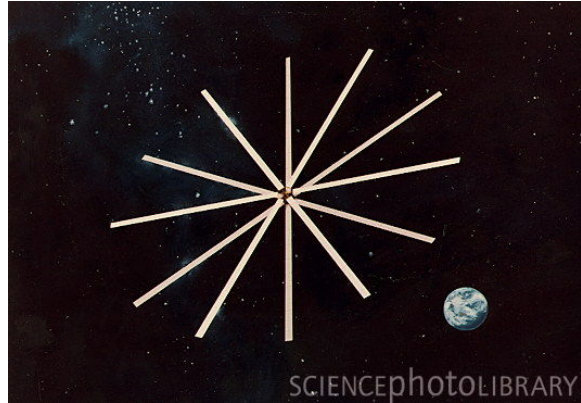


Figure 2.6: Rotating Solar Sail. Rotating solar sail or heliogyro. The sail film's shape is kept through centripetal force caused by rotation. Courtesy Sciencephotolibrary.

2.3.1 Solar Sailing Principle

The principle mechanism of solar sailing is the transfer of the sunlight photons' momentum to the highly reflective solar sail film facing the Sun, and thus to the spacecraft. The flux of solar photons causes the so-called solar radiation pressure (SRP) p_{SRP}

$$p_{\text{SRP}} = \frac{S}{c}, \quad (2.22)$$

where S is the solar radiation flux and c is the speed of light in vacuum. The SRP must not be confused with the Solar Wind, which is also radially emitted by the Sun but consists of charged particles, e.g., electrons, protons or α -particles. The solar radiation flux is inversely dependent on the square of the Sun-spacecraft-distance r_s so that

$$S = S_0 \left(\frac{r_0}{r_s} \right)^2. \quad (2.23)$$

Here, S_0 is the solar constant S_0 at mean Earth-Sun distance $r_0 = 1 \text{ AU}$. Using (2.23) in (2.22) gives

$$p_{\text{SRP}} = \frac{S_0}{c} \left(\frac{r_0}{r_s} \right)^2 = p_{\text{SRP},0} \left(\frac{r_0}{r_s} \right)^2 = 4.563 \text{ Pa} \left(\frac{r_0}{r_s} \right)^2. \quad (2.24)$$

The result of p_{SRP} acting on the solar sail area A is a force, and the reflection of the photons gives a second force of the same size. The resulting absolute force is called Solar Radiation Pressure Force F_{SRP}

$$F_{\text{SRP}} = 2Ap_{\text{SRP}}. \quad (2.25)$$

Although F_{SRP} always points away from the Sun, it can be directed on the complete hemisphere for which the vector product of the sunlight direction vector \mathbf{e}_1 and the sail normal unit vector \mathbf{e}_n is ≥ 0 (see Fig. 2.7).

As an example, one can conclude from (2.25) that a square-shaped solar sail must have a side length $l \approx 330 \text{ m}$ to achieve an acceleration of 1 N at $r_s = r_0$. Although his example is only valid for a sail that is oriented perpendicular to the Sun, it shows a peculiarity of solar sails; their dimensions need to be large for reasonable thrust values. Generally, direction and

magnitude of F_{SRP} do not only depend on the distance from the Sun but also on the spatial orientation of the sailcraft (see 2.3.3).

Solar sails must not only have large dimensions and at the same time maintain attitude control. To increase acceleration, they must also be as lightweight as possible. The space environment and the intense sunlight thereby pose additional challenges to sailcraft design. As the front side is never a perfect reflector, a small fraction of the incoming solar radiation energy is usually absorbed. The backside should therefore have an emissivity high enough to radiate excess thermal energy into space. This way a solar sail's damage from temperature-induced degradation of optical and mechanical properties, or both, can be reduced. Another technical challenge associated with large structures on spacecraft in deep space is to assure the required pointing accuracy for RF communication with the operations team. Big solar sails can only turn at very slow angular rates. Attaining and stabilizing the spacecraft structure for communication and then turning back into the required optimal direction for thrust generation is not trivial. Thus must therefore be accounted for during mission analysis.

2.3.2 Performance Metrics

Performance figures allow the comparison of the capabilities of different solar sail concepts. The metrics described in the following are the sail assembly loading σ_{SA} , the sailcraft loading σ , the characteristic acceleration a_c , and the lightness number λ .

The stiffness of the sail film is insufficient to allow a solar sail to maintain its shape by itself. The sail film is therefore usually attached to a structure that spans the sail film¹ to reduce wrinkles. Wrinkles do not only reduce reflection efficiency but can also cause local hot spots. If the temperature in such hot spots exceeds the sail film material's limit temperature, the sail film will be persistently damaged. A structure is also necessary to mount the payload, i.e., the remaining bus structure including instruments, and to transfer the sail radiation pressure force from the sail film to the entire spacecraft. The sum of the solar sail film mass and the structure mass is called sail assembly mass m_{SA} . Dividing m_{SA} by the sail area A yields the sail assembly loading σ_{SA}

$$\sigma_{\text{SA}} = \frac{m_{\text{SA}}}{A}. \quad (2.26)$$

The sail assembly loading does not include the payload mass m_{PL} . It is therefore a metric only for the comparison of the structural efficiency of different solar sail designs and not their overall performance.

The solar sail mass m_s is the sum of the payload mass m_{PL} and the sail assembly mass m_{SA} . Dividing m_s by the sail area A gives the sail loading σ , which is independent of sailcraft design and technology

$$\sigma = \frac{m_s}{A} = \frac{m_{\text{SA}} + m_{\text{PL}}}{A} = \sigma_{\text{SA}} + \frac{m_{\text{PL}}}{A}. \quad (2.27)$$

High performance solar sails, i.e., those with a high acceleration capability, have lower σ -values than less competitive sail designs.

A solar sail at the distance r_0 , with its reflecting side oriented perpendicular to the sunlight, experiences the characteristic acceleration a_c [55]. It is calculated from the SRP at Earth-Sun mean distance $p_{\text{SRP}}(r_s = r_0) = p_{\text{SRP},0}$, the sail area A , and m_s

$$a_c = \frac{2Ap_{\text{SRP},0}}{m_s} = \frac{2p_{\text{SRP}}}{\sigma}. \quad (2.28)$$

The lightness number λ is defined as the ratio of the maximum SRP acceleration a_{SRP} to the gravitational acceleration $a_g = \mu_s/r_s^2$, with the Sun's gravitational parameter μ_s . The

¹This does not hold true for rotating sail structures, e.g., the heliogyro concept shown in Fig. 2.6.

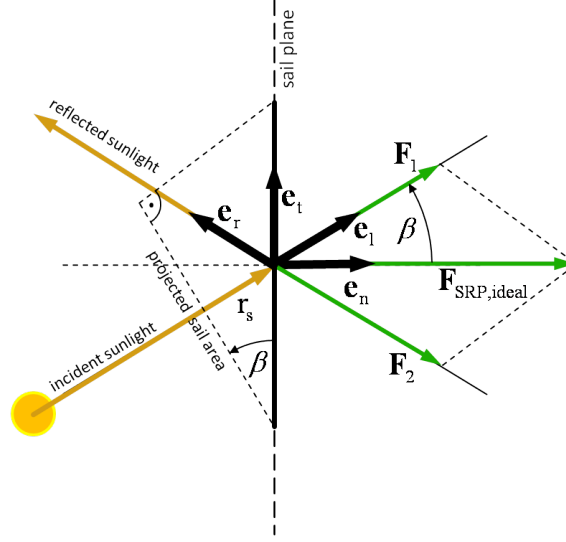


Figure 2.7: Ideal SRP Force model. Perfect reflection of sunlight photons at the front side of a flat solar sail having the position vector \mathbf{r}_s w.r.t. the Sun. The resulting vectors \mathbf{F}_1 and \mathbf{F}_2 sum to the SRP force vector $\mathbf{F}_{\text{SRP,ideal}}$. The pitch angle β is defined between the sail plane unit vector \mathbf{e}_n and the unit vector of incident sunlight \mathbf{e}_1 . The transversal unit vector $\mathbf{e}_t = (\mathbf{e}_n \times \mathbf{e}_1) \times \mathbf{e}_n$ is in the sail plane and in the plane spanned by \mathbf{e}_n and \mathbf{e}_1 . The unit vector \mathbf{e}_r shows the reflection direction.

maximum acceleration a_{SRP} occurs when the sail unit vector \mathbf{e}_n and the incoming sunlight unit vector \mathbf{e}_1 are aligned ($\mathbf{e}_1 \cdot \mathbf{e}_n = 1$). Under this condition, the lightness number is

$$\lambda = \frac{a_{\text{SRP}}}{a_g} = \frac{a_c \left(\frac{r_0}{r_s} \right)^2}{\frac{\mu_s}{r_s^2}} = \frac{a_c r_0^2}{\mu_s} = \frac{a_c}{a_0}. \quad (2.29)$$

Equation (2.29) also gives the relation between a_c and the gravitation acceleration a_0 of the Sun acting on a body at the distance r_0 .

2.3.3 SRP Force Models

A variety of solar sail force models of different complexity exists, and three of them are presented in the following. These are the ideal force model, the simple force model, and the standard force model.

2.3.3.1 Ideal SRP Force Model

The cause of the SRP force is a two momentum transfer between sunlight photons and the solar sail film. The first one takes place when the photons impinge on the sail surface, and the result is a force component in sunlight direction, described by $\mathbf{e}_1 = \sin \beta \mathbf{e}_t + \cos \beta \mathbf{e}_n$ as shown in Fig. 2.7

$$\mathbf{F}_1 = p_{\text{SRP}} A \cos \beta \mathbf{e}_1. \quad (2.30)$$

The so-called pitch angle

$$\beta = \arccos(\mathbf{e}_1 \cdot \mathbf{e}_n) \quad (2.31)$$

is defined between \mathbf{e}_1 and the sail plane normal unit vector \mathbf{e}_n . The momentum transfer due to back reflection of the photons in direction $\mathbf{e}_r = \sin \beta \mathbf{e}_t - \cos \beta \mathbf{e}_n$ results in a second force of opposite direction

$$\mathbf{F}_2 = -p_{\text{SRP}} A \cos \beta \mathbf{e}_r. \quad (2.32)$$

Vector addition of \mathbf{F}_1 and \mathbf{F}_2 yields

$$\mathbf{F}_{\text{SRP,ideal}} = \mathbf{F}_1 + \mathbf{F}_2 = p_{\text{SRP}} A \cos \beta (\mathbf{e}_1 - \mathbf{e}_r), \quad (2.33)$$

which can be simplified with $\mathbf{e}_1 - \mathbf{e}_r = 2 \cos \beta \mathbf{e}_n$ as

$$\mathbf{F}_{\text{SRP,ideal}} = f(r_s, \beta, \mathbf{e}_n) = 2P_{\text{SRP},0} \left(\frac{r_0}{r_s}\right)^2 A \cos^2 \beta \mathbf{e}_n \quad (2.34)$$

$$\mathbf{a}_{\text{SRP,ideal}} = \frac{\mathbf{F}_{\text{SRP,ideal}}}{m_s} = 2p_{\text{SRP},0} \left(\frac{r_0}{r_s}\right)^2 \frac{A}{m_s} \cos^2 \beta \mathbf{e}_n. \quad (2.35)$$

An alternative expression of (2.35) is possible by using the sail loading (2.27)

$$\mathbf{a}_{\text{SRP,ideal}} = 2 \frac{p_{\text{SRP},0}}{\sigma} \left(\frac{r_0}{r_s}\right)^2 \cos^2 \beta \mathbf{e}_n. \quad (2.36)$$

For $\mathbf{e}_n \cdot \mathbf{e}_1 = \cos \beta = 1$ and $r_s = r_0$, the characteristic acceleration (2.28) and the corresponding acceleration vector are

$$a_{c,\text{ideal}} = \frac{2p_{\text{SRP},0}}{\sigma} \quad (2.37)$$

$$a_{\text{SRP,ideal}} = a_{c,\text{ideal}} \left(\frac{r_0}{r_s}\right)^2 \cos^2 \beta \quad (2.38)$$

$$\mathbf{a}_{\text{SRP,ideal}} = a_{\text{SRP,ideal}} \mathbf{e}_n. \quad (2.39)$$

2.3.3.2 Simple SRP Force Model

The simple SRP force model is a compromise between the ideal SRP force model and more complex force models, such as the standard SRP force model (2.3.3.3). Sailcraft configuration and real-world effects like wrinkles and sail billowing are only coarsely approximated with an overall sail efficiency parameter $\eta \in (0, 1)$. The respective equations are

$$\mathbf{F}_{\text{SRP,simple}} = 2\eta p_{\text{SRP},0} \left(\frac{r_0}{r_s}\right)^2 A \cos^2 \beta \mathbf{e}_n \quad (2.40)$$

$$a_{c,\text{simple}} = \frac{2\eta p_{\text{SRP},0}}{\sigma} \quad (2.41)$$

$$\mathbf{a}_{\text{SRP,simple}} = a_{\text{SRP,simple}} \mathbf{e}_n = \eta a_{\text{SRP,ideal}} \mathbf{e}_n = \frac{2\eta p_{\text{SRP},0}}{\sigma} \left(\frac{r_0}{r_s}\right)^2 \cos^2 \beta \mathbf{e}_n. \quad (2.42)$$

This force model combines the simplicity of the ideal SRP force model with a higher degree of realism. However, the efficiency parameter must be chosen carefully and dependent on the solar sail technology.

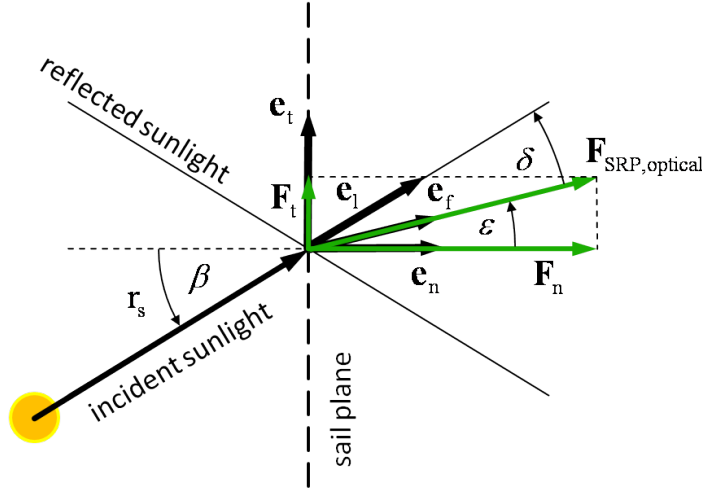


Figure 2.8: The Optical SRP Force Model. The pitch angle β is defined between the sail plane unit vector \mathbf{e}_n and the unit vector of the incident sunlight \mathbf{e}_1 . The vector \mathbf{r}_s is the solar sail's position vector w.r.t. the Sun. The unit vector $\mathbf{e}_t = (\mathbf{e}_n \times \mathbf{e}_1) \times \mathbf{e}_n$ is in the plane spanned by \mathbf{e}_n and \mathbf{e}_1 perpendicular to \mathbf{e}_n and $(\mathbf{e}_n \times \mathbf{e}_1)$.

2.3.3.3 Standard SRP Force Model

The standard SRP force model describes the physics of a solar sail in space more accurately than the ideal SRP force model, and therefore it has become a standard in solar sail-related literature. As it principally relies on the sail film's optical parameters, it is also named optical force model [55].

Even the best sail film does not completely reflect incoming light but absorbs a fraction of the photon's energy. This energy flux leads to an increase of the sail film temperature and causes a temperature dependent irradiation of energy. The SRP optical force model vector $\mathbf{F}_{\text{SRP,optical}}$ therefore consists of components resulting from reflection, absorption, and emission (see appendix C)

$$\mathbf{F}_{\text{SRP,optical}} = \mathbf{F}_r + \mathbf{F}_a + \mathbf{F}_e. \quad (2.43)$$

Magnitude and direction of $\mathbf{F}_{\text{SRP,optical}}$ depends on the sail film's optical properties, which are described with the absorption coefficient α , the transmission coefficient τ , and the reflection coefficient ρ . Energy conservation demands the sum of absorption, transmission, and emission to equal the incoming energy. The associated coefficients must therefore fulfill the relation

$$\alpha + \rho + \tau = 1. \quad (2.44)$$

Within this force model, the energy impinging on the solar sail front side is assumed to be absorbed completely first and then the major part of this energy is either reflected or re-radiated. Under the assumption of zero transmission ($\tau = 0$), α can be expressed with ρ

$$\alpha = 1 - \rho. \quad (2.45)$$

Reflection divides into a specular and a diffuse part.

$$\rho = \rho_s + \rho_d \quad (2.46)$$

The force vector $\mathbf{F}_{\text{SRP,optical}}$ in Fig. 2.8 consists of a normal component \mathbf{F}_n and a transversal component \mathbf{F}_t (see appendix C)

$$\mathbf{F}_{\text{SRP,optical}} = \mathbf{F}_n + \mathbf{F}_t \quad (2.47)$$

Table 2.2: Optical parameters for an ideal solar sail, a Al|Cr-coated JPL square sail, and JPL heliogyro [55].

	ρ	ρ_s	ϵ_f	ϵ_b	B_f	B_b
Ideal sail	1	1	0	0	2/3	2/3
Square sail/Heliogyro	0.88	0.83	0.05	0.55	0.79	0.55

$$\mathbf{F}_n = p_{\text{SRP}} A \cos \beta \left((1 + \rho_s) \cos \beta + \rho_d B_f + (1 - \rho) \frac{\epsilon_f B_f - \epsilon_b B_b}{\epsilon_f + \epsilon_b} \right) \mathbf{e}_n \quad (2.48)$$

$$\mathbf{F}_t = p_{\text{SRP}} A \cos \beta (1 - \rho_s) \sin \beta \mathbf{e}_t. \quad (2.49)$$

Table 2.2 provides values of ρ , ρ_s , the emissivity factors ϵ_f and ϵ_b , and the non-Lambert coefficients B_f and B_b of the sail's front and back side for different sailcraft configurations. With the centerline angle

$$\epsilon = \arctan \left(\frac{F_t}{F_n} \right) = \arctan \left(\frac{(1 - \rho_s) \sin \beta}{(1 + \rho_s) \cos \beta + \rho_d B_f + (1 - \rho) \frac{\epsilon_f B_f - \epsilon_b B_b}{\epsilon_f + \epsilon_b}} \right) \quad (2.50)$$

and the unit vector in SRP force direction

$$\mathbf{e}_f = \sin \epsilon \mathbf{e}_t + \cos \epsilon \mathbf{e}_n \quad (2.51)$$

an equivalent formula of the optical SRP force model is

$$\mathbf{F}_{\text{SRP,optical}} = \sqrt{F_n^2 + F_t^2} \mathbf{e}_f. \quad (2.52)$$

Alternatively, also the thrust cone angle $\delta = \beta - \epsilon$ is used to describe the thrust direction. It is defined between \mathbf{e}_f and \mathbf{e}_1

$$\delta = \arccos (\mathbf{e}_f \cdot \mathbf{e}_1). \quad (2.53)$$

3

Trajectory Optimization

At a specific point in time t , a spacecraft's position $\mathbf{r}(t)$ and velocity $\mathbf{v}(t)$ completely describe its astrodynamic state

$$\mathbf{y}(t) = \begin{pmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \end{pmatrix}, \quad (3.1)$$

and on trajectory begin, at the so-called epoch t_0 , this state vector is consequently

$$\mathbf{y}_0 = \mathbf{y}(t_0) = \begin{pmatrix} \mathbf{r}(t_0) \\ \mathbf{v}(t_0) \end{pmatrix}. \quad (3.2)$$

The spacecraft motion is governed by Newton's physical laws¹. That means, its state of motion changes through the application of forces. Those can be external ones, like gravitational attraction, or forces generated by the spacecraft itself, e.g., thrust forces from propulsion systems. Depending on \mathbf{y}_0 and on the forces acting on the spacecraft, the state vector $\mathbf{y} = \mathbf{f}(t)$ changes over time and assumes the final state $\mathbf{y}(t_f)$ after the flight time Δt at $t_f = t_0 + \Delta t$. The three-dimensional, continuous position vector function $\mathbf{r}(t)$ with $t \in (t_0, t_f)$ is called trajectory or flight path. The general problem is thereby to determine $\mathbf{r}(t)$ for a given initial state and the known acting forces.

Analytical formulations of a solution to this problem exist only for special cases. The classical Kepler problem is an example for such a case. It requires the determination of a body's trajectory for a given initial state and under the Newtonian gravitational influence of a central body. The resulting analytical trajectory is called Keplerian orbit and described with six so-called Keplerian parameters (see also appendix B). As the determination of a body's trajectory for the general case is impossible, a time-discretized approximation of $\mathbf{r}(t)$, obtained through numerical integration of the equations of motion, is therefore normally used instead. The resulting position vector set $\mathbf{r}[t_k]$ with $t_0 \leq t_k \leq t_f$ is then also called trajectory.

The problem gets more difficult if a certain trajectory type is required. That is the case if it should fit a specific purpose, such as bringing a spacecraft from one orbit to another orbit or celestial body. To find the trajectory that fits this purpose best w.r.t. a given criterion, e.g., with the lowest energy demand, is even more difficult. An example of such a minimum-energy transfer, which is only valid for coplanar circular orbits, is the so-called Hohmann transfer [41].

From a flight dynamics point of view, space missions, especially interplanetary ones, therefore depend on whether the target body or target orbit can be attained at all, and on the necessary effort. This becomes evident when thinking of the compromise each mission designer has to

¹Relativity effects are neglected for this introduction but must be respected for physical correctness at very large velocities close to the speed of light. Within the scope of this thesis, $v \ll c$ is assumed.

make between schedule, cost, required technology, and risk. A space mission's chances for realization therefore implicitly depend on whether a trajectory for the posed objectives exists in the given mission time frame. Furthermore, the influence of potentially several trajectory alternatives on the overall mission design, its technical implementation, and the necessary effort is often considerable. The chosen trajectory influences key mission design parameters such as ΔV , mission duration, and the spacecraft's launch mass m_0 . The latter is primarily important if propellant-dependent propulsion systems are employed, as the propellant mass accounts for a major fraction of a spacecraft's total launch mass. Also for the by principle propellant-less solar sails, the trajectory optimization can yield a required minimum characteristic acceleration. For example, if the spacecraft mass without the solar sail is fixed, bigger and technologically more challenging sail structures may become necessary to achieve that acceleration.

Trajectory optimization is therefore essential already at early mission analysis and design stages. It is an indispensable tool for mission designers to balance the mission requirements with the available resources and technological capabilities.

This chapter addresses different, general aspects of low-thrust trajectory optimization. It begins with possible optimization objectives and the concept of Pareto-optimality in section 3.1, followed by the formulation of the general single-phase and multiphase, low-thrust optimization problem in section 3.2. Section 3.3 shows how low-thrust steering can be achieved through the application of local steering laws. Different control methods and optimization methods and the difference between global and local optimization techniques are covered in section 3.4. Finally, section 3.5 provides a concept of a smart, multiphase, low-thrust optimization method.

3.1 Objectives and Optimality

The optimality of a spacecraft trajectory expresses if certain problem-dependent boundary conditions or constraints are matched or if they are not violated. An example of such a constraint is the allowed minimum distance to the Sun under which the spacecraft should not fall below during a transfer in order to keep its thermal balance. If all conditions were met, the optimality is furthermore an expression for how good the actual optimization criterion has been fulfilled. Optimality criteria can thus be understood as a superset of conditions, constraints, and metrics. Such optimality criteria can be the transfer time Δt , the ΔV that is required for a certain trajectory, or, alternatively, the required propellant mass m_p ¹. Possible optimization objectives for propellant-dependent missions are therefore:

1. minimization of Δt for a given m_0 ,
2. minimization of Δt for a given m_{PL} ,
3. maximization of m_{PL} through minimization of m_p for a given Δt and m_0 ,
4. minimization of m_p for a given Δt and m_{PL} , which effectively reduces the launch system requirements through the minimization of m_0 .

For solar sails, the following optimization objectives can be thought of:

1. minimization of Δt for a given sailcraft performance metric, e.g., a_c ,
2. minimization of a specified performance metric for a given Δt ,

¹The optimization for m_p is impossible for solar sails.

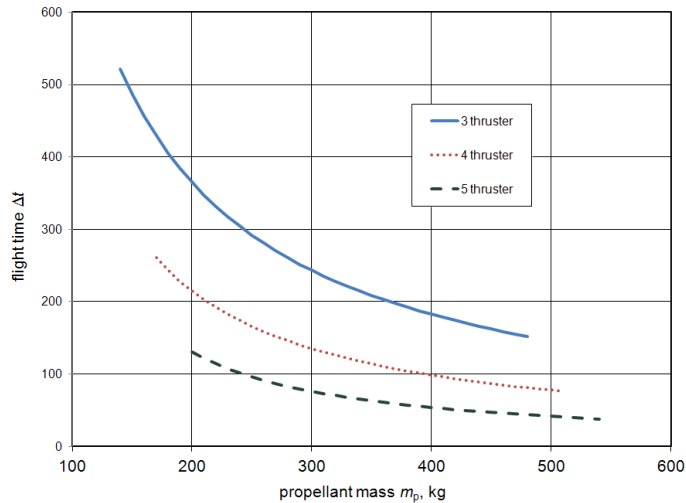


Figure 3.1: Pareto-optimality Curves of an Example EP Mission with Three, Four, or Five Thrust Units. This figure shall help explaining the generic problem of concurrent, mutually affecting mission parameters, such as flight time, complexity, and mass. Implicitly, and together with other factors, these parameters ultimately also affect mission cost. The two parameters of the number of thrust units and m_p significantly drive technical design and implementation decisions.

Each curve shows the flight time Δt over the propellant mass m_p of the pareto-optimal solutions for a fixed number of thrust units with equal maximum thrust. It qualitatively shows the benefit of a having more thrust units, but the reduction in Δt becomes smaller with each additional thruster. Furthermore, the reduction of Δt due to increased m_p becomes negligible beyond a certain mass.

3. minimization of the solar sail film's solar radiation dose (SRD) for a given maximum Δt .

An optimization carried out w.r.t. a single criterion is called single-objective optimization, while multi-objective optimization involves multiple criteria. Most real-world optimization problems, e.g., finding an optimal spaceflight trajectory, are of multiobjective nature. A problem solution generally cannot satisfy all criteria to best possible extent, i.e., if one criterion is optimally fulfilled the others are not and vice versa. Applied to the trajectory optimization problem, the solution with the smallest Δt is often suboptimal in terms of ΔV and thus m_p . Solutions to multi-objective problems may therefore be treated with the concept of pareto optimality, developed by the Italian sociologist and economist Vilfredo Pareto (1848-1923). A pareto-optimal solution is not dominated by any other solution in all criteria, or, in other words, an improvement of a pareto-optimal solution in one criterion inevitably results in worsening it in at least one other criterion. If all but two problem parameters are fixed, the resulting plot of pareto-optimal solutions is the pareto optimality curve. For trajectory optimization problems, this is typically Δt and either m_0 or m_p , as can be seen in Fig. 3.1. The example pareto-optimality curves show the sensitivity of one criterion to the others, and such plots can be helpful for the assessment of possible design trade-offs. One option is the decision against increasing m_p because the pareto-curve tells that Δt is less sensitive, i.e., decreases only insignificantly, beyond a critical propellant mass. Another option is the determination of the optimum number of EP thrusters, because the relative improvement of Δt reduces with each additional thruster.

Hartmann showed with the example of low-thrust Earth-Mars transfers how to transform a multiobjective problem into a single-objective problem [39]. For a two-criteria problem, this

is typically done on two ways:

- Only one criterion is subject to optimization, with the other criteria being constraints.
- Using weighting factors. The fulfillment of both criteria is combined into one scalar value to be optimized. To decide on proper weighting factors is not trivial and requires profound knowledge of the particular problem. This often turns the determination of adequate weighting factors into a separate optimization task.

In practice, it may depend on the particular problem or on the employed optimization methodology, or both, which one of the two ways leads to acceptable results.

3.2 The Multiphase, Low-thrust Optimization Problem

The optimization of multiphase low-thrust trajectories requires addressing of several subproblems, which are treated within the following sections. After the explanation of the fundamental difference between low- and high-thrust trajectories in subsection 3.2.1, the problem is viewed from the perspective of optimal control in subsection 3.2.2. Subsection 3.2.3 finally explains the difference between single-phase and multiphase transfers.

3.2.1 Low-thrust and High-thrust Trajectories

The shape of low-thrust trajectories differs significantly from that of spacecraft driven by chemical propulsion systems because of different acceleration characteristics of these propulsion technologies w.r.t. thrust magnitude and thrust duration. While low-thrust propulsion systems apply a highly propellant-efficient thrust of a few Newtons at maximum for months or even years, chemical propulsion systems can operate only for minutes to hours but with absolute thrust values that are magnitudes higher.

In the following subsection 3.2.1.1 and subsection 3.2.1.2, the effect of these different operation characteristics on the resulting trajectories is described. Then, in subsection 3.2.1.3 and subsection 3.2.1.4, the pros and cons of low-thrust trajectories in comparison to chemical ones are discussed from a mission design perspective.

3.2.1.1 High-thrust Trajectories

Compared to the duration of the entire transfer, the acceleration phase of a chemically propelled spacecraft is negligibly short, which makes it basically a free-flight phase. Such a transfer is determined by the state $\mathbf{x}(t_0)$ at burnout of the rocket's upper stage and the launch time t_0 ¹. Along the trajectory, the spacecraft is subject to the dominating gravitational attraction of the respective central body and the perturbing accelerations of other celestial bodies, e.g., of Jupiter - our solar system's second-largest body. Eventual small deviations from the predetermined $\mathbf{x}(t_0)$ are correctable with midcourse correction maneuvers (MCM) [21]. These are small- ΔV maneuvers, executed at certain trajectory points with small rocket engines or cold gas systems. For a particular mission, the spacecraft must achieve the six predetermined parameters of $\mathbf{x}(t_0)$ at t_0 to reach its target. If the spacecraft exhibits no

¹In this context, t_0 means date and time-of-day.

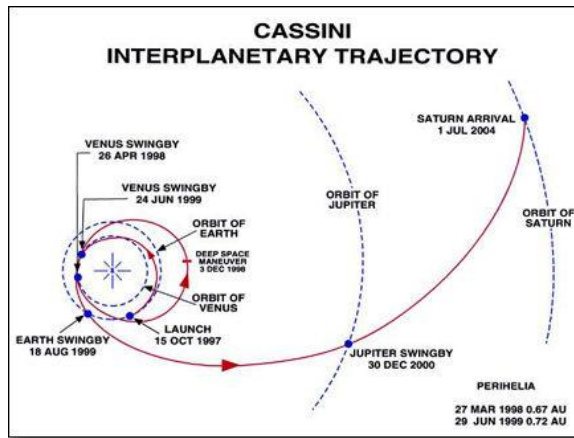


Figure 3.2: Example of a Multiphase High-thrust Trajectory. The trajectory of the ESA/NASA mission Cassini-Huygens serves as an example of a multiphase, high-thrust transfer with the typical ballistic, free-flight phases. Additionally, this transfer includes a Venus-Venus-Earth-Jupiter gravity assist sequence. Courtesy JPL/NASA.

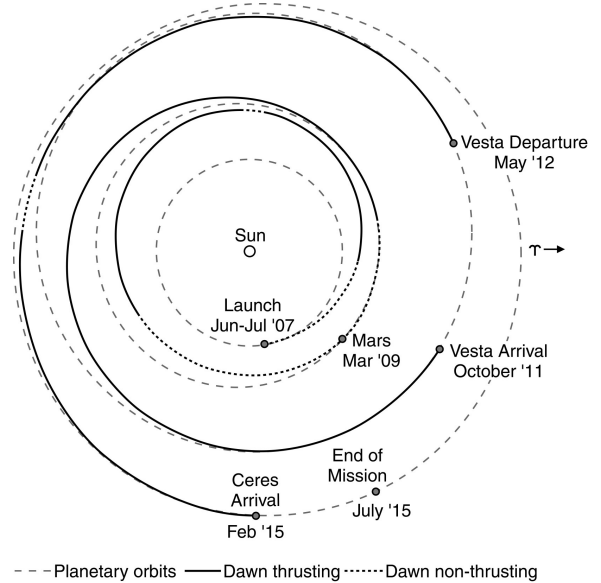


Figure 3.3: Low-thrust Transfers. The trajectory of NASA's deep space mission Dawn shows the principle difference of low-thrust missions, with their long thrust phases, to high-thrust missions. It also includes a gravity assist at Mars. Courtesy JPL/NASA.

hyperbolic excess velocity, i.e., it barely leaves the launch body's gravity field, then the initial state $\mathbf{x}(t_0)$ of the spacecraft equals¹ that of the launch body

$$\mathbf{x}(t_0) = \begin{pmatrix} \mathbf{R}_{\text{Earth}}(t_0) \\ \mathbf{V}_{\text{Earth}}(t_0) \end{pmatrix}. \quad (3.3)$$

Although t_0 is a fixed point in time, there might be a mission-dependent set of feasible $(\mathbf{x}(t_0), t_0)$ within a so-called launch window. It can be several weeks long, with daily launch opportunities, but it may also be only a few days long if the mission involves multiple gravity assists at planets, as can be seen in Fig. 3.2. It shows the heliocentric trajectory of NASA's Saturn system mission Cassini-Huygens, with the typical features of chemical interplanetary mission design. After launch at Earth, a series of gravity assists at Venus, Earth, and Jupiter is necessary to attain the final transfer orbit to Saturn. After having reached the target body, a second velocity increment is needed. This so-called insertion burn can be optimized together with the launch ΔV . Finding the optimal² balance between flight time, required propellant, mass budget, launch opportunities, and necessary margins requires compromises and also profound experience.

Neglecting gravity assists, the high-thrust transfer problem between two celestial bodies is defined with the launch and arrival time, and the 12 parameters of the four, 3-dimensional vectors of position and velocity at launch and arrival. There is not an a priori best transfer trajectory between two celestial bodies but an indefinite number, each requiring a specific mission- ΔV and having a resulting Δt . The problem of determining the ΔV for a given two-body transfer problem and the Δt is known as Gauss problem and can be solved with an

¹The spacecraft position is of course not exactly that of the launch body, this assumption is however valid for preliminary mission analysis.

²Optimality is almost impossible to define for this type of problem. It can therefore hardly ever be mastered, as design of a space mission continues and less variables can be modified with ongoing project time.

algorithm given by Bate et al. [5]. The method is also known as Lambert-solver and gives a mission designer an overview of the ΔV -surface over the 2-dimensional Δt - t_0 -parameter space. The assessment of the resulting solutions to this multiobjective optimization problem and the selection from potentially arising mission design options is the task of mission designers and not discussed here.

3.2.1.2 Low-thrust Trajectories

Low-thrust trajectories typically consist of long thrust phases and small coast phases. In addition to the initial state $\mathbf{x}(t_0)$, the gravitational acceleration of the central body and third bodies, the acceleration due to an operating low-thrust propulsion system determines the trajectory shape. Contrary to chemical propulsion systems, this low-thrust acceleration is active for a major fraction of the mission time, and its vector can basically point into any¹ direction. The actual thrust duration however depends on the employed propulsion system technology and on the carried-with m_p , if a propellant is necessary. A low-thrust trajectory is thus mainly determined through a steering strategy which is the guidance that provides the thrust acceleration vector at any time t . The optimal solution of a low-thrust trajectory is therefore a combination of an optimal $\mathbf{x}(t_0)$ at the optimal launch date t_0 and an optimal steering strategy. This makes low-thrust trajectory optimization in general a difficult task. Characteristic for low-thrust trajectories is their many-revolutions shape, which results from the low acceleration level which allows only for small orbit changes per revolution. Figure 3.3 shows an example of an heliocentric low-thrust trajectory. It is the trajectory of the NASA mission New Millennium Deep Space 1 which relies on EP and a gravity assist at Mars to visit the two asteroids Vesta and Ceres.

3.2.1.3 Advantages of Low-thrust Propulsion For Mission Design

Low-thrust propulsion enables high- ΔV missions that are hardly realizable with chemical, high-thrust propulsion systems. Even if chemical propulsion is viable for the respective mission, its design will almost inevitably comprise gravity assists. For the design of a deep space mission, the trajectories resulting from low-thrust propulsion therefore have several advantages:

- **New mission types.** Due to their unique feature of only sunlight-dependent SRP acceleration, solar sails offer new applications and mission types. They could, for example, use non-Keplerian polar Sun orbits for continuous monitoring of the Sun's poles.
- **Fuel-efficient thrust due to the high I_{sp} .** For a given ΔV , the efficient propellant utilization of EP allows for a reduced m_p , an increased m_{PL} , or a mixture of both. Potentially, if the spacecraft launch mass m_0 reduces through a reduction of m_p , also a smaller launch vehicle becomes possible.
- **Wider launch windows.** Low-thrust propulsion systems enable high- ΔV space missions without the gravity assist maneuvers that might be necessary for the corresponding high-thrust mission design alternative. This, together with the capability to “shape” the trajectory during flight with a suited steering strategy, not only widens the launch window but also enables more frequent launch opportunities.

¹There are limits for the thrust direction vector for solar sails.

- **Reduced sensitivity on injection accuracy.** The $\mathbf{x}(t_0)$ is never accurately matched, i.e., injection errors do inevitably always occur. A low-thrust stage can however correct for larger injection errors than a spacecraft on the ballistic flight phase of a chemical mission would be able to. Low-thrust propulsion can therefore help to lower the risk of mission failure.
- **Reduced flight time Δt .** If a chemical, high- ΔV mission designs contains gravity assists and a concurrent low-thrust mission design alternative does not, the Δt of the latter is often smaller.

The influence of these advantages can not be generalized in the sense that low-thrust is beneficial to all mission types. An increased flexibility in launch date and window, however, reduces mission failure risk especially for deep space missions. A current example for this is ESA's comet rendezvous and in situ measurement mission ROSETTA, which suffered from launch delay caused by the launch segment. Its original target object, the comet 46P/Wirtanen, could therefore not be met any more. The redesign of the mission resulted in the new primary target 67P/Tschurjumow-Gerasimenko and also different asteroid flybys on the way to that comet [4].

3.2.1.4 Disadvantages of Low-thrust Propulsion For Mission Design

Low-thrust propulsion systems are despite their advantages not a priori the best propulsion technology for every deep space mission and not for every mission phase. The actual mission-specific criteria, how they are best met, and at which cost, determine whether low-thrust propulsion should be favored over chemical propulsion. The following disadvantages of low-thrust systems and their effect on mission design must therefore be considered:

- **High P_e -demand.** EP-systems have an electrical power demand P_e in the range of several kW to MW, especially if multiple thrust units are operated simultaneously. While solar-electric power generation systems can provide up to ≈ 50 kW, P_e -values in the MW-regime inevitably requires an NPGS. Space-qualified, ready-for-flight nuclear reactors do not exist and solar-electric power generation is not only limited but also strongly dependent on the distance to the Sun.
- **Technological readiness level.** Up until now, the practical in-space experience with low-thrust propulsion systems covers approximately a decade. This is hardly comparable to the decades of heritage with chemical propulsion systems. For space missions requiring a very high reliability and a very low mission failure risk, this can exclude the usage of certain low-thrust propulsion systems.
- **Increased flight time Δt .** Due to the different thrust characteristic, the transfer time of low-thrust missions is often but not necessarily higher than their chemical alternative if both do not use gravity assists. For example, ESA's low-thrust, lunar mission SMART-1 flew to the Moon in about one year while the Apollo spacecraft crossed that distance in only about three days on a ballistic trajectory.
- **Increased mission ΔV .** A low-thrust trajectory close to a celestial body, e.g., an EP-powered Earth system escape trajectory beginning on a geostationary transfer orbit (GTO), involves many revolutions about that body. During this time, the spacecraft's velocity vector is not exactly anti-parallel to the gravitational acceleration vector. Due to the long thrust duration, the resulting, so-called gravity loss ΔV_{loss} is not negligible and increases the total mission ΔV .

- **Longer exposure to “no go” areas.** The low-thrust propulsion peculiarity of small orbit changes rates can result in multiple revolutions around the central body. This can lead to long dwell times in regions of strong radiation near some celestial bodies, e.g., the Van-Allen belts of the Earth. Together with the potential disadvantage of long flight times, this could prohibit low-thrust systems from being used, e.g., for manned spaceflight in that environment.

Each potential mission design must be assessed w.r.t. those disadvantages. Eventually one or more of these can rule out the use of low-thrust propulsion for a particular spacecraft and mission.

3.2.2 Low-thrust Trajectories From an Optimal Control Perspective

Subsection 3.2.1.2 showed that a low-thrust trajectory depends on the employed steering strategy, which defines the direction and magnitude of the thrust acceleration vector over time. In terms of optimal control theory, the steering strategy therefore equates the control function \mathbf{U} . It determines the values of the control vector $\mathbf{u} \in \mathbb{R}^{n_u}$, e.g., two steering angles and the magnitude of the thrust vector.

Within this work, two elementary low-thrust transfer types were used: the flyby (FB) and the rendezvous (RV). In simple terms, a flyby requires to match the spacecraft’s position with those of the target, while a rendezvous at the same time additionally requires the respective velocity to match. The target states of both transfer types can be manifold, although there are mostly used with celestial bodies as target states. Nevertheless, possible target types are:

- a celestial body,
- the orbit of a celestial body,
- a free orbit described by (incomplete) set of orbital elements,
- a fixed state vector,
- a fixed position vector.

A verbal description of each of these elementary problem types simple, e.g., steer a low-thrust spacecraft time-optimally from the initial state to the target state. Depending on the point of view or the context in which the problem is treated, different expressions are possible as well. This section states these elementary transfers from a optimal control perspective, i.e., the FB and RV problems are formulated as optimal control problems in continuous time formulation in subsection 3.2.2.1 and in discrete time formulation in subsection 3.2.2.2. Later, these transfers will then be reformulated as artificial intelligence and machine-learning problems.

3.2.2.1 Problem Formulation in Continuous Time

The following definitions of the low-thrust FB problem and the low-thrust RV problem are adapted versions of the ones given by Dachwald [13]. They were originally formulated in the context of heliocentric, low-thrust trajectories, but their general formulations also hold true for non-heliocentric problems.

RV problem from the perspective of optimal control theory:

Find a spacecraft control function $\mathbf{U}: t \in [t_0, t_f] \rightarrow \mathbf{u} \in \mathbb{R}^{n_u}$, which forces the state $\mathbf{x}_{\text{SC}}(t) = (\mathbf{r}_{\text{SC}}(t)^\top, \dot{\mathbf{r}}_{\text{SC}}(t)^\top)^\top$ of the spacecraft from its initial value $\mathbf{x}_{\text{SC}}(t_0)$ to the state $\mathbf{x}_{\text{T}}(t)$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\text{SC}}(t) = \mathbf{G}(\mathbf{x}_{\text{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{x}_{\text{SC}}(t_f) = \mathbf{x}_{\text{T}}(t_f)$, and at the same time minimizes a specific cost function J for that transfer.

FB problem from the perspective of optimal control theory:

Find a spacecraft control function $\mathbf{U}: t \in [t_0, t_f] \rightarrow \mathbf{u} \in \mathbb{R}^{n_u}$, which forces the position $\mathbf{r}_{\text{SC}}(t)$ of the spacecraft from its initial value $\mathbf{r}_{\text{SC}}(t_0)$ to the position $\mathbf{r}_{\text{T}}(t)$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\text{SC}}(t) = \mathbf{G}(\mathbf{x}_{\text{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{r}_{\text{SC}}(t_f) = \mathbf{r}_{\text{T}}(t_f)$ ¹, and at the same time minimizes a specific cost function J for that transfer.

The definition of the cost function J depends on the actual optimization goal. If the flight time Δt was to be minimized, its definition could be

$$J_{\text{T}} = \int_{t_0}^{t_f} dt = t_f - t_0 = \Delta t. \quad (3.4)$$

For a propellant mass optimization it could be

$$J_{m_p} = \int_{t_0}^{t_f} \dot{m}_p dt = m_p(t_f) - m_p(t_0). \quad (3.5)$$

Both dates t_0 and t_f can either be fixed or also subject to optimization. The problem of finding the optimal trajectory \mathbf{x}_{SC}^* is thus transformed into finding the optimal control $\mathbf{u}^*(t)$, which is determined through the optimal spacecraft control function \mathbf{U}^* .

3.2.2.2 Problem Formulation in Discrete Time

The continuous n_u -dimensional control $\mathbf{u}(t)$ and the control function \mathbf{U} are element of an infinite function space. Finding an optimal low-thrust trajectory therefore means to find an optimal \mathbf{U}^* in a per definition infinite solution space. This is possible only in rare cases, and analytical solutions to the general FB and RV problem are in fact unknown.

Numerical solutions are however feasible, and the first step to obtain them is the reduction of the problem complexity through transformation into a finite-dimensional problem. A discretization of $[\underline{t}_0, \bar{t}_f]$ in τ intervals with $\underline{t}_0 \leq t_0 = \bar{t}_0, \dots, \bar{t}_f = t_f \leq \bar{t}_\tau \leq \bar{t}_f$ reduces the dimension of the control function parameter space to $n_u \tau$. However, with a dimension of $n_u \tau$, the search space is still substantial. Through the discretization it is not necessary to find \mathbf{U}^* but the spacecraft control vector history $\mathbf{u}^*[\bar{t}] \in \mathbb{R}^{n_u \tau}$. Applying $\mathbf{u}^*[\bar{t}]$ then yields the optimal trajectory $\mathbf{x}_{\text{SC}}^*[t] = \mathbf{x}_{\text{SC}}^*[\bar{t}_0, \bar{t}_f]$. The discrete formulations of the FB (\overline{FB}) and the RV problem (\overline{RV}) are given below:

¹If the target state is a celestial body, this constraint cannot be kept up but must be altered to $r_{\text{min}} \leq |\mathbf{r}_{\text{SC}}(t_f) - \mathbf{r}_{\text{T}}(t_f)| \leq r_{\text{max}}$, due to the target body's dimensions. The distances r_{min} and r_{max} stand for the boundaries within which the respective constraint is achieved.

Discrete RV problem from the perspective of optimal control theory:

Find a spacecraft control vector history $\mathbf{u}[\bar{t}]$, with $\bar{t} \in [\bar{t}_0, \dots, \bar{t}_{f-1}]$, which forces the state $\mathbf{x}_{\text{SC}}(t) = (\mathbf{r}_{\text{SC}}(t)^\top, \dot{\mathbf{r}}_{\text{SC}}(t)^\top)^\top$ of the spacecraft from its initial value $\mathbf{x}_{\text{SC}}(\bar{t}_0)$ to the state $\mathbf{x}_{\text{T}}(\bar{t})$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\text{SC}}(t) = \mathbf{G}(\mathbf{x}_{\text{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{x}_{\text{SC}}(\bar{t}_f) = \mathbf{x}_{\text{T}}(\bar{t}_f)$, and at the same time minimizes a specific cost function J for that transfer.

Discrete FB problem from the perspective of optimal control theory:

Find a spacecraft control vector history $\mathbf{u}[\bar{t}]$, with $\bar{t} \in [\bar{t}_0, \dots, \bar{t}_{f-1}]$, which forces the position $\mathbf{r}_{\text{SC}}(t)$ of the spacecraft from its initial value $\mathbf{r}_{\text{SC}}(\bar{t}_0)$ to the position $\mathbf{r}_{\text{T}}(\bar{t})$ of the target, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{\text{SC}}(t) = \mathbf{G}(\mathbf{x}_{\text{SC}}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{r}_{\text{SC}}(\bar{t}_f) = \mathbf{r}_{\text{T}}(\bar{t}_f)$, and at the same time minimizes a specific cost function J for that transfer.

The definitions of the cost function remain unchanged.

3.2.3 Single-phase Versus Multiphase

The distinction between a single-phase trajectory and a multiphase is not as straightforward as it seems. One is tempted to say that any trajectory leading to a single target object or state is of single-phase type. However, such a transfer can be divided into further subphases. Consider, for example, an heliocentric solar sail transfer beginning at the Earth and ending in a circular heliocentric orbit of 0.48 AU radius and an inclination of 75 deg. Due to $a_{\text{SRP}} \sim 1/r^2$, Sauer suggested to subdivide the transfer into two phases for better optimization [79]: a phase of spiraling into the inner solar system to a circular 0.48 AU heliocentric orbit and an orbit cranking phase to the target inclination of 75 deg. A subdivision into three phases is also possible, as shown by Dachwald and the author in [14]. In the third phase the spacecraft thereby spirals out to 0.48 AU again, because, to facilitate faster orbit cranking during the second phase, it got in closer to the Sun than 0.48 AU in the first phase. The results obtained by Dachwald and the author were faster than the two-phase scenario but still inferior to the single-phase global optimization results obtained with InTrance [14].

Despite the possibility to compose a single-target trajectory of multiple phases, the number of targets remains a primary criterion for multiphase transfers. For the scope of this work, a single-phase transfer therefore has only a single target that is met on a FB or a RB trajectory without gravity assists. Additionally, the phase must take place within one influence region or SOI, i.e., the central body does not change throughout the entire transfer.

If a trajectory leads to more than one target body, or if the flightpath crosses a SOI-boundary, it is called a multiphase trajectory. Examples of such trajectories are the ones of the Voyager deep space probes or the Apollo spacecraft. While Voyager's trajectories have intermediate targets for gravity assists¹, the Apollo capsules crossed the Moon's SOI on their way.

Today's low-thrust missions are mostly of multiphase type. This is either to gain additional ΔV at a GA-body, or because the mission's objective is to visit more than one body, or the flight path leads across an SOI-boundary. This is shown with two examples from Tab. 3.1. NASA's low-thrust mission Dawn, takes place in interplanetary space and comprises a gravity assist at Mars, two asteroid FBs, and two RVs at the primary targets asteroids Vesta and Ceres. This results in five phases and each phase having one single target. ESA's low-thrust Moon orbiter mission SMART-1 serves as example for a SOI-border-crossing mission without heliocentric phases. Its trajectory leads from an Earth-bound orbit into a Moon-bound

¹Effectively, they also cross the respective GA-body's SOI-boundary twice: first, when they approach the GA-body, and, second, when they leave it again on a hyperbolic trajectory

Table 3.1: Examples of high-thrust and low-thrust transfers of single-phase and multiphase nature. It gives for each mission the respective mission phases and corresponding targets. Gravity assists are part of most missions, namely at Venus (VGA), Earth (EGA), and Mars (MGA). The objectives of the chosen interplanetary missions are often flybys or rendezvous at asteroids (AFB, ARV) or comets (CFB,CRV), but also a capture in the Saturnian system (SCPT). Although considered of interplanetary type, the missions Apollo and SMART-1 comprise of planetary phases when having reached the Moon’s sphere-of-influence (SOI) (LSOI) or attained a Moon-bound orbit (LO).

Phase	High-thrust Apollo	High-thrust CASSINI	High-thrust Rosetta	Low-thrust DS 1	Low-thrust SMART-1	Low-thrust Dawn
1	LSOI	VGA	EGA	AFB	LSOI	MGA
2	LO	VGA	MGA	CFB	LO	AFB
3	n/a	EGA	EGA	n/a	n/a	AFB
4	n/a	SCPT	AFB	n/a	n/a	ARV
5	n/a	n/a	EGA	n/a	n/a	ARV
6	n/a	n/a	AFB	n/a	n/a	n/a
7	n/a	n/a	CRV	n/a	n/a	n/a

orbit and thereby crossed the Moon’s SOI. While the first part of the trajectory takes place under the dominating gravitational influence of the Earth, with the Moon’s and the Sun’s gravitation acting as disturbances, the scene changes within the Moon’s SOI. Then, Luna’s gravitation becomes dominant, and turns the Sun and the Earth into disturbing bodies. The entire transfer from Earth to the Moon can therefore be divided into the two phases of a capture within the Moon’s SOI and a second phase during which the spacecraft attains the final orbit about the Moon.

3.2.3.1 The Patched-conics Approximation

The problem of solving multiphase trajectory problems emerged as soon as progress in propulsion technology enabled spaceflight beyond Earth-bound orbits. The trajectories of high-thrust missions are primarily ballistic free-flight arcs, and, assuming no disturbance acceleration, each arc corresponds to an analytically solvable two-body problem. This is exploited within the Patched-Conic Approximation or Patched Two-Body Approximation, as shown by Bate et al. [5]. In its simplest form, it transforms the n -body problem into several two-body problems, whose resulting trajectories are then “patched together” at the respective SOI-boundaries according to trajectory optimization expert knowledge. The method is applicable for preliminary mission analysis and gives solutions for heliocentric missions as well as for transfer trajectories to the Moon.

However, what is a viable way for high-thrust trajectories, does not transfer easily to low-thrust trajectories. This is evident when recalling the fundamental difference between both propulsion technologies: the very low but long-lasting thrust, and the fact that ballistic, two-body flight arcs have analytical solutions and low-thrust ones generally have not. Furthermore, third-body disturbance accelerations are not negligible for multiphase low-thrust trajectories in the vicinity of an SOI border. However, the underlying principle of subdividing a mission into its subphases, finding solutions for them independently, and patching the resulting trajectories together will be picked up later and derived further in the context of global optimization.

3.2.3.2 Multiphase Transfers as Single-phase Optimization Problem

In the context of multiphase trajectory optimization, it is worth to ask why to distinguish between single-phase and multiphase transfers at all. Why not assume a single-phase transfer with intermediate and final targets, constraints, and central bodies that change along the trajectory? The big advantage of this approach is the physical correctness of every solution at every stage of the optimization, whether it fulfills the mission’s constraints or not. Furthermore, there is no need to “enforce” continuous state vectors at eventual phase boundaries because there are no such phase boundaries.

This approach however complicates optimization due to the following:

- **Steering strategy.** The steering strategy can differ significantly from one mission phase to another. An example would be a low-thrust mission that comprises an escape from an Earth-bound orbit into an heliocentric orbit, a consequent rendezvous transfer to the SOI of Mars, and a final transfer onto a low-altitude, polar orbit about a Martian moon.
- **Third-body accelerations.** The determination of when to include the third-body disturbance acceleration of which celestial body is not trivial and hard to determine before trajectory integration. Including all of them throughout the entire transfer would be a complete but inefficient alternative.
- **Problem description.** Each phase of a mission may be different from the others, and so may be the optimal description of each one. The inclination of the osculating orbit of a spacecraft on an interplanetary flight, for example, often changes only little throughout the transfer. Therefore the respective state vector and state equations could utilize orbital elements or polar coordinates, together with the associate rates of change. For planetary problems, however, cartesian coordinate formulations may be more appropriate.
- **Discretization.** Section 3.2.2.2 showed the time-discrete formulation of the low-thrust optimization problem. When “forcing” a multiphase low-thrust problem into a single-phase problem, one is inevitably confronted with the challenge of finding a suited time discretization. The necessary step size ranges from minutes, if the spacecraft is near the pericenter of a highly-elliptic orbit around the Earth, to several days for the same spacecraft during the following heliocentric flight phase. Proper a priori choice of suited step sizes is thus difficult if not impossible.¹
- **Parameter Search Space.** Assuming a three-dimensional control $\mathbf{u}[t]$, the search space dimension scales with the number of steps that is necessary for problem discretization. In fact, the search space dimension of low-thrust optimization problems can be tens of thousands.
- **Optimization.** Finding an optimal trajectory for a low-thrust transfer that requires a “complicated” steering strategy is already difficult. The optimization of multiphase transfers of this type is even more challenging and can require a combination of strategies of different complexity. Treating the entire problem as one phase, requires an optimization method, which shall be equally well suited for various low-thrust problems.

¹A dynamic determination of the control step size, which the is time after which a new control variable value is obtained through the application of a steering strategy, is part of problem modeling. Step size control for keeping integration accuracy is however task of appropriate integration algorithms.

- **Performance.** This topic is related to the time discretization of the transfer problem. Multiphase low-thrust problems with heliocentric and planetary phases often require many control steps. This is especially true for fixed-size control steps, because the step size here depends on the smallest step size required for accurate modeling.

The practical problems when trying to cope with the listed eventual problems often leads to the, in principle, suboptimal approach of subdivision into several mission phases.

3.2.3.3 Challenges of Optimizing Multiphase, Low-thrust Trajectories

Many of the problems of subsection 3.2.3.2 are alleviated by treating the multiphase problem as concatenated, inter-dependent problems in the context of an overall optimization. The transfer problem of each phase may be described with a suited set of state variables, only the dominating third-bodies, an optimized discretization scheme, and control variables. The optimization framework would be more flexible in general, as it could also cope easier with phase-dependent propulsion systems or instantaneous mass changes, e.g., caused by dropped-off propulsion stages or surface probes. Finally, the parameter search space of each phase's problem would also reduce significantly.

These improvements go along with challenges. The biggest one for this approach is thereby to assure final physical correctness, i.e., the steady and continues state vector transition between consecutive trajectory phases¹. Therefore and additional to an appropriate subdivision of the transfer problem into several solvable phases, also the putting back together of the phases' solutions to a physically correct and global-optimal result must be achieved. Proper setup of the subproblems can thereby be a challenge that requires experts in orbit mechanics as well as optimal control.

3.3 Low-thrust Steering Using Local Steering Laws

Local steering law (LSL)s are analytical expressions or algorithms that provide the thrust acceleration vector \mathbf{a}_{thr} at the current time t so that one or several orbital elements change with a desired rate. Lagrange's planetary equations in Gauss form is often the foundation of such LSLs, as they express the rate of change of the orbital elements (see appendix B) dependent on an acceleration vector $\mathbf{a} = (a_r, a_t, a_h)^\top$ in orbit frame (\mathcal{O} -frame) components². With the angular momentum per unit mass $h = \sqrt{\mu a(1 - e^2)}$ and the semilatus rectum, or orbit parameter, $p = h^2/\mu$, these equations are [6]:

$$\frac{da}{dt} = \frac{2a^2}{h} \left(ea_r \sin \nu + \frac{p}{r} a_t \right) \quad (3.6)$$

$$\frac{de}{dt} = \frac{1}{h} (pa_r \sin \nu + [(p + r) \cos \nu + re] a_t) \quad (3.7)$$

$$\frac{di}{dt} = \frac{r \cos(\omega + \nu)}{h} a_h \quad (3.8)$$

$$\frac{d\Omega}{dt} = \frac{r \sin(\omega + \nu)}{h \sin i} a_h \quad (3.9)$$

$$\frac{d\omega}{dt} = \frac{1}{eh} [-p \cos \nu a_r + a_t (p + r) \sin \nu] - \frac{r \sin(\omega + \nu) \cos i}{h \sin i} a_h \quad (3.10)$$

¹This does not necessary include the spacecraft mass, which can reduce instantaneously from one phase to another.

²see appendix A.3 for an explanation of the \mathcal{O} -frame

$$\frac{d\nu}{dt} = \frac{h}{r^2} + \frac{1}{eh} [pa_r \cos \nu - (p + r) a_t \sin \nu]. \quad (3.11)$$

Dachwald showed how LSLs can be used to change and adjust the orbital elements during a low-thrust transfers with maximum rate [13] [14] and how to blend pure LSLs into one another to time-efficiently change more than one orbital element at a time.

Petropoulos developed a local feedback steering law for low-thrust trajectory optimization based on a Lyapunov function, termed “proximity quotient” Q [68]. The proximity quotient quantifies the proximity of the current orbit to the target orbit in terms of the respective orbital elements. This control law also employs (3.6) to (3.11) to determine how quickly Q can be reduced. It is a semi-analytical method that uses a user-provided cut-off parameter η_{cut} for the decision on when to thrust and when to introduce a coast phase. This so-called Q -Law has been successfully applied to various many-revolution, low-thrust orbit transfer problems. The application to low-thrust body RV or FB problems is not possible, as it currently does not include phasing and also lacks support for disturbing accelerations.

The first advantage of LSLs is their independence on necessary optimization processes as they are purely mathematical. If they involve control parameters that require separate optimization, like the cut-off value η_{cut} of the Q -law, this advantage however becomes relative. The second advantage is that LSLs are mathematically well understood and therefore can be proven.

Despite these advantages, LSLs are of limited use for the general multiphase, low-thrust problem. For particular mission phases they could however be beneficial. For example, if zero-inclination or zero-eccentricity orbits could be prevented¹, and if only a single orbital element must be altered, then the application of LSLs is useful. The inevitable problem with multiphase trajectories is that it is often not known when and where a mission phase ends and the next one begins. It is also unknown beforehand which order or combination of orbital element changes is optimal. Furthermore, this optimal combination or sequence also depends on the actual optimization goal, e.g., flight time or propellant consumption. As LSLs have no inherent strategy or intelligence to address these problems, they were not used within this work.

3.4 Trajectory Optimization Methods

Optimization techniques and methods exist today for many kinds of engineering problems and thus also for high-thrust and low-thrust trajectory optimization problems. These methods generally divide into local trajectory optimization methods (LTOM) and global trajectory optimization methods (GTOM). Both are explained in the following subsections, with their advantages and disadvantages as well as their application fields.

3.4.1 Local Trajectory Optimization Methods

LTOMs employ mathematically extensive but well-understood, numerical, optimal control methods that are based on the calculus of variations. Further classification is possible in direct methods and indirect methods. While nonlinear programming (NLP) is an example for direct methods, gradient methods like Hill Climbing and neighboring extremal methods are examples of indirect methods. Both were not employed for this work, and the reader is referred to Betts [9] for a comprehensive survey on LTOMs. Common to all LTOMs is that they need an initial guess, which is either an initial control vector history $\mathbf{u}[\bar{t}]$, or, in the case of neighboring extremal methods, the starting adjoint vector of Lagrange multipliers $\lambda(\bar{t}_0)$.

¹This is due to the singularities of (3.9), (3.10), and (3.11)

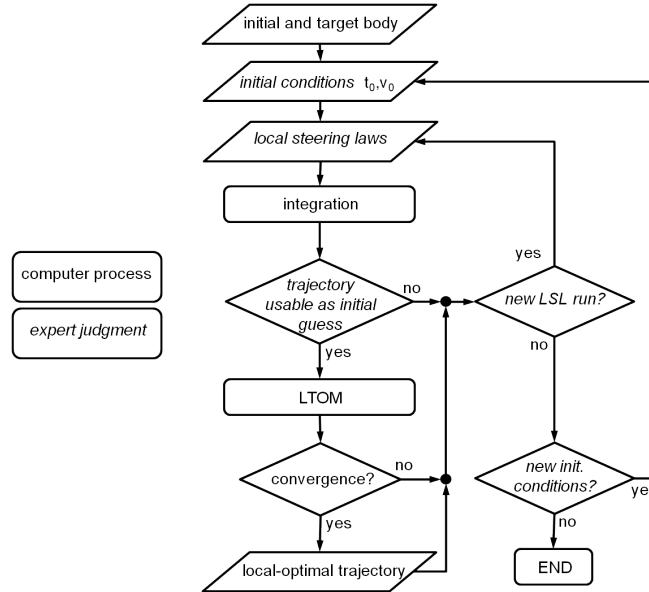


Figure 3.4: Traditional Trajectory Optimization Using Local Optimization Methods. [13] The basic algorithm or procedure of LTOMs starts with the problem setup and defines the initial and terminal conditions as well as the timely discretization. Application of LSLs then gives an initial guess of the control history, which is again input to the LTOM. The result is an improved, local-optimal solution, which is often close to the initial guess.

Starting from there, LTOMs try to find an optimal $\mathbf{u}^* [t]$ in the sense of finding “a better solution than the initial guess” than finding “the best solution”.

Figure 3.4 shows a generic sketch of a LTOM algorithm. Its first step is the setup of the transfer, i.e., initial conditions, arrival conditions, and a timely discretization. This requires expert knowledge in mission design and astrodynamics. A simulation is then carried out during which the application of LSLs changes the orbital elements until the spacecraft’s terminal state is close enough to the target state or body. This is again done under supervision of an expert and finally results in an initial control vector history that can be input to an LTOM. Convergence of the following LTOM application with this initial guess is however not guaranteed because similar initial guesses can produce dissimilar results.¹ The generation of the initial guess must be repeated if the LTOM does not converge. This makes regular user interaction and supervision indispensable. The result of a successful LTOM run is an improved solution, which will however be often close to the initial guess solution. That means, if the initial guess solution was far away from the global optimum, then the LTOM-improved solution will most likely also be far away from the global optimum. An LTOM is by principle not able to cross big distances in the solution parameter space, but that is exactly what is required to move from a far-off input solution towards a globally optimal one.

The advantages of LTOMS are:

- **Proven mathematical foundations.** Being purely mathematical, LTOMs are well understood and deterministic, which gives confidence in their results.
- **Accuracy.** The results of an LTOM are of very high fidelity, which is not surprising as they are mathematical optimization methods. A given transfer is often treated as a boundary value problem (BVP) and therefore the initial and final conditions matched exactly.

¹This is no contradiction to the well understood mathematical foundations of LTOMs.

Despite these advantages, LTOMs also have the following disadvantages:

- **Initial guess dependence.** This is maybe one of the biggest drawbacks of LTOMs when viewed from a low-thrust trajectory optimization perspective. To find a suited initial guess might be still possible for transfers with less complex control strategies. Even then, it requires experts and special tools, like shape-based methods, to obtain them. For complex trajectories that involve changes of all orbital elements it becomes almost impossible to find an initial guess close to the global optimum.
- **Convergence behavior.** If LTOMs converge to a solution, it is often a local optimum close to the initial guess. Iterative variation of the initial guess, however, to improve such a solution, is difficult, as similar inputs to an LTOM can produce very dissimilar results.
- **Necessary expert knowledge.** LTOMs can practically only be used by trajectory optimization experts who know how to generate an initial guess, how to interact with the optimization methods, and how to tune the respective parameters.
- **Requirement for differentials.** Depending on the actual LTOM, the user must provide differential expressions at the discretization nodes. In practice, there may be no analytical expressions for these differentials, which necessitates additional effort to generate them numerically.
- **No optimization of initial conditions.** The initial conditions of a trajectory problem have significant influence on the overall quality of the solution. As the choice of these parameter values is part of the initial guess generation, their optimization is naturally not subject of the respective LTOM.
- **Expert attendance.** As LTOMs practically require permanent supervision and interaction of experts, to find an optimal solution can become a time-consuming task.
- **Discretization.** The time discretization of an LTOM's solution is basically determined with the initial guess. The LTOM can thus not arbitrarily vary the time step size, if it can vary them at all.
- **Computational effort.** Many LTOMs transcribe the trajectory problem into a system of linear equations. The dimension of the resulting system matrix can reach up to many thousands. Although such matrices are sparsely set, the computational effort for their solution in terms of storage and CPU time is substantial. It can be of such extent that it puts a practical limitation on the discretization. This drawback is however relative, given the rapid improvement of computing power.

An LTOM can thus yield high-fidelity solutions whose global optimality depends on the quality of the provided initial guess solution. If this initial solution is close enough to the global optimum, the LTOM can find it with high accuracy.

3.4.2 Global Trajectory Optimization Methods

A variety of global optimization techniques¹ exists today. These methods have been applied to problems in numerous fields, e.g., biology, economy, science, and engineering. Examples of

¹A survey of global optimization methods was carried out at Sandia National Laboratory in 1997 and can be found at <http://www.cs.sandia.gov/opt/survey>

these methods are Dynamic Programming, Branch and Bound algorithms, Differential Evolution (DE), Particle Swarm Optimization (PSO), Genetic Programming, Simulated Annealing, and Multiple Shooting.

Many of these optimization methods have also been applied to low-thrust trajectory optimization. Alemany gives a survey of GTOMs for the application to low-thrust asteroid rendezvous [2]. The number of existing methods indicates that no superior-to-all-others algorithm does exist and each method has its pros and cons.

GTOMs have the following advantages:

- **Global search behavior.** If a global optimum exists for a particular problem, and this problem is adequately coded, a GTOM can potentially find it. That means, the search space in which a GTOM searches for the optimum is not constrained to the vicinity of a predefined local optimum.
- **Independence from initial guess.** This crucial difference to LTOMs eases optimization significantly because their global search behavior often enable GTOMs to start from a randomly initialized parameter set.
- **Optimization of initial conditions.** If the problem is adequately coded, a GTOM cannot only optimize the control vector history $\mathbf{u}[\bar{t}]$ but also the initial conditions, e.g., launch time t_0 , or a launcher-provided hyperbolic excess velocity vector \mathbf{v}_∞ .
- **No expert supervision required.** Without the need for an initial guess and if the problem formulation is suited to the respective global optimization method, this method can run without supervision of experts and interaction with them until the optimal solution is found.
- **Convergence behavior.** This is a relative advantage because it depends on the problem formulation, the problem itself, and on the optimization algorithm's control parameters. Although a global optimization method can principally find the global optimum, it might have problems in finding it quickly. However, GTOMs are generally more robust than LTOMs.

Besides their advantages, GTOMs also have disadvantages, such as:

- **Difficult problem formulation.** While the application of an LTOM, with its necessary permanent expert supervision, can sometime become more art than science, the same holds true for the problem formulation to solve that problem with a GTOM. The way the problem is coded is often crucial for the algorithm's performance in finding the global optimum. Consequently, this is the part for which the expert for the respective problem spends most of the time.
- **Search space dimension.** Compared to LTOMs, the underlying problem to find the optimal control vector history $\mathbf{u}^*[\bar{t}]$ is unchanged and thus also the problem's parameter space is the same. The number of necessary parameters can however be reduced by choosing a different description. This is discussed in detail in Chapter 4.
- **Lower accuracy than LTOMs.** Due to their focus on global parameter space search, the results of GTOMs can lack the accuracy of LTOM-solutions. For preliminary mission analysis this is no disadvantage as high-fidelity solutions are not required at this stage.
- **Nondeterministic.** Unlike LTOMs, GTOMs are often heuristic methods and their behavior is therefore not deterministic. It can require multiple runs of the respective GTOM with the same initial conditions to confirm a previously found optimum.

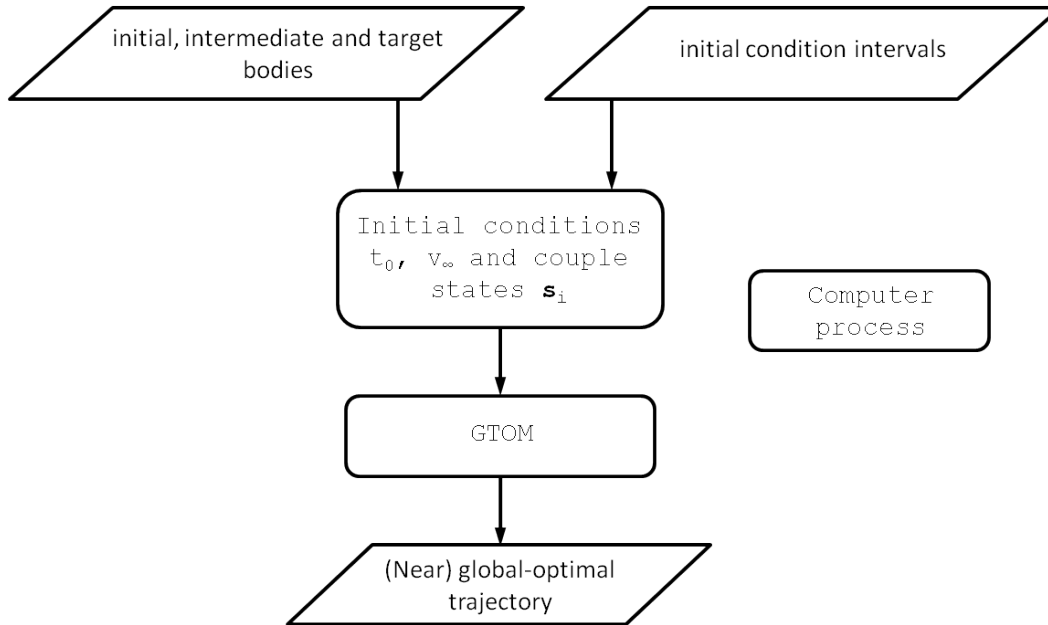


Figure 3.5: Smart Global Optimization Of Multiphase Low-thrust Trajectories. Dachwald sketched the general concept of smart global optimization of low-thrust trajectories [13] and successfully applied the method of ENC to single-phase, interplanetary low-thrust transfers. The concept basically holds true also for multiphase transfers. Instead of one initial body and the respective initial conditions, e.g., the launch time t_0 , a user must now also name eventual intermediate bodies or states, and also the respective intervals of coupling states s_i .

3.5 Smart Multiphase, Low-thrust Trajectory Optimization

The drawbacks of LTOMs can be addressed with GTOMs, which has been successfully demonstrated by Dachwald, who applied ENC for single-phase, low-thrust transfers [13]. This method requires no experts and, after provision of the basic transfer problem parameters, runs autonomously until it has obtained a near-globally optimal solution for the given problem.

One can also solve multiphase low-thrust problems with this method but the solution would most likely not be globally optimal. A mission designer therefore must divide the mission into its phases and optimize each phase individually with a GTOM. The concatenation of the results remains as a manual task, which is often solved iteratively. Such a procedure can by principle not result in globally optimal trajectory as the optimization of transition from one phase to the next is not an explicit component of the optimization method.

Figure 3.5 outlines the functionality of a truly global optimization method for multiphase low-thrust transfers. Such a method must

1. be independent on the knowledge of low-thrust trajectory optimization experts and their attendance throughout the optimization process,
2. not require any input prior to optimization than a description of the mission, i.e., which celestial bodies to meet or which astrodynamical states to assume, and intervals for the initial conditions,
3. automatically optimize the transition of the spacecraft's flight path between adjacent phase boundaries, e.g., SOIs of celestial bodies.

Chapter 4 and Chapter 5 describe the enabling principles, mechanisms, and an implementation of such a methodology.

Multiphase Trajectory Optimization With Evolutionary Neurocontrol

Chapter 3 showed the low-thrust trajectory optimization problem as an optimal control problem in continuous and discrete time. This chapter treats the same problem from the perspective of artificial intelligence (AI) and machine learning (ML). It describes the combined application both concepts to the problem of finding the optimal control function $\mathbf{u}^* [t]$, which leads to the optimal trajectory $\mathbf{r} [t]$. This trajectory $\mathbf{r} [t]$ results from the application of a time independent strategy \mathbf{S} . It maps an input domain, e.g., the state vectors of the spacecraft \mathbf{x}_{SC} and the target body \mathbf{x}_T , to the domain of the spacecraft control variables \mathbf{u}

$$\mathbf{S} : \{\mathbf{x}_{SC}, \mathbf{x}_T\} \mapsto \mathbf{u} \in \mathbb{R}^{n_u}. \quad (4.1)$$

Finding the optimal trajectory thus requires to find the optimal strategy \mathbf{S}^* . Within ENC, a strategy \mathbf{S} is incorporated by an ANN with the corresponding network parameter vector $\boldsymbol{\pi} \in \mathbb{R}^{n_\pi}$, which defines the transfer function $N(\boldsymbol{\pi})$ and therefore the problem solving capability of the ANN. An evolutionary algorithm (EA) then tries to find the optimal parameter set $\boldsymbol{\pi}^*$. This is possible through mapping of candidate parameter sets $\boldsymbol{\pi}_i$ onto corresponding EA chromosomes ξ_i ¹. Each parameter set $\boldsymbol{\pi}$ therefore defines a strategy \mathbf{S}_π . The relations between the optimal chromosome, the optimal trajectory, and the transformations between them are shown in Fig. 4.1.

Multiphase Evolutionary Neurocontrol (MENC) extends ENC by the capability of concurrent optimization of multiple, dependent or independent, heliocentric or non-heliocentric flight phases. If the flight phases depend on each other, then physical correctness inevitably requires the end points and start points of adjacent phases to match in time, spacecraft mass, and astrodynamic state, i.e., position and velocity. Intended and mission-specific exceptions are thereby possible. An example for this is the instantaneous spacecraft mass loss caused by a jettisoned propulsion stage or a dropped surface probe. The transition conditions are thereby constraints that are also subject to optimization. The point at which a spacecraft enters a target planet's SOI, for example, can influence the transfer time or propellant consumption until attaining the final orbit around that planet.

The structure of this chapter is as follows. A general introduction into AI and ML is given in section 4.1 before section 4.2 and section 4.3 explain the details of the concepts that form the method of ENC. This method was invented by Dachwald [13] and successfully

¹The index i denotes a particular vector set of variables; it is not a particular vector element.

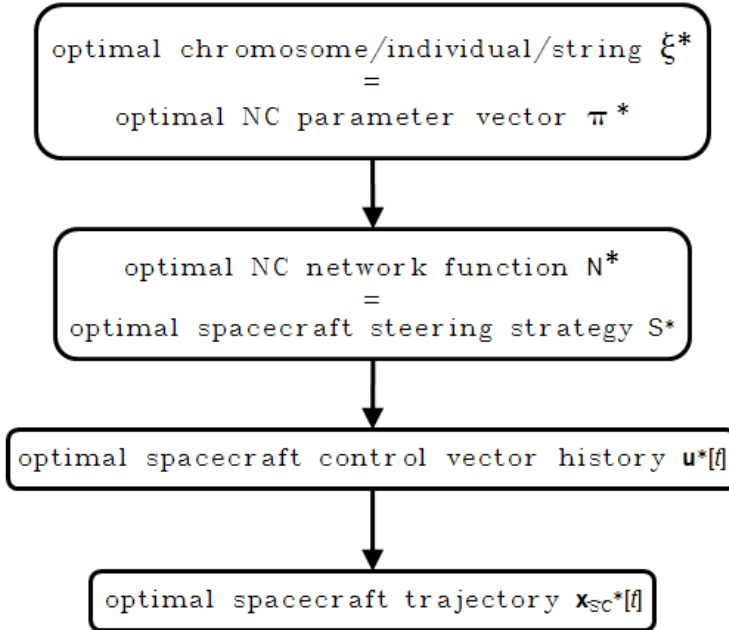


Figure 4.1: From The Optimal Chromosome To The Optimal Trajectory.

used for the optimization of heliocentric, single-phase, low-thrust transfers. A description of ENC is provided in section 4.4, with the focus on the application of ENC on planetary and interplanetary, multiphase low-thrust transfers, for which the method has not been used before.

4.1 Artificial Intelligence and Machine Learning

4.1.1 Artificial Intelligence

As a multidisciplinary research field of computer science, aims at giving machines the capabilities that distinguish man from animals: the ability to think, to learn, to generalize, to solve problems, and to perpetually adapt to a dynamic environment. Shapiro defined AI as follows [82]:

“Artificial Intelligence is a field of science and engineering concerned with the computational understanding of what is commonly called intelligent behavior, and with the creation of artifacts that exhibit such behavior.”

AI is dominated by computer science but also touches philosophy, psychology, and linguistic sciences. It divides into several branches, and, so far, only niche solutions for certain applications are found. The problem of finding general AI, or strong AI remains unsolved. Recreating something as unique and complex as human intelligence is difficult because the involved concepts and mechanisms and their mutual interactions are not yet completely understood. These mechanisms are perception of the environment, reasoning, knowledge, planning, learning, and the manipulation of the environment. A number of today’s applications result from AI research. Examples are chess computers and programs; image, motion, voice, and character recognition systems; car assist systems; or medical expert systems for the support of medical treatments and surgeries. AI systems are thereby superior to humans in some fields, e.g., chess playing. They can however fail easily on other tasks trivial to humans, such as recognizing a familiar person from distance only by seeing the way of walking.

Still being in its infancy, AI research however achieves progress in all of its branches, which touches our daily life in a generally positive but potentially also controversial way. Intelligent and robust humanoid service robots, for example, are most probably welcome in physically demanding and life-threatening positions. When it comes to work force, the threat of intelligent robots replacing work force is existing and discussed controversial. AI research and the resulting applications and products therefore also have a social component besides their philosophical aspect.

AI research divides into the subfields Natural Language, Problem Solving and Search, Knowledge Representation and Reasoning, Learning, Vision, and Robotics. The optimization of low-thrust trajectories thereby primarily falls into the second subfield. The method ENC, which was employed in this work to solve multiphase problems, uses ANNs, which also stem from AI research. ANNs have been successfully applied on a number of classification problems, such as pattern recognition, and to intelligent control problems, such as controlling a robotic element. The EA, which trains the ANNs and is ENC's second essential component, belongs to the AI subfield of Learning.

4.1.2 Machine Learning

It is difficult to formulate a universally accepted definition of ML, as many different learning problems, methods, and systems exist in this AI subfield. Some of them share distinct features or exhibit overlaps, and for this reason a rigorous categorization is virtually impossible. However, common to most of them is according to De Jong [18] their ability to change themselves with the intention to improve their performance on given tasks, whereas that performance is evaluated by the problem environment.

An important learning problem class, which is immanent for this work, is reinforcement learning (RL). The learning system in RL is called agent, and an associative mapping $S : \mathcal{X} \rightarrow \mathcal{A}$ from a situation domain \mathcal{X} onto an action domain \mathcal{A} defines the agent's behavior. That mapping is called policy in RL literature, but within the scope of this work it is called strategy. The reaction of the environment on the agent's actions is called evaluation, reward, or reinforcement. Evaluation results in a scalar fitness value J . The fitness "measures" the quality of the agent's action. An agent thus learns through the immediate or delayed response of the environment to its actions. If the environment responds directly after each of the agent's actions, then this is an immediate reinforcement learning problem. If the response to a number of actions is delayed, and the environment returns only a single J for all actions during a certain time period, then this is a delayed reinforcement problem. Latter problem type is more challenging but many control problems of dynamical systems fall into this category. The learning process uses the reinforcement J to improve the agent's strategy until an optimum is found. The optimal strategy S^* is the one that makes the agent receive more positive reinforcements over time than negative ones. In other words, it maximizes the sum of positive reinforcements and minimizes the sum of negative ones.

4.1.3 Low-thrust Trajectory Optimization From Machine Learning Perspective

This section reformulates the problem of finding optimal low-thrust trajectories from the perspective of ML. Each trajectory $\mathbf{r}[t]$ is in this perspective the result of the application of a steering strategy S . A strategy is thereby an associative mapping from a problem-dependent set of input variables on the spacecraft control vector $\mathbf{u} \in \mathbb{R}^{n_u}$. The quality of S is assessed after the trajectory has been obtained through the application of S . This delayed reaction makes the search for the optimal strategy S^* a delayed reinforcement problem. The RV and FB problems of subsection 3.2.2 can thus be formulated as:

RV problem from the perspective of machine learning:

Find a spacecraft steering strategy S , which forces the spacecraft state vector $\mathbf{x}_{SC}(t) = (\mathbf{r}_{SC}(t)^\top \dot{\mathbf{r}}_{SC}(t)^\top)^\top$ from its initial value $\mathbf{x}_{SC}(t_0)$ to the state $\mathbf{x}_T(t)$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{SC}(t) = \mathbf{G}(\mathbf{x}_{SC}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{x}_{SC}(t_f) = \mathbf{x}_T(t_f)$, and at the same time minimizes a cost function J .

FB problem from the perspective of machine learning:

Find a spacecraft steering strategy S , which forces the spacecraft position vector $\mathbf{r}_{SC}(t)$ from its initial value $\mathbf{r}_{SC}(t_0)$ to the position $\mathbf{r}_T(t)$ of the target body, along a trajectory that obeys the dynamic constraint $\dot{\mathbf{x}}_{SC}(t) = \mathbf{G}(\mathbf{x}_{SC}(t), \mathbf{u}(t))$ and the terminal constraint $\mathbf{r}_{SC}(t_f) = \mathbf{r}_T(t_f)$, and at the same time minimizes a cost function J .

The problems RV and FB thus reduce to the problem of finding an optimal strategy S^* . Due an ANN's suitability to address mapping problems, ENC uses ANNs to incorporate a strategy S .

4.2 Artificial Neural Networks

Natural neural networks are the inspiration for the development of ANNs. It is therefore helpful to know about the composition and function of such biological nervous systems. Subsection 4.2.1 gives a brief introduction. The transfer of the natural information processing principles and mechanisms to ANNs is treated in subsection 4.2.2. Successful applications of ANNs are given in Subsection 4.2.3. ANNs are useless without an appropriate parameter set, which depends on the actual problem and is obtained through a process called training. Training is the content of subsection 4.2.4.

4.2.1 Inspiration by Natural Neural Networks

A biological nervous system is a natural neural network. It consists of nerve cells, or neurons - the anatomic, genetic, trophic, and functional entity and the basic building block of all natural neural networks. Nerve cells are distributed across the whole body, with the brain and the spinal cord being the place where they are concentrated. The human brain, for example, comprises billions of neurons. They are responsible for the transmission of information from receptors cells, e.g., temperature sensitive skin cells, via the spinal cord to the brain and, conversely, from the brain back via the spinal cord to muscles and organs. Depending on the actual task, they divide into sensory neurons, relay neurons, and motor neurons. Table 4.1 gives details of these neuron types, their appearance, location, and distinctive features. Depending on the actual type, neurons also differ in shape, but they share common features. Figure 4.2 gives an illustration. A neuron is connected to other neurons via dendrites, which are extensions leading to the neuron's cell body called soma. Not all details about the neurons' functional principles are known. The general understanding is that a neuron's electrical output depends on the level of exciting and inhibitive inputs. If the neuron excitement exceeds the sum of inhibitive inputs, the neuron "fires" an electrical signal along its output extension. This extension is called axion and can be of varying length. It either connects directly to muscles or organs or splits up into several synaptic endings which connect to dendrites of other neurons via synapses. An axion of one neuron can also connect directly to the soma of another neuron where it acts primarily inhibitive. The degree of interconnectivity and the processing capability of a natural neural network is thus determined by the dendrites and the connectivity at the synapses. Chemical substances called neurotransmitters in the

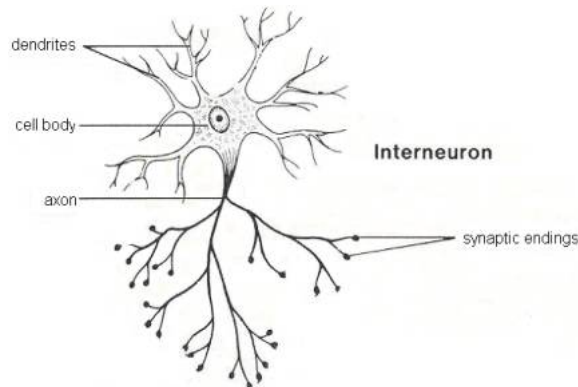


Figure 4.2: Natural Relay Neuron or Interneuron. This type of neuron is found in the brain and the spinal cord. The typical neuron features are the cell body (soma), the extensions leading to the soma (dendrites), and an extension leading away from it (axon). The latter splits up into several synaptic endings which connect to dendrites of other neurons via synapses.

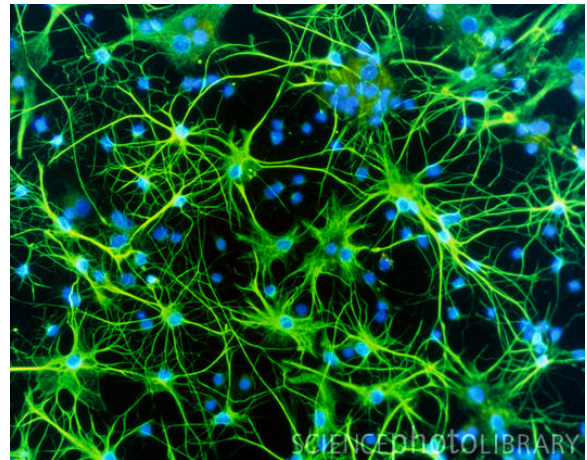


Figure 4.3: Natural Neural Network. Immunofluorescent light micrograph of brain cells from the cortex of a mammalian brain. The nucleus of each cell is stained blue and cytoplasm stained green. Courtesy Science Photo Library.

Table 4.1: Natural neuron types. [47]

	Sensory neuron	Relay neuron	Motor neuron
Dendrite length	long	short	short
Axon length	short	short/long	long
Location	spinal cord	brain	spinal cord
Connects to	relay neuron	relay/motor neuron	organ/muscle

synapses control the conductivity between an axon of a neuron and the dendrites of other neurons that are connected to that particular neuron. Figure 4.3 gives an impression of the degree of interconnectivity in a cluster of brain cells.

4.2.2 Composition and Internal Structure

ANNs are the result of the attempt to exploit the advantageous features of natural nervous systems for technical problems. These features are information storage and processing, fault tolerance, generalization, and decision making. Like its biological prototype, an ANN consists of a number of neurons as simple processing units that exchange information via interneuron connections. That means, a neuron receives information from other neurons, and its output is again input to other neurons. The output is determined by the sum of all input information and the transfer function or activation function, which is inherent to each neuron. The amount by which the output of a neuron i contributes to the total input of a neuron j thereby depends on the weight factor $w_{ij} \in \mathbb{R}$. The neuron transfer function can be relative simple, e.g., a step or linear function, but throughout this thesis the commonly employed sigmoid transfer function is used. Figure 4.4 shows the shape of this function in dependence on a slope parameter or temperature parameter γ . As shown in Fig. 4.5, the output of a neuron i that uses a sigmoid transfer function is calculated from the output y_j of its input neurons j

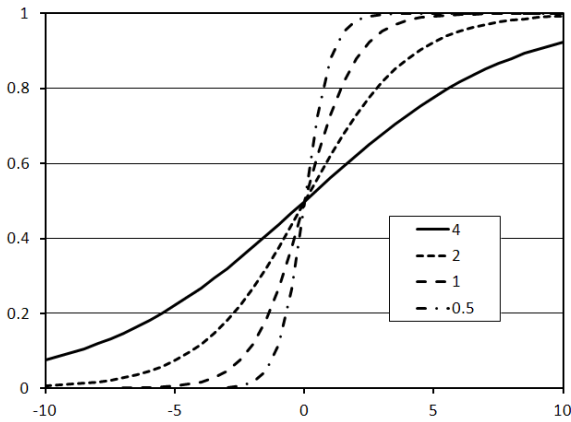


Figure 4.4: Sigmoid Neuron Activation Function. The transfer behavior of this differentiable, continuous activation function depends on the slope or temperature parameter γ . It controls the sigmoid's shape and makes it either resemble a linear function ($\gamma = 4$) or a step function ($\gamma = 0.5$).

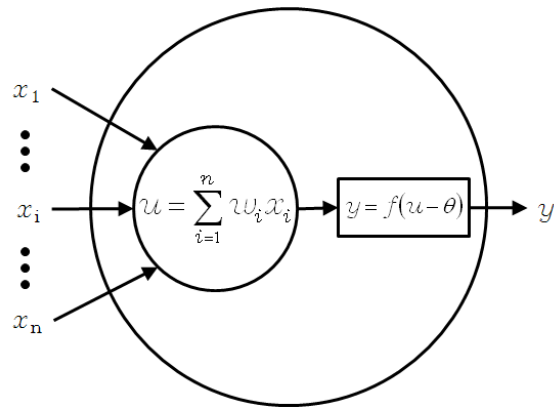


Figure 4.5: Artificial Neuron. The sum of the input variables x_i with the corresponding weight factors w_i is passed through a nonlinear function f , whereby the activation threshold θ is subtracted before. The result is the output variable y . Such a neuron is the atomic building block of an ANN.

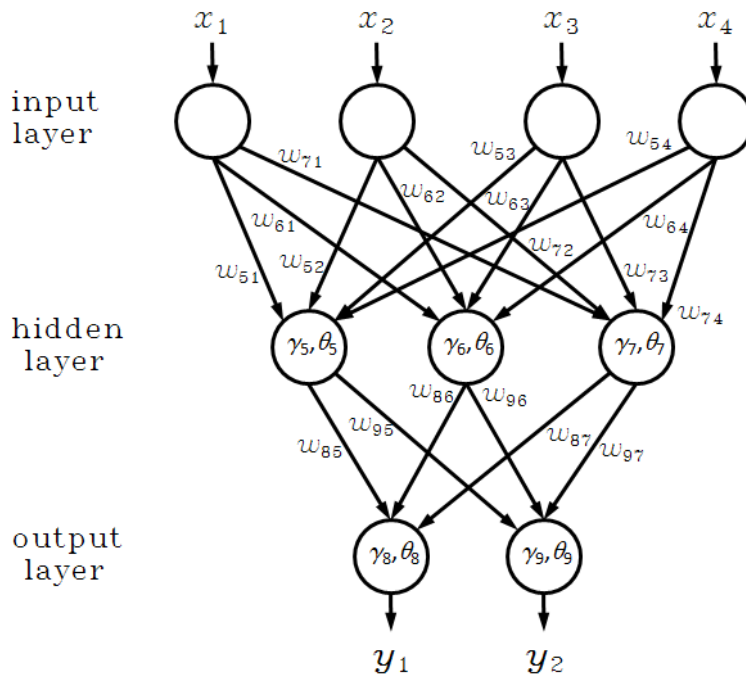


Figure 4.6: Artificial Neural Network. Due to its structure with an input layer, an output layer, and potential hidden layers, a feedforward, layered ANN is also called multilayer perceptron (MLP). It takes the input variables' data at the input neurons, propagates these data via the intermediate layers to the output layer, and provides the transfer function's results y_i at the output neurons. As the input neurons only take the pure input values x_i , no activation threshold θ_i or temperature parameter γ_i is associated to them.

Table 4.2: Natural and artificial neural networks [75].

	ANN	Natural neural network (nervous systems)
Interconnectivity	feedforward	recurrent
Connection type	fully-connected	mostly local connections
Structure	uniform structure	functional modules
No. of node types	a few node types	hundreds of node types
No. of nodes	10-1000	human brain: $O(10^{11})$, $O(10^{15})$ synapses

via

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

with the connection weights w_{ij} , the threshold or bias θ_i , and the temperature parameter γ_i . An advantage of using the sigmoid is its ability to mimic the step function and linear functions through adjustment of the temperature parameter. The sigmoid is thereby always steady and limits its output to a minimal and maximal value.

A single neuron's capability is limited to its transfer function, but complex transfer functions become viable through the interconnection of multiple neurons. The only approximately 100 000 cells of a fly brain, for example, enable it not only to sense its environment, to find food, and to reproduce, but also to manage its challenging flight control system. The key to success is the optimal transfer behavior of the fly's nervous system resulting from a high degree of interconnectivity between the proper neurons.

The transfer behavior of an ANN that uses only sigmoid-type neurons is determined by the parameters w_{ij} , θ_i , and γ_i of all neurons i . They may be arranged to the m -dimensional ANN parameter vector $\boldsymbol{\pi} = (\pi_1, \dots, \pi_k, \dots, \pi_m)$, with π_k being one of the respective ANN parameters (w_{ij} , θ_i , and γ_i). The vector $\boldsymbol{\pi}$ thus completely defines the ANN network function $N_{\boldsymbol{\pi}}$. For this work, only ANNs whose internal neurons are organized in layers were used, i.e., an input layer receiving the problem dependent information (the environment), an output layer returning the network function's result (the control), and eventually existing hidden layers between them. Feedforward connections transport the information between two consecutive layers and thus render data propagation through the ANN deterministic. Figure 4.6 shows this type of layered, feedforward ANN, which has successfully been applied to a number of classification and control problems. Table 4.2 compares ANNs with natural nervous systems.

4.2.3 Application

ANNs are used in real-world applications and neuroscience. Firstly, their ability to infer unknown functions from examples turned out useful. ANNs have therefore found their application where data are either too big to process with other methods, or when it is impossible to model the desired function otherwise. Exemplary problems are regression or function approximation, classification or recognition of pattern and sequences, data processing, e.g., filtering and compression, and the control of dynamic systems or processes. ANNs generally need a rigorous training for the particular problem prior their application, which is addressed in subsection 4.2.4.

Besides practical applications in data modeling and control of dynamical systems, ANNs are also employed in theoretical and computational neuroscience where they model biological neural networks and serve the investigation on natural behavior and understanding of natural neural processes.

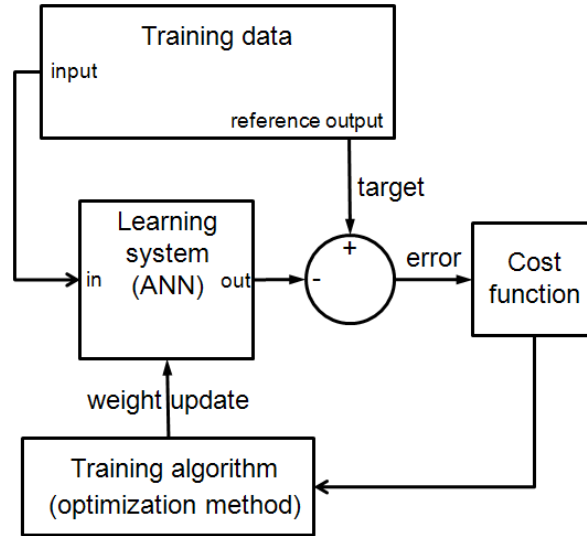


Figure 4.7: Supervised Learning.

4.2.4 Training of Artificial Neural Networks

To train an ANN means to find suited parameter vectors π that enable the ANN to solve the given task. In general, the objective of the training is to make the ANN achieve generalization, i.e., the ANN “understands” the structure of the problem and its solution rather than memorizing the answers of training data. If an ANN, for example, should serve for character recognition in an optical character recognition (OCR) software, it must be trained beforehand for more than only correctly recognizing characters under ideal conditions. It must reliably recognize the letter “A”, regardless of varying optical quality of the presented character, which could result from different font sets, character sizes, fore- and background colors, nonuniform backgrounds, scan image qualities, or rotated characters. Decent training with relevant examples beforehand is therefore decisive for robust behavior during later application.

The determination of the optimal parameter vector π^* is called training and can be done on several ways. A training method is called off-line if the ANN training is finished prior the actual application and the ANN is not further trained later. Training and application therefore happen at different times. Contrary to off-line methods, on-line methods continue with the training process of the ANN when it is already fielded, which allows the ANN to acquire new knowledge about a possibly changed application environment. Some training methods use samples $\mathbf{s} = (\mathbf{x}, \mathbf{t})$, which consist of input vectors \mathbf{x} and the corresponding, correct output vectors or target vectors \mathbf{t} . The difference between \mathbf{t} and the ANN-output vector \mathbf{y} is the error that must be minimized by the training algorithm through modification of π . The sum of square errors (SSE) over all samples n_s is a commonly used error metric [75]

$$E_{\text{SSE}} = \sum_{i=1}^{n_s} (\mathbf{t}_i - \mathbf{y}_i)^2. \quad (4.3)$$

Figure 4.7 shows the elements of a generic, supervised learning algorithm. The first is the preparation of a set of representative training samples, which is often difficult and sometimes impossible if the correct output is not known. This is generally the case for low-thrust trajectory optimization problems. The following application of the ANN to all input vectors \mathbf{x}_i of the training data set and comparison of the resulting output vectors \mathbf{y}_i with the reference output \mathbf{t}_i results in the SSE (4.3). Based on this error metric, the training algorithm modifies

π and initializes the ANN with the resulting, new parameters. This procedure repeats until E_{SSE} , the cost function in this example, is smaller than a predefined limit.

The most widely used training algorithm for MLPs is the Back-Propagation algorithm. It was developed by Rumelhart, Hinton, and Williams [77] and is applicable to any feedforward ANN¹ having neurons with differentiable transfer functions. It uses the chain rule to determine the derivatives of the error function w.r.t. the w_{ij} . The optimization technique of gradient descent then modifies the w_{ij} to reduce the error function value. Back-propagation allows training of ANNs for a wide range of applications and thus renewed the interest in ANNs when it was declining after the first enthusiasm. However, its application requires experience as training success also depends on proper choice of certain control parameters like the learning rate. It also principally fails for problems for which no training data exists or cannot be generated.

4.3 Evolutionary Algorithms

The term EA stands representative within this work for all methodologies, concepts, and algorithms that use the principles and mechanisms of natural evolution to find the optimal solution for a given problem. Such concepts are genetic algorithms (GA), genetic programming (GP), and evolutionary strategies (ES), for which De Jong [19] also proposed the umbrella term evolutionary computing (EC). All of these methods share the same enabling principles but focus on different problem classes. Their common feature is the simulation of the evolution of individual structures through reproduction, inheritance, and selection mechanisms. Selection refers to a candidate solution's adaption to the respective environment and its resulting higher probability to reproduce. Hybrid methods exist as well and combine distinctive features of two or more concepts.

This section describes the elements and mechanisms of EAs and the mutual interaction of these elements with a focus on the application within ENC. A more general introduction into the topic of simulated evolutionary optimization is given by Spears et al. [83], Fogel [28], and Whitley [96], the latter with the focus on the concept GA. De Jong also gives a more recent, comprehensive study of EC in [19]. For a better understanding of how and why EAs are suited for optimization tasks, subsection 4.3.1 provides a brief introduction into the mechanisms of natural evolution. Subsection 4.3.2 then explains how these principles transfer to EAs, whereby the focus is on the EA that was used for this work. Subsection 4.3.3 summarizes the advantages of EAs and also lists their drawbacks, before subsection 4.3.4 shows possible applications of EAs.

4.3.1 Natural Evolution

Natural Evolution is the continuous adaption process of all biological life forms to their respective environment. It is nature's optimization method, and its principles and mechanisms have proven effective over millions of years. The diversity of species on Earth and the manifold of environments in which they survive demonstrate this effectiveness impressively. Although not fully understood in every detail, the basic mechanisms of evolution are known today, and pioneer work on this field was conducted by Darwin [16].

Within this introduction to natural evolution, a single life form is called an individual. A group of individuals of the same type is called population, and the type is called species. Following this nomenclature, a single lion is an individual of the species lion, and a pack

¹The ANN does not need to be of layered structure.

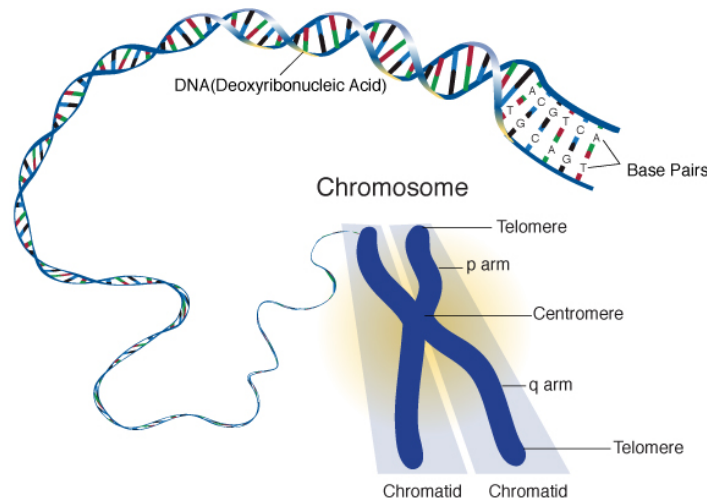


Figure 4.8: Diploid Chromosome and its DNA. Courtesy National Human Genome Research Institute (NHGRI).

of lions is a population. The pack itself is again a subpopulation of the population that comprises all lions on Earth.

Survival of a species is generally assured by the inner drive of its individuals to reproduce, i.e., to mate and to raise offspring. However, not all individuals and not all species have equal chances to do so. “Natural selection” [16] prefers the strong and healthy individual over the weak or unhealthy, and the better adapted species over the less adapted ones. That means, physically disabled or ill individuals are discriminated by mating candidates and also have less chances to hunt or to find food. They most probably die of starvation or get killed by predators before they have a chance to mate. This holds basically true for all species in a changing environment in which they compete for the available resources. Less adapted ones, or species that cannot adapt quickly enough to an environmental change, will become replaced by other, better adapted ones.

The pace of evolution is generally slow, taking thousands of years for significant changes to a species’ features. It can however increase on sudden catastrophic events that change the environment drastically. On one hand, events like this can lead to mass extinction of entire species. On the other hand they can give birth to a manifold of new ones. An example for such a environment change is the rapid climate change on Earth approximately 65 million years ago. It was most probably caused by the impact of a larger asteroid and lead to a mass extinction of the majority of species on Earth at that time. Among them were the dinosaurs, which could not adapt quickly enough to the declining average temperature and the related change in vegetation. Entire food chains collapsed, and other, less specialized species with shorter life cycles suddenly had less competitors for the remaining resources. Additionally, they were less threatened by predators. Especially mammals, which can regulate their body temperature and are thus less dependent on the environment temperature, were not only able to adapt quicker to these environment changes. They also did not have to compete any more with the now extincted dinosaurs when vegetation started to reconquer the planet.

The more beneficial the individuals’ traits and capabilities are for their survival, the higher are the chances of the entire species to survive. Those features, e.g., whether it is a mammal or a reptile, it can fly or not, it is carnivorous or herbivorous and thereby its primary type of food, are determined through its genome. The genome information of all individuals of a population is called the genome pool of the population. The genome is encoded with the genome code in the chromosomes. Located in the cell nucleus of every life form, chromosomes consist

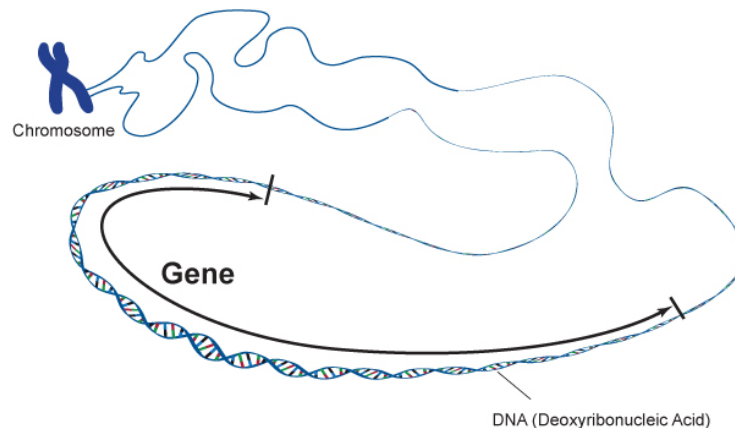


Figure 4.9: Natural Gene on a Chromosome. Courtesy NHGRI.

of two pairs of deoxyribonucleic acid (DNA) molecules that hold the genome information in genes. An exemplary illustration of an X-shape chromosome and of the DNA is shown in Fig. 4.8 and Fig. 4.9. Genes are the smallest units of inheritance and encode an individual's features, like the eye color. They are variable-length sequences of the four bases adenine (A), cytosine (C), guanine (G), and thymine (T). The position of a base pair on the chromosome is called allele. Complementary base pairing assures only A-T and C-G combinations, which is essential during cell division to pass the complete genome information to both daughter cells. The DNA splits before the actual cell division and the resulting half-chromosomes complement themselves again through base pairing. Shortly before cell division, the mother cell therefore holds a complete chromosome copy for each daughter cell.

A particular sequence of bases of a gene is called the genotype of that gene. The individual's observable trait resulting from that genotype is called phenotype. Chromosomes that hold the complete genome information twice are called diploid and haploid if not. The diploid structure becomes vital for evolution as it allows inheritance through the fact that offspring individual chromosomes consist of the genome information of one half of each of their parents' chromosomes. An offspring can therefore potentially benefit from the positive features of its both parents and their ancestors. If the parents have different, beneficial-for-survival features and inherit them to their offspring, these offspring may be even better adapted to their environment through the combination of positive features.

During reproduction, a diploid chromosome's halves split and exchange gene information through the genetic operation of recombination. Recombination breaks up DNA chains and reconnects the resulting fragments to other DNA fragments. The recombination of entire chromosomes is also called crossover. Crossover increases the genetic diversity of a population, which helps to adapt to future environment changes. During the life of an individual its genome also experiences random changes. These are called mutation and they introduce new genotypes to the genome. This results in new phenotypes, with positive or negative effects on an individual's struggle for survival. They can also have no influence at the moment but may become important in later generations. In both cases they increase the population's genetic diversity independently on the current genome pool of a species and therefore allow evolution of that species to proceed into new directions.

4.3.2 Elements and Mechanisms

The principles of natural evolution are the foundations of EAs, which do not simulate every detail of evolution. To find optimal solutions to technical problems, they model only the

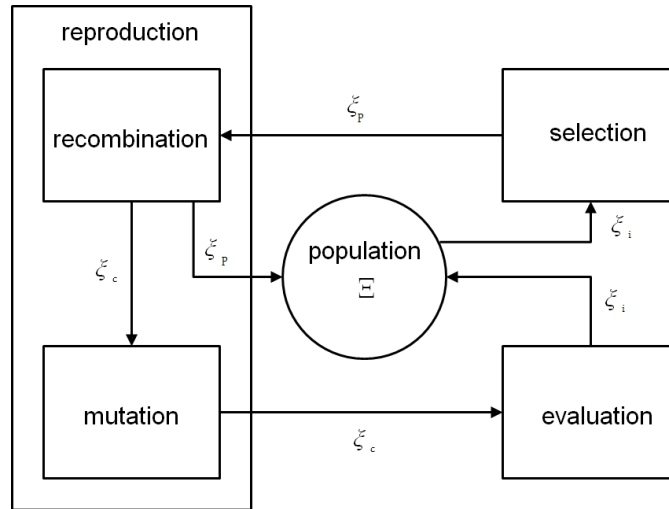


Figure 4.10: Evolutionary Algorithm Elements. Although differing in detail, EAs mostly have four essential components: a population Ξ , which holds the candidate solutions; a mechanism to select the candidates for reproduction; the reproduction itself, consisting of the genetic operators recombination and mutation; and evaluation for assessment of an offspring’s adaption to the environment. The arrows show the “reproduction cycle” of the population individuals ξ_i . Individuals selected for reproduction become parent individuals ξ_p from whose genome information reproduction generates the child individuals ξ_c .

required basic elements and principles of evolution to exploit its robust optimization capability. EAs are problem-independent, heuristic, parallel search methods. Information about the actual problem is encoded within the individuals ξ_i the EA acts upon. It therefore depends on proper encoding of the problem solution parameters on the ξ_i if the particular problem can be solved by the EA. Additional but unnecessary data is thereby less important than missing essential information. Figure 4.10 shows the elements and mechanisms of a generic EA: the population Ξ , a selection mechanism, genetic operators simulating recombination and mutation, and a problem dependent evaluation. The implementation and the importance of the EA elements differ from one EA to another. Within the used ENC implementation, a GA trains an ANN parameter set, and the following description and explanation therefore concentrates on the particular implementation in the optimization software InTrance.

Like natural populations consist of individuals of the same species, the data structure of an EA’s population Ξ holds candidate solutions or individuals ξ_i of the same type, with the population dimension q and $i \in \mathbb{N}, 1 \leq i \leq q$. The same type means that each ξ_i , which is also called a chromosome or a string, is a potential or candidate solution for the same problem. A ξ_i presents a solution candidate to the GA. There is no ideal generic representation type, which makes the choice of representation dependent on the actual problem, and in practice bit strings, real-value vectors, or graphs are used. Each ξ_i corresponds to exactly one candidate solution, and the individual’s scalar fitness value J measures the overall quality of this incorporated solution. The fitness J is a metric for the fulfillment of the actual objective function, e.g., minimum flight time or minimum propellant mass. It however also serves to express violations of additional boundary constraints. The fitness is by convention subject to maximization. Greater J -values are therefore assigned to those individuals that are better adapted to their environment and are consequently better solutions of the respective problem. If necessary, this can be reversed with $-J$.

The fitness $J(\xi_i)$ of an individual ξ_i determines its chances during selection for taking part in reproduction. Selection is the second EA element and decides which ξ_i of Ξ may reproduce

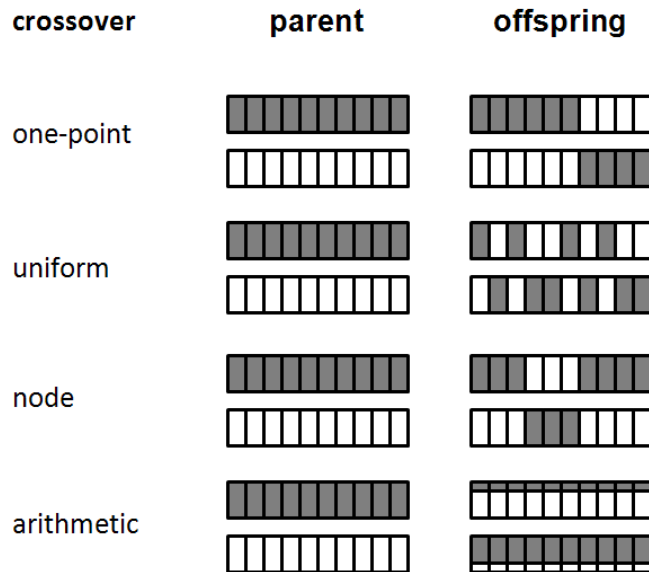


Figure 4.11: Crossover Operators.

and have offspring and which not. There are two major types of selection schemes: deterministic and probabilistic (or stochastic) [19]. Deterministic selection schemes, e.g., truncation selection, allow only the subset of the best population individuals to reproduce and exhibit a strong selection pressure. This is especially true for the case if only the best population individual survives the selection process.

Probabilistic selection methods, e.g., roulette wheel selection and tournament selection, choose reproduction candidates based on a probability that is assigned to each individual. They further divide into uniform selection methods, if all individuals have the same selection probability, and fitness-biased selection methods, if selection probability depends on $J(\xi_i)$. The drawback of these selection methods is however that they do not guarantee the survival of the best population individual when using non-overlapping¹ populations. This is the case even if the best individual's fitness is large compared to the average population fitness J_{avg} . For overlapping populations this is no problem as the best individuals do not get lost between reproduction cycles. Roulette wheel selection, as the name implies, spins a virtual roulette wheel with q slices; each slice thus corresponds to a ξ_i of Ξ . Depending on whether the slices are of equal size or related to the relative fitness of the respective ξ_i , roulette wheel selection is uniform or fitness-biased.

Another fitness-biased selection method is tournament selection and this selection method is implemented in the EA of InTrance. It randomly selects μ individuals ξ_i and takes the one with the highest fitness as winner. The tournament size μ controls the selection pressure. The bigger the tournaments the larger is the probability to select the best individual of the population, which results in strong selection pressure as the other ξ_i have less chances for reproduction. Overlapping populations require at least two tournament rounds per reproduction cycle to select two parent individuals ξ_p . To keep q constant, the ξ_i that were inferior during selection are replaced by the new individuals resulting from reproduction. InTrance holds two tournaments for each reproduction cycle with a tournament size of $\mu = 2$, which is also called binary tournament.

¹Non-overlapping populations generate a complete new population in one step through repeated application of selection and reproduction. The old population is then discarded. After evaluation of all individuals of the new population the next cycle starts.

Following selection, the element of reproduction is crucial for the optimization capability of an EA. It serves two purposes. First, it must create new children ξ_c through recombination of genome information stored in their parent individuals' chromosomes. Genes are passed on this way from parents to offspring and with them also the phenotypes that are potentially beneficial for survival. The parents are put back into the population, ready for the next selection round. Genetic crossover is the next step and Fig. 4.11 shows different crossover types that have evolved in GA-practice: one-point crossover, uniform crossover, and arithmetic crossover. Node crossover is a special type of uniform crossover derived for the training of ANNs with an EA. The crossover loci are chosen specifically for the training of the particular ANN, as it is beneficial for the learning rate to keep information related to a neuron intact.

The second purpose of reproduction is to maintain a high number of genotypes in the population's genome pool, i.e., to assure a high genetic diversity. As selection prefers superior individuals, and without a mean to introduce new genes, genetic diversity would inevitably reduce with each individual that permanently leaves the population. Mutation is the genetic operator that helps to increase genetic diversity again. Executed after recombination, mutation introduces random changes at randomly chosen offspring chromosome loci, which helps to explore the solution space more thoroughly and less dependent on the population's initial genome pool.

The generated offspring proceed to the final EA-element of evaluation. The problem-dependent evaluation exposes the ξ_c to the environment. Its adaption to this environment is expressed in the fitness $J(\xi_c)$. The new individuals ξ_j are then put back to the population where they are available for selection like their parents.

This cycle of selection, recombination, mutation, and evaluation repeats until the population has converged to a single solution, or no improvement could be achieved for a predetermined number of reproduction cycles, or another abort criterion is met.

4.3.3 Advantages and Disadvantages Compared to Other Optimizing Techniques

Compared to other optimization techniques, especially to gradient-based methods, EAs have a number of advantages. The first one is their global search behavior, which allows a more thorough parameter space search than local optimization techniques can achieve. In a multimodal problem environment, EAs are also less prone to premature convergence to local optima than local search techniques. Further advantages of EAs are their blindness and their robustness. Blindness means an EA can improve a solution without additional information, like problem-specific derivatives. This makes them applicable to a wider range of problems than gradient based methods. Back-propagation, for example, can only optimize feedforward ANNs with differentiable neuron activation functions. Robustness means that EAs need no initial guess solution to start an optimization. They can basically find the optimal solution after having started from a randomly initialized parameter set. EAs are therefore suited to solve problems for which initial solutions are difficult or impossible to generate.

Although EAs can solve numerous problems, they also have disadvantages and are not the optimization method of choice for any problem. The inclusion of non-trivial constraints, for example, is difficult and can require experts in the respective field of the particular problem and several attempts to find a suited problem formulation. Another drawback is their inability to find high-fidelity solutions, compared to local optimization techniques. A local optimization of an initial guess solution obtained from a previous EA run is therefore a valid method leading to global-optimal solutions of high accuracy. EAs are computational expensive, as they require the evaluation of many intermediate solutions to find the global optimum. The number of evaluations and the robustness of the EA for the respective problem thereby

strongly correlates to the problem's transcription into a chromosome. An advantageous mapping of the problem with all its necessary variables may enable an EA to find the global optimum rather quickly. A problem formulation that is poorly suited for the optimization with an EA, or a formulation missing a crucial problem variable, can not only increase the runtime of an EA considerably. It can moreover reduce the chances of finding the global optimum at all.

4.3.4 Applications of Evolutionary Algorithms

GAs and GP methods have been successfully applied to various optimization problems in engineering, management, or economics. The following examples give a brief impression of the range of possible EA-applications.

The first example is the support of air traffic control personnel by an EA-based software tool called ROGENA (free ROuting with GENTic Algorithms) [33] [34], which helps to optimize airspace usage. Under obeying the safety constraints and separation distance margins, this tool optimizes the airplanes' entry and exit points of the respective airspace control box in order to maximize the number of air traffic per hour in that control box. It is a typical example of the important problem class of scheduling and routing, for which EAs have been applied successfully.

Design optimization in engineering applications, especially where shape influences functionality, is another application field of EAs. This includes 2-dimensional and 3-dimensional shapes in basically all engineering branches. An example in which a small efficiency gain of one percent can have considerable effect on complexity, and thus on production and operating cost, is the shape of a compressor blade of stationary and non-stationary turbo machinery. EAs have been applied in conjunction with computational fluid dynamics (CFD) software tools to optimize blade shapes for higher efficiencies through higher pressure ratios per blade [12]. The objective is thereby the development of turbo machinery having fewer blades per compressor stage, or maybe even fewer compressor stages. Both yields in the desired higher fuel efficiency and cost saving.

GAs were also applied to the optimization of control systems [49] or industry processes synthesis [90] and for the design and optimization of controllers for animated motion, often in conjunction with ANNs. Genetic optimization methods proved also successful in game theory, where they optimize the behavior and strategy of agents, for example. EAs have been used also for artificial life simulations and for the development of machine learning rule-based classifier systems.

Finally, EAs are employed in evolutionary biology research, where they help to model natural systems and serve the understanding of natural evolutionary processes.

4.4 Multiphase Evolutionary Neurocontrol for Trajectory Optimization

The optimization method ENC combines the concepts of ANNs and EAs. MENC is the extension of ENC on mission designs that consist of more than one transfer phase, which must not necessarily be of low-thrust type. A mission optimization criterion therefore affects multiple flight legs.

This section describes the functional principles of ENC and MENC. It comprises the use of an ANN for spacecraft steering and of an EA for the optimization of the ANN parameter set and initial conditions. With this method, ENC and its extension MENC enable the optimization

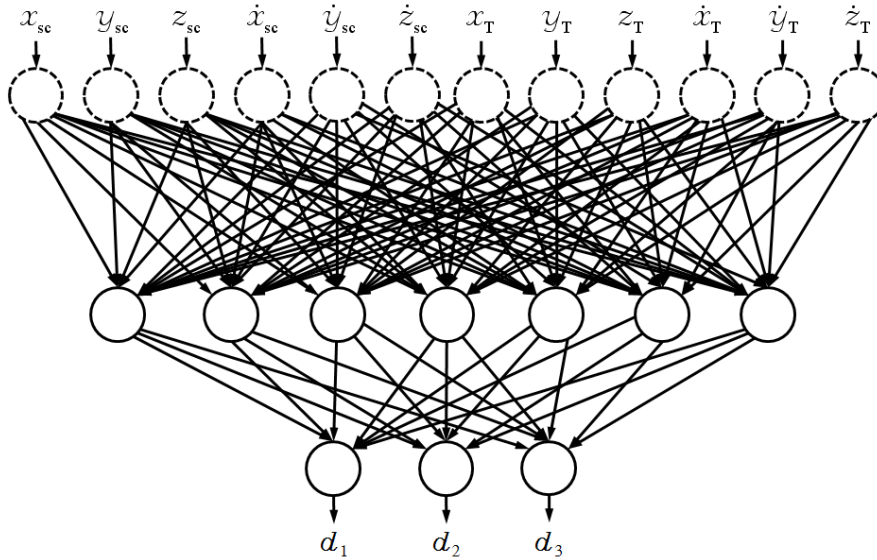


Figure 4.12: Neurocontroller For Spacecraft Steering.

of single-phase and multiphase mission design also for non-experts in astrodynamics and optimization.

First, subsection 4.4.1 introduces the neurocontroller concept, followed by the description of neurocontroller training in subsection 4.4.2. If each mission phase employs a separate neurocontroller for spacecraft steering, training of these neurocontrollers must take place concurrently, as the flight phases affect each other. How this can be done, is explained in subsection 4.4.3. Subdividing a mission into separate phases, and optimizing those phases concurrently, raises the requirement on appropriate transition conditions that must be satisfied to yield physically valid solutions. This and the optimization of such phase transitions w.r.t. to the overall optimization criterion is subject of subsection 4.4.4.

4.4.1 Spacecraft Steering With Neurocontrollers

A neurocontroller (NC) is an ANN that is used for control of a dynamic system. Within this work, the low-thrust-propelled spacecraft under the influence of its environment and internal forces is the dynamic system that must be controlled optimally. Its thrust acceleration vector must point in the local-optimal direction at any time in order to follow the desired trajectory. The determination of that direction requires a strategy. Within ENC, that strategy is incorporated by an NC. The NC commands direction and magnitude of the local-optimal thrust vector through mapping of problem specific information from its input neurons via its transfer function to its output neurons. An important problem is hence which input data to use, how to present these data to the NC, and how to decode an NC's output to obtain the needed control variables.

An optimal input data set contains only the variables that are necessary for optimal control of the dynamic system. Additional variables would not further improve control behavior. Instead, leaving out a variable of that optimal set would decrease the NC's capabilities or even disable them. The question of the transfer-type-dependent optimal set of input parameters was not pursued during this work, as many parameter sets lead to the desired result as long as they contain the necessary variables.

A natural assumption for the input data of a spacecraft-steering NC are the state vectors of the spacecraft \mathbf{x}_{sc} and the target \mathbf{x}_T or the relative state $(\mathbf{x}_{sc} - \mathbf{x}_T)$. However, to make the

NC's steering strategy independent on time and therefore more robust, the current time t should not be used. The usage of m_p as input for an NC that controls a propellant dependent spacecraft¹, on the other hand, could provide implicit information about remaining thrust time. As \mathbf{x}_{SC} and \mathbf{x}_T may be expressed in cartesian or polar coordinates of rotating or inertial frames, there is a great number of possible input data combinations.

The choice of input data also affects the training progress of an ANN parameter set $\boldsymbol{\pi}$, because input parameter values can exhibit considerably differing absolute magnitudes. The ratio of the Earth's mean distance to the Sun ($1.5 \cdot 10^8$ km) and its mean orbital velocity (28 km/s), for example, is approximately $5.4 \cdot 10^6$ seconds. An ANN can be trained to handle such input data differences, but it is time consuming and may also require additional ANN nodes.

Another aspect of very large absolute parameter values becomes evident from Fig. 4.4. A transfer function, like the depicted sigmoid, can have an "operational range". Above and below this range, changes to the input have no significant effect on the function's output value, which influences the training method that is employed for the optimization of $\boldsymbol{\pi}$. If input parameter changes do not result in noticeable changes of the output values, training progress will be slow, or not converge at all. Normalizing the ANN's input vector data can improve the optimization process and, through less required ANN nodes, reduce the number of optimization parameters.

Another problem arises when using angular input parameters, e.g., the polar azimuth angle φ . Close to 0 or π and depending on the actual representation, small changes of the actual angle can result in parameter value changes close to 2π . ANNs can adapt to this peculiarity, but it often requires additional internal nodes, or longer training, or both. The transformation of angular parameters prior their usage as ANN input is therefore recommended.

The transfer function maps this input on the ANN output neurons from where it is transcribed to the control vector \mathbf{u} . The control is the local-optimal thrust vector expressed in the orbit frame \mathcal{O} (see appendix A.3). For the duration of a control step, i.e., until the next NC call, this vector is constant in that frame. It consequently changes in the inertial frame according to the spacecraft's orbital motion. If the ANN output gives the thrust vector directly, it is called a direct steering strategy and an indirect steering strategy if another process, e.g., an LSL, is interlinked. For the calculations within this work only direct steering strategies were used. The chapter covering the implementation of MENC details further on the generation of the ANN input parameters and how the control variables deduce from the NC output.

4.4.2 Optimization of Neurocontrollers Through Training

Assuming the same transfer function for each neuron, the number of layers and the number of neurons in each layer defines the topology of the NC's ANN. Topology and the respective parameter set $\boldsymbol{\pi}$ thus define the overall network transfer function $\mathbf{N} = \mathbf{f}(\boldsymbol{\pi}) = \mathbf{N}_{\boldsymbol{\pi}}$. If topology is fixed, i.e., not part of the optimization, the optimal parameter set $\boldsymbol{\pi}^*$ determines the optimal transfer function $\mathbf{N}^* = \mathbf{N}_{\boldsymbol{\pi}^*}$, whose application results in the optimal trajectory $\mathbf{r}^*[\bar{t}]$.

ENC employs a GA to train the spacecraft-steering NC through optimization of its parameters $\boldsymbol{\pi}$, which requires a mapping from $\boldsymbol{\pi}$ to the GA chromosomes. The example depicted in Fig. 4.13 shows for a small ANN how the respective parameters of all but the input neurons sequentially arrange on a preferably real-value chromosome ξ . Figure 4.14 shows the functional principles of ENC with its two main loops: an outer GA loop, and an embedded trajectory integration loop. The latter incorporates the GA's evaluation element and complements "newborn" individuals ξ_i with the fitness $J(\xi_i)$.

¹This is not necessary for solar sails.

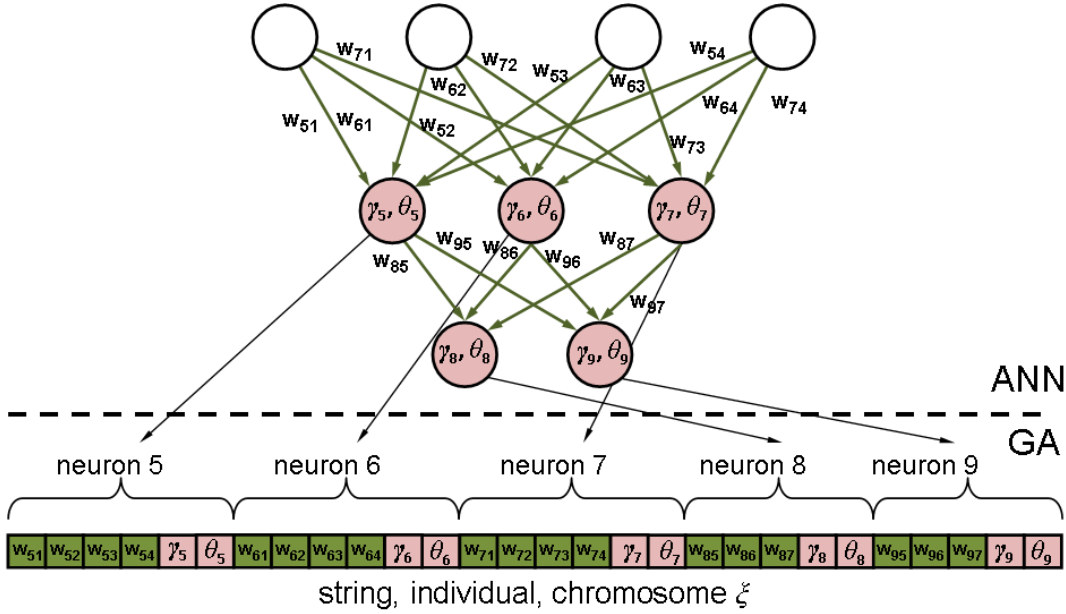


Figure 4.13: Mapping of the NC Parameters on a Chromosome. Optimizing an NC means to find the optimal values of its ANN weight factors w_{ij} , its input bias or thresholds θ_i , and its neuron transfer function parameters γ_i . Mapping these parameter on a string or chromosome ξ allows the application of a GA for their optimization.

The inner loop starts with the initialization of the NC, using the information encoded on ξ_i . In addition to π , ξ_i also contains the simulation's initial conditions. These are the spacecraft's initial state $\mathbf{x}_{SC,0}$, which may depend on the launch date t_0 , and, for spacecraft with propellant-dependent propulsion, on the initial propellant mass $m_{p,0}$. Both are therefore also subject of the optimization. The simulation is the numerical integration of a differential equation system (DES) of the state equations, also called equations of motion (EOM). Each integration step from \bar{t}_i to $\bar{t}_{i+1} = \bar{t}_i + h$, with the control step size¹ h , precedes an NC application. The NC provides the control $\mathbf{u}(\bar{t}_i)$, based on \mathbf{N}_π and the input data $\mathbf{x}(\bar{t}_i)$, which comprises at least the astrodynamics states of the spacecraft $\mathbf{x}_{SC}(\bar{t}_i)$ and the target $\mathbf{x}_T(\bar{t}_i)$. This simulation model neglects realistic rotation dynamics and allows instantaneous attitude alteration at each control step. The control vector² $\mathbf{u}(\bar{t}_i)$ is constant in \mathcal{O} -frame during h but changes in the \mathcal{J} -frame according to the relative motion of \mathcal{O} w.r.t. \mathcal{J} . Numerical integration of the EOMs is the next step, followed by a check that decides about whether to continue integration or not. If the trajectory integration should continue, the new $\mathbf{u}(\bar{t}_{i+1})$ of the next step is obtained through another NC application.

Trajectory integration stops upon fulfillment of the accuracy constraints, or violation of boundary constraints, or after a preset maximum integration time. Based on the resulting $\mathbf{r}[\bar{t}]$, an assessment w.r.t. the optimization criterion gives the individual's fitness $J(\xi_i)$. The evaluated individual is passed on to the calling outer loop, where it is put back to the population. This process repeats until the outer loop achieves population convergence, i.e., no ξ_i with a better trajectory and resulting higher fitness could be generated. If the problem was properly coded and all the parameters relevant for the problem were chosen appropriately,

¹The control step size not necessarily equals the integration step size. Most integrators automatically adapt the integration step size to meet predefined accuracy limits.

²As the optimal control in this case is the direction and magnitude of the thrust direction vector, this vector is three-dimensional.

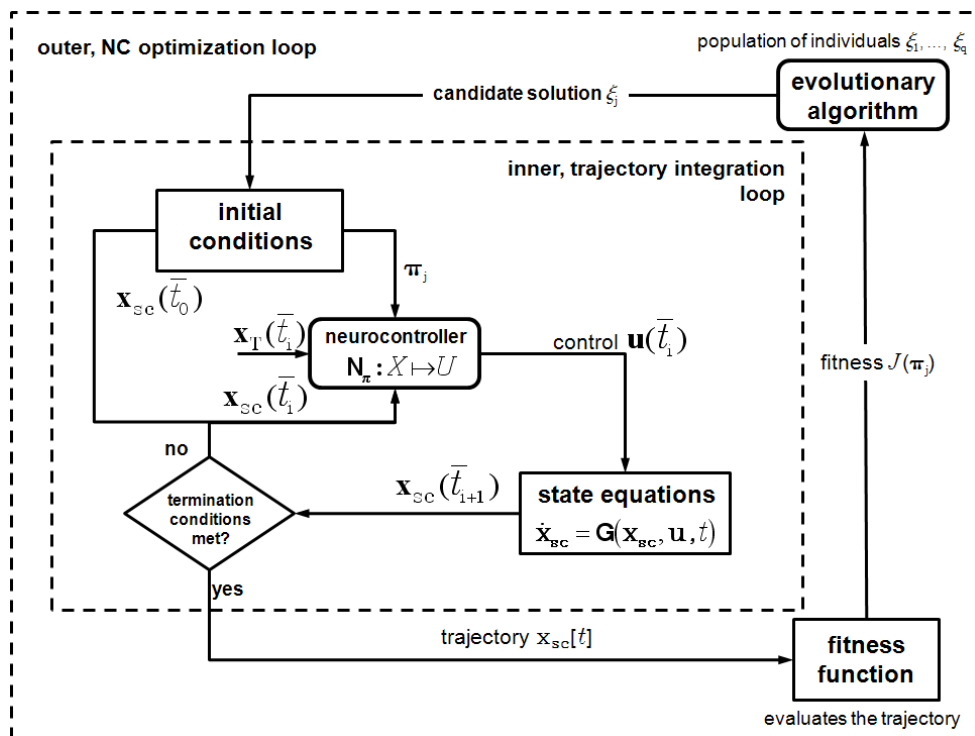


Figure 4.14: Trajectory Optimization Through Optimization of the Initial Conditions and the Spacecraft-Steering NC. This is an extended version of the figure from Dachwald [13]. It shows the solution ξ_j holding not only the NC's parameter set π_j but also the trajectory integration initial conditions.

e.g., initial propellant mass and launch windows, the best population individual ξ^* carries the globally-optimal solution within its genome.

4.4.3 Optimization of Multiphase Missions at the Example of an Earth-Mars-Earth Double Rendezvous

The optimization of space missions with more than one transfer phase exhibits in general similar but also new problems, regardless of the employed propulsion technology. This is because the mutually dependent but separate mission phases of such multiphase missions affect each other and their optimization. Objective functions that assess the entire mission, e.g., minimum total mission duration, are therefore indirect constraints for the optimization of each phase. The terms “mission phase” or “flight leg” are used here to describe certain, mission -specific elements of a logical breakdown and can be used interchangeably.

A SEP-powered Earth-Mars-Earth double RV mission shown in Fig. 4.15 serves in the following for the explanation of potential problems and constraints that may hamper the optimization of such mission designs. The strategies to alleviate those problems and the chosen approach for multiphase mission optimization with ENC are exemplified in this scenario as well. It was chosen for the several reasons. Firstly, a two-phase mission design is a logical next step from a single-phase mission. Secondly, low-thrust sample-return missions from Mars are of scientific interest and study results can be found in literature [20]. Thirdly, both low-thrust transfers from Earth to Mars and from Mars to Earth are of limited complexity in terms of trajectory optimization. This would not be the case for a low-thrust Earth-Moon-Earth mission design, as the Moon's gravitation acting as disturbance acceleration cannot be neglected

for low-thrust transfer optimization. Finally, this example allows an easy explanation of the concepts of dwell time, flight time, and total mission duration, as well as the dependencies between them.

A spacecraft with the dry mass $m_d = 624$ kg, a propulsion subsystem comprising two NSTAR engines with variable throttling, and a solarelectric power generation subsystem with $P_{e,SP,0} = 10$ kW and $\kappa = 1.6$ should first traverse on a RV-trajectory to Mars. The scientific operations there are accounted for with a stay time or dwell time

$$\Delta t_{dw} = \sum_{i=2}^{p_f} {}^{(i)}\Delta t_{dw} = {}^{(i)}t_0 - {}^{(i-1)}t_f = {}^{(2)}\Delta t_{dw} \quad (4.4)$$

in [100, 150] d before the second flight leg leads the spacecraft back to the Earth. The symbol i in parenthesis and upper left of the respective symbol denotes the flight phase $1 < i \leq p_f \in \mathbb{N} : p_f > 1$ of that variable. For the two-phase example $p_f = 2$. The RV conditions are set to one percent of the target's mean Sun distance and mean orbital velocity, respectively, which is sufficient for the purpose of illustration. Maximum allowed values for the distance $\Delta r_f = \|\mathbf{r}_T(t_f) - \mathbf{r}_{SC}(t_f)\| \leq \Delta r_{f,max}$ and the relative velocity $\Delta v_f = \|\mathbf{v}_T(t_f) - \mathbf{v}_{SC}(t_f)\| \leq \Delta v_{f,max}$ are therefore ${}^{(1)}\Delta r_{f,max} = 2.28 \cdot 10^6$ km and ${}^{(1)}\Delta v_{f,max} = 240$ m/s for the first flight phase, and ${}^{(2)}\Delta r_{f,max} = 1.5 \cdot 10^6$ km and ${}^{(2)}\Delta v_{f,max} = 300$ m/s for the second phase. The launch date windows are [54 426, 55 523] MJD and [54 827, 55 923] MJD, respectively.

A multiphase scenario like this offers additional optimization options compared to those of single-phase missions. Besides minimization of the total time

$$\Delta t_m = {}^{(p_f)}t_f - {}^{(1)}t_0 = {}^{(2)}t_f - {}^{(1)}t_0 \quad (4.5)$$

or the total propellant consumption

$$\Delta m_p = \sum_{i=1}^{p_f} \left({}^{(i)}m_{p,f} - {}^{(i)}m_{p,0} \right) = \sum_{i=1}^{p_f} {}^{(i)}\Delta m_p = {}^{(1)}\Delta m_p + {}^{(2)}\Delta m_p = {}^{(2)}m_{p,f} - {}^{(1)}m_{p,0}, \quad (4.6)$$

the total stay time could be maximized

$$\Delta t_{dw} = \sum_{i=2}^{p_f} \left({}^{(i)}t_0 - {}^{(i-1)}t_f \right) = {}^{(2)}t_0 - {}^{(1)}t_f, \quad (4.7)$$

or the sum of the transfer days minimized

$$\Delta t_t = \sum_{i=1}^{p_f} {}^{(i)}t_f - {}^{(i)}t_0. \quad (4.8)$$

The latter strives only for the shortest possible sum of all transfer times and neglects Δt_m and Δt_{dw} . Depending on the chosen objective function, the steering strategies ${}^{(i)}\mathbf{S}$, the launch dates ${}^{(i)}t_0$, and the initial propellant masses ${}^{(i)}m_{p,0}$ will be different, and these are the variables to be optimized in an overall mission context. As for single-phase transfers, the approach to optimize an entire mission scenario is to employ a GA. This time, however, the chromosome holds the parameters of multiple phases, as shown in Fig. 4.16. In this example, the steering strategy of the two phases is incorporated by an ANN for each phase. Their parameter vectors $\boldsymbol{\pi}$ and the initial conditions of each phase therefore become substrings of $\boldsymbol{\xi}$. This way the solutions of phase one and phase two compete with each other to form the optimum combination, i.e., the optimum overall mission design.

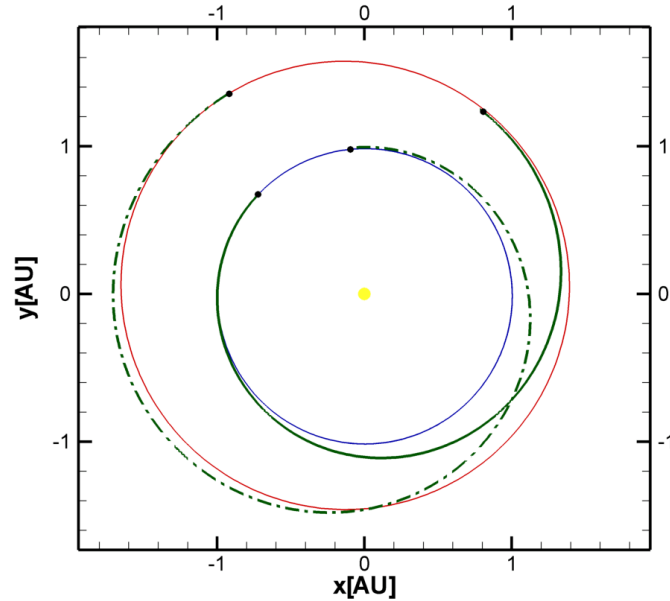


Figure 4.15: Low-thrust Earth-Mars-Earth Double Rendezvous. The inner circle is the orbit of the Earth and the outer one that of Mars. The dots denote the positions of Earth and Mars at departure and arrival, respectively. After launch at Earth, the spacecraft traverses on a low-thrust arc to Mars, stays there for the so-called dwell time for science operations. After conclusion of this task, it flies on a low-thrust trajectory back to Earth.

This corresponds to natural species that are mutually dependent but at the same time compete with each other for the limited resources of their common environment. Natural co-evolution lets both species evolve in such a way that both profit from each other at the expense of taking compromises. Co-evolution is the most likely reason for the survival of both species, because neither of them would survive the selection pressure of their common environment without the other.

The context is similar for the low-thrust, double RV mission. Neither transfer alone is a valid problem solution. A second phase taking place before the first one would obviously invalidate the mission as well, even if this wrong sequence order resulted in less transfer days. The same holds true for a second phase starting earlier than 100 days or later than 150 days past the end of the first phase, i.e., ${}^{(1)}t_f$ must obey $100 \text{ d} \leq ({}^{(2)}t_0 - {}^{(1)}t_f) \leq 150 \text{ d}$.

Without this dwell time constraint, optimization would be feasible through manual optimization of each phase individually, with basically two options. The first one starts with the optimization of the second transfer, which gives triplets $({}^{(2)}t_0, {}^{(2)}m_p, {}^{(2)}\Delta t)_i$ for each found transfer within the launch date interval. The corresponding $({}^{(1)}t_0, {}^{(1)}m_p, {}^{(1)}\Delta t)_i$ result from the subsequent optimization of the first phase, respecting ${}^{(2)}m_p$ for the dry mass ${}^{(1)}m_d$ and ${}^{(1)}t_f \leq {}^{(2)}t_0$. For launch intervals that allow more than one solution, several combinations of transfers for phase one and two result from this approach. The one having the smallest transfer days sum $\Delta t_t = ({}^{(1)}\Delta t + {}^{(2)}\Delta t)$ is the final, best solution, which, depending on the total launch date interval, can nevertheless have a significant dwell time at Mars.

The second option for manual optimization of this mission type is to start with the first flight leg. This is an intuitive approach but one needs to find a reasonable guess for the a priori unknown variable ${}^{(2)}m_p$. Again, the triplets $({}^{(1)}t_0, {}^{(1)}m_p, {}^{(1)}\Delta t)_i$ are the result of a trajectory

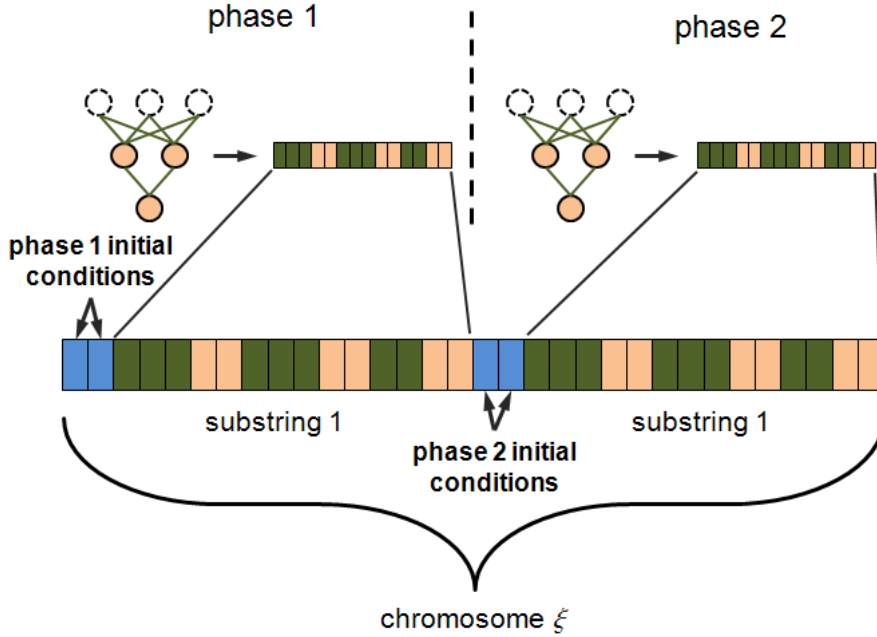


Figure 4.16: Multiphase Mission Parameters on a Single Chromosome.

optimization. The subsequent optimization of the second transfer leg, obeying

$${}^{(2)}t_0 \geqslant {}^{(1)}t_f, \quad (4.9)$$

provides $({}^{(2)}t_0, {}^{(2)}m_p, {}^{(2)}\Delta t)_i$. With the true $({}^{(2)}m_p)$, a second optimization of the first transfer leg gives an updated $({}^{(1)}t_0, {}^{(1)}m_p, {}^{(1)}\Delta t)_i$. If $({}^{(1)}t_f)$ would violate (4.9), a second optimization gives a new $({}^{(2)}m_p)$, etc. The combination of valid triplets that has fewest transfer days is the final solution. Both options are viable ways to obtain solutions for this problem but involve manual interaction and several iterations to find the global-optimal solution, although parts can certainly be semiautomated.

Respecting also the additional dwell time constraint, however, does not invalidate the described manual approach but turns it into a tedious process, even for this simple mission design. That is because of the mutual effect of each phase on the other and on the optimization objective. On one side, the second phase should have minimal duration but must begin within a time frame specified by the dwell time limits and $({}^{(1)}t_f)$ of the preceding phase. On the other side, this preceding phase is “coupled” to the second phase via $({}^{(2)}m_p)$, which has an effect on $({}^{(1)}\Delta t)$ and on $({}^{(1)}t_0)$, due to the required, flight-time-dependent, initial phase angle between the launch and target bodies. Manually finding of a valid solution requires the determination of the respective launch dates and to adjust them according to the results of the trajectory optimization of each phase. Depending on the actual problem, the launch body, the target, and on the posed constraints, this optimization can be time consuming.

MENC exploits the principle of coevolution for the automatic optimization of multiphase transfers and thus mitigates tedious, manual optimization. As the following description is not restricted on two-phase transfers, the generic notations i for a phase and $(i + 1)$ for its successor phase are used in the following.

An important decision for MENC concerns the evaluation of the individuals, which now contain solutions of more than one phase. Precisely, it is about the simulation and its trajectory integration’s initial conditions. There are two options, and both have their pros and cons.

The first option is to evaluate them one by one, with phase $(i + 1)$ always starting exactly at the final state of phase i . This requires chromosome-encoded initial conditions only for the first phase and reduces the number of necessary parameters. Using the preceding phase's state, mass, and time information also always guarantees physically valid trajectories, which is a considerable advantage. However, it reduces optimization efficiency, as it constrains the solutions space of phase $(i + 1)$ to a subspace that starts directly at the end of phase i . This could be imagined as optimizing phase $(i + 1)$ through a solution space "keyhole" determined by the final conditions of phase i . Moreover, only phase i can effect phase $(i + 1)$ at the begin of the optimization when boundary or accuracy constraints are not yet fulfilled and the optimization criterion is not taken into account for fitness determination. Driving the optimization into an undesired solution space can be a consequence, which is difficult to correct later upon boundary constraint match.

The second option also evaluates the substrings of an individual ξ one after another. In contradiction to option one, the phase-specific initial conditions of each phase stem from the chromosome. At the expense of additional optimization parameters, the solution space of each mission phase is therefore unconstrained right from optimization start and can be searched more thoroughly. However, this results in the disadvantage of discontinuous trajectories as initial conditions of phase $(i + 1)$ now generally differ from the final conditions of phase i . Assuring a continuous and smooth final trajectory is therefore a task of the optimization procedure, and often implies respecting the optimization criterion from the very beginning. This is not a strict requirement for all mission designs, as can be seen in the two-phase, Earth-Mars-Earth mission example. In this, the required dwell time prohibits continuous and smooth transition between the phases.

The initial conditions of all but the first phase are also termed transition conditions as they determine the transition between two subsequent phases. Encoding them on ξ , where they are subject to optimization by the GA, allows implicit optimization of these parameters as follows. If the trajectory of phase i fails to match the transition conditions of phase $(i + 1)$, but phase $(i + 1)$ would achieve its objective also with slightly changed initial conditions, such a change could be done in favor of phase i . The transitions conditions of all phases thus would continuously evolve to those of the optimal solution. However, they are unlikely to match perfectly upon optimization stop. A mechanism that uses a predecessor phase's final conditions as initial conditions for the following phase, if the respective deviation is smaller than a problem-dependent threshold, finally guarantees a physically valid final solution.

4.4.4 Phase Transition Conditions

The transition conditions between consecutive flight phases depend on the actual mission design. This section defines the condition types used within this work, as well as the mission types for which they were applied. The first transition condition variable is the time t . For mission designs that require a continuous trajectory that leads from one phase to another, being the majority of space missions, the following equation holds

$${}^{(i)}t_f = {}^{(i+1)}t_0. \quad (4.10)$$

That means, except for missions that involve a dwell time at a celestial body before they continue traversing to the next target body, the end time of one phase ${}^{(i)}t_f$ must equal the start time ${}^{(i+1)}t_0$ of the subsequent phase.

The next transition variable is that of the spacecraft mass m_{SC} . Its transition condition is

$${}^{(i)}m_{SC,f} = {}^{(i+1)}m_{SC,0} + \Delta m. \quad (4.11)$$

Table 4.3: Transition Condition Types for Three Scenarios. Here, (applicable) means that mass continuity is required but instantaneous mass changes may be allowed, e.g., to account for jettisoned propulsion stages or surface probes.

Scenario	${}^{(i)}\mathbf{x} = {}^{(i+1)}\mathbf{x}$	${}^{(i)}m_{\text{SC}} = {}^{(i+1)}m_{\text{SC}}$	${}^{(i)}t_f = {}^{(i+1)}t_0$
A	applicable	(applicable)	applicable
B	n/a	(applicable)	n/a
C	applicable	applicable	applicable

This formulation accounts for planned changes of spacecraft mass Δm , which can be surface probes dropped on close flyby at a target or jettisoned propulsion stages.

The third transition variable is the astrodynamic state of the spacecraft \mathbf{x}_{SC} . A continuous trajectory requires identical state vectors at the end of phase i and the begin of phase $(i + 1)$.

$${}^{(i)}\mathbf{x}_{\text{SC},f} = {}^{(i+1)}\mathbf{x}_{\text{SC},0} \quad (4.12)$$

An exception to (4.12) arises from missions including dwell times or instantaneous velocity changes, e.g., resulting from chemically induced ΔV . The latter was not used within this work.

Three scenarios and the respective transition conditions should be defined for the transition between two adjacent mission phases. Table 4.3 shows how the different transition conditions (4.10), (4.11), and (4.12) would be defined. Scenario A is either a flyby at a celestial body or a rendezvous at that body without stay time. A rendezvous scenario including a stay time at the target body before starting the second phase is scenario B. Scenario C involves a SOI-crossing, i.e., a change of the central body.

5

Implementation

This chapter shows and explains the essential elements of the implementation of MENC in the software InTrance. Dachwald developed the first version of this optimization tool [13] exclusively for heliocentric, single-phase low-thrust transfers. It was the starting point for the author's work on the extension of ENC to the optimization of non-heliocentric and heliocentric multiphase low-thrust transfers. The existing version was during this work extensively revised and extended to enable the optimization of multiphase transfers in heliocentric space. The version of InTrance developed by the author supports both single-phase, non-heliocentric transfers and continuous, multiphase, low-thrust transfers comprising non-heliocentric as well as heliocentric phases. InTrance is written in C++ following the principles of object-oriented programming (OOP). Its input is a user provided, text-file based configuration. Its output is a set of comma-separated-value data files, plain text report files, and data files for visualization of the resulting transfers¹.

Enhancement of InTrance required the implementation of several mechanisms described in this chapter. They are of three general types: type A, mandatory for functionality; type B, to increase simulation fidelity for more detailed analysis; and type C, to improve optimization robustness. Type A mechanisms are essential, as they transcribe the problem of multiphase, low-thrust trajectory optimization to the EA such that in principle it can solve it. Type B mechanisms allow refined mission analysis through trajectory integration in a more realistic simulation environment, e.g., with eventual planet shadows and third-body disturbance forces. Most of these mechanisms are not essential for multiphase trajectory optimization. Type C mechanisms increase the robustness of the optimization, i.e., they help to find the globally-optimal solution more reliably. An example for such a feature is the dynamic determination of the control step size.

Depending on the actual transfer problem, some of the type B mechanisms also belong to type A. For example, although basically a simulation fidelity-enhancing feature, third-body disturbance accelerations must be respected for low-thrust trajectory problems that involve crossing of a SOI because there the gravitational pull of the central body and the disturbing body are in balance. The support of third-body accelerations is a mandatory technique in that case. Table 5.1 gives an overview of the different implementation elements and their classification.

Five sections describe these mechanisms and techniques in detail. Section 5.1 describes the necessary features to carry out non-heliocentric astrodynamics simulations. The following section 5.2 describes the developed multiphase framework and its implications on the optimization process. Section 5.3 treats simulation components in more detail, e.g., equations of motion, integration schemes, disturbance bodies influence. The elements and mechanisms of

¹The visualization format is the Virtual Reality Modeling Language (VRML).

Table 5.1: Classification of the mechanisms and techniques in InTrance. Type A mechanisms are mandatory for functionality, type B mechanisms increase simulation fidelity, and type C mechanisms improve the robustness.

Element/mechanism	Type A	Type B	Type C
Non-heliocentric simulation	x	n/a	n/a
Planetary shadows	n/a	x	n/a
Third-body perturbation	x	x	n/a
Excess energy optimization	n/a	x	n/a
Dynamic control step size	n/a	n/a	x
Parameter range adaption	n/a	n/a	x
Search space scan	n/a	n/a	x
Hypercube size control	n/a	n/a	x
Variable boundary constraints	n/a	n/a	x

the EA employed in InTrance are the content of section 5.4. Section 5.5 treats the usage of ANNs as NCs within MENC and explains the encoding of the NC input information and the transcription of NC output data into spacecraft control variables.

5.1 Support of Non-heliocentric Low-thrust Transfers

One of the major changes to InTrance’s astrodynamics simulation implementation was the added support for non-heliocentric low-thrust transfers. This was necessary as the existing version was limited to heliocentric trajectory integration using an EOM formulation in polar coordinates. It further neglected third-body disturbances, planetary shadows, and assumed the sunlight direction vector to be coincident with the radial distance vector of the spacecraft. The assumptions underlying that implementation therefore prohibited generic non-heliocentric simulations, e.g., those of an EP-powered spacecraft on a near-polar orbit about the Moon. The reasons for enhancing InTrance with non-heliocentric simulation capability were therefore as follows.

With a few exceptions, heliocentric orbits of most space missions are more or less constrained to the ecliptic plane. This is mainly for two reasons. First, the orbits of the respective mission targets, e.g., the major celestial bodies, are within an maximum inclination range of about seven degrees¹. Second, the required energy to increase the heliocentric inclination is significant. Missions to high-inclination orbits or to targets on such orbits therefore almost inevitably turn into high-energy missions. The resulting ΔV -requirement necessitates either larger launch systems or the inclusion of GAs into trajectory design. Neither one is always feasible. For non-heliocentric orbits, e.g., around Earth, Mars, Jupiter, or one of their moons, there is generally no such “limitation” on the orbit’s inclination. In fact, spacecraft whose objective is to monitor or to map the surface of a body often fly on polar or near-polar orbits. This has implications for the simulation framework because polar state equations exhibit singularities for certain inclination values. Singularity-free astrodynamics state equations are therefore preferable for non-heliocentric calculations. InTrance therefore uses a cartesian formulation of the state equations for non-heliocentric simulations. As the polar state equation formulation of InTrance has advantages in terms of performance over the cartesian form, it was kept for heliocentric simulations.

¹The orbit inclination of Pluto, however, which is now categorized as dwarf planet, is about 17 deg.

The second important peculiarity is the influence of disturbance accelerations, resulting from the gravitational attraction of bodies other than the central body. In the context of preliminary mission analysis, these third-body disturbance accelerations are negligible for heliocentric motion. For non-heliocentric orbits, however, their influence on a spacecraft and on the spacecraft's orbit can be considerable. This influence is dependent on the spacecraft's distances to the central body and to the disturbing bodies. If a low-thrust spacecraft solely relies on its thrust acceleration and is not supported with additional v_∞ , it is principally longer exposed to such accelerations than chemically propelled spacecraft. A simulation framework must therefore account for third-body accelerations for all low-thrust mission phases whose trajectories lead close to or across a body's SOI.

Another peculiarity of non-heliocentric simulations is the influence of the shadow of the central body, caused by the obstruction of the Sun by that body. This is of significant importance for spacecraft whose propulsion system relies either directly (solar sails) or indirectly (EP) on sunlight. If a spacecraft is in planetary shadow, its solarelectric power generation subsystem is inherently disabled. If it has not other power supply, thrust generation ceases as well and reduces the means for orbit alteration. This effect decreases with increasing orbit heights. However, its influence on the transfer time of a sunlight-dependent spacecraft from a non-heliocentric lower orbit to a higher orbit, or even to an escape trajectory, can be significant.

The fourth difference is related to the sunlight direction vector. This is the vector that points from the Sun to the spacecraft. For heliocentric simulations, the sunlight direction vector coincides with the radial position vector. This is not the case for non-heliocentric motion. Especially solar sail steering strategies around celestial bodies are different than on heliocentric orbits and can demand rapid attitude changes. For EP-powered spacecraft, this difference may be neglected as well, if one assumes the spacecraft's solar panels always optimally aligned to the sunlight vector.

5.2 Multiphase Framework

A solution candidate ξ in InTrance is evaluated via an astrodynamics simulation. The simulation result is a trajectory obtained through numerical integration of the EOMs. Multiphase transfers can differ considerably from one phase to the next, a fact any simulation framework for such transfers must take care of. InTrance was originally designed to support single-leg missions with a C++ simulation object for the entire mission. This was changed during this work to a chain of concatenated simulation objects that represents the mission. Each object thereby represents a phase of that mission and the objects call up themselves in sequential order. One advantage of this approach is its transparency to the outer optimization loop. The EA running there, "sees" only the first evaluator object to which it passes an unevaluated solution candidate ξ . Figure 5.1 shows this concept for a mission that comprises three mission phases, with the first and the last phase employing an NC for spacecraft steering. After the trajectory integration, the first simulation object then passes ξ on to the second simulation object and also receives it back again later from that object.

The simulation object chain is implemented as a doubly linked list. That means each object can access its eventually existing predecessor and its successor. Furthermore, by iterating through that list, each simulation can access all other list objects. This assures that each simulation chain object can access public information of all other simulation objects of the same chain, whether they have yet carried out the trajectory integration for the current solution candidate ξ or not. This is important for the optimization of the transition or couple conditions, which is covered in subsection 5.2.1.

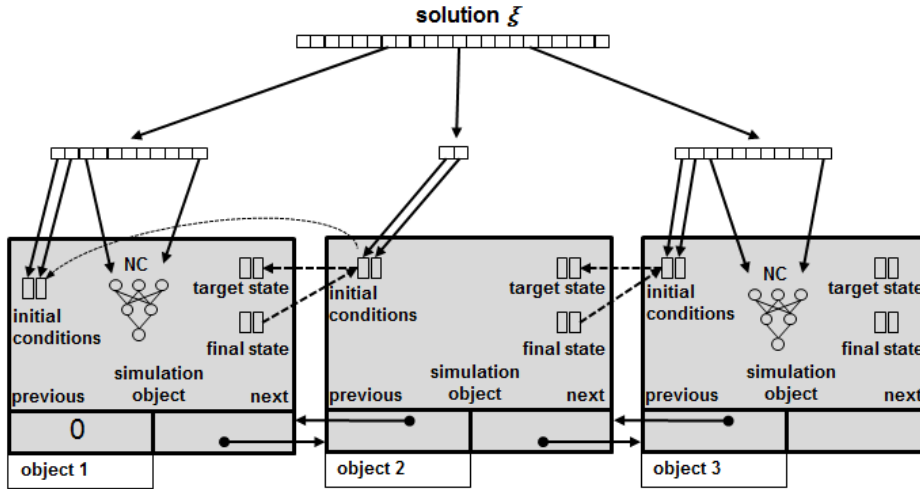


Figure 5.1: Chained Simulation Objects.

An individual's evaluation with this simulation object chain is as follows. It starts with the call to the first object of that chain, with the candidate solutions passed as parameter. The mission phase covered by the simulation object is then evaluated and the solution is assigned a phase-specific subfitness value¹. The simulation object then calls its successor object, with the complete solution chromosome ξ again passed as parameter. These steps repeat until the last simulation chain object was called and has finished its evaluation. In addition to the phase-specific subfitness, the last simulation also assigns the absolute fitness $J(\xi)$ to the candidate solution, i.e., it assesses the fulfillment of the optimization criterion w.r.t. the entire mission². Recursively going back to the first simulation object then returns the completely evaluated solution candidate, which is given back to the calling EA.

5.2.1 Implications on Initial and Final Conditions

Simulation is a core component of InTrance and corresponds to the EA element of evaluation. It receives candidate solutions during the EA's reproduction cycle and yields a trajectory, whose shape and transfer duration is determined by the initial conditions and the information encoded in the candidate solution's chromosome ξ .

A simulation run comprises the three stages initialization, state equations integration, and postprocessing. The last one gives the phase-specific subfitness. The simulation objects in InTrance are implemented with C++ objects, with specific initial, final, and boundary conditions, and each phase of a multiphase mission corresponds to exactly one simulation object. The simulation parameters are in general externally provided, e.g., by a user. Instead of fixed values, the initial conditions may also be given as parameter ranges, from which the optimizer must find the optimal value from. The requirement for physically valid final trajectories additionally leads to coupling conditions, which can translate into initial and final conditions of adjacent mission phases. That means, the final spacecraft state of phase i can be the initial state of phase $(i+1)$. On the other hand, the initial state of phase $(i+1)$ can be the target state of phase i . The couple conditions can hardly be guessed, yet be chosen optimally by an operator. It is better to let the optimizer choose promising parameter ranges and also let it modify them during the optimization process.

¹The concept of phase-specific, so-called proximity values is covered in subsection 5.3.7.

²See also subsection 5.4.7 for more details on the fitness calculation in InTrance.

A simulation object is therefore logically linked to its predecessor simulation and to the successor simulation object. Figure 5.1 shows an example of such a simulation object chain. In order to set up a simulation object correctly, it must obtain information about certain simulation parameters from the preceding simulation and from its successor. To calculate the spacecraft launch mass for the phase i , for example, the initialization stage of that phase must determine the propellant masses required for the following mission phases. This is feasible because of the chosen approach to encode the initial conditions of all phases on the solution chromosome. This information contains the launch or arrival date, the initial or final state, and the spacecraft or propellant mass, respectively. Each simulation object also contains a control strategy, which is principally initialized with the candidate solution data but may also be a completely different steering strategies, e.g., constant thrust vector angles. The second simulation object in Fig. 5.1 is an example for such a mission phase. The figure also shows the relation between coupling conditions of adjacent phases. The initial conditions of the second simulation object are at the same time the target conditions of the first simulation object, and the assessment of the complete solution must also include fulfillment of these target conditions. Later in the optimization process, when the difference between the final state of phase i and the initial state of phase $(i+1)$ has become negligibly small, phase i 's final state can be used directly as initial state of phase $(i+1)$ to assure a continuous trajectory, as indicated in Fig. 5.1.

5.2.2 Spacecraft Design

A spacecraft is characterized by a set of specific parameters. Important ones are for example the mass, electrical input or output power, or the maximum thrust level. Enabled by the developed multiphase simulation framework, InTrance can optimize some of these parameters within user-provided limits. It can therefore specifically optimize the design of a spacecraft and, via the steering strategy, the trajectory for a space mission.

The most simple spacecraft configuration in InTrance is that of a solar sail. It is configured primarily via its characteristic acceleration or sail loading, and, for the optical SRP force model, its sail film optical properties. For electrical propulsion systems, InTrance offers configuration and optimization options for the structure mass, tank, propellant, power supply systems, etc. Besides the steering strategy and the launch conditions, e.g., the hyperbolic excess velocity vector \mathbf{v}_∞ , the optimization thus also includes the spacecraft design. For a spacecraft with a staged propulsion subsystem, InTrance can find the optimal compromise between each phase's propellant mass. If configured with a power-specific mass, InTrance can also optimize the size of the spacecraft solar generator size. More details about the respective configuration parameters can be found in appendix D.

5.2.3 Potential Applications of the Developed Framework

The developed simulation framework can help to solve numerous problem types. The optimization of multiphase low-thrust transfer problems is thereby the only one that was used within this work. Such mission designs are thereby principally unconstrained in terms of propulsion systems and steering strategies. Using this framework without any steering strategy would even allow the optimization of chemically-propelled multiphase missions; the trajectories of each mission phase would thereby be treated as coast arcs. Additionally, this framework allows combinations of low-thrust and ballistic phases within a mission.

However, with this framework and an EA's optimization capability the treatment of other problems than trajectory optimization for a particular mission become viable. Such problems address more fundamental questions w.r.t. NCs that are used for spacecraft steering. One

of them is the question of the smallest NC, i.e., the one with the fewest neurons, that is still capable find the optimal transfer. Based on this, the logical next question is whether it is possible to train an NC so that it can find the optimal transfer for a particular mission regardless of the launch date. In other words, once properly trained, such an NC would steer a low-thrust spacecraft optimally to the respective target state for every given launch date without further training. A further complication is the usage of more than a single target for the training of a NC. A single NC that masters the two low-thrust transfers from the Earth to Mars and back would already possess better generalization than an NC that can handle only one transfer. Finally, such a training setup in combination with different launch dates within one synodic cycle could yield an NC with a better general capability of treating low-thrust transfer problems.

Potential applications of the developed simulation framework are numerous. However, the focus of this work is on the optimization of multiphase low-thrust trajectories. Therefore the approach of one NC for each phase was chosen as a first logical step to prove the functionality of MENC. All the sketched applications have not been pursued further¹. Potential later work can however use the gained results as reference for the described more complex problems.

5.3 Simulation

Simulation is an essential EA component and yields the trajectory $\mathbf{r}[\bar{t}]$, whose assessment determines the fitness $J(\xi_i)$ of the solution candidate ξ_i . The following subsections describe the important enabling mechanisms of the simulation objects within InTrance. They cover topics such as numerical integration, spacecraft structure and mass breakdown, state equations, third-body perturbations or planetary shadows, and problem-dependent constraints. As the number of InTrance's simulation features has grown over the time, only those important for this work are presented.

5.3.1 Spacecraft Mass Breakdown

At a time t , the maximum acceleration of a spacecraft is determined through its available maximum thrust force $F_{\max}(t)$ and its total mass $m_{\text{SC}}(t)$. The maximum thrust force and the total mass are dependent on each other in manifold ways. An EP-powered spacecraft, for example, has several components that are decisive for thrust generation. At the same time these components also contribute to the overall mass budget, e.g., propellant, propellant tank, solarelectric power generation system, and the thrust unit(s). To support the optimization of these components for a particular mission design in conjunction with the optimal transfer strategy, a more detailed spacecraft mass model was developed for InTrance's simulation model.

In this model, the spacecraft wet mass or launch mass $m_{\text{SC},0}$ is the sum of the dry mass m_{d} and the propellant mass m_{p}

$$m_{\text{SC},0} = m_{\text{d}} + m_{\text{p}}, \quad (5.1)$$

with the spacecraft dry mass comprising the payload mass m_{PL} and the bus mass m_{bus}

$$m_{\text{d}} = m_{\text{PL}} + m_{\text{bus}}. \quad (5.2)$$

InTrance offers two principle options to declare the mass of the bus components structure, propellant tank, and power generation: either explicitly, via the component's total mass, or,

¹see also section 7.2

implicitly, in relation to a second parameter of the respective component. Using a power-specific mass factor α_{PGS} , for example, the mass of the electric power generation system m_{PGS} is determined by its reference power output $P_{e,0}$.

$$m_{\text{PGS}} = \alpha_{\text{PGS}} P_{e,0}. \quad (5.3)$$

Depending on the actual optimization criterion, this allows to implicitly optimize for the optimal compromise between less powerful but lighter and heavier solar panels with higher electrical power output.

Similarly, the tank mass of the spacecraft is determined by a propellant mass-specific factor α_{prop} and the propellant mass m_p .

$$m_{\text{tank}} = \alpha_{\text{prop}} m_p. \quad (5.4)$$

The structure mass of the spacecraft m_{st} depends on the propulsion subsystem mass m_{PS} , the masses of the other components, and a mass-specific factor α_{st} . In practice, α_{st} is between 0.15 and 0.25.

$$m_{\text{st}} = \alpha_{\text{st}} (m_{\text{PGS}} + m_{\text{PS}} + m_{\text{tank}} + m_p) \quad (5.5)$$

The sum of the system masses provides the bus mass

$$m_{\text{bus}} = m_{\text{PGS}} + m_{\text{PS}} + m_{\text{tank}} + m_{\text{st}}. \quad (5.6)$$

Inserting (5.3)- (5.5) in (5.6), the bus mass may be expressed as function of $P_{e,0}$ and m_p

$$m_{\text{bus}} = (1 + \alpha_{\text{st}})(\alpha_{\text{PGS}} P_{e,0} + m_{\text{PS}}) + (\alpha_{\text{prop}} + \alpha_{\text{prop}} \alpha_{\text{st}} + \alpha_{\text{st}}) m_p. \quad (5.7)$$

If configured to use given factors α_i and intervals for $P_{e,0}$ and m_p , InTrance can thus optimize the entire spacecraft mass breakdown together with the steering strategy.

5.3.2 Equations of Motion

The dynamic spacecraft state is determined through the position vector $\mathbf{r}_{\text{SC}}(t)$ and the associated velocity vector $\mathbf{v}_{\text{SC}}(t)$. Together with the spacecraft mass $m_{\text{SC}}(t)$, these vectors form the 7-dimensional state vector $\mathbf{s}(t)$ of the state variables s_i

$$\mathbf{s}(t) = (s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6 \ s_7)^T = \begin{pmatrix} \mathbf{r}_{\text{SC}}(t) \\ \mathbf{v}_{\text{SC}}(t) \\ m_{\text{SC}}(t) \end{pmatrix}. \quad (5.8)$$

The type of these state variables is thereby not crucial as long as the state description is complete and explicit. Commonly used representations contain cartesian, polar, or orbital elements (see appendices A and B). The minimum number of variables that is required to completely describe the state of a dynamic system determines the state vector dimension and depends on the system's number of degree-of-freedom (DOF). A simulation that considers only the translational motion of a spacecraft and neglects rotational motion has three DOF for position and velocity, respectively, which requires six state variables for a complete description. As the mass of a spacecraft that operates a propellant-dependent propulsion system changes, the absolute mass m_{SC} increases the state vector dimension to seven.

The values of the state variables s_i change under the effect of internal and external forces. State equations quantify the resulting rate of change w.r.t. time of each s_i . In this case, the state equations primarily describe the spacecraft motion, which is why they are also called EOMs. External factors are accelerations caused by external forces acting on the spacecraft.

These are, for example, the accelerations due to the central body's gravitation \mathbf{a}_g and due to third-body disturbance acceleration \mathbf{a}_{dis} . The acceleration caused by the thrust force \mathbf{a}_{thr} and the spacecraft mass reduction due to expelled propellant are examples of internal factors. The absolute acceleration vector $\mathbf{a}_t = \mathbf{a}_g + \mathbf{a}_{\text{thr}} + \mathbf{a}_{\text{dis}}$ is the sum of internal and external forces. This vector usually depends on a number of further parameters. A propulsion system's thrust force, for example, may depend on a throttle setting or on the Sun-spacecraft distance, or even on both, as is the case for SEP. The state vector's first time derivative $\dot{\mathbf{s}}$ is in general a function of the spacecraft's state \mathbf{s} , the independent variable t , the flight time $\Delta t = t - t_0$, and \mathbf{a}_t

$$\dot{\mathbf{s}} = \frac{d\mathbf{s}}{dt} = \frac{d}{dt} \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \\ m_{\text{SC}} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \dot{\mathbf{v}} \\ \dot{m}_{\text{SC}} \end{pmatrix} = \mathbf{f}(t, \Delta t, \mathbf{s}, \mathbf{a}_t). \quad (5.9)$$

The actual EOM formulation depends on the chosen state variables representation, and cartesian, polar, or orbital elements are the most commonly used. Cartesian EOMs have the advantage of a rather simple formulation and, unlike EOMs in of polar coordinates or orbital elements, they are free of singularities.

$$\dot{\mathbf{s}}_c = \frac{d\mathbf{s}_c}{dt} = \begin{pmatrix} \dot{s}_1 \\ \dot{s}_2 \\ \dot{s}_3 \\ \dot{s}_4 \\ \dot{s}_5 \\ \dot{s}_6 \\ \dot{s}_7 \end{pmatrix}_c = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \dot{m}_{\text{SC}} \end{pmatrix} = \begin{pmatrix} s_4 \\ s_5 \\ s_6 \\ \mathbf{a}_t \mathbf{e}_x \\ \mathbf{a}_t \mathbf{e}_y \\ \mathbf{a}_t \mathbf{e}_z \\ -\dot{m}_p \end{pmatrix} \quad (5.10)$$

For potentially highly-inclined orbits about celestial bodies, e.g., for polar Earth orbits, InTrance uses (5.10) with the cartesian state vector $\mathbf{s}_c = (x, y, z, \dot{x}, \dot{y}, \dot{z}, m_{\text{SC}})^\top$.

The corresponding polar state vector is $\mathbf{s}_p = (r, \varphi, \vartheta, \dot{r}, \dot{\varphi}, \dot{\vartheta}, m_{\text{SC}})^\top$, and the polar EOMs result from multiplying (A.12) with the polar frame unit vectors \mathbf{e}_r , \mathbf{e}_φ , and \mathbf{e}_ϑ , and resolving for \ddot{r} , $\ddot{\varphi}$, and $\ddot{\vartheta}$

$$\dot{\mathbf{s}}_p = \frac{d\mathbf{s}_p}{dt} = \begin{pmatrix} \dot{r} \\ \dot{\varphi} \\ \dot{\vartheta} \\ \ddot{r} \\ \ddot{\varphi} \\ \ddot{\vartheta} \\ \dot{m}_{\text{SC}} \end{pmatrix} = \begin{pmatrix} s_4 \\ s_5 \\ s_6 \\ \mathbf{a}_t \mathbf{e}_r + r\dot{\vartheta}^2 + r\dot{\varphi}^2 \cos^2 \vartheta \\ \frac{1}{r \cos \vartheta} (\mathbf{a}_t \mathbf{e}_\varphi + 2\dot{\varphi}(r\dot{\varphi} \sin \vartheta - \dot{r} \cos \vartheta)) \\ \frac{1}{r} (\mathbf{a}_t \mathbf{e}_\vartheta - 2\dot{r}\dot{\vartheta} - r\dot{\varphi}^2 \cos \vartheta \sin \vartheta) \\ -\dot{m}_p \end{pmatrix}. \quad (5.11)$$

A spacecraft on an heliocentric trajectory rarely experiences high inclination values and thus also the polar elevation angle ϑ is unlikely to assume values close to $\pi/2$. InTrance therefore uses the polar EOMs for heliocentric simulations.

5.3.3 Third-body Perturbation

The motion of a spacecraft with mass m_{SC} is determined by its current position and velocity and the gravitation of the central body, whose mass m_{CB} is in general magnitudes larger than the spacecraft mass $m_{\text{CB}} \gg m_{\text{SC}}$. Without any other acceleration than the central body's

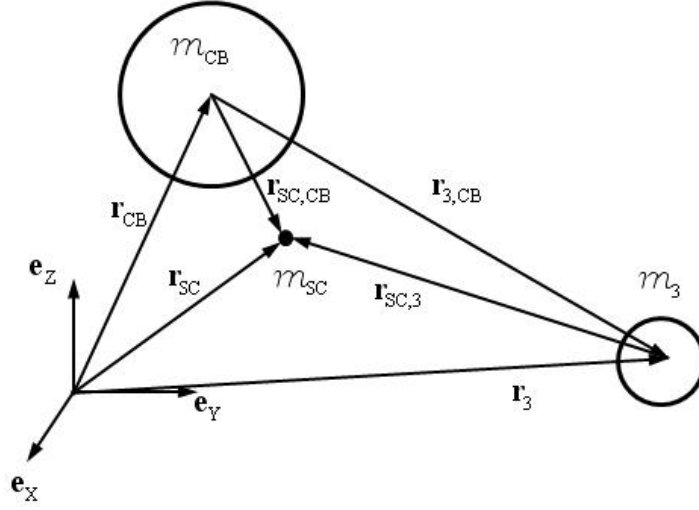


Figure 5.2: Three-Body Environment in an Inertial Reference Frame. A spacecraft at the position \mathbf{r}_{SC} and with the mass m_{SC} orbits a central body, which has the mass m_{CB} and is located at \mathbf{r}_{CB} . The gravitational acceleration of a third body of the mass m_3 at the location \mathbf{r}_3 disturbs the orbit of the spacecraft. The vectors $\mathbf{r}_{SC,CB}$ and $\mathbf{r}_{3,CB}$ are the spacecraft's and the disturbing body's position w.r.t. the central body, and $\mathbf{r}_{SC,3}$ is the spacecraft's position w.r.t. the perturbing body.

gravitational pull, the spacecraft would move on a Keplerian orbit.¹ However, in reality other bodies also act on the spacecraft through their gravitation and therefore influence its orbit. The inducing accelerations depend on the respective distances of the perturbing body and the spacecraft and the perturbing body and the central body. This effect is called third-body perturbation and the resulting accelerations are normally magnitudes smaller than the dominating central body pull. Third-body perturbation is negligible for preliminary mission analysis of heliocentric missions, as long as a trajectory does not lead close to a mass-rich planet, like Jupiter or Saturn. For the calculation of Earth-bound high-altitude orbits, however, accounting for the gravitation of the Moon and the Sun is required to achieve an acceptable fidelity. For low-thrust transfers from the Earth to the Moon, the inclusion of third-body perturbation is mandatory.

Figure 5.2 shows a representative three body setup, with the central body m_{CB} at the inertial position \mathbf{r}_{CB} , the spacecraft at \mathbf{r}_{SC} with the mass m_{SC} , and a perturbing body m_3 at \mathbf{r}_3 . The spacecraft position w.r.t. the central body $\mathbf{r}_{SC,CB}$ and its second time derivative are

$$\mathbf{r}_{SC,CB} = \mathbf{r}_{SC} - \mathbf{r}_{CB} \quad (5.12)$$

$$\ddot{\mathbf{r}}_{SC,CB} = \ddot{\mathbf{r}}_{SC} - \ddot{\mathbf{r}}_{CB}. \quad (5.13)$$

Inserting the inertial acceleration vectors

$$\ddot{\mathbf{r}}_{SC} = -\mu_{CB} \frac{\mathbf{r}_{SC,CB}}{r_{SC,CB}^3} - \mu_3 \frac{\mathbf{r}_{SC,3}}{r_{SC,3}^3} \quad (5.14)$$

and

$$\ddot{\mathbf{r}}_{CB} = \mu_{SC} \frac{\mathbf{r}_{SC,CB}}{r_{SC,CB}^3} + \mu_3 \frac{\mathbf{r}_{3,CB}}{r_{3,CB}^3} \quad (5.15)$$

¹For pure two-body motion, the gravitation of the central body must be uniform, i.e., a perfect sphere with homogeneous mass distribution.

into equation (5.13) yields the spacecraft's acceleration vector w.r.t. the central body

$$\ddot{\mathbf{r}}_{\text{SC,CB}} = -\mu_{\text{CB}} \left(1 + \frac{m_{\text{SC}}}{m_{\text{CB}}} \right) \frac{\mathbf{r}_{\text{SC,CB}}}{r_{\text{SC,CB}}^3} - \mu_3 \left(\frac{\mathbf{r}_{\text{SC},3}}{r_{\text{SC},3}^3} + \frac{\mathbf{r}_{3,\text{CB}}}{r_{3,\text{CB}}^3} \right), \quad (5.16)$$

with the gravitation constant γ and the gravitational coefficients $\mu_{\text{CB}} = \gamma m_{\text{CB}}$, $\mu_{\text{SC}} = \gamma m_{\text{SC}}$, and $\mu_3 = \gamma m_3$. The two terms result from the gravitational acceleration due to the central body and the perturbing body, respectively. Taking into account $m_{\text{SC}}/m_{\text{CB}} \approx 0$ and n perturbing bodies, a general formulation of (5.16) is

$$\ddot{\mathbf{r}}_{\text{SC,CB}} = -\mu_{\text{CB}} \frac{\mathbf{r}_{\text{SC,CB}}}{r_{\text{SC,CB}}^3} - \sum_{i=1}^n \mu_i \left(\frac{\mathbf{r}_{\text{SC},i}}{r_{\text{SC},i}^3} + \frac{\mathbf{r}_{i,\text{CB}}}{r_{i,\text{CB}}^3} \right). \quad (5.17)$$

With $\mathbf{r}_{i,\text{SC}} = -\mathbf{r}_{\text{SC},i}$, (5.17) can be rewritten to the form given by Vallado [91]

$$\ddot{\mathbf{r}}_{\text{SC,CB}} = -\mu_{\text{CB}} \frac{\mathbf{r}_{\text{SC,CB}}}{r_{\text{SC,CB}}^3} + \sum_{i=1}^n \mu_i \left(\frac{\mathbf{r}_{i,\text{SC}}}{r_{i,\text{SC}}^3} - \frac{\mathbf{r}_{i,\text{CB}}}{r_{i,\text{CB}}^3} \right). \quad (5.18)$$

The first term is the gravitational acceleration of the central body and responsible for two-body motion. The second term is the perturbation acceleration

$$a_{\text{pert}} = \sum_{i=1}^n \mu_i \left(\frac{\mathbf{r}_{i,\text{SC}}}{r_{i,\text{SC}}^3} - \frac{\mathbf{r}_{i,\text{CB}}}{r_{i,\text{CB}}^3} \right). \quad (5.19)$$

5.3.4 Numerical Integration

ENC consists of the outer EA reproduction cycle as optimization loop and an inner trajectory integration loop within the EA's evaluation component. The integration loop numerically integrates the state equations, which form the dynamic constraint of the optimization. These equations are a set of first-order ordinary differential equations (ODE) of the form

$$\dot{\mathbf{s}} = \mathbf{f}(t, \mathbf{s}), \quad (5.20)$$

and they describe the spacecraft's orbital motion under the influence of gravitational, disturbance, and thrust forces. Information on the spacecraft position and velocity is contained in the generic state vector \mathbf{s} . The underlying astrodynamics equations of motion are a second-order differential equation system of the form $\ddot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$ ¹. Efficient integration schemes exist for that type of differential equations. However, as additional time-dependent first-order variables complement \mathbf{s} , e.g., the propellant mass m_p (see also subsection 5.3.2), a first order integrator was required.

Numerous integration methods exist for the integration of the resulting equation system. During this work either the RUNGE-KUTTA-FEHLBERG 4(5) scheme [26] for lower-precision calculations or a DORMAND-and-PRINCE 8(7) scheme [57] for high simulation fidelity optimization runs were used. Both methods are one-step integration schemes, i.e., they calculate the new state vector at $t + h$ from the current state \mathbf{s} , the state equations, the independent variable t , and the step size h . Such integration methods therefore need no information of previous state vectors. The integration methods were chosen because in comparison to other numerical integration methods [5] they:

- exhibit good stability and robustness,

¹Drag cannot be respected with this formulation. This would require dependency on the first time derivative of the spacecraft's position.

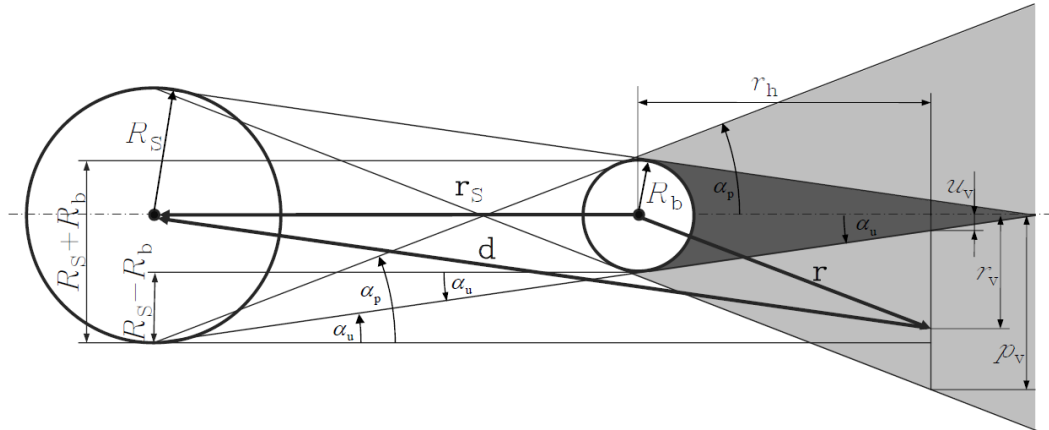


Figure 5.3: Eclipse Geometry. The shown generic setup of the eclipse geometry is exaggerated for better understanding. The Sun with the radius R_S is on the left, and the eclipse-causing body b with the radius R_b is on the right. The distance between them equals the Sun distance r_S . Occultation of the Sun disc by the body b results in two shadow cones: the penumbra with a cone angle α_p and the umbra with a cone angle α_u . For the Sun and the Earth the shadow angles are $\alpha_u = 0.2641$ deg and $\alpha_p = 0.2690$ deg [91, p. 301]. The other variables are the position vector \mathbf{r} of a spacecraft w.r.t. body b , the Sun vector \mathbf{d} , and the temporary variables r_h , r_v , p_v , and u_v .

- allow easy step size control, which is important for a dynamic control step size,
- have acceptable round-off errors.

The implemented integrator objects control the internal step size via user-provided limits for the absolute error tol_{abs} and the relative error tol_{rel} . Hard-coded default values of 10^{-6} were used for both boundaries if no other limits are given.

5.3.5 Eclipse Treatment

Low-thrust propulsion systems generally¹ rely on sunlight for the generation of thrust. This dependency is either implicit, if the sunlight's energy is converted into electrical energy to power EP thrusters, or explicit, if the sunlight is employed directly for propulsion. SEP systems are of the first kind while solar sailing is of the second one. Without means for temporary energy storage, sunlight-dependent propulsion systems have limited or no thrust capability if the incident sunlight is obstructed by another body. This eclipse effect is negligible for heliocentric trajectories. For a realistic simulation of trajectories close to planets or moons, e.g., for spacecraft in low Earth orbit (LEO), an eclipse model is however mandatory. Figure 5.3 shows the eclipse geometry, with the penumbra cone angle α_p where the Sun is partially occulted

$$\alpha_p = \frac{R_S + R_b}{r_S} \quad (5.21)$$

and the umbra cone angle α_u of complete shadow

$$\alpha_u = \frac{R_S - R_b}{r_S}. \quad (5.22)$$

¹NEP systems are an exception. They can operate without sunlight until depletion of their nuclear propellant.

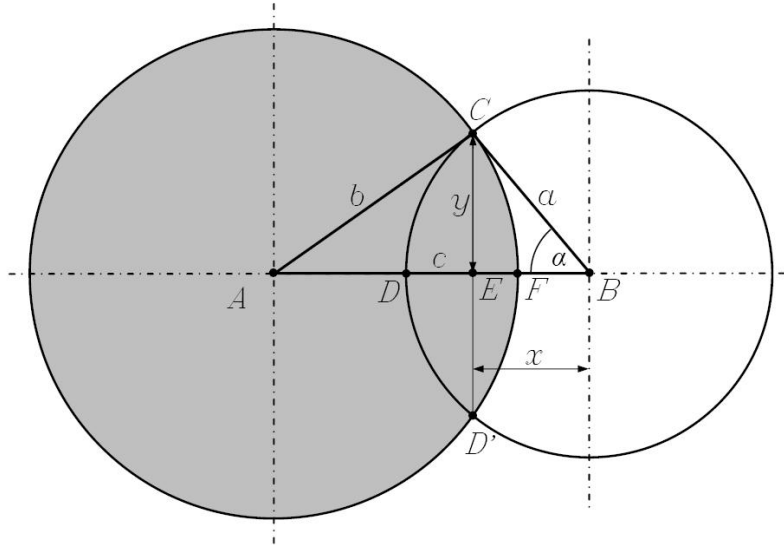


Figure 5.4: Occultation of the Sun Disc by a Spherical Body. The Sun is on the right hand side and the occulting body, e.g., the Earth, is on the left hand side. The distances a , b , and c are the apparent radius and separation of the centers of both bodies as seen from within the penumbra shadow cone. Courtesy Montenbruck and Gill [57, p. 82]

The vertical component

$$r_v = \sqrt{\mathbf{r}^\top \mathbf{r} - r_h^2} \quad (5.23)$$

of the position vector \mathbf{r} is required for the determination whether the spacecraft is in full sunlight or shadow. It may be obtained using the associated horizontal component

$$r_h = \frac{-\mathbf{r}_s^\top \mathbf{r}}{|\mathbf{r}_s|}. \quad (5.24)$$

With the shadow cone radii p_v and u_v at the spacecraft's horizontal distance r_h

$$p_v = R_b + r_h \tan \alpha_p \quad (5.25)$$

$$u_v = R_b - r_h \tan \alpha_u \quad (5.26)$$

a coarse determination of the spacecraft's illumination becomes feasible. The spacecraft is in umbra if $r_v \leq u_v$, it is outside umbra but still within penumbra if $u_v < r_v \leq p_v$, and it is in full sunlight if $p_v < r_v$.

For higher precision trajectory integration it is not sufficient to know if the spacecraft is in umbra or penumbra. Being in penumbra, one also needs to know how much sunlight is still available, i.e., the fraction of solar disc that is visible from the spacecraft's current location. InTrance uses the shadow function $f_s \in \mathbb{R} : 0 \leq f_s \leq 1$ from Montenbruck and Gill [57]

$$f_s = 1 - \frac{A}{\pi a^2} \quad (5.27)$$

to quantify the occultation of the apparent Sun disc area by another body, with the occulted Sun disc area A and the apparent Sun disc radius a as seen from the spacecraft's location. The shadow function f_s is zero for umbra, one for full sunlight, and between zero and one for penumbra conditions. According to Fig. 5.4, A is

$$A = a^2 \arccos\left(\frac{x}{a}\right) + b^2 \arccos\left(\frac{c-x}{b}\right) - cy, \quad (5.28)$$

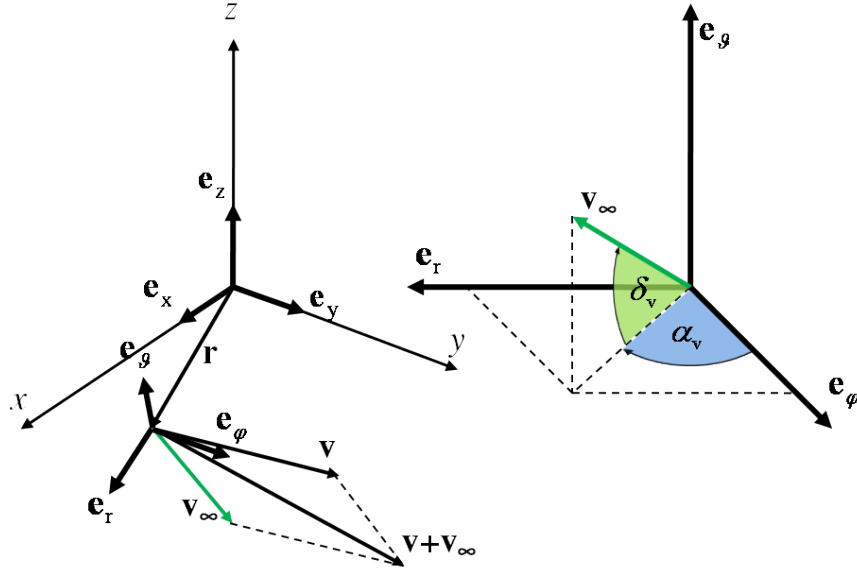


Figure 5.5: Launch System-provided Hyperbolic Excess Velocity v_{∞} . The direction of v_{∞} is described with the azimuth angle $\alpha_{v_{\infty}}$ and the declination angle $\delta_{v_{\infty}} \in (-\pi/2, \pi/2)$.

with the auxiliary variables

$$x = \frac{c^2 + a^2 - b^2}{2c} \quad (5.29)$$

and

$$y = \sqrt{a^2 - x^2}. \quad (5.30)$$

The apparent radius of the occulting body b , the apparent distance between this body and the Sun c , and the Sun's apparent radius a are calculated with R_s , R_b , the Sun's position relative to the spacecraft \mathbf{d} , and \mathbf{r}

$$a = \arcsin \frac{R_s}{d} \quad (5.31)$$

$$b = \arcsin \frac{R_b}{r} \quad (5.32)$$

$$c = \arccos \frac{-\mathbf{r}^T \mathbf{d}}{rd}. \quad (5.33)$$

The computation of f_s is included in the EOMs of InTrance and therefore part of every integration sub-step.

5.3.6 Launch Rocket Excess Energy

Launch rockets have large absolute thrust values and need no external oxidizer, which allows their operation also in vacuum. Practically all today's spacecraft are therefore brought from Earth into space via chemical launchers. A mission's target orbit or its trajectory initial conditions thereby determine the ΔV that a launcher model must provide at least to become a launch system candidate for that mission. For each potential launcher, ΔV translates to a maximum payload mass $m_{\text{PL}}^{\text{max}}$ that can be delivered to the specified target orbit. The criterion $m_{\text{SC}} \leq m_{\text{PL}}^{\text{max}}$ is the logical primary requirement for any potential launcher. However, other

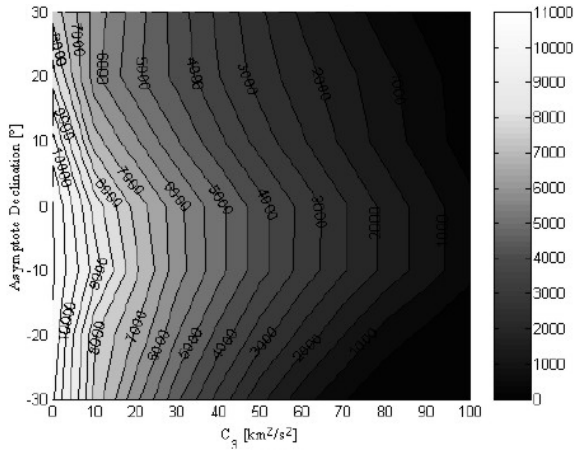


Figure 5.6: Ariane V ECB Launch Performance Diagram. It shows the dependence of the hyperbolic excess energy C_3 on the payload mass m_{PL} and on the angle δ_∞ between the ecliptic and the asymptote of the escape trajectory. Courtesy Ancarola [3].

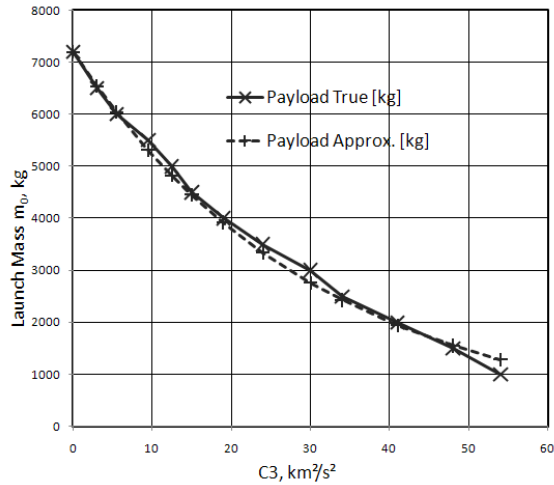


Figure 5.7: Approximation of Ariane V ECB's Launch Performance, $\delta_\infty = 0$. It shows the dependence of the hyperbolic excess energy C_3 on the spacecraft launch mass m_0 . The “true” values are taken from Fig. 5.6 [3]

criteria exist as well and influence the final decision. Exemplary criteria are cost, availability, launch opportunities, launch success rate, and political constraints.

Several factors contribute to a launcher's maximum payload mass m_{PL}^{\max} , such as the number of rocket stages and the I_{sp} of each stage. Only in rare occasions, a launcher's m_{PL}^{\max} matches m_{SC} for a given ΔV exactly. For most missions $m_{SC} < m_{PL}^{\max}(\Delta V)$, which offers trade space for the following three options. The first option is to minimize the surplus payload mass ($m_{PL}^{\max} - m_{SC}$), corresponding to choosing the smallest launcher that still allows mission execution. This option can potentially reduce launch cost.

A second option is to launch more than one spacecraft with a single launcher. So-called piggyback launches became popular with high-performance launch systems and when technology progress allows to build smaller spacecraft. It allows to share launch cost among the payload's owners and reduces individual mission's cost. Launching multiple spacecraft with a single rocket is however more complex. It often requires restartable upper stage engines, which has implications on the risk assessment of each participating mission.

The third option is to include the available, launcher-provided excess energy into mission design for an additional acceleration of the spacecraft and thus a potential reduction of necessary on-board propellant. This is promising especially for deep space probes, as it results in a higher hyperbolic excess velocity v_∞ w.r.t. the Earth, which may help to reduce the transfer time. The square of v_∞ is the specific hyperbolic excess energy¹

$$C_3 = v_\infty^2. \quad (5.34)$$

Figure 5.5 shows how v_∞ adds to the spacecraft's initial velocity \mathbf{v} . For example, for a launch from Earth onto a parabolic trajectory, i.e., $C_3 = 0$, \mathbf{v} would equal the Earth's heliocentric orbit velocity at the launch date t_0 . The direction of v_∞ is described in the polar reference frame with the azimuth angle α_∞ and the declination angle δ_∞ . Figure 5.6 shows the launch performance diagram of the Ariane V ECB for a launch into an escape orbit. Assuming an

¹Precisely, hyperbolic excess energy is a mass-normalized energy, i.e., its unit is km^2/s^2 .

escape trajectory declination angle $\delta_\infty = 0$, the dependence of v_∞ on m_{PL} can be approximated with

$$v_\infty(m_{\text{PL}}) = \sqrt{\frac{1}{\gamma} \ln \left(\frac{m_{\text{PL}}^{\text{max}}}{m_{\text{PL}}} \right)}, \quad (5.35)$$

with the maximum escape orbit payload mass $m_{\text{PL}}^{\text{max}} = 7\,200$ kg, the payload mass $m_{\text{PL}} = m_{\text{SC}}$, and the fit parameter $\gamma = 0.032$. Figure 5.7 shows the resulting fitting curve¹ and the original values given by Ancarola [3].

The exploitation of C_3 allows for an interesting optimization because any gain in v_∞ results in a reduction of m_{SC} and hence m_p ², and vice versa. Within the capabilities of the actual launcher and for a given mission design, a spacecraft design can therefore be optimized w.r.t. an overall optimization criterion, e.g., the total mission duration. However, it strongly depends on the actual mission if a high v_∞ is beneficial for the optimal fulfillment of the optimization criterion, or if it's better to use the spacecraft's propulsion system to generate the necessary ΔV .

5.3.7 Target State, Deviation, and Proximity

The term target state describes within the context of this work the state that the spacecraft should have attained at the end of a mission phase. In other words, a mission phase ends if the spacecraft's state coincides with the predefined target state. If the mission design comprises a single flight phase only, the target state of this phase equals the mission's target state. Different target state types maybe thought of, but flybys or rendezvous at celestial bodies are used most widely. Nevertheless, abstract target states are possible as well, e.g., achieving escape or capture conditions w.r.t. a specified body. For the description of the target states that are implemented in InTrance, the term target is therefore used universally. It can refer to a celestial body moving on its orbit, to a fixed point in space, or to an orbit described by an incomplete³ set of orbital elements.

5.3.7.1 Proximity

Prior assessment of the quality of a given candidate ξ , the simulation of a mission phase i integrates the state equations until matching a stop criterion. Exemplary criteria are the maximum integration time or the minimum Sun distance. Whatever criterion stopped the integration loop, the final state vector is unlikely the best one in terms of being close to the target state. Therefore a check for this "being close to the target" is necessary at each trajectory point. The simulation conducts this check through calculation of the proximity $P \in \mathbb{R}$ after each integration step, with P being the required measure for the distance between the spacecraft state⁴ and the target state. Depending on the phase-specific target state, P is determined through one or more so-called deviations δ_i

$$P \in \mathbb{R} : P(\delta_i) := \begin{cases} -\sqrt{\sum \delta_i^2} & , \text{ if } \exists \delta_i : \delta_i < 0 \\ \min \delta_i & , \text{ if } \forall \delta_i : \delta_i \geq 0 \end{cases} . \quad (5.36)$$

The definition of (5.36) shows that P is negative if at least one criterion is negative. If all $\delta_i \geq 0$, then P corresponds to the smallest positive δ_i .

¹Following a reformulation of (5.35) and using (5.34), the fitting curve gives the launch mass in dependency of C_3 .

²This is not applicable for solar sails.

³As six elements are necessary to fully describe a Keplerian orbit in space, "incomplete" means any set of less than six independent Keplerian or other orbital elements.

⁴"State" is in this context a general term and, depending on the respective target state, not constraint to the pure astrodynamic state.

5.3.7.2 Flyby

A successful flyby requires the distance Δr_f between the spacecraft position $\mathbf{r}_{\text{SC}}(t_f)$ and the target position $\mathbf{r}_{\text{T}}(t_f)$

$$\Delta r_f = |\mathbf{r}_{\text{SC}}(t_f) - \mathbf{r}_{\text{T}}(t_f)| \quad (5.37)$$

at the time of closest approach t_f to be smaller or equal than a predefined maximum distance Δr_{max} . For a time-independent target vector \mathbf{r}_{T} , Δr_f simplifies to $\Delta r_f = |\mathbf{r}_{\text{SC}}(t_f) - \mathbf{r}_{\text{T}}|$. The associated relative distance deviation δ_r is

$$\delta_r \in (-\infty, 1] : \delta_r := \frac{\Delta r_{\text{max}} - \Delta r_f}{\Delta r_{\text{max}}}. \quad (5.38)$$

Positive δ_r -values indicate achieved flyby conditions, with the maximum value one denoting a perfect match of $\mathbf{r}_{\text{SC}}(t_f)$ and \mathbf{r}_{T} . Consequently, δ_r is negative as long as $\Delta r_f > \Delta r_{\text{max}}$. The flyby proximity $P_{\text{FB}} = P(\delta_r)$ is a function of δ_r and calculated with (5.36).

5.3.7.3 Rendezvous

For a successful rendezvous with a target, apart from the distance criterion δ_r , the one of relative velocity δ_v must be fulfilled as well. This requires the absolute velocity difference

$$\Delta v_f = |\mathbf{v}_{\text{SC}}(t_f) - \mathbf{v}_{\text{T}}(t_f)| \quad (5.39)$$

between the spacecraft velocity $\mathbf{v}_{\text{SC}}(t_f)$ and the target state's velocity $\mathbf{v}_{\text{T}}(t_f)$ at a time t_f to be smaller than a problem-dependent upper limit Δv_{max} . For static targets, Δv_f reduces to $\Delta v_f = |\mathbf{v}_{\text{SC}}(t_f) - \mathbf{v}_{\text{T}}|$. The associated relative velocity deviation is

$$\delta_v \in (-\infty, 1] : \delta_v = \frac{\Delta v_{\text{max}} - \Delta v_f}{\Delta v_{\text{max}}}. \quad (5.40)$$

The rendezvous proximity $P_{\text{RV}} = P(\delta_r, \delta_v)$ is a function of δ_r and δ_v and calculated with (5.36).

5.3.7.4 Capture

A capture is in general the transition from an orbit about one celestial body to a closed orbit about another celestial body. The transfer from a heliocentric trajectory into a Mars-bound orbit is an example for such a capture. Within this work, however, not the transitions but the result, the final state, was defined as a capture. This means the capture of a spacecraft by the central body was successful if the spacecraft's orbit is closed, i.e., an ellipse, and resides completely within the influence region of that body. The capture proximity $P_{\text{CP}} = P(\delta_r, \delta_v)$ is defined with (5.36), using the distance deviation δ_r from (5.38) and the velocity deviation δ_v from (5.40); only the definitions of the differences Δr_f and Δv_f and the associated limits Δr_{max} and Δv_{max} differ from those in subsection 5.3.7.2 and subsection 5.3.7.3. The distance limit $\Delta r_{\text{max}} = \max(r_{\text{H}}, r_{\text{SOI}})$ is the maximum of the Hill Sphere radius and the radius of the SOI. Hamilton and Burns [38] give the Hill Sphere

$$r_{\text{H}} \approx a(1 - e) \left(\frac{m_{\text{CB}}}{3m} \right)^{1/3}, \quad (5.41)$$

with the body mass m , the central body mass m_{CB} , and the eccentricity e and semimajor axis a of the body's orbit. The radius of the SOI is [5]

$$r_{\text{SOI}} \approx r \left(\frac{m_{\text{CB}}}{m} \right)^{2/5}, \quad (5.42)$$

Table 5.2: SOI radii and Hill sphere radii of selected celestial bodies. The radii r_H and r_{SOI} of a body with the mass m derived from the body's semimajor axis a , the eccentricity e of the body's orbit, and the ratio $\frac{m_{CB}}{m}$ of the central body mass m_{CB} and m . The radii r_H and r_{SOI} are expressed in multiples of the body radius R_b . Note that r_H is smaller than r_{SOI} for the Moon; in all other cases $r_H > r_{SOI}$.

Body	R_b [km]	$\frac{m_{CB}}{m}$	a [10^6]	e [km]	r_{SOI} [R_b]	r_H [R_b]
Mercury	2 440	6 023 600	57.91	0.206	46	90
Venus	6 052	408 524	108.21	0.007	102	167
Earth	6 371	328 901	149.60	0.017	146	236
Moon	1 738	81.3	0.38	0.055	38	35
Mars	3 390	3 098 708	227.04	0.093	170	320
Jupiter	69 911	1 047	778.28	0.049	690	760
Saturn	58 232	3 498	1 427.39	0.056	937	1 120
Uranus	25 362	22 903	2 870.48	0.047	2 041	2 763
Neptun	24 622	19 412	4 498.34	0.009	3 520	4 713
Vesta	560	$\approx 7.45 \cdot 10^9$	353.35	0.089	71	224
Ceres	975	$\approx 2.13 \cdot 10^9$	413.94	0.080	79	229

with current relative distance r between the body and the central body. Table 5.2 lists both radii for selected celestial bodies. Depending on the eccentricity of the spacecraft's orbit w.r.t. the target body, the definition of the relative distance Δr_f is

$$\Delta r_f = \begin{cases} |\mathbf{r}_{SC}(t_f) - \mathbf{r}_T(t_f)| & , \text{if } e \geq 1 \\ r_a & , \text{if } e < 1 \end{cases} . \quad (5.43)$$

In the first case, (5.43) equals (5.37), and it is replaced with the apocenter distance r_a of the spacecraft's orbit w.r.t. the target body in the second case.

Contrary to the regular rendezvous velocity limit, the second limit

$$\Delta v_{\max} = \begin{cases} \sqrt{\frac{\mu}{\Delta r_{\max}}} & , \text{if } e \geq 1 \\ \sqrt{\frac{\mu}{r_a}} & , \text{if } e < 1 \end{cases} \quad (5.44)$$

and the associated difference

$$\Delta v_f = \begin{cases} |\mathbf{v}_{SC}(t_f) - \mathbf{v}_T(t_f)| & , \text{if } e \geq 1 \\ v_a = \sqrt{\mu \left(\frac{2}{r_a} - \frac{1}{a} \right)} & , \text{if } e < 1 \end{cases} \quad (5.45)$$

also depend on the eccentricity of the spacecraft's orbit w.r.t. the target body.

5.3.8 Dynamic Adaption of Accuracy Boundary Constraints

Successful evaluation of a candidate solution ξ_i results in a trajectory $\mathbf{r}[\bar{t}]$. This trajectory should preferably give a high objective function value. Before that, it must however comply to problem-dependent boundary or accuracy constraints. An example for such a constraint is the maximum spacecraft-target-distance Δr_{\max} in FB problems. For RV problems, the

constraint of a maximum relative velocity Δv_{\max} must additionally be fulfilled. Once $\mathbf{r}[t]$ satisfies all constraints, the optimization objective function determines the fitness $J(\xi_i)$.

Especially at the begin of an optimization run, an EA's search behavior benefits from using more relaxed accuracy constraint values than the final ones. That means to begin an optimization of an heliocentric flyby problem with a larger maximum flyby distance, e.g., $\Delta r_{\max} = 10^6$ km. Upon each constraint-matching solution, this distance is then successively reduced to a final value, e.g., $\Delta r_{\max} = 10^4$ km. This procedure is more promising for finding the optimal solution than using only the latter limit. The final solution should however match reasonable and representative accuracy constraints. In a FB scenario, a spacecraft's trajectory should therefore not end millions of kilometers away from the target, and for a successful rendezvous, Δv_{\max} should not be several kilometers per second. The values of Δr_{\max} and Δv_{\max} depend on the particular problem. In practice, it may therefore require a few trials to obtain robust settings.

To support both requirements, the relaxed initial and the final constraint values, InTrance implements dynamic boundary constraints. These constraints adapt themselves according to the ongoing optimization progress and controlled by user-specified rules. The algorithm for the adaption of both constraints is as follows.

The trigger for the alteration of these constraints is three successful EA epochs¹, i.e., three epochs that each resulted in a new best solution compliant to current Δr_{\max} and Δv_{\max} . Depending on the configuration, InTrance then calculates new limits for Δr_{\max} and Δv_{\max} with either an exponential or a linear law.

$$\Delta r_{\max, \text{new}} = c_{r,e} \cdot \Delta r_{\max, \text{old}} \quad (5.46)$$

$$\Delta v_{\max, \text{new}} = c_{v,e} \cdot \Delta v_{\max, \text{old}} \quad (5.47)$$

$$\Delta r_{\max, \text{new}} = \Delta r_{\max, \text{old}} - c_{r,l} \quad (5.48)$$

$$\Delta v_{\max, \text{new}} = \Delta v_{\max, \text{old}} - c_{v,l} \quad (5.49)$$

The factors $c_{r,e} \in (0, 1)$ and $c_{v,e} \in (0, 1)$ for the exponential decrease of the distance and relative velocity and the decrements $c_{r,l}$ and $c_{v,l}$ for the associated linear decrease must be provided by the user. The algorithm continues until the respective final value of each limit is met. Both limits have individual and independent configurations, i.e., one may be configured for exponential decrease and the other for linear reduction.

5.3.9 Multiphase Constraints

In addition to general constraints that apply to most space mission designs, e.g., an allowed minimum safety distance from the Sun, multiphase missions have a few specific constraints. Some of them result only from the fact that a mission has more than one flight phase. Others result from the approach of concurrent optimization of each flight leg and reconnecting them to a complete trajectory.

One of the constraints of the first type is the stay-time, or dwell time. It comes in effect during multi-RV missions and is defined with (4.4). Figure 5.8 shows the stay-time window over time and the three launch dates ${}^{(i)}t_{0,a}$, ${}^{(i)}t_{0,b}$, and ${}^{(i)}t_{0,c}$. Only the second date is between the earliest and the latest launch date. The other two conflict with the launch window that is defined through the preceding phase's arrival date ${}^{(i-1)}t_f$ and the stay-time window

¹Within InTrance, the number of successful epochs before adapting the accuracy limits is configurable. However, during this work three epochs turned out as robust setting. It allows InTrance to optimize w.r.t. the actual optimization criterion for a while before adapting the boundary constraints again.

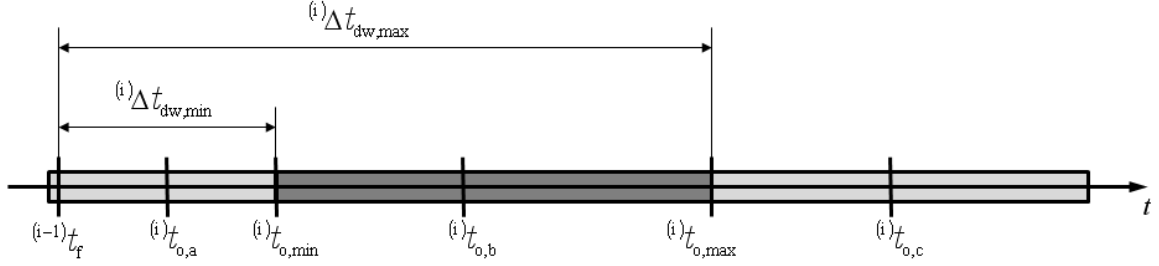


Figure 5.8: Stay-time Constraint and its Violation. The dark gray bar in the middle is the interval of valid launch dates for phase i . Launch dates within the light gray bars are either too early or too late and therefore invalid. The earliest allowed launch date of phase i $(i)t_{0,\min}$ and the corresponding latest allowed launch date $(i)t_{0,\max}$ are determined through the arrival date of the preceding phase $(i-1)t_f$ and the stay-time interval $(i)\Delta t_{dw,\min}$, $(i)\Delta t_{dw,\max}$. $(i)t_{0,a}$, $(i)t_{0,b}$, and $(i)t_{0,c}$ are example launch dates, whereas only the second is within the dark gray bar in the middle and thus a valid one; the other two dates $(i)t_{0,a}$ and $(i)t_{0,c}$ result in a stay-time constraint violation, expressed with scalar value V_{dw} .

$(i)\Delta t_{dw,\min}$, $(i)\Delta t_{dw,\max}$). The scalar violation $(i)V_{dw}$ counts the days by which a launch date is outside the specified launch date window

$$(i)V_{dw} = \begin{cases} (i)t_{0,\min} - (i)t_0 & , \text{ if } (i)t_0 < (i)t_{0,\min} \\ 0 & , \text{ if } (i)t_{0,\min} \leq (i)t_0 \leq (i)t_{0,\max} \\ (i)t_0 - (i)t_{0,\max} & , \text{ if } (i)t_{0,\max} < (i)t_0 \end{cases} . \quad (5.50)$$

Different to the stay-time constraint, the next constraint violations result from optimizing multiple mission phases concurrently. This is because the resulting trajectories are generally discontinuous w.r.t. the astrodynamical state, spacecraft mass, and time. The scalar state violation

$$(i)V_s = \left((i)\mathbf{x}_0 - (i-1)\mathbf{x}_f \right)^2 \quad (5.51)$$

measures the difference between the states of the spacecraft at the end of a predecessor phase $(i-1)$ and the begin of a successor phase i . The indices f and 0 denote the arrival or final state and the launch or initial state of the spacecraft in the respective phase. Violation of potentially required mass continuity is expressed with the mass violation

$$(i)V_m = \left((i)m_{SC,0} - (i-1)m_{SC,f} \right)^2 - (i)\Delta_m^2. \quad (5.52)$$

The mass $(i)\Delta_m$ accounts for any intended mass change, e.g., a jettisoned propulsion stage, collected sample material, or docked-on other spacecraft.

The requirement of steady and smooth trajectories also demands a continuous time history along the entire mission trajectory. This is automatically supported with the fulfillment of the stay-time constraint $(i)V_{dw}$ if $(i)\Delta t_{dw,\min} = (i)\Delta t_{dw,\max} = 0$.

5.3.10 Dynamic Control Step Size

A spacecraft with a propulsion system can change its orbit by applying thrust. To do this efficiently, the thrust vector must be chosen optimally, i.e., its direction and magnitude must change the respective orbital element with maximum effect. For low-thrust transfers, the adaption of the thrust vector must be performed repeatedly along the trajectory, for some

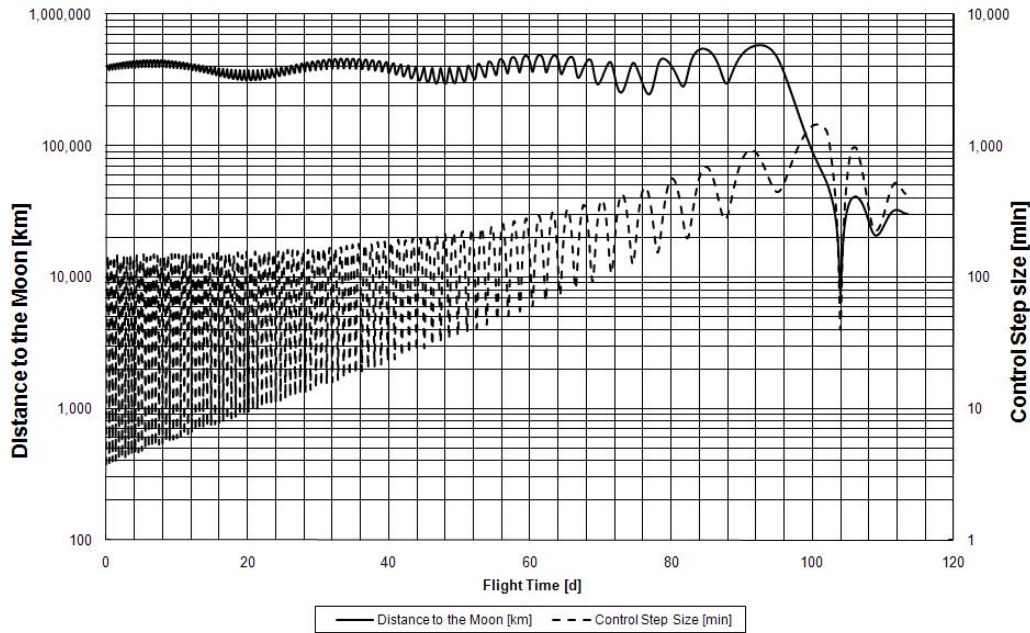


Figure 5.9: Control Step Size and Moon Distance over Flight Time.

transfers even continuously throughout the entire transfer. For the sake of an easier integration or because an optimization method may only handle uniform step sizes, low-thrust trajectories are often discretized with constant steering step sizes. This step size can also be associated with a control step size because the control variables can be altered only after each such step. The problem-dependent decision about what control step size to use is often taken by astrodynamics experts. For example, for a transfer from the Earth to Mars it can be sufficient to change the acceleration vector daily, resulting in a step size of one day. For a high-elliptical heliocentric orbit, e.g., one with a pericenter at Mercury’s orbit and a pericenter at Mars orbit distance, this may be unfavorable. For efficient control, a step size of one day may be too large near the orbit’s pericenter and unnecessary small when the spacecraft is near the apocenter of its orbit. Consequently, if the control step size needs to be constant, one has to determine the smallest required time step. For low-complexity transfers it may be possible to estimate a lower boundary for this parameter. However, for many low-thrust transfers suitable steering strategies are completely unknown beforehand. This complicates an a priori determination of a fixed control step size. It is therefore better to determine the control step size dependent on the respective astrodynamics problem. That means to reduce the control step size if the rate of change of the orbital elements increases and vice versa. The true anomaly ν was chosen as the step size-controlling orbital element for InTrance. Its rate of change is the orbital angular velocity $\dot{\nu}$ and is calculated from the spacecraft’s current linear velocity \mathbf{v} and the distance from the central body \mathbf{r} [87]

$$\dot{\nu} = \frac{|\mathbf{r} \times \mathbf{v}|}{r^2}. \quad (5.53)$$

This algorithm is called dynamic step size control (DSSC). The control step size Δt_s is calculated with

$$\Delta t_{s^*} = \frac{\nu_{max}}{\dot{\nu}}. \quad (5.54)$$

$$\Delta t_s = \begin{cases} \Delta t_{s-} & \text{if } \Delta t_{s^*} < \Delta t_{s-}, \\ \Delta t_{s^*} & \text{if } \Delta t_{s-} \leq \Delta t_{s^*} \leq \Delta t_{s+}, \\ \Delta t_{s+} & \text{if } \Delta t_{s^*} > \Delta t_{s+}. \end{cases} \quad (5.55)$$

The angle ν_{max} is the maximum angle that the spacecraft is allowed to traverse through during a control step. The upper and lower boundary of Δt_s is Δt_{s-} and Δt_{s+} . Control step size adaption according to the shown algorithm is not constrained to the central body alone but also includes any disturbance body.

Figure 5.9 shows the effect of DSSC on the control step size for a transfer from an Earth-bound orbit into an orbit around the Moon. During the transfer with many revolutions around the Earth, the control step size changes by several magnitudes as the spacecraft recedes the Earth and reaches the Moon's SOI. The transfer starts from a high-elliptical orbit, e.g., a geostationary transfer orbit, and the control step sizes vary between three minutes around pericenter and two and a half hours around apocenter. Then the apocenter and pericenter distances increase. Consequently, also the respective control step sizes are adapted to higher values. As soon as the gravitational pull of the Moon becomes dominating, that attraction becomes the main driver for step size control. This occurs at a flight time of approximately 100 days. The spacecraft attains a bound orbit about the Moon and the orbit period drops significantly. This is also reflected in the visible drop of the Moon distance, which coincidences with the drop in the control step size.

Values of one to ten degrees have turned out to be reasonable values for ν_{max} , depending on the desired control resolution.

5.3.11 Dynamic Parameter Encoding

EAs robustly find the global optimum if the problem is properly encoded, i.e., in an adequate form for the EA, but they lack the precision that local optimization can achieve. Schraudolph and Belew [81] developed a dynamic parameter encoding (DPE) method to alleviate this problem. DPE increases the precision of the solution but is prone to premature convergence if the global optimal falls outside the new, smaller parameter range. The implementation of DPE in InTrance therefore does not only shrink the parameter range but also shifts its center if the current optimum parameter value is close to the respective lower or upper parameter limit. This allows the EA to move away again from a local optimum if another, better optimum is in the vicinity.

DPE within InTrance applies to initial propellant masses and launch dates, as these two are primary initial conditions for most trajectory optimizations. The range of initial propellant mass Δm_p was calculated with (5.56) [81] from the current best propellant mass m_p , the coefficient $c = 1.6$ kg, and $b \in \mathbb{N}$ and $b \leq \frac{\ln m_p}{\ln c}$

$$\Delta m_p = \text{int} \left(\frac{c^{b-1}}{4} \right). \quad (5.56)$$

The lower and upper boundary of m_p are accordingly $\underline{m}_p = m_p - \frac{1}{2}\Delta m_p$ and $\overline{m}_p = m_p + \frac{1}{2}\Delta m_p$. The second parameter using DPE is the launch date t_o . At the begin of an EA optimization, the range of that simulation parameter is the entire user-provided launch window. Later usage of smaller parameter ranges, when the EA has settled in the vicinity of the final solution, could however improve local search behavior. Furthermore, it is beneficial if the respective limit is formulated in an astrodynamic context instead of time units. InTrance defines the

launch window in degrees of the launch body orbit that the initial body, e.g., the Earth, is allowed to traverse through during the specified time span

$$\Delta t_0 = \frac{\Delta \nu_{\max}}{\dot{\nu}(t_0)}. \quad (5.57)$$

In other words, current orbital velocity at t_0 of the celestial body the spacecraft starts from and the user-provided launch window angle determine the time window. For high-elliptic orbits of that body, the launch window therefore broadens when getting closer to the apocenter. Close to the pericenter of that body, where the rate of change in true anomaly is bigger, the launch window consequently shrinks.

5.4 Evolutionary Algorithm

Every implementation of an EA has its specific peculiarities. This section is about the EA implemented in InTrance. During the course of this work, the source code of InTrance's original EA was revised and extended by new features. The following description covers the essential elements and principles that are necessary for its understanding. Subsection 5.4.1 describes the encoding of an individual. The dynamic control of the current search space hypercube is covered in subsection 5.4.2. Subsection 5.4.3 explains how the new implementation of a mechanism called Search Space Scans exploits niche evolution to obtain the initial partial solution for the following further optimization with the EA. The basic EA elements of selection, crossover, mutation, and evaluation are content of subsection 5.4.4 until subsection 5.4.6. Subsection 5.4.7 details on the calculation of fitness values, and the final subsection 5.4.8 details the encoding of the the additional simulation parameters on the chromosome.

5.4.1 Real Delta Coding

Whitely et al. [98], [97] proposed **delta coding** (DC) as iterative search strategy to increase precision and convergence behavior of genetic search with binary strings. Rather than encoding the complete solution, a DC-chromosome encodes the distance δ to another solution \mathbf{h} , called partial solution or interim solution. In this context, the best solution found so far ξ^* is used for \mathbf{h} . A population individual ξ_i thus comprises a \mathbf{h} -chromosome and a δ -chromosome

$$\xi_i = \mathbf{h} + \delta_i \Leftrightarrow \langle b_1, \dots, b_i \rangle_i = \langle h_1, \dots, h_i \rangle + \langle \delta_1, \dots, \delta_i \rangle_i. \quad (5.58)$$

DC thus limits the EA to effectively explore only a dynamically selected subspace around \mathbf{h} and not the entire search space.

At the begin of the algorithm, no partial solution exists, i.e., all $b_i = 0$. A first \mathbf{h} may however be obtained through a regular GA-run. After convergence of this randomly-initialized initial GA-population, the resulting best δ is the initial partial solution¹. From then on, new populations are always initialized with the best found solution so far. If the best population individual is better than the previous solution, it will become the new partial solution. DC avoids premature convergence by periodic reinitialization and also controls the size of the current binary hypercube. A reduction mechanism shrinks the hypercube for promising search space regions, and an expansion mechanism increases the hypercube again to explore also previously not searched regions. The algorithm continues until either achieving of user-defined stop criteria or exceeding the maximum allowed number of unsuccessful trials. Further details and a comparison of DC against a standard GA and a hill-climbing algorithm on a set of GA test functions can be found in [54].

¹For this first GA-run, \mathbf{h} of all individuals is set to null, i.e., $\xi_i = \delta_i$.

Tsinas and Dachwald [88] extended binary DC to floating-point delta coding (FPDC), and Dachwald further revised FPDC to real delta coding (RDC) [13]

$$\langle r_1, \dots, r_i \rangle_i = \langle h_1, \dots, h_i \rangle_i + \langle \delta_1, \dots, \delta_i \rangle_i.$$

RDC runs in cycles called epochs e_i , during which the GA explores a dynamically selected parameter subspace, the hypercube H , around the most recent solution. The hypercube of the first epoch e_0

$$H_0 = [-\delta_{\max}(e_0), \delta_{\max}(e_0)]^1 \in \mathbb{R}^1$$

is constructed around the null solution $\mathbf{h}(e_0) = \mathbf{0}$. The population of the first time step t_0 of that epoch e_0 is randomly initialized

$$\Xi^{t_0}(e_0) = (\delta_1^{t_0}(e_0), \dots, \delta_q^{t_0}(e_0)). \quad (5.59)$$

The EA runs until convergence of that epoch's population, which has occurred at the so-called convergence time step $t = t_c$ if the relative fitness improvement over the last ν time steps was less than a preset limit ϵ

$$\frac{J(\xi_1^{t_c}) - J(\xi_1^{t_c - \nu})}{|J(\xi_1^{t_c - \nu})|} < \epsilon. \quad (5.60)$$

The best solution $\xi_1^{t_c}(e_0)$ found for this epoch becomes the partial solution $\mathbf{h}(e_1)$ of the subsequent EA search epoch e_1 , whose new delta-value range is

$$\delta_{\max}(e_1) = \kappa \cdot \delta_{\max}(e_0), \quad (5.61)$$

with the user-defined parameter $0 < \kappa < 1$. Initialization of the new population $\Xi^{t_0}(e_1)$ takes place with random values within the updated search space hypercube

$$H_1 = [h_1(e_1) - \delta_{\max}(e_1), h_1(e_1) + \delta_{\max}(e_1)] \times \dots \times [h_i(e_1) - \delta_{\max}(e_1), h_i(e_1) + \delta_{\max}(e_1)] \subset \mathbb{R}^1.$$

Subsection 5.4.2 details on how the $\delta_{\max}(e_i)$ is controlled depending on optimization progress and current simulation parameters.

The algorithm continues until achievement of RDC convergence, i.e., until the relative improvement between two consecutive epochs is smaller than a preset limit ϵ

$$\frac{J(\mathbf{h}(e_{i+1})) - J(\mathbf{h}(e_i))}{|J(\mathbf{h}(e_i))|} < \epsilon. \quad (5.62)$$

For an exhaustive search of the parameter space, small values for ϵ and large values for ν should be chosen, but this also strongly depends on the actual problem and the fitness function.

Dachwald [13] further introduced two mechanisms to improve robustness of the optimization process on limited available calculation time. The first is a mechanism called **Search Space Scan** (SSS), and the second is the shifting of the search space hypercube. As the original implementation of SSS was modified during this work, its description is content of the separate subsection 5.4.3. Shifting the effective search space is done through the usage of the best solution $\xi_1^{t_c}(e_{i-1})$ of the previous epoch for initialization of the new epoch population, regardless if e_{i-1} resulted in a new best solution ξ^* or not. This allows the EA to search also parameter subspaces more distant to ξ^* . If two, subsequent applications of this mechanism do not result in a new best solution, the stored ξ^* is used for initialization of the next epoch with a new, usually smaller, hypercube size.

5.4.2 Hypercube Size Control

RDC assured convergence of the optimization algorithm by using (5.61) with a constant hypercube shrink factor $\kappa < 1$. The shrink factor reduces the hypercube size H after every successful epoch. In order to improve local search behavior within the preset, fixed parameter ranges, the hypercube size is also reduced after two consecutive unsuccessful epochs. DPE dynamically modifies the parameter boundaries according to the current best solution and the configured range of the respective parameter. However, it can become necessary to choose such ranges quite narrow, e.g., only one degree for an angular parameter, and to let DPE shift that range if the current optimal parameter value is close to the chosen boundaries. Without also dynamically altering H , RDC could make the EA get stuck in a local optimum after such a range adaption, because of a then eventually too small parameter search space to leave a local optimum. The existing implementation of RDC in InTrance was therefore extended with an expansion mechanism. Figure 5.10 shows the resulting control logic. The dynamic

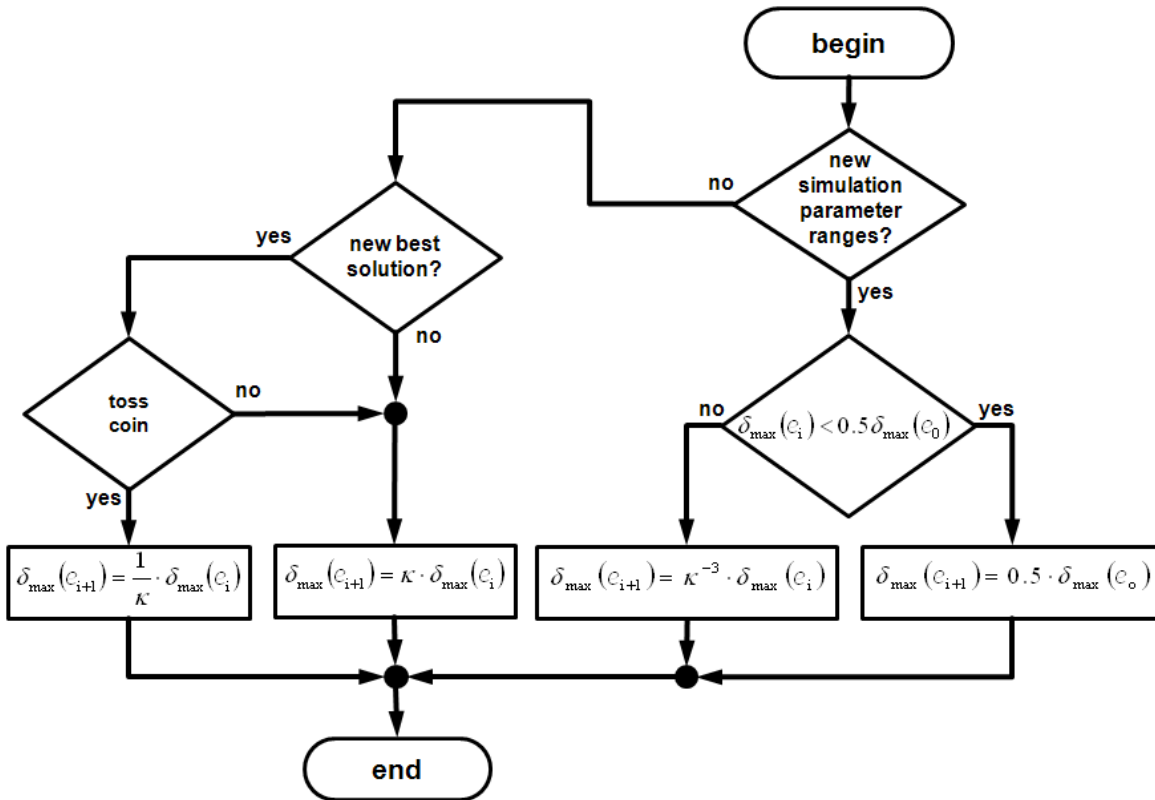


Figure 5.10: EA Parameter Hypercube Control

hypercube size control increases H on any change of the range of simulation parameters that are subject to optimization. Such parameters are the launch date window, the initial propellant mass range, and the initial state vector. Being also a change of the simulation environment, a modified boundary constraint upon a new best solution triggers an increase of H as well. The hypercube is thereby increased to half of $\delta(e_0)$ if $\delta(e_i)$ was smaller than that limit. As Fig. 5.10 shows, if $\delta(e_i) \geq \delta(e_0)$, then H is increased through a multiplication with κ^{-3} . This enables the EA to keep H at the initial size as long as possible and also to increase H again, if necessary.

Furthermore, it turned out beneficial to the fidelity of the final solution, to increase δ with a probability of 50 percent also upon a successful epoch. At the expense of increased runtime, that way the hypercube shrinks slower after having found the new best solution while still

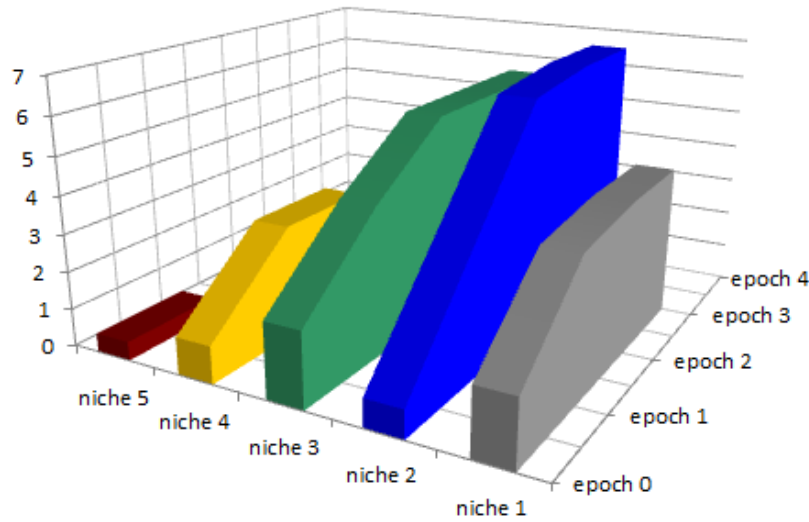


Figure 5.11: The Evolution of the Initial Individuals' Fitness During Search Space Scan. A niche is the solution subspace in the vicinity of an initial individual. Here, the number of niches is five. An EA run for each initial individual potentially gives a new individual with a higher fitness value. If that is the case, the new individual replaces the one that was used for initialization of the EA population. The first epoch ends after five EA runs with the removal of individual of niche five, which has the lowest fitness. The algorithm stops if only one individual is left, which is the one of niche two in this example. This solution is the starting point for the subsequent optimization run.

assuring convergence. The algorithm ends when consecutive epochs without triggering a simulation environment change have reduced $\delta(e_i)$ below a preset lower boundary.

5.4.3 Search Space Scan

InTrance features two options to start an optimization run. The first option is called warmstart. Any optimization thereby starts from an user-provided solution, which mostly stems from another InTrance run. To enable quicker evaluation of the EA individuals, such a previous optimization could use a less-fidelity simulation. Successive InTrance warmstart-initialized runs would then be carried out, each with a more complex simulation environment than the previous one. Third body disturbance, nonuniform gravity fields, or a smaller control step size are elements and control variables which can thus be activated step by step.

The second option is called coldstart and requires no externally provided solution. Instead, InTrance generates the needed initialization solution prior the actual optimization via a heuristic parameter search called Search Space Scan. This algorithm uses niching and concurrent evolution or co-evolution to explore the parameter space to find a suitable starting point. Figure 5.11 shows a generic example of the best fitness values of five niches during SSS. At first, a number of randomly initialized individuals is evaluated and ordered according to their fitness. The lowest line connects the fitness values of the first epoch's individuals at the begin of the SSS algorithm. Each of these individuals initializes a population which are then exposed to the EA epoch until convergence with a hypercube size according to the algorithm in subsection 5.4.2. If the fitness of the individual resulting from an epoch is better than the one that initialized that epoch, then it replaces the old one in the list of individuals. After all individuals were subject to an optimization epoch, they, or their better offspring individuals,

are ordered again w.r.t. to their fitness and the one with the worst fitness is removed. This procedure repeats until the list of individuals has reduced to a single individual, which is the result of the SSS and the starting point for the subsequent optimization run.

5.4.4 Selection

The two mutually-influencing mechanisms of selective pressure (SP) and population diversity effect the performance of evolutionary search methods. The SP of a selection method is the expected number of individuals of a new population Ξ^{t+1} that descend from the best individual ξ_1^t of the previous population Ξ^t . Population diversity is the number of different genome material in a population. Increasing the SP reduces the population diversity and vice versa. Some selection methods, e.g., fitness proportional selection, calculate the probability $p(\xi_j)$ of an individual for being selected for reproduction from its fitness $J(\xi_j)$ in relation to the other population members' fitness values. The resulting SP for this selection scheme is

$$SP^t = p_1^t \cdot q = \frac{J(\xi_1^t)}{\sum_{i=1}^q J(\xi_i^t)} \cdot q = \frac{J(\xi_1^t)}{J_{avg}^t}, \quad (5.63)$$

with the population dimension q . A high SP for a few superindividuals can be the consequence and a low SP for the majority of individuals. Superindividuals are population members with much higher fitness values than the average population fitness. A potential result is premature convergence to local optima because individuals dominated by superindividuals have little chances to contribute to the genome pools of new generations. Carefully chosen, problem-dependent fitness functions or fitness scaling methods can alleviate this problem. They are however not recommended as both approaches refer the decision on if and when to use them to the user. This is undesirable for an intended robust method.

Other selection methods use the individuals' relative fitness instead of their absolute fitness for the calculation of the corresponding selection probabilities. The relative fitness is the rank of the individuals, i.e., their position if they were arranged in ascending order according to their fitness values, with the best individual being at first position. Such selection methods do not need fitness scaling, as they do not suffer from potentially existing superindividuals and premature convergence. Their drawback is however their computationally cost. A computationally more efficient selection method, called tournament selection, was therefore chosen for InTrance. It creates a group of $\mu \geq 1$ randomly chosen Ξ^t -individuals and copies the best individual of that group into the new population Ξ^{t+1} . Corresponding to the population size, this process repeats q times to fill Ξ^{t+1} . InTrance implements tournament selection with the commonly used tournament size $\mu = 2$, which is also named binary tournament selection. The left part of Figure 5.12 shows the principle. Selection probabilities of the individuals for tournament selection are [13]

$$p_i = \frac{(\mu - i - 1)^\mu - (q - i)^\mu}{q^\mu}, \quad (5.64)$$

which gives the best individuals selection probability for $\mu = 2$

$$p_1 = \frac{2q - 1}{q^2}. \quad (5.65)$$

Using (5.65) in (5.63) gives the corresponding SP

$$SP_{\mu=2} = p_1 \cdot q = \frac{2q - 1}{q}. \quad (5.66)$$

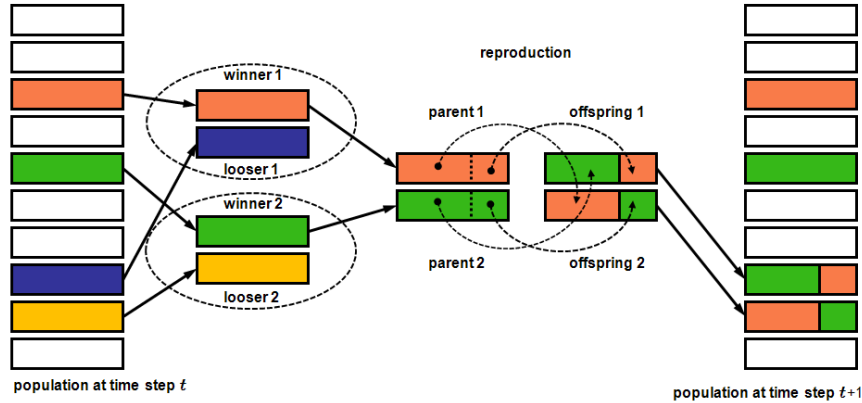


Figure 5.12: One-at-a-time Reproduction with Tournament Selection. This is a modified version of the figure from [13] and shows how two binary tournaments determine two winners and two losers out of four randomly picked individuals. The winner individuals recombine their genome material to create two new offspring, which then replace the individuals that lost the tournament in the new population.

The SP of tournament selection is constant for the entire EA run, if $q = \text{const.}$, which approaches 2 for $q \rightarrow \infty$. The best individual of a large population therefore contributes on average and independent on its absolute fitness value to two individuals to a new population. Another advantage of tournament selection is its implicit support for multiobjective optimization. Using a different optimization criterion for each tournament, this selection scheme prefers individuals that incorporate desired compromises between the respective criteria.

InTrance uses a different reproduction mechanism than traditional EA implementations. Its one-at-a-time reproduction or steady-state reproduction mechanism lets reproduction happen at each time step, which is computationally more efficient than generational reproduction and also combines well with tournament reproduction.

5.4.5 Crossover

InTrance implements the four crossover types one-point crossover, uniform crossover, arithmetic crossover, and loci crossover. The actual operator is chosen randomly with an initial probability $p = 0.25$ and applied to the parent individual's chromosomes, which potentially contain an ANN for each mission phase and the allele holding the transition conditions. Similar to node crossover, which assures that ANN node-related data is kept together during crossover, a specialized operator now keeps together the data fragments that belong to a mission phase.

The crossover probability is constant at the begin of the algorithm. Later, after > 100 successful reproductions, i.e., reproductions that resulted in a new best solution, crossover operator probabilities are determined dynamically by InTrance. Therefore a counter for each crossover operator keeps track of the number of successful reproductions after application of that operator. Once the sum of all operator counters exceeds 100, the ratio of an operator's counter value and the sum of all counters is taken as probability. This dynamic adaption of EA control parameters helps to adapt to the respective problem and reduces an EA's customization to a particular application or problem type, which in this case is the optimization of low-thrust trajectories.

5.4.6 Mutation

According to Whitely, mutation is not necessary when using DC, as populations are regularly reinitialized. Dachwald [13] found this unfavorable for real-value strings because it may lead to premature convergence within an epoch in case of small population sizes. InTrance therefore implements the so-called fast uniform mutation, a mutation scheme based on uniform mutation. Uniform mutation determines for each locus of the chromosome whether the allele at that locus should be modified or not. The typical mutation probability of $p_m \approx 10^{-3}$ is very small, resulting in the generation of many random numbers for potentially only a few mutation operations per chromosome. To prevent these time-consuming calculations, InTrance's fast mutation operator applies a random change with a probability of $0 \leq p_m \leq 1$ only to a single allele at a randomly chosen locus. The decision on the application of mutation is therefore made for the entire chromosome and not for each locus. If the decision is to mutate the string ξ_i , the operator picks a random locus j and replaces allele δ_{ij} with a new $\delta_{ij} \in [-\delta_{\max}, +\delta_{\max}]$.

5.4.7 Fitness in Multiphase Problems

The evaluation of a fitness function gives a scalar fitness J , which drives the optimization process of the EA by fulfilling multiple tasks. It expresses the conformity to problem dependent constraints or, equivalently, the violation of one or more constraints. Beyond that, J guides the EA search towards the optimal problem solution.

Multiphase missions introduced new constraints to the simulation component of InTrance. This necessitated a revision of the existing fitness function implementation. Basis for the calculation of J is an intermediate fitness J_P

$$J_P = \begin{cases} -\sqrt{\frac{\sum_{i=1}^{n_P} \min(0, P_i)^2}{n_P}} & , \text{if } \exists i \in (0, n_P) : P_i < 0 \\ \min(P_i) & , \text{if } \forall i \in (0, n_P) : P_i \geq 0 \end{cases} \quad (5.67)$$

It is an expression for the degree to which the trajectory fulfills the respective phases' constraints. For each phase, any of the constraints expressed with the proximity variables defined in subsection 5.3.7 may be used. If all proximity variables $P_i \geq 0$, then all constraints of the phases i have been achieved and $J_P \geq 0$. Once this has happened, $J(\xi_i)$ is primarily determined with the objective function O . Table 5.3 gives the objective functions implemented in InTrance. Note that the O_i are subject of maximization, regardless whether the underlying base variable should be minimized or maximized. The final fitness is

$$J(\xi) = \begin{cases} J_P & , \text{if } J_P < 0 \\ O \cdot \prod_{i \in (\text{dw}, \text{m}, \text{s})} \frac{1}{1+V_i} & , \text{if } J_P \geq 0 \end{cases} \quad (5.68)$$

Once the phase-specific constraints have been achieved, any fitness improvement results from an increased value of the respective objective function O^1 or from a reduction of the transition condition violations V_i defined in subsection 5.3.9, or both.

¹The objective functions O of Tab. 5.3 are mission specific and not phase specific. They cannot be used concurrently.

Table 5.3: InTrance objective functions. Using the variables that are available to the simulation, other objective functions may be implemented in the future.

No.	Symbol	Definition	Description
1	$O_{\Delta t, \min}$	$1 - \frac{\sum_{i=0}^{n_p} \Delta t_i}{\sum_{i=0}^{n_p} \Delta t_i}$	transfer time minimization
2	O_{m_p}	$\frac{1}{(0)m_{p,0} - (n_p)m_{p,f}}$	minimum propellant consumption
3	$O_{m_{SC} \Delta t}$	$\frac{(n_p)m_{SC}}{\Delta t}$	maximum final mass over transfer time
4	$O_{m_{SC} \Delta t_m}$	$\frac{(n_p)m_{SC}}{(n_p)t_f - (0)t_0}$	maximum final mass over mission duration
5	$O_{\Delta t_{dw}}$	$\frac{\sum \Delta t_{dw,i}}{\sum \Delta t_{dw,i}}$	maximum stay-time
6	$O_{\Delta t_m}$	$\frac{\Delta t_m - \Delta t_m}{\Delta t_m}$	minimum mission duration
7	$O_{t_0, \min}$	$\frac{(0)t_0 - (0)t_0}{(0)\bar{t}_0 - (0)t_0}$	earliest launch date
8	$O_{t_0, \max}$	$\frac{(0)t_0 - (0)t_0}{(0)\bar{t}_0 - (0)t_0}$	latest launch date

5.4.8 Encoding of Variable Initial Conditions on a Chromosome

An EA chromosome ξ_i in InTrance contains more than the parameter sets $^{(i)}\boldsymbol{\pi}$ that determine the different mission phases' NCs. It also holds the simulation parameters that should be optimized and which either affect spacecraft design or mission design. Of the first parameter type are the launch propellant mass $m_{p,0}$ and the characteristic power output of the electric power generation subsystem $P_{e,0}$. Mission-design-influencing parameters are the launch date t_0 , the hyperbolic excess velocity vector \mathbf{v}_∞ at t_0 , and the initial state vector $\mathbf{x}_{SC}(t_0)$. There are however a few peculiarities to each of these parameters when it comes to choosing a suited encoding scheme for storing them on ξ_i . In general, a parameter is bound to a range of values determined either by the user or by InTrance. The begin of the trajectory, for example, must take place in a launch date window $\underline{t}_0 \leq t_0 \leq \bar{t}_0$. A few parameters additionally exhibit discontinuities, e.g., angular parameters at 0 or π , which must be taken care of. Other parameters offer more than one possible representation; \mathbf{v}_∞ is such a parameter that can be expressed with cartesian or polar coordinates, for example. The same holds true for $\mathbf{x}_{SC}(t_0)$, and the representations chosen for InTrance are $\mathbf{v}_\infty = (v_\infty, \alpha_\infty, \delta_\infty)^\top$ and $\mathbf{x}_{SC}(t_0) = (r, \varphi, \vartheta, v, \zeta, \Phi)^\top$.

Table 5.4 provides the loci of the chromosome allele that determine the respective parameter. These loci are relative to the begin of a parameter block that follows directly on the $\boldsymbol{\pi}$ -parameters of that mission phase. They are described in the sequel.

A linear mapping of the first allele onto a range $(\underline{t}_0, \bar{t}_0)$ defined by the minimum and maximum launch dates gives the actual launch date

$$t_0 = \underline{t}_0 + \frac{r_0 - r}{\bar{r} - r} (\bar{t}_0 - \underline{t}_0). \quad (5.69)$$

The lower and upper boundary of allele values at all loci are \underline{r} and \bar{r} . Subsection 5.3.6 introduced the hyperbolic excess velocity vector \mathbf{v}_∞ . The definition of this launcher-provided velocity increment in the \mathcal{P} -frame is

$$\mathbf{v}_\infty = v_\infty \begin{pmatrix} \cos \alpha_\infty \cos \delta_\infty \\ \sin \alpha_\infty \cos \delta_\infty \\ \sin \delta_\infty \end{pmatrix}. \quad (5.70)$$

Table 5.4: Simulation parameter encoding.

Relative locus	Allele	Associated variable	Affected initial condition
0	r_0	t_0	t_0
1	r_1	v_∞	\mathbf{v}_∞
2	r_2	α_∞	\mathbf{v}_∞
3	r_3	δ_∞	\mathbf{v}_∞
4	r_4	$m_{p,0}$	$m_{p,0}$
5	r_5	$P_{e,0}$	$P_{e,0}$
6	r_6	r, v	$\mathbf{x}_{SC}(t_0)$
7	r_7	r, v	$\mathbf{x}_{SC}(t_0)$
8	r_8	r, v	$\mathbf{x}_{SC}(t_0)$
9	r_9	φ, ϑ	$\mathbf{x}_{SC}(t_0)$
10	r_{10}	φ, ϑ	$\mathbf{x}_{SC}(t_0)$
11	r_{11}	φ, ϑ	$\mathbf{x}_{SC}(t_0)$
12	r_{12}	ζ, Φ	$\mathbf{x}_{SC}(t_0)$
13	r_{13}	ζ, Φ	$\mathbf{x}_{SC}(t_0)$
14	r_{14}	ζ, Φ	$\mathbf{x}_{SC}(t_0)$

The mappings of the required three components v_∞ , α_∞ , and δ_∞ are linear as well

$$v_\infty = \underline{v}_\infty + \frac{r_1 - \underline{r}}{\bar{r} - \underline{r}} (\bar{v}_\infty - \underline{v}_\infty) \quad (5.71)$$

$$\alpha_\infty = \underline{\alpha}_\infty + \frac{r_2 - \underline{r}}{\bar{r} - \underline{r}} (\bar{\alpha}_\infty - \underline{\alpha}_\infty) \quad (5.72)$$

$$\delta_\infty = \underline{\delta}_\infty + \frac{r_3 - \underline{r}}{\bar{r} - \underline{r}} (\bar{\delta}_\infty - \underline{\delta}_\infty). \quad (5.73)$$

A linear mapping was also applied for the initial propellant mass

$$m_{p,0} = \underline{m}_{p,0} + \frac{r_4 - \underline{r}}{\bar{r} - \underline{r}} (\bar{m}_{p,0} - \underline{m}_{p,0}) \quad (5.74)$$

and the characteristic electrical power

$$P_{e,0} = \underline{P}_{e,0} + \frac{r_5 - \underline{r}}{\bar{r} - \underline{r}} (\bar{P}_{e,0} - \underline{P}_{e,0}). \quad (5.75)$$

The remaining alleles $r_6 - r_{14}$ are first interpreted as cartesian vectors $\mathbf{r}_{6/7/8} = (r_6 \ r_7 \ r_8)^\top$, $\mathbf{r}_{9/10/11} = (r_9 \ r_{10} \ r_{11})^\top$, and $\mathbf{r}_{12/13/14} = (r_{12} \ r_{13} \ r_{14})^\top$. From the equivalent polar representation

$$\begin{pmatrix} r_6 \\ r_7 \\ r_8 \end{pmatrix} = r_{6/7/8} \begin{pmatrix} \cos \varphi_{6/7/8} \cos \vartheta_{6/7/8} \\ \sin \varphi_{6/7/8} \cos \vartheta_{6/7/8} \\ \sin \vartheta_{6/7/8} \end{pmatrix} \quad (5.76)$$

result the corresponding polar azimuth and elevation angles for each vector¹. Their squares

¹See also appendix A.2.

are the base for the determination of the components of $\mathbf{x}_{\text{SC}}(t_0)$ via quasi-linear mappings.

$$r = \underline{r} + (\bar{r} - \underline{r}) \sin^2 \varphi_{6/7/8} \quad (5.77)$$

$$v = \underline{v} + (\bar{v} - \underline{v}) \sin^2 \vartheta_{6/7/8} \quad (5.78)$$

$$\varphi = \underline{\varphi} + (\bar{\varphi} - \underline{\varphi}) \sin^2 \varphi_{9/10/11} \quad (5.79)$$

$$\vartheta = \underline{\vartheta} + (\bar{\vartheta} - \underline{\vartheta}) \sin^2 \vartheta_{9/10/11} \quad (5.80)$$

$$\zeta = \underline{\zeta} + (\bar{\zeta} - \underline{\zeta}) \sin^2 \varphi_{12/13/14} \quad (5.81)$$

$$\Phi = \underline{\Phi} + (\bar{\Phi} - \underline{\Phi}) \sin^2 \vartheta_{12/13/14} \quad (5.82)$$

This encoding scheme uses nine alleles to determine six initial state variables and is therefore overdetermined but guarantees always valid variables and prevents saturation.

5.5 Artificial Neural Networks

One of the two elementary elements of ENC are ANNs, which are used as NCs for the control of low-thrust powered spacecraft. ANNs thereby map from a set of problem-specific input data to the output variables, which then must be translated into control variables. Section 5.5.1 describes encoding and scaling of NC input information, and section 5.5.2 describes how the required control data derives from the ANN output.

5.5.1 Encoding and Scaling of Input Data

An NC is given input data of different types from which it determines the desired control via its internal ANN. The input data thereby range from position and velocity variables, over angles and their change rates, to mass, time and acceleration variables. An NC can generally handle all these data types but it needs a minimum number of internal nodes and an appropriately trained parameter set to do so. Problems may arise if input values are outside the reasonable range of neuron transfer functions. For example, if a neuron's transfer function is the sigmoid (see Fig. 4.4), and the spacecraft's heliocentric x-coordinate in kilometer is input to that neuron, then this input almost never contributes to the neuron's output. This is evident as the input is practically never inside the range of reasonable sigmoid input values. Using the distance unit AU would alleviate this problem but not mitigate it. An NC that should steer optimally along a transfer that spans over large distance ranges still suffers from it. Normalization of the different input data types is therefore encouraged.

Another problem occurs when using angle input data. Depending on the respective definition, at 0 or 2π their values can exhibit a change of 2π although the actual angle might change only by a few degrees. Assuming a sufficient number of neurons and proper training, an NC can handle even such input data peculiarities. The training of the NC for the required steering strategy might however be time consuming.

To alleviate the latter problem through the usage of a different encoding scheme and therefore "help" the NC to focus on the actual control problem, InTrance uses the sine and cosine of an angle instead of the actual angle value. The values of both, singularity-free functions are bound within $(-1, 1)$ and thus require no normalization. Solving the first problem means to find normalization values for distance, time, and velocity. However, only the first two must be defined because the velocity norm is derived from them. InTrance uses the current spacecraft distance w.r.t. the current central body r_{SC} as unit distance \hat{r} and scales all linear distance input values with this norm value

$$\hat{r} = r_{\text{SC}}. \quad (5.83)$$

Table 5.5: NC input data.

Variable	NC input	Norm	Frame
Spacecraft position x	x/\hat{r}	\hat{r}	(rotating),cartesian
Spacecraft position y	y/\hat{r}	\hat{r}	(rotating),cartesian
Spacecraft position z	z/\hat{r}	\hat{r}	(rotating),cartesian
Spacecraft velocity v_x	v_x/\hat{v}	\hat{v}	(rotating),cartesian
Spacecraft velocity v_y	v_y/\hat{v}	\hat{v}	(rotating),cartesian
Spacecraft velocity v_z	v_z/\hat{v}	\hat{v}	(rotating),cartesian
Control step size h	h/\hat{t}	\hat{t}	n/a
Spacecraft azimuth angle φ	$\sin \varphi$	n/a	polar
Spacecraft azimuth angle φ	$\cos \varphi$	n/a	polar
Spacecraft elevation angle ϑ	$\sin \vartheta$	n/a	polar
Spacecraft elevation angle ϑ	$\cos \vartheta$	n/a	polar
Spacecraft azimuth rate $\dot{\varphi}$	$\dot{\varphi}/\hat{t}$	\hat{t}	polar
Spacecraft elevation rate $\dot{\vartheta}$	$\dot{\vartheta}/\hat{t}$	\hat{t}	polar
Range ρ	$\frac{\ \mathbf{r}_T - \mathbf{r}_{SC}\ }{\hat{r}}$	\hat{r}	n/a
Range rate $\dot{\rho}$	$\frac{(\mathbf{v}_T - \mathbf{v}_{SC})(\mathbf{r}_T - \mathbf{r}_{SC})}{\hat{v}\ \mathbf{r}_T - \mathbf{r}_{SC}\ }$	\hat{v}	n/a
Abs. target position x_T	x_T/\hat{r}	\hat{r}	(rotating),cartesian
Abs. target position y_T	y_T/\hat{r}	\hat{r}	(rotating),cartesian
Abs. target position z_T	z_T/\hat{r}	\hat{r}	(rotating),cartesian
Abs. target velocity $v_{x,T}$	$v_{x,T}/\hat{v}$	\hat{v}	(rotating),cartesian
Abs. target velocity $v_{y,T}$	$v_{y,T}/\hat{v}$	\hat{v}	(rotating),cartesian
Abs. target velocity $v_{z,T}$	$v_{z,T}/\hat{v}$	\hat{v}	(rotating),cartesian
Rel. target position $x_T - x$	$\frac{x_T - x}{\hat{r}}$	\hat{r}	(rotating),cartesian
Rel. target position $y_T - y$	$\frac{y_T - y}{\hat{r}}$	\hat{r}	(rotating),cartesian
Rel. target position $z_T - z$	$\frac{z_T - z}{\hat{r}}$	\hat{r}	(rotating),cartesian
Rel. target velocity $v_{x,T} - v_x$	$\frac{v_{x,T} - v_x}{\hat{v}}$	\hat{v}	(rotating),cartesian
Rel. target velocity $v_{y,T} - v_y$	$\frac{v_{y,T} - v_y}{\hat{v}}$	\hat{v}	(rotating),cartesian
Rel. target velocity $v_{z,T} - v_z$	$\frac{v_{z,T} - v_z}{\hat{v}}$	\hat{v}	(rotating),cartesian
Propellant mass m_p	$m_p/m_{p,0}$	$m_{p,0}$	n/a

The time norm \hat{t} is derived with \hat{r} and the velocity v_{circ} of a body on a circular orbit with the radius \hat{r}

$$\hat{t} = \frac{\hat{r}}{v_{\text{circ}}} = \hat{r} \sqrt{\frac{\hat{r}}{\mu}}. \quad (5.84)$$

The velocity norm \hat{v} is calculated from \hat{r} and \hat{t}

$$\hat{v} = \frac{\hat{r}}{\hat{t}} = v_{\text{circ}}(r_{SC}). \quad (5.85)$$

Table 5.5 lists the definitions of the different NC input data variables and the physical variables they are derived from.

5.5.2 Decoding of Output Values

The spacecraft control vector \mathbf{u} comprises the local-optimal thrust direction vector \mathbf{e}_f for all propulsion systems and for propellant-dependent systems additionally the throttle factor χ .

Both control variables derive from the NC output vector $\mathbf{d} \in \mathbb{R}^{n_d}$, with $n_d = 3$ for solar sails and $n_d = 6$ for propellant-dependent propulsion systems. The first three components of \mathbf{d} determine the thrust direction unit vector $\hat{\mathbf{e}}_f$

$$\hat{\mathbf{e}}_f = \frac{\mathbf{e}_f}{\sqrt{\mathbf{e}_f \mathbf{e}_f}}, \quad (5.86)$$

with the intermediate vector

$$\mathbf{e}_f = 2 \begin{pmatrix} d_1 - 0.5 \\ d_2 - 0.5 \\ d_3 - 0.5 \end{pmatrix}. \quad (5.87)$$

The remaining three components determine the throttle χ

$$\chi = \begin{cases} 0, & \hat{\chi} \leq 0 \\ \hat{\chi}, & 0 < \hat{\chi} < 1 \\ 1, & \hat{\chi} \geq 1 \end{cases}. \quad (5.88)$$

It is calculated from the intermediate vector

$$\mathbf{p} = 2 \begin{pmatrix} d_4 - 0.5 \\ d_5 - 0.5 \\ d_6 - 0.5 \end{pmatrix} \quad (5.89)$$

through centering and rescaling with a variable a and the scale factor $b = 1.1$

$$\hat{\chi} = ab - \frac{1-b}{2}, \quad (5.90)$$

with

$$a = 1 - \frac{2 \left\| \arcsin \left(\frac{p_3}{\sqrt{\mathbf{p} \mathbf{p}^T}} \right) \right\|}{\pi}. \quad (5.91)$$

This encoding scheme is overdetermined and assures that:

1. the control is always valid, regardless of the respective d_i ,
2. no saturation effects occur, which might arise, for example, if only the range $(-1, 1)$ is taken as control,
3. various combinations of d_4 , d_5 , and d_6 yield the same χ .

Effectively, the throttle is the ratio of the angle $\angle(\mathbf{e}_1, \mathbf{p})$ and $\frac{\pi}{2}$, scaled by b and constrained in $(0, 1)$.

6

Validation and Mission Analysis

This chapter describes the validation of MENC and its implementation in InTrance, as well as its application to the analysis of multiphase deep space missions. The validation took place in several stages and levels of detail. Basic software features, such as OOP-related functionality, were validated with software test procedures to ensure logical correctness of implemented features. These low-level tests shall not be described here. More complex software components, e.g., numerical integration routines, required other validation methods. Section 6.1 provides results of two examples of the carried-out validation at this next-higher stage. The first one, presented in subsection 6.1.1, validates the results of the tests of the numerical integration routines and the second, in subsection 6.1.2, explains and demonstrates the validation of the GA implementation.

The third validation stage contained mission analysis, i.e., the optimization of low-thrust trajectories. Through comparison of the obtained results with analytical solutions of the respective problems or reference solutions from literature, new InTrance features were verified for correctness. One of these features is the support of non-heliocentric or planetocentric simulations. Correct simulation is a prerequisite for the optimization of low-thrust trajectories in such environments. Section 6.2 describes the corresponding calculations and presents the obtained results.

Solutions of heliocentric multiphase problems belong to the fourth validation stage. It contained the validation of the newly developed capability of InTrance to optimize multiphase trajectory problems. This was constrained to two-phase problems, i.e., trajectories that either lead to a single target, but were divided into two mission phases, or two target bodies. The first type is covered with section 6.3, which presents the results of several calculations of Earth-Moon low-thrust transfers. Section 6.4 is on the validation of the second problem type. It provides the results of a double-rendezvous low-thrust deep space mission for which corresponding trajectories have been calculated. NASA's deep space mission Dawn was chosen as reference and the results compared with its reference trajectory. This section further shows how global optimal solutions for that problem type may now be calculated with InTrance and explains the difference to a manually optimized multiphase trajectories.

Following validation, section 6.5 describes the application of InTrance to the analysis of a complex and challenging deep space mission. It is a solar system escape mission to the boundary of our solar system. Relying on a SEP and a REP stage, its mission design thereby leads the two-stage spacecraft to a distance of 200 AU within a defined maximum mission duration of 25 years. The obtained trajectory exhibits a gravity assist at Jupiter, which is an implicit result of the inclusion of third-body disturbances into simulation.

6.1 Implementation Validation

The following subsection 6.1.1 and subsection 6.1.2 describe the validation of the numerical integration algorithms and the GA that were used for this thesis. Both are only examples of the carried-out stage-2 validation activities during the development of InTrance, but further tests were conducted as well. This includes tests for the newly implemented support of third-body perturbing forces, reference system transformation and planetary shadows. A complete description of all of these validation activities is however beyond the scope of this work. Other simulation elements were already validated during Dachwald's work on the first version of InTrance [13] and can thus be considered as validated. An example of this is the computation of the SRP.

6.1.1 Validation of Numerical Integration Schemes

Numerical integration schemes, or numerical integrators, are essential components of InTrance's simulation objects. They provide the current state variables based on the state variables of the previous step, the state equations, and the control variables. It is immanent that numerical integration schemes should be free of any implementation errors. To assure correct integration of a given DES, the integration schemes implemented in InTrance were used on test DES, for which analytical results exist. Hull et al. [42] gives the test function

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{pmatrix} = \begin{pmatrix} y_3 \\ y_4 \\ -y_1/(y_1^2 + y_2^2)^{3/2} \\ -y_2/(y_1^2 + y_2^2)^{3/2} \end{pmatrix}. \quad (6.1)$$

The initial state vector for all calculations was

$$\mathbf{y}(0) = \begin{pmatrix} 1 - e \\ 0 \\ 0 \\ \sqrt{\frac{1+e}{1-e}} \end{pmatrix}. \quad (6.2)$$

It describes the planar orbital motion of an object under a pure Newtonian central force. For the five test runs, the eccentricity parameter was $e \in [0.1, 0.3, 0.5, 0.7, 0.9]$. The integration was carried out for $x \in (0, 20)$ with integrators Runge-Kutta (RK)4, Runge-Kutta-Fehlberg (RKF)5(4), and RK8(7)DP¹. Absolute and relative accuracy tolerance was set to 10^{-10} . The allowed cartesian deviation from the analytical reference solution was always less than 10^{-3} .

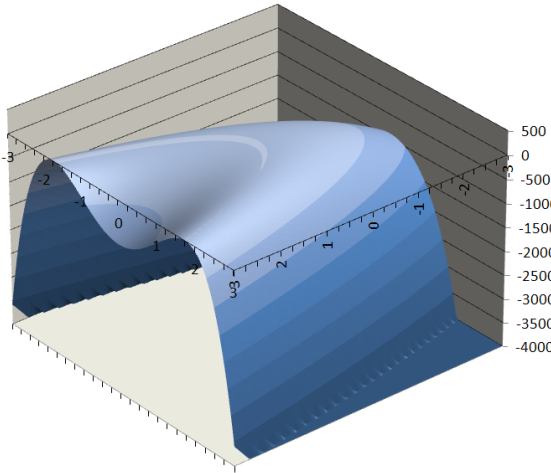
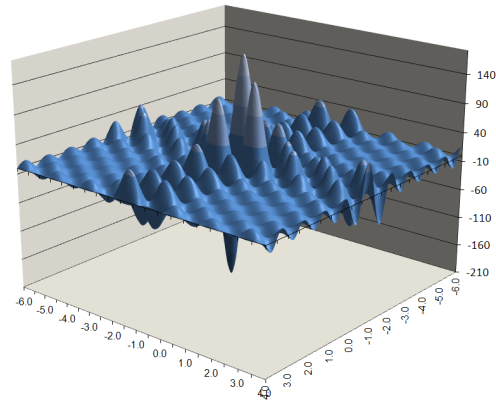
The respective numbers of calls to (6.1) to achieve the required accuracy limits are listed in Tab. 6.1. It shows an increasing number of DES calls with higher orbit eccentricities because the integrators had to insert more internal step subdivisions to keep the configured accuracy tolerances. Table 6.1 also shows a common feature of higher-order integration routines, such as RK87DP. They need fewer DES calls for achieving high accuracy than lower-order integrators, e.g., RK54F. For low accuracy requirements, however, this does not generally hold true any more, as both integrators may be able to achieve the accuracy tolerances without internal subdivision.

The results of the validation calculations were within the defined allowed deviation from the analytical solution, and the implementation of the integrators in InTrance was therefore considered to be correct.

¹RK4 is an integration scheme of order four, RKF5(4) is one of fifth order with an embedded method of order four for step size control based on the estimation of the local truncation error, and RK87DP is of order eight with an embedded method of order seven according to Prince and Dormand. [57]

Table 6.1: Calls to a DES function during integration of Hull’s class D test function.

Test Case	RK4	RKF54	RK87DP
D1 ($e = 0.1$)	4 400	3 561	1 309
D2 ($e = 0.3$)	15 200	7 290	2 903
D3 ($e = 0.5$)	32 400	11 612	4 861
D4 ($e = 0.7$)	56 000	17 038	7 416
D5 ($e = 0.9$)	85 996	24 563	11 178

**Figure 6.1:** Rosenbrock 2D Test Function.**Figure 6.2:** Multimodal 2D Test Function.

6.1.2 Genetic Algorithm Validation

A second essential InTrance component is the GA. It was therefore tested separately before using it for optimization runs. For verification, reference test functions with known optimal solutions were chosen. These known solutions were compared with the solutions that were obtained through using the GA on the test functions.

A standard test function in function optimization literature is the two-dimensional Rosenbrock function [76]

$$f(x_1, x_2) = - \left(100(x_1^2 - x_2)^2 + (1 - x_1)^2 \right). \quad (6.3)$$

It is also known as Rosenbrock’s valley, Rosenbrock’s banana, or as De Jong’s Second Function [17, 19]. Its global optimum $f = 0$ is at $(1.0, 1.0)$. As the EA’s objective is the maximization of an objective function, the inverted Rosenbrock function (6.3) was used for the validation runs. Figure 6.1 shows the maximum of (6.3) on a wide parabolic “mountain” with relatively small derivatives w.r.t. x_1 and x_2 . This is a fundamental problem for gradient-based methods. They either experience difficulties in finding the optimum at all or need many function evaluations for success. A separate test program using the GA of InTrance to find the optimum of the inverse of (6.3) was written. Multiple execution of that program with an accuracy of 10^{-3} showed that the GA-implementation within InTrance found the optimum of that test function in every run.

Table 6.2: Results of the EA runs for a multimodal, two-dimensional test function. The deviation Δf^* is the difference between the worst solution found by InTrance and the reference solution. The last column gives the standard deviation of the solutions found by the GA w.r.t. to the analytical reference solution.

q	Avg. epochs	Max. no. of eval.	Δf^* [10^{-8}]	dev. [10^{-7}]
25	136	203 615	4.3	3.6
50	122	228 339	3.4	4.7
75	122	251 727	5.6	6.1
100	119	282 100	7.8	5.1
125	119	310 487	11	4.2

A second GA validation test used the two-dimensional, multimodal function

$$f(x_1, x_2) = - \sum_{i=1}^5 i \cdot \cos((i-1)x_1 + i) \cdot \sum_{j=1}^5 j \cdot \cos((j+1)x_2 + j). \quad (6.4)$$

Figure 6.2 shows this function in $x_1, x_2 \in [-6, 4]^2$. It has numerous local optima and a global optimum $f(x_1^* = -1.306708, x_2^* = -1.4251284) = 176.5417931367$. The multimodal character of this function makes (6.4) suited for the validation of the GA's ability to find the global optimum in such solution spaces. The configuration parameters of the GA were varied to investigate on their effect for finding the global optimum. Therefore, the number of SSS epochs were varied from one to nine with a step size of one. The mutation probabilities were 0.2 to 0.8 in steps of 0.2. The hypercube shrink factor κ was varied from 0.2 to 0.8 also in steps of 0.2, and the population size q from 25 to 125 with a step size of 25. To test the GA's robustness, each optimization was run five times with identical settings of q . The optimum of the test function was found every time, whereby the average number of required GA epochs per run reduces with increasing q . The absolute number of function evaluations however increases with bigger populations. The average difference Δf^* to the optimum as well as its deviation was always below the preset limits.

6.2 Non-heliocentric Single-phase Transfers

One of the new developments in the course of enabling multiphase low-thrust trajectory optimization with InTrance was the support for non-heliocentric or planetocentric simulations. As that type of simulation was going to be used for the optimization of mission designs that involve respective phases, it needed to be validated beforehand. Several transfers from literature thereby served as reference for comparison and the assessment of simulation correctness. These test cases were four Earth-bound transfer problems and one having the asteroid Vesta as central body. The chosen cases therefore cover a representative set of orbit transfer problems. They range from coplanar circular orbit transfers, i.e., only the semimajor axis changes, to trajectories during which all orbital elements must be changed optimally.

6.2.1 Test Cases

Five test cases were chosen for the validation of the capability of InTrance to optimize non-heliocentric low-thrust trajectories. They are given by Petropoulos who used them for the

Table 6.3: Orbit specification of test cases for non-heliocentric orbit changes. The test cases were taken from [68, 69]. The orbits are specified with Keplerian elements (see appendix B). The respective spacecraft characteristics are given with the maximum thrust F_{thr} , the launch mass m_0 , and the specific impulse I_{sp} of the propulsion subsystem.

Test Case	Orbit Type	a [km]	e	i [deg]	ω [deg]	Ω [deg]	F_{thr} [mN]	m_0 [kg]	I_{sp} [s]	Central Body
A	Launch	7 000	0.01	0.05	0	0	1 000	300	3 100	Earth
A	Target	42 000	0.01	n/d	n/d	n/d	1 000	300	3 100	Earth
B	Launch	24 505.9	0.725	7.05	0	0	350	2 000	2 000	Earth
B	Target	42 165	0.001	0.05	n/d	n/d	350	2 000	2 000	Earth
C	Launch	9 222.7	0.2	0.573	0	0	9 300	300	3 100	Earth
C	Target	30 000	0.7	n/d	n/d	n/d	9 300	300	3 100	Earth
D	Launch	944.64	0.015	90.06	156.9	-24.6	45	950	3 045	Vesta
D	Target	401.72	0.012	90.01	n/d	-40.73	45	950	3 045	Vesta
D*	Launch	950.00	0.000	90.00	n/d	-102	45	950	3 045	Vesta
D*	Target	375.00	0.000	90.00	n/d	-126	45	950	3 045	Vesta
E	Launch	24 505.9	0.725	0.06	0	0	2 000	2 000	2 000	Earth
E	Target	26 500	0.7	116	270	180	2 000	2 000	2 000	Earth

validation of a heuristic steering law based on a Lyapunov function, which he named Q-law [67, 68, 69]. Table 6.3 lists the initial and target orbit parameters for each case as well as the spacecraft specifications. Petropoulos also compared the flight times and propellant mass consumptions of his solutions with reference trajectories found by others with different optimization techniques.

Case A is a coplanar, circle-to-circle, time optimal transfer from a LEO to a geostationary orbit (GEO). For a low-thrust system, the chosen thrust-to-mass ratio of 1 N/300 kg is rather high but still less than for chemical propulsion. For this transfer, there exists an optimal solution found by Edelbaum [23], and the fastest Q-law transfer found by Petropoulos showed good coincidence with it. Any minimum-time Case A solution found by InTrance should consequently be close to these two reference solutions in order to be valid.

Case B is more complex than Case A because three instead of one target orbit element had to be matched. It is a propellant-optimal, low-thrust transfer from a slightly inclined GTO to a GEO. Geoffrey and Epenoy [32] found further solutions of that transfer using orbit averaging and the calculus of variations. Their results were included into the comparison.

Case C is a propellant-optimal, coplanar transfer from a low-eccentricity orbit to a larger, high-eccentricity orbit with a thrust-to-mass ratio bigger than in Case A. Petropoulos compared his results to fixed-time, propellant-optimal solutions for this problem that was found with an optimization software tool called Mystic. This software is based on the static/dynamic control (SDC) algorithm, which was developed by Whiffen and Sims [95]. Mystic is one of the primary mission analysis tools at JPL and, as of today, restricted by International Traffic and Arms Regulations (ITAR).

The fourth test Case D is the only one that was not bound to the Earth. It was roughly a circle-to-circle transfer around the asteroid Vesta with a small plane change. Again, the solutions found by Petropoulos were compared with a minimum-time, minimum-propellant solution found with Mystic. However, the initial and final conditions of the transfer published by Whiffen [94] is not exactly the one given by Petropoulos. The original transfer was therefore included as test case D_mod.

Table 6.4: Results of non-heliocentric, low-thrust orbit transfers. The flight time Δt and the consumed propellant m_p of the solutions found by InTrance are listed for each test case. Respective figures of reference solutions are provided for comparison. The figures of Case B of Petropoulos [68] were taken from Fig. 4.

Test Case	Δt [d]	m_p [kg]	Reference
A	14.60	41.50	Petropoulos [69]
A	14.53	41.28	InTrance
A	14.42	40.98	Edelbaum [23]
B	≈ 145	≈ 220	Petropoulos [68]
B	142.27	219	InTrance
B	137.5	212	Geoffreoy & Epenoy [32]
C	4.4	18.7	Petropoulos [68]
C	4.4	18.7	InTrance
C	4.4	18.2	Whiffen [68]
D1	25	3.22	Petropoulos [68]
D1	25	3.26	InTrance
D2	25	3.26	InTrance
D2	25	3.22	Whiffen [94]
E	96.6	677.2	Petropoulos [68]
E	81.61	719.0	Petropoulos [69]
E	72.3	637.2	InTrance

Test Case E is the most challenging one and requires changing all orbit elements except mean anomaly, which was always free. It was a transfer from an inclined GTO into a Molniya-type orbit. The required inclination change was 116 deg. The objective was to find the time-minimal transfer, and the shortest transfer found in [68] was 96.6 days. Later, with a refined Q-law, Petropoulos could reduce the transfer time to 81.61 days. The required propellant mass of that solution was 719 kg [69].

6.2.2 Results

The recalculation of test cases A to E were conducted by following a two-stage approach. A so-called “coldstart” optimization run yielded the first solution. The obtained solution was then refined with a second optimization run called “refine”. During coldstart runs, the EOMs were integrated with an RKF45 integrator. The control step size was dynamically determined with a discretization in true anomaly of six degrees for the first run and of three degrees for the refinement run. The relative and absolute integration error tolerances were 10^{-6} and the allowed final distance 200 km and relative velocity 100 m/s. For the subsequent refinement of the coldstart solution, an Dormand and Prince 7(8) integrator was used with relative and absolute error tolerances of 10^{-10} . The respective optimization goal was set for each problem.

Table 6.4 gives the flight time Δt and the consumed propellant mass m_p of the obtained results and the corresponding figures of the reference transfers from literature. The associated trajectory plots are given in Fig. 6.3 to Fig. 6.8. The found InTrance-solution for case A was between those found with the Q-law and the solution found by Edelmann, whereby the InTrance-solution was closer to one of Edelmann. For case B, the solution found by

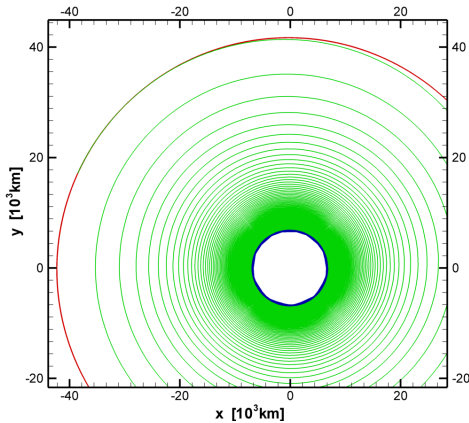


Figure 6.3: Case A.

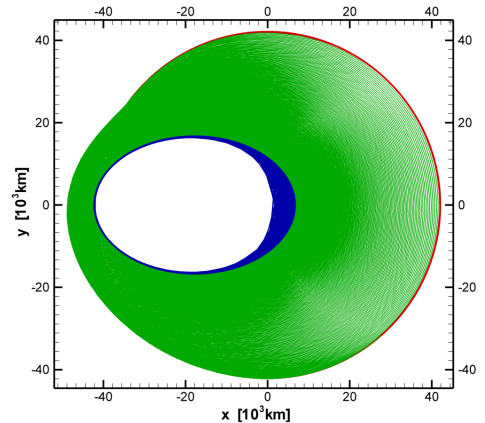


Figure 6.4: Case B.

InTrance was inferior to those found by Geoffroy and Epenoy but better than the Q-law solution. InTrance's solution for case C matched the Q-law solutions but was inferior to the one found with Mystic. The fourth test case result was also close enough to the reference solution to be acceptable. For the most challenging transfer, case E, InTrance found a solution which was faster and consumed less propellant than the best obtained Q-law result. The Q-law, which showed acceptable performance for the other four test cases and good coincidence with respective reference solutions, therefore seem to be less suited for case-E-like transfers. This may however depend on the choice of the user for the Q-law control parameter. The comparison of the results that were obtained with InTrance for various non-heliocentric test cases showed very good coincidence with the chosen reference solutions. For the most challenging transfer, Case E, InTrance even found a significantly better result.

With only small deviations of less than 2% from the respective reference solutions A-D, InTrance found comparable solutions for these test cases. The deviations in flight time and mass consumption are explainable with differences in the respective simulation and optimization setups. Elimination of those is hardly possible, as publications do rarely contain all the details of carried-out calculations that would be required for exact comparison. Using a slightly different control step size, for example, can sum up along the trajectory to small deviations of such figures as the flight time. This is however acceptable for the purpose of preliminary mission analysis. The explanation of the better solution found by InTrance for Case E is Petropoulos' Q-law. Dependent on a user-provided so-called cut-off parameter, this control law allows for coast phases. This parameter was not optimally chosen, as InTrance found a steering strategy that uses the thruster during the entire transfer, resulting in a better solution to the problem. Concluding from these results, the implementation of non-heliocentric simulations within InTrance is considered validated and usable for optimization.

6.3 Earth-Moon Transfers

This section is about the optimization of two-phase low-thrust transfers during which a SOI must be crossed. The chosen problem is a Δt -optimal transfer from an Earth-bound orbit into an orbit about the Moon. The resulting trajectory should qualitatively resemble those of SMART-1, one of the few real-life low-thrust missions so far and the only low-thrust mission to the Moon. This mission type has been treated extensively in literature, for example by Betts and Erb [10] or Herman and Conway [40].

Earth-Moon transfers exhibit several peculiarities. Some of them stress the simulation framework which is inevitably associated to trajectory optimization. Others can in fact ease the

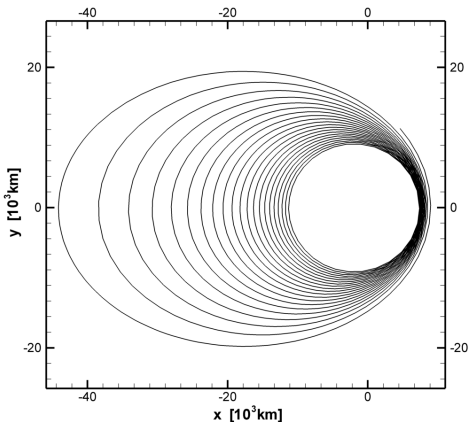


Figure 6.5: Case C.

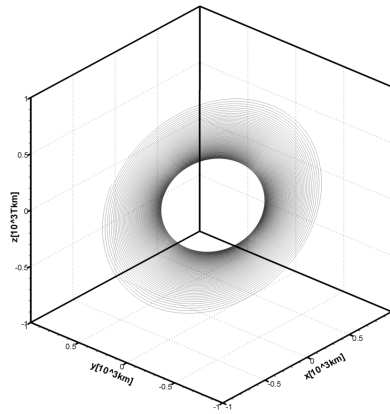


Figure 6.6: Case D1.

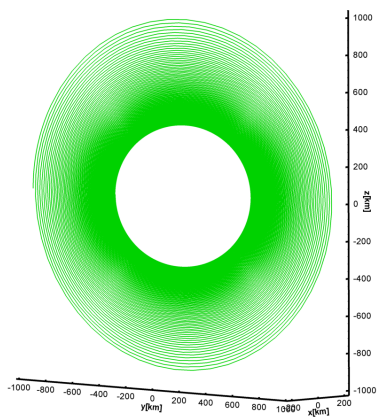


Figure 6.7: Case D2.

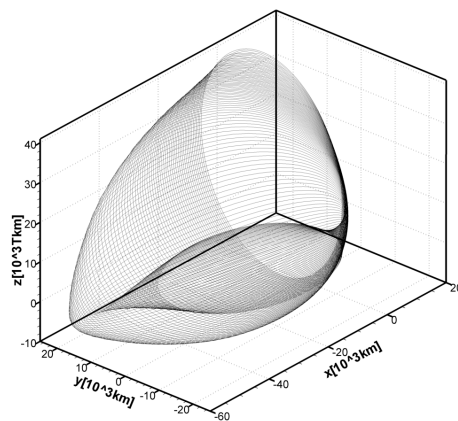


Figure 6.8: Case E.

optimization process. Compared to the natural satellites of other planets in our solar system, the Moon is a big and mass-rich. A spacecraft in this environment is therefore subject to strong gravitational forces of the Earth and the Moon. Additional gravitational pull results from the Sun. An astrodynamics simulation must therefore at least include the gravitational accelerations caused by these three celestial bodies. As third-body accelerations cannot be neglected for Earth-Moon transfers, it was implemented in InTrance before such transfers were calculated (see subsection 5.3.3 for details). The calculation of the transfer cases of this section therefore also validates this feature of InTrance.

The mass ratio of the Earth and the Moon is 81.3 and thus smaller than the corresponding ratio of any other planet-satellite constellation¹ of the solar system. This mass ratio makes the radius of the Moon's SOI exceptionally big. In fact, it is nearly 1/6 of the semimajor axis of its orbit. With approximately 1/400, the corresponding SOI-to-semimajor-axis ratio of Mars is significantly smaller. From the perspective of trajectory optimization, the Moon is therefore a large, heavy, and near target.

The advantage of a close and heavy target comes with a downside. Low-thrust transfer trajectories from LEO typically exhibit many revolutions about the Earth. The reason is that at such orbit heights the gravitation acceleration is magnitudes higher than acceleration resulting from propulsion. This often requires more complex steering strategies which are difficult to optimize.

The recalculation of this transfer problem was conducted via two approaches. The first treated the complete transfer as a single-phase problem while the second splits it into two phases. The second approach eases simulation but requires the optimization of the transition point, i.e., finding the point in state space from which the problem turns from a geocentric into a Moon-centric one. The next two sections describe the test cases, the simulation and optimization setup, and the obtained results. Finally, the remarkable ENC-feature of parameter space reduction is explained with the presented results.

6.3.1 Problem Description and Setup

Three transfer time-minimal Earth-Moon transfers, called Case A, Case B, and Case C, were chosen. Orbital elements of the respective initial Earth-bound orbit and the target orbit of each case are listed in Tab. 6.5. The transfers Case A and Case B were optimized as single-phase transfers, and transfer Case C was configured as two-phase transfer. The EOMs in cartesian, Earth-centered inertial (ECI) coordinates were integrated with a RK 8(7) integrator (Dormand and Prince) using error tolerances of 10^{-6} . The step size of DSSC was set to 20 deg. JPL's DE405 ephemerides catalog was used for position and velocity information of the Earth and the Moon. The spacecraft was modeled with a NEP subsystem, $m_d=283$ kg, $m_p=84$ kg, $I_{sp}=3714$ s and $F^*=135$ mN. The EA settings were $q=35$ and $p_m=0.93$. Tests carried out before the optimization showed that the spacecraft-steering NC required no hidden-layer neurons. Its configuration was therefore set to 37 input neurons and five output neurons. Spacecraft ECI position and velocity in cartesian and polar expression were provided to the first 14 NC input neurons. To prevent discontinuities at null and 360 deg, angular parameters, e.g., polar azimuth and elevation, were first transformed to sine and cosine before using as input. Next 14 input elements were the Moon's cartesian and polar position and velocity, processed as the ones for Earth. Moon-centered orbit elements a , e , and i of the spacecraft and of the target orbit were the input of the next eight input neurons. Input to the last input neuron was the relative propellant, i.e., as fraction of current propellant mass to initial propellant mass.

¹See Tab. 5.2. The mass ratio next in size is that of the Sun and Jupiter with 1047.

Table 6.5: Orbit specification of Earth-Moon transfers.

Test case	Orbit type	Central body	a [km]	e	i [deg]	Ω [deg]	ω [deg]	M [deg]
A,B	Launch	Earth	24 500	0.73	0	0	0	0
A	Target	Moon	20 000	0	40	free	free	free
B	Target	Moon	7 238	0.62	90	free	free	free
C	Launch	Earth	24 460	0.73	6	279.2	177.9	3.5
C	Target	Moon	2 000	0	84	free	free	free

Table 6.6: Results of single-phase Earth-Moon transfers. For each case, the flight time Δt , the consumed propellant mass m_p , and the achieved distance Δr to the respective target and relative velocity Δv are listed.

Transfer	Δt [d]	m_p [kg]	Δr [km]	Δv [m/s]
A	120	35	670	33
B	194	56	500	25
C	183	105	694	91

The first leg of Case C was computed in cartesian ECI coordinates and the second in cartesian Moon-centered inertial (MCI)-frame coordinates. The handover was at the SOI, whereby the state vector at the end of phase one was taken as initial state of phase two. Therefore, the resulting trajectory was always physically valid. This approach was possible because of the big radius of the Moon's SOI. Initial mass was set to 671 kg, the maximum thrust was 271 mN, and the specific impulse 3714 s. Each phase used a dedicated NC for spacecraft steering. Phase one used a (26-15-6) NC with 26 input neurons, 15 hidden-layer neurons, and 6 output neurons for spacecraft steering. The NC of phase two had a (23-35-6) topology.

6.3.2 Results and Discussion

The results of the single-phase transfers are listed in Tab. 6.6 and were also published by the author [63]. Figure 6.9 and Fig. 6.9 show the respective trajectories in ECI-frame coordinates.

The trajectory of Case A from GTO into selenocentric circular orbit takes $\Delta t \approx 120$ d and consumes $m_p \approx 35$ kg of propellant. For Case B, the respective figures are $\Delta t \approx 193$ d and $m_p \approx 56$ kg. As expected, the flightpath of both transfers exhibits many revolutions about the Earth before the spacecraft enters the SOI of the Moon.

Table 6.7: Earth-Moon transfer: Comparison of the Number of Node Parameters and ENC Parameters of Case C.

	# of steps	# of node variables	# of ENC variables
Leg 1	5 068	15 204	527 (3.5%)
Leg 2	17 556	52 668	1 102 (2.1%)
Total	22 624	67 872	1 629 (2.4%)

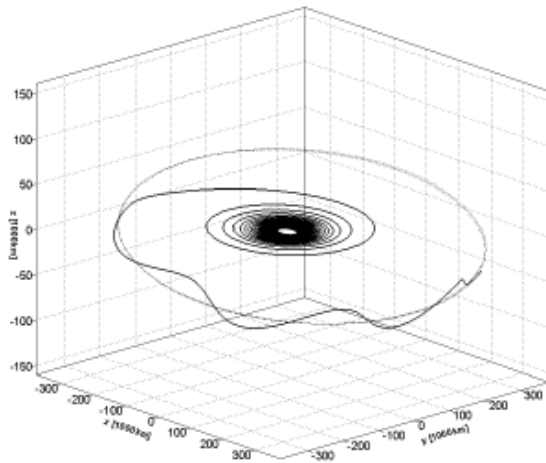


Figure 6.9: Trajectory of Earth-Moon Case A.

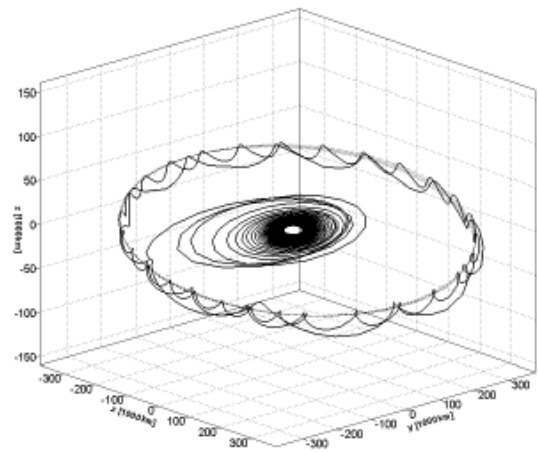


Figure 6.10: Trajectory of Earth-Moon Case B.

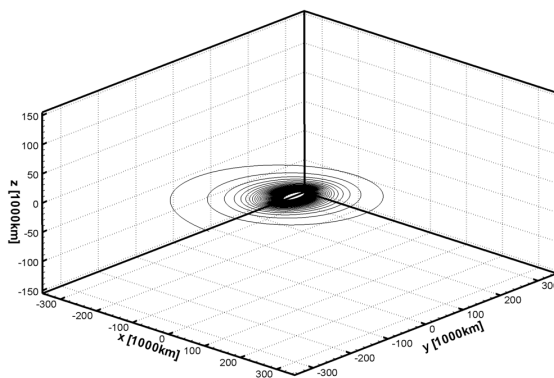


Figure 6.11: First Flight Leg from Earth Orbit to the Moon's SOI

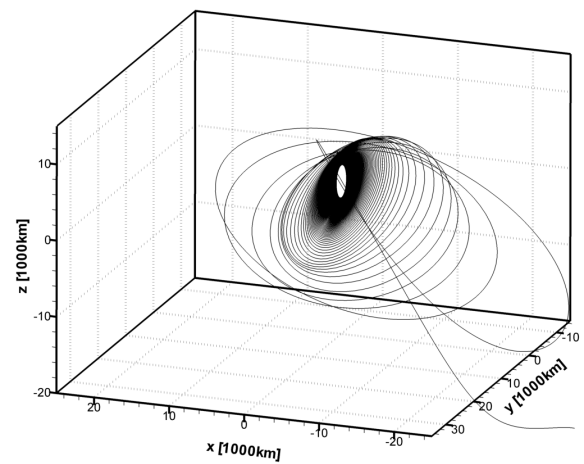


Figure 6.12: Second Transfer Phase from SOI Boundary to Low Lunar Orbit.

The resulting trajectories of the test cases highlight a substantial advantage of ENC, which becomes evident with trajectories having multiple revolutions about a central body. This advantage is the reduction of the number of required parameters and shall be explained with the solution of Case C. Table 6.7 gives the number of nodes for each flight leg of that transfer. A node is a trajectory point at which the thrust vector is determined and then held constant for the integration of the EOMs until the next node. As a three-dimensional vector \mathbf{F}_{thr} is associated to each node, the resulting parameter count is three times the number of nodes. For the first leg, and without using ENC, 15 204 variables would have been required for modeling and optimization of that trajectory part. The corresponding number of ENC parameters, which also describe the solution of that leg completely, was however only 527, corresponding to a reduction of about 97 percent. The solution of the complete transfer comprised 1 629 parameters, and the number of node parameters in a non-ENC treatment would have been of 67 872. The achieved total reduction is about 98 percent.

6.4 Multiple Asteroid Rendezvous – Dawn

NASA’s deep space low-thrust mission Dawn is an example for low-thrust missions that comprise multiple asteroid rendezvous. It uses SEP throughout the entire mission and for all phases, i.e., interplanetary cruise, target body approach, orbit changes, departure from Vesta, and attitude control. Based on the original mission specification, InTrance was used to recalculate the respective flight legs and to optimize the entire mission for minimum duration. This example thus demonstrates how InTrance can be applied to preliminary analysis of heliocentric, low-thrust, multiple-rendezvous missions.

Subsection 6.4.1 briefly describes the scientific objectives and key figures and facts of Dawn. For more information on that fascinating and novel low-thrust mission, please see Rayman et al. [70], Rayman and Patel [74], Rayman and Mase [73], and Russel et al. [78]. Subsection 6.4.2 describes the trajectory optimization problem and the setup of InTrance that was used for the optimization runs. The obtained results, which are also in Ohndorf and Dachwald [62], are then presented and discussed in subsection 6.4.3.

6.4.1 Mission Description, Objectives, and Spacecraft Design

Dawn is the first mission whose objective is to orbit a main belt asteroid and the first that orbits two extraterrestrial bodies. The project is designed to increase our understanding of the conditions and processes acting at the solar system’s earliest epoch. Examining the geophysical properties of Ceres and Vesta with panchromatic and multi-spectral imagery is therefore one of Dawn’s primary objectives. Table 6.8 lists physical and orbit characteristics of both asteroids. Vesta and Ceres are by far the largest asteroids in our solar system. Because of that size, they have survived the collisional history of our solar system largely intact. They are virtually “records” of the physical and chemical conditions during the early epochs of our solar system. This renders them particularly interesting for astrogeologists, astrobiologists, and astrophysicists. Rayman et al. [70] hence list the following scientific objectives for Dawn:

1. Determination of the bulk density of Vesta and Ceres to better than 1%;
2. Determination of the spin axis orientation of Vesta and Ceres to better than 0.5%;
3. Determination of the gravity fields of Vesta and Ceres;
4. Optical mapping of the surface of Vesta and Ceres;

Table 6.8: Physical parameters and orbit elements of asteroids Vesta and Ceres. The orbits are specified with perihelion distance r_p , aphelion distance r_a , and inclination i . Physical parameters are the principle radii, density ρ , and sidereal rotation period Δt_{rot} . Values taken from Rayman et al. [70] and Taylor et al. [86]

Body	Principle Radii [km]	ρ [g/cm ³]	Δt_{rot} [h]	r_p [AU]	r_a [AU]	i [deg]
Vesta	289 × 280 × 229	3.7	5.34	2.15	2.57	7.1
Ceres	487 × 487 × 455	2.1	9.08	2.55	2.99	10.6

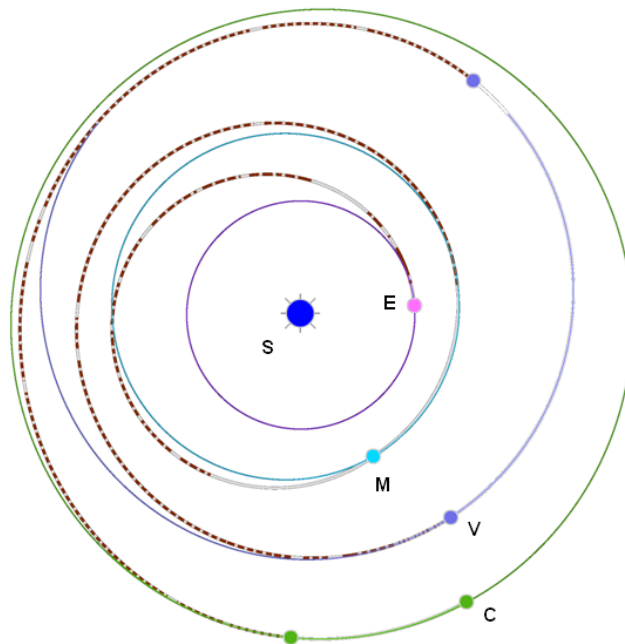


Figure 6.13: Reference Trajectory of the Multi-asteroid-rendezvous Low-thrust Mission Dawn. The letters denote the Sun (S), the Earth (E), Mars (M), Vesta (V), and Ceres (C). Taken from [72]. Courtesy NASA/JPL

5. Creation of a topographical map of the surface of Vesta and Ceres;
6. Measure and map the abundances of major rock-forming elements and the elements H, K, Th, and U of the surface of Vesta and Ceres;
7. Obtain spectral frames of the surfaces of Vesta ($\geq 10\,000$ frames) and Ceres ($\geq 8\,000$ frames).

Dawn launched on September 27 2007 (54370 MJD) with a Delta II 7925H-9.5 rocket from Cape Canaveral, Florida. The launch mass was $m_0 = 1\,217$ kg, with 425 kg being Xe propellant. Dawn's IPS is based on DS1 heritage and uses the same NSTAR thruster type. A single thruster of this type has a maximum thrust of 92 mN [70], a PPU input power $P_{\text{PPU}} = 2.6$ kW, and a maximum propellant throughput of 150 kg. The Dawn spacecraft's IPS comprises three NSTAR engines, with maximum one being operational at a time. EPGS output at 1 AU distance from the Sun and beginning-of-life (BOL) conditions is 10.3 kW. The reference trajectory is shown in Fig. 6.13. It features a Mars gravity assist (MGA), which,

according to Rayman, was not mandatory for mission success but increased system margin and thus reduced mission failure risk.

6.4.2 Problem Description and Setup

To determine a trajectory that would be comparable to the reference trajectory, the boundary conditions needed to equal to those of Dawn. The following assumptions were therefore made for the recalculation of the reference trajectory with InTrance. The launch should be not later than September 30 2007 ($t_0 \leq 54\,373\text{MJD}$). InTrance lacks a robust and generic support of gravity assists. Contrary to the real Dawn mission, the transfer should therefore not contain such a flyby maneuver. A maximum hyperbolic excess energy of $C_3=11.3\text{ km}^2/\text{s}^2$ was allowed [74] ($|\mathbf{v}_\infty|=3.362\text{ km/s}$), with the direction of \mathbf{v}_∞ being subject to optimization by InTrance. Duty cycle limits were not included, i.e., the IPS was allowed for continuous operation directly from launch. Launch and early operations phase (LEOP), in-orbit check-out, and commissioning were therefore ignored or ignored for operating the IPS. The stay-time at Vesta should be at least 250 days ($\Delta t_{\text{dw}} \geq 250\text{ d}$), and the arrival at Ceres should be not later than at the end of March 2015 ($t_{\text{f,Ceres}} \leq 57\,112\text{MJD}$). An asteroid was successfully met if $\Delta r \leq 500\,000\text{ km}$ and $\Delta v \leq 500\text{ m/s}$. The dry mass was set to $m_d=790\text{ kg}$ and the propellant mass was optimized by InTrance for both flight legs. Mass reductions due to the depletion of Hydrazine, the propellant of Dawn's attitude control subsystem (ACS), was not modeled. To account for the load of other spacecraft components, the available electrical power at 1 AU was set to $P_{\text{e,SP},0}(t_0)=9.8\text{ kW}$ instead of the 10.3 kW supplied at that distance by the real Dawn spacecraft's solar generator. Degradation was not simulated ($P_{\text{e,SP},0}(t_0)=P_{\text{e,SP},0}(t_{\text{f,Ceres}})$). The power law exponent was $\kappa=1.7$, according to Rayman et al. [71]. Thrust and propellant consumption curves of the NSTAR thruster were taken from Williams and Coverstone-Carroll [99]. The maximum thruster input power was $P_{\text{T}}=2.6\text{ kW}$.

As InTrance can globally optimize multi-rendezvous deep space missions, it was used to investigate on whether such mission designs could be improved w.r.t. minimum mission duration. Two design options were calculated for this purpose. The first alternative A comprised the individual optimization of each leg for minimum Δt , i.e., one after another, obeying a preset minimum stay-time at Vesta. A globally optimized complete mission, using the developed multiphase framework, was called Alternative B. The resulting mission durations and the consumed propellant masses should be compared. Thus, and besides recalculation, the interest was also on the optimization of the actual Dawn reference trajectory. Therefore, slightly different values or ranges of the simulation parameters were chosen. One of them were the launch windows, which are less important for the planned comparison. Furthermore, dry mass was $m_d=790\text{ kg}$ and the maximum thrust $F_{\text{thr}}=72\text{ mN}$, the original value of an NSTAR thruster. The propellant mass m_p should be optimized by InTrance. Launcher-provided hyperbolic excess velocity was set to 1.732 km/s [70] and the EPGS power output at one AU Sun-distance to 10.3 kW. The error tolerances of the RKF 5(4) integrator were set with 10^{-5} .

6.4.3 Results and Discussion

Figure 6.14 shows the trajectory resulting from the optimization with InTrance. The launch is on August 21 2007 (54 333 MJD) with the predefined $v_\infty=3.362\text{ km/s}$. Launch mass is 1 139 kg, with 346 kg propellant. Its direction was however determined by InTrance for maximal benefit for the transfer to Vesta, where it arrives after 1 472 days on September 1 2011 (55 805 MJD). Arrival conditions are $\Delta r=162\,000\text{ km}$, $\Delta v=195\text{ m/s}$, and 264 kg of propellant are consumed. The stay-time at Vesta is $\Delta t_{\text{dw}}=251\text{ d}$, and on May 8 2012 (56 055 MJD) Dawn leaves Vesta towards Ceres. After 1 047 days, the spacecraft arrives at Ceres on March 21

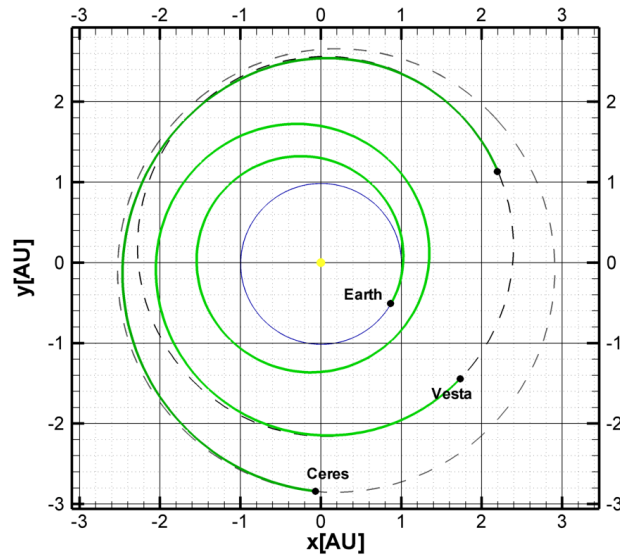


Figure 6.14: Dawn's Trajectory, Recalculated with InTrance.

2015 (57 054 MJD) with $\Delta r=480\,000$ km and $\Delta v=239$ m/s. The total mission duration until arrival at Ceres is 2770 days.

Except for the excluded MGA, the trajectory on Fig. 6.14 looks almost identical to the reference trajectory on Fig. 6.13. Its propellant consumption is however 20 percent less than the reference value $m_p=425$ kg. With the original propellant mass, the transfers from Earth to Vesta and from Vesta to Ceres are not achievable without violation of set constraints, e.g., mission duration or stay-time at Vesta. Dawn however requires these additional 87 kg of Xe for orbit maneuvers at the asteroids and for safety margin of its propellant budget. The implementation of ENC for heliocentric multi-rendezvous missions in InTrance is nevertheless considered valid, as the described recalculation led to qualitatively equal results. Due to the MGA of the reference trajectory, further comparison is difficult. Both solutions have specific consequences. The MGA gives a higher margin for the ΔV -budget. It also reduces the mission duration but at the same time limits the number of launch dates or narrows launch windows. The solution obtained through InTrance lacks that ΔV -margin and therefore requires the entire propellant but has a wider launch date window. Furthermore, the duty cycle of the IPS was set to unrealistic 100 percent. The found solution may therefore not be realized without different launch and arrival dates. It may even require an adaption of the mission's mass budget, i.e., reduction of m_a or increase of m_p . However, for preliminary mission analysis, the result is adequate and comparable with the reference mission design. It furthermore demonstrates the advantage of multiphase optimization with InTrance, as it has been found requiring significantly less effort than for the reference trajectory¹

Due to the validation purpose, i.e., for the comparison with a reference, the recalculation of the Dawn trajectory included several constraining assumptions. These assumptions do however limit a global optimization, which is a key feature of InTrance. The optimizations of two alternative transfers, called A and B, were therefore conducted with less Dawn-derived constraints, as described in subsection 6.4.2. Table 6.9 gives the results of these optimization

¹The determination of the Dawn reference trajectory required a team of astrodynamics and optimization experts using JPL-proprietary, non-freely available tools. One of them is the ITAR restricted software Mystic.

Table 6.9: Results of two options of a Dawn-like transfer to Vesta and Ceres. The symbols denote launch date t_0 , flight time Δt , propellant mass m_p , and the stay-time Δt_{dw} at Vesta. The numerals 1 and 2 denote the first transfer from Earth to Vesta and the second transfer from Vesta to Ceres.

	Unit	A	B	B-A
t_0	[MJD]	54 103	54 150	+47
Δt_1	[d]	1 744	1 749	+5 (0.3%)
$m_{p,1}$	[kg]	327	328	+1 (0.3%)
Δt_{dw}	[d]	528	341	-241 (41.4%)
Δt_2	[d]	679	790	+111 (16.3%)
$m_{p,2}$	[kg]	114	111	-3 (2.6%)
Δt_{1+2}	[d]	3 005	2 879	-126 (4.2%)
$m_{p,1+2}$	[kg]	441	439	-2 (0.5%)

runs. The resulting trajectory plots in heliocentric ecliptic projection are shown in Fig. 6.15 and Fig. 6.16.

Alternative A takes 3 005 days and 441 kg of propellant. The globally optimized Alternative B needs 2 879, a saving of 126 days or four percent, while the required propellant is practically unchanged. Its difference is only 0.5% and therefore within margin of any real mission's mass budget.

The reason for the differing mission durations becomes apparent when comparing the launch and arrival dates. Clearly, a minimal total mission duration requires early departure at Earth, a minimum stay-time at the intermediate target Vesta, and a time-minimal second transfer from Vesta to Ceres. Due to its global-optimizing character, InTrance can find the compromise between these mission phases that leads to minimal mission duration. This can also be seen in the trajectories shown in Fig. 6.15 and Fig. 6.16. Compared to Alternative A, launch of Alternative B takes place 47 days later. The transfer to Vesta takes five more days, resulting in a net saving of 42 days. Transfer leg 2 of Alternative B takes 111 days longer than for the individual-leg-optimized alternative. This is however over-compensated by the shorter dwell-time at Vesta. While Alternative A takes 679 days, Alternative B requires 241 days less.

Two conclusions can be drawn from the presented results. First, the optimization of multi-rendezvous missions requires global optimization tools. As shown with the Dawn mission design, they yield better results than individually optimizing the trajectories of each phase and putting them together manually. Second, the MENC-implementation InTrance can solve that particular problem type and yield global-optimal solutions.

6.5 Solar System Escape to the Heliosphere Bowshock

This section describes the use of InTrance for trajectory optimization of a proposal for a future deep space mission and presents the results. The objective of that mission is to bring a spacecraft within a predefined time frame on a solar system escape course to the heliosphere boundary region. This mission analysis was included in this thesis because:

1. It shows how the newly developed framework in InTrance can be applied to the optimization of such multiphase trajectories and for preliminary mission analysis.

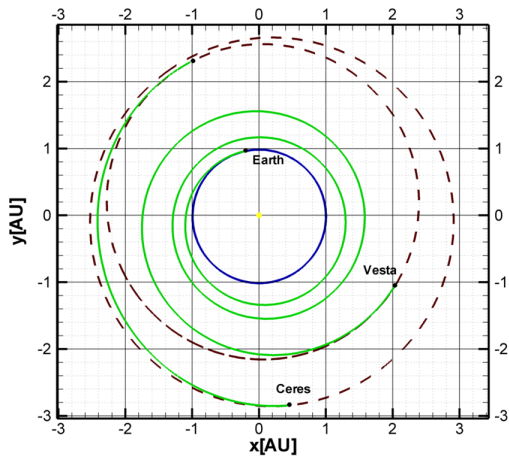


Figure 6.15: Trajectory of Alternative A of a Dawn-like Transfer.

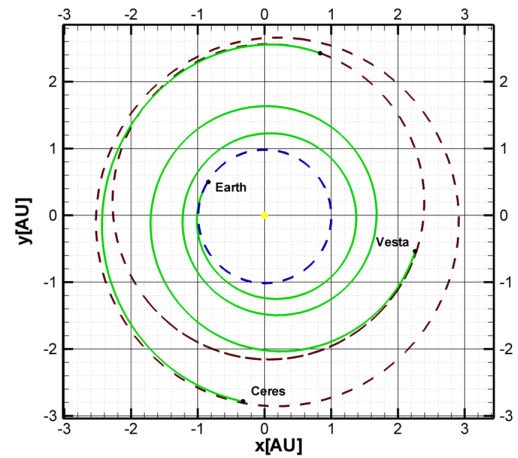


Figure 6.16: Trajectory of Alternative B of a Dawn-like Transfer.

2. It does not involve rendezvous conditions at the target body. The resulting trajectory is therefore different to the examples shown before and its optimization requires a different steering strategy than for multiple rendezvous-type transfer problems.
3. The spacecraft design comprises two propulsion subsystem stages of different technology: a primary SEP-stage and a second stage based on REP. When the spacecraft is too far from the Sun for the generation of solar-electrical power, the jettison of the SEP stage instantaneously reduces the mass of the spacecraft. Other optimization methods can have problems with the modeling of such changes, and therefore the optimization of this transfer also demonstrates the respective new capability of InTrance.
4. The mission concept poses a challenging flight-time requirement. It is however achievable with low-thrust propulsion and one gravity assist at Jupiter. The mission concept is therefore an example for deep space missions that are impossible for chemical propulsion but become feasible with low-thrust propulsion, although it still involves a GA. Subsection 6.5.2 will discuss whether the disadvantages for mission design that result from using GAs can be eliminated by a relaxation of the flight time requirement.
5. The inclusion of a Jupiter gravity assist shows how this technique can be included for low-thrust trajectory optimization, although InTrance lacks a generic support of gravity assist. For certain cases, and using mass-rich celestial bodies such as Jupiter, it is however implicitly possible.

The outer solar system and the region beyond are of scientific interest since decades [45]. The heliosphere bow shock is one of those particularly interesting areas. In the heliocentric ecliptic reference frame, it is in flight direction of our solar system at a distance of approximately $r = 200$ AU, a longitude of $l = 105.5$ deg longitude, and a latitude of $b = 7.5$ deg [100]. Within this region, the then-subsonic solar wind hits the local interstellar medium (LISM). Complex interaction of electromagnetic fields decelerates the solar wind particles, which results in the so-called bow shock wave, which can be seen in Fig. 6.17. To prove this model and its validity¹, it is necessary to probe this region using a scientific deep space probe. The key scientific questions of such a mission would be [53]:

¹Zank [101] gives more information on these models.

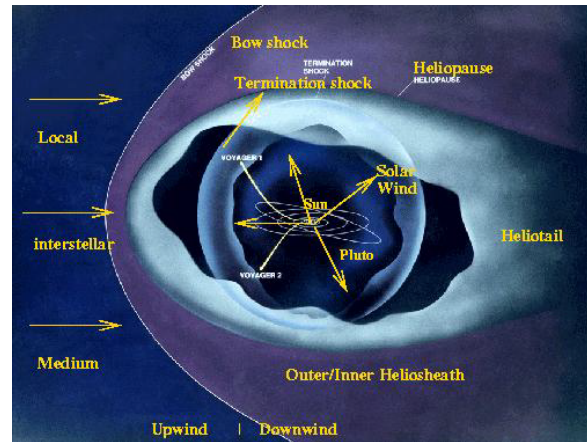


Figure 6.17: The Heliosphere in the Local Interstellar Medium. The main components of the heliosphere are the inner and outer heliosheath, the bow shock, the termination shock, the heliotail and the heliopause. The solar system is moving from right to left, and this movement presumably generates a bow shock through electromagnetic interaction of the local interstellar medium with the outer heliosheath. This figure also shows the flight path and approximate distances of the Voyager deep space probes, the two man-made objects that have crossed the largest distance so far. They are however useless for investigations of the bow shock for three reasons: first, their instruments were developed for investigations of planetary magnetospheres; second, their flightpath will not lead them towards the bow shock; and, third, they will run out of energy before reaching the required distances. For Voyager 1, this will be latest in 2020 at a distance of approximately 148 AU. Courtesy ESA [53].

- What are the characteristics and precise location of the termination shock and the heliopause, and what is the temporal variation?
- How are (anomalous) cosmic rays accelerated in the heliosheath?
- What are the processes in the heliosheath that (possibly) affect the characteristics of the interstellar neutral gas and dust as well as galactic cosmic rays interactions?
- What are the characteristics of the local interstellar medium beyond the heliopause?

Answering these questions has been prevented by the large distance and the available propulsion technologies. However, the development of new propulsion techniques, like EP and solar sails, offer viable options, and NASA and ESA have included such interstellar heliopause probe (IHP) missions into their strategic programs. The set maximum allowed mission durations are 25 yrs for ESA missions and 15 yrs for NASA missions, respectively. The corresponding average velocities are 8 AU/yr and 13.3 AU/yr. Solar sailing is a promising concept, as it theoretically provides the required ΔV . This was investigated in past IHP studies by Liewer et al. [50] and Lyngvi et al. [53]. Other studies by Bramanti et al. [11] concentrated on very high-power SEP or NEP propulsion for that transfer type. The use of REP in combination with hyperbolic excess energy and gravity assists was investigated as well, for example by McNutt et al. [56] and Wimmer-Schweingruber et al. [100]

In the course of a study [51], a team including the author investigated whether this transfer is feasible with a combination of SEP and REP propulsion technology. The maximum flight time was set to the ESA Δt -requirement of 25 yrs. Only the optimization of the trajectory with InTrance shall however be treated here, and the reader is referred to [51, 52, 64] for more details on this particular IHP mission analysis.

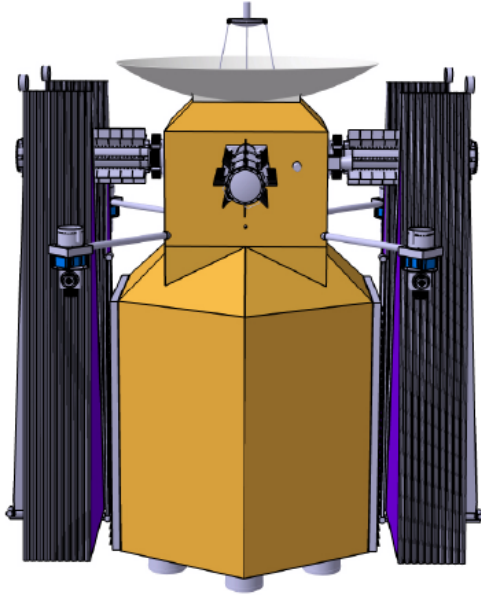


Figure 6.18: Interstellar Heliopause Probe with Folded Solar Power Generator Panels. [64]

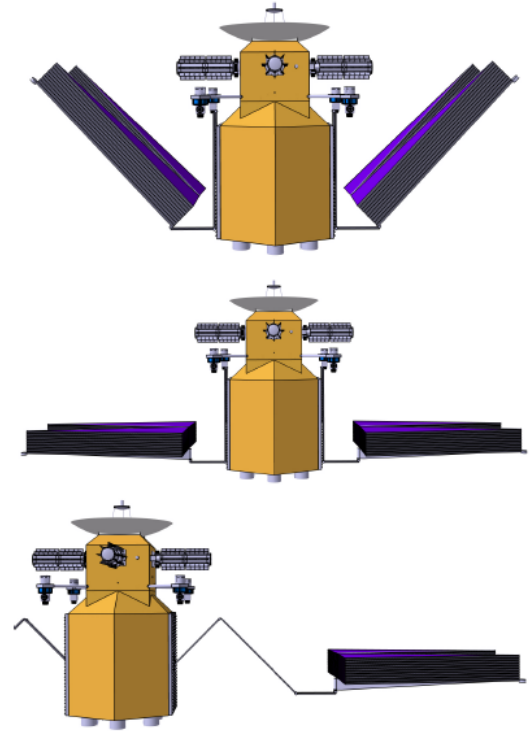


Figure 6.19: Interstellar Heliopause Probe with Solar Power Generator Panels Unfolding. [64]

6.5.1 Mission Design and Optimization Setup

The objective of the IHP mission is to leave the solar system towards the heliosphere bow shock with a velocity that allows for latest arrival there within maximum 25 yrs. The mission should use SEP for primary propulsion, with the option of a GA at a major celestial body, if necessary. Potentially, a second, REP-powered stage may be used for further acceleration to achieve the required terminal velocity.

The spacecraft design consisted of the probe itself, a SEP stage for primary propulsion, and an optional secondary REP stage, both using Xenon as propellant. The SEP stage comprised six RIT-22 engines with $I_{sp}=7377$ s and a solar-electric power generation subsystem with $P_e=53$ kW. At distances greater than three AU, the EPGS output is insufficient to supply the SEP stage. Mission design therefore foresaw the jettison of that first propulsion stage and to utilize the second one, which contained four RTG units with a total BOL power output of 648 W, enough to operate the RIT-10 unit with an input power $P_T=592$ W. The resulting thrust is 21 mN. An illustration of the spacecraft is given in Fig. 6.18 and Fig. 6.19. From a flight dynamics point of view, the mission's trajectory requirement corresponds to a flyby scenario at a fix point in space. That point is defined through a vector defined with the spherical coordinates $r = 200$ AU, $l = 105.5$ deg, and $b = 7.5$ deg in the heliocentric ecliptical frame. For the investigation on whether SEP is a suited propulsion option for an IHP mission or not, gravity assists were neglected first. With this assumption, and with the Earth as launch body, one could simplify the problem even more and use an orbit flyby as target condition. This means that InTrance had to find a trajectory on which the IHP spacecraft hits an ecliptical, circular, 200 AU orbit with $\Delta r_{max} \leq 10 \cdot 10^6$ km and $\Delta t_{max} \leq 25$ yrs. To save mass, the IHP design did not include a thermal control subsystem specifically designed

Table 6.10: Data of the fastest IHP transfer, including a gravity assist at Jupiter.

	Unit	Value	
Launch date t_0	MJD	58 254.59	(17 May 2018)
m_0	kg	1 692	
m_p	kg	440	
C_3	km^2/s^2	45.1	
Δt_{SEP}	d	831	(1.31 yrs)
r_{SEP}	AU	3.05	
v_{SEP}	km/s	30.5	(6.44 AU/yr)
v_{J-}	km/s	26.8	(5.65 AU/yr)
v_{J+}	km/s	39.3	(8.29 AU/yr)
d	R_J	1.34	
t_{REP}	MJD	59 225.48	(11 Jan 2021)
Δt_{REP}	d	3 283	(8.98 yrs)
r_{REP}	AU	79.60	
v_{REP}	km/s	47.4	(10 AU/yr)
t_f	MJD	66 929.80	(15 Feb 2042)
Δt	d	8 675	(23.8 yrs)

for operations near the Sun. An allowed minimum Sun-spacecraft-distance of 0.7 AU was therefore set as an optimization boundary condition. A NC steered the IHP during the first, SEP-propelled mission phase on a flyby course towards Jupiter. During the second phase, the REP-acceleration phase, the thrust pointed in flight direction. This decision based on the fact, that the acceleration of the REP-stage was insufficient for a significant modification of the direction of the velocity vector, even if it was applied for years. Consequently, the spacecraft had to be on its final course already before this phase started. InTrance was configured to determine the required propellant mass and to utilize the excess energy of the foreseen launcher Ariane 5 ECA. The combination of both features should find the mission-dependent optimal compromise between launcher-provided ΔV and the ΔV resulting from operation of a EP-stage.

6.5.2 Results and Discussion

Table 6.10 shows the resulting figures with $\Delta t=23.8$ yrs significantly below the 25 yrs-limit. Figure 6.20 and Fig. 6.21 show the associated trajectory plots in ecliptical heliocentric inertial (HCI)-frame. The launcher Ariane 5 ECA provided an additional C_3 of $45.1 \text{ km}^2/\text{s}^2$, which corresponds to $v_\infty=6.72 \text{ km/s}$. This additional ΔV is used to raise the aphelion. The thrust of the SEP-stage then lowers the perihelion while the spacecraft traverses to its current orbit's aphelion. This way, the spacecraft has maximum electrical power during the following perihelion passage and can operate its propulsion subsystem with maximum thrust. The launch mass m_0 of 1692 kg contains 440 kg of Xe propellant, and the Jupiter gravity assist (EGA) provided an inertial ΔV of 12.5 km/s (2.64 AU/yr). The flyby distance from Jupiter was only 1.34 Jupiter radii. Operation of the REP-stage ceased after approximately 10 yrs, corresponding to a mission-elapsed-time (MET) of 11.6 yrs. The IHP then traverses on a ballistic trajectory towards the target with a heliocentric, inertial velocity of 10 AU/yr. At the time of REP-burnout at $r=80$ AU, the distance to the target is 120 AU.

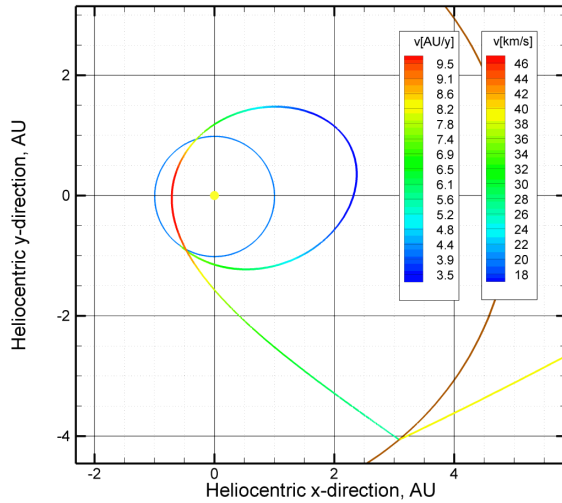


Figure 6.20: SEP+REP Transfer to 200 AU with a Swing-by at Jupiter.

The influence of the flyby maneuver, i.e., the increase in heliocentric velocity, is visualized in the sudden trajectory color change at Jupiter.

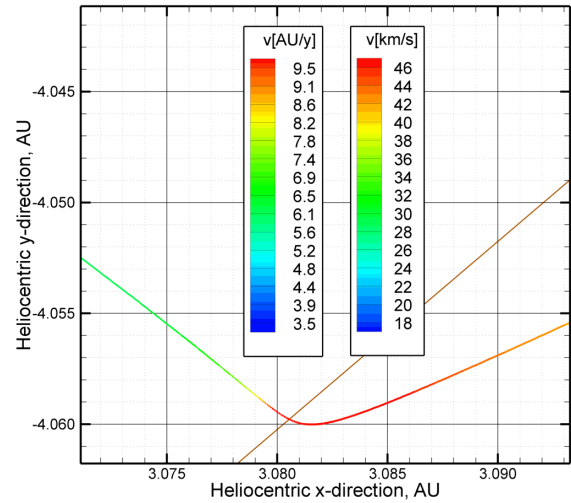


Figure 6.21: Close-up of the IHP Gravity Assist at Jupiter.

The color coding of the trajectory shows how the spacecraft's velocity is increased when passing by behind Jupiter, whose trajectory is shown in brown.

The figures of Tab. 6.10 show that this mission satisfies the posed requirements. Due to the required relative positions of the target and Jupiter, suited launch opportunities do however exist only once per decade. The JGA in fact reduces mission flexibility and thus increases mission execution risk through eventual missing the required launch date. The author et al. [64] showed that a pure SEP+REP mission, based on already or soon existing technologies, can achieve a flight time of 27.5 yrs. A reduction of the mission execution risk is therefore possible through relaxation of the Δt -limit by approximately 10 percent. This results in a 28-yr-limit, but the benefit would be more frequent launch opportunities. That is because the launch date only depends on the relative position of the Earth to the heliopause nose. This geometry repeats more often than for IHP mission designs that include gravity assists at outer planets. More frequent launch opportunities and risk reduction could potentially compensate for increased mission times.

Conclusions

This chapter concludes this thesis by summarizing the problem statement and the novel contributions to research and by providing recommendations and an outlook to further research fields. It describes the problem, lists the obtained results, and gives a summary about what has been done to achieve these results. The thesis ends with a discussion of still unsolved problems and recommendations of associated research fields worth while investigation with the presented methods.

7.1 Summary

This subchapter describes the problem handled in this thesis. Then the newly developed method to solve that problem is briefly described, followed by a description of the results that were obtained in the validation and application process.

7.1.1 Problem Description and State-of-the-Art Solutions

Deep space missions are essential for the exploration of our solar system and beyond. The scientific interest thereby shifts to more and more distant objects. Space missions to the outer planets (Cassini-Huygens), to asteroids (Hayabusa), and to comets (Rosetta) are examples, as well as solar polar orbiter (Ulysses) or solar-system-escape missions (Voyager, New Horizons). Maximizing the scientific return of costly deep space missions also necessitates the investigation of more than a single object per mission, if possible. NASA's Dawn mission achieved this by visiting the two asteroids Vesta and Ceres. As a consequence, the required velocity increment of such deep space missions is constantly growing.

Up to a certain limit, the generation of the required velocity increment is achievable with chemical propulsion. That limit is determined by the energy contained in the propellant of the launcher or carried on-board the spacecraft. Therefore, this technology often cannot provide the required orbit energy completely or the remaining usable spacecraft mass would be insufficient for reasonable science. To enable deep space missions with chemical propulsion, the respective mission design consequently must include externally-provided additional energy. This is possible through so-called gravity assists, which are flybys at celestial bodies at a distance close enough to bend the trajectory of the spacecraft through gravitational attraction. Planned and carried out correctly, a spacecraft can gain or loose orbital energy from the flyby body. Including gravity assists into a mission however constrains the design of the transfer trajectory. This affects the mission itself because the realization of a gravity assists depends on several factors. These are the velocity of the spacecraft and the relative

positions of the assisting body and the target at the time of the flyby. Multiple gravity assists can prolong a mission's duration further because of the required cruise phases to meet the desired subsequent flyby conditions. Depending on the constellation of the involved bodies, this can be achieved only every few years. The effect on possible launch dates in case of program schedule delays can be severe. Despite the provision of the needed additional orbit energy, gravity assists can thus be inapplicable for a particular mission.

An alternative to including gravity assists into mission design for higher velocity increments is the use of a more capable propulsion technology. Low-thrust propulsion is such a technology, with electrical propulsion being the most prominent one. The exhaust velocities of electrical thrusters are not inherently limited and primarily dependent on the electrical input power. The propellant efficiency of electric propulsion systems, expressed with the specific impulse, is therefore significantly higher than the one of chemical propulsion systems. Advanced low-thrust propulsion concepts, such as solar sails, even operate propellantless through the generation of a thrust force solely from solar radiation pressure. Low-thrust propulsion can therefore enable the described high-energy mission types without or with fewer gravity assists. This reduces or eliminates mission design constraints resulting from these flybys. The benefit of a high thrust efficiency of low-thrust propulsion comes along with a very small absolute thrust level. It ranges from of a few milli-Newtons to a few Newtons and is significantly less than the kilo-Newtons of thrust force chemical propulsion can provide. High-thrust applications, such as a launch from the Earth or a planetary capture, are consequently infeasible for low-thrust propulsion. Its usefulness for propellant-efficient transfers in planetary and interplanetary space is nevertheless obvious.

The different thrust characteristics of chemical and low-thrust propulsion affect the design of transfer trajectories and its optimization w.r.t. criteria like flight time or propellant consumption. High-thrust trajectories can be regarded as pure free-flight phases with instantaneous velocity changes at the begin and the end of a transfer and a few small midcourse correction maneuvers. In its simplest case, an optimal chemical transfer trajectory only requires an optimal injection velocity vector at the optimal launch time, which results in four variables that must be optimized. Various optimization methods have been successfully developed for this purpose. They range from gradient-based local optimization methods, often based on Newton iteration schemes, to heuristic and probability-based global optimization methods, like simulated annealing, ant simulation, particle swarm optimizers, and genetic algorithms. Local optimization methods are precise, deterministic and their mathematical foundations well understood. This means, if there is an optimum near the current solution, then such methods will find it with high fidelity. Their need for an initial guess solution is, however, a known drawback, as well as their inability to escape from any potentially existing local optimum. Global optimization methods can, by principle, escape local optima in solution space, although it is not guaranteed. Furthermore, they need no initial guess solution, which is a considerable advantage over local optimization methods. Especially for complex transfers, such an initial guess is difficult to generate and often requires a team of flight mechanics and optimization experts.

Unlike the quasi-instantaneous flight path changes caused by chemical propulsion, the orbital elements of low-thrust-propelled spacecraft change permanently through the continuously applied thrust acceleration. Assuming three-dimensional local optimal thrust vectors and depending on the chosen time-discretization, this can lead to thousands of variables to optimize. A few local optimization methods have been adapted for this purpose. Common to most of them is a resulting set of many equations. Arranged in a matrix and provided with an initial-guess solution sufficiently close to the optimum, these optimization methods find the optimum through variation techniques. Stochastic methods have been applied as well to low-thrust trajectory optimization. One of them is Evolutionary Neurocontrol, which was

used by Dachwald [13] to optimize single-phase, interplanetary, low-thrust transfer problems. It combines an optimal control mechanism called Artificial Neural Networks with the optimization method Evolutionary Algorithms for the training of neural network parameter sets. Evolutionary neurocontrol uses artificial neural networks for the steering of a spacecraft. That means it derives the necessary control variables from a set of problem-specific input variables according to the strategy stored within its internal parameters. An evolutionary algorithm then searches for an optimal strategy to solve the particular transfer problem. The implementation of ENC resulting from Dachwald's work is a computer program named Intelligent spacecraft Trajectory Optimization through Evolutionary Neurocontrol (InTrance).

A deep space mission however often comprises more than only a single flight phase. Dawn, for example, features a rendezvous transfer from the Earth to Vesta and, after finishing of science operations at Vesta, a second rendezvous transfer to the final destination Ceres. This rather simple multiphase transfer shows the problems associated with global optimization of multiphase low-thrust transfers. For the optimization of such a mission design w.r.t. short mission duration, for example, it does not suffice to optimize the two low-thrust rendezvous transfers for a short transfer duration. The launch dates at the Earth and Vesta must be chosen optimally as well. Whether it is favorable for a minimum total mission duration to concentrate on the reduction of the flight time of the first flight leg or of the second is principally unknown beforehand. The dwell time at the first mission target as well as the propellant mass introduce further variables that contributes to mission duration and must be accounted for in a global optimization method. Moreover, an optimization of the first transfer requires knowledge of the propellant mass required for the second transfer. In summary, global optimization of high-thrust and single-phase low-thrust transfers is no problem for existing optimization schemes and methods. However, the problem of finding a generic methodology for true global optimization of multiphase low-thrust trajectories therefore remained unsolved so far.

7.1.2 Approach and Results

The starting point for the development of a methodology called Multiphase Evolutionary Neurocontrol (MENC) for the global optimization of multiphase low-thrust trajectories was Evolutionary Neurocontrol and its implementation for heliocentric single-phase transfers InTrance. This software was substantially revised and extended to support that objective. The core component of trajectory simulation, which serves as the evaluation element of the Evolutionary Algorithm, was redesigned from a single-phase simulation to a multiphase simulation framework. The basic approach was thereby to implement each mission phase through a specific simulation object. Each of these objects contained a steering strategy incorporated by a Neurocontroller. As in the real transfer, each simulation was also linked to the simulation of the preceding phase and the subsequent phase. Coupling conditions assured physical integrity at the boundary of two phases in terms of mass, time, position, and velocity variables. To support also non-heliocentric mission analysis, the simulation implementation was further extended with new features. Third-body disturbances, which are not negligible for Earth-Moon transfers, for example, were implemented. As low-altitude orbits about the Earth or another planet are affected by shadowing, and as this has a crucial effect on solar-electric propulsion systems, models of planetary shadows were implemented as well. Using Neurocontrollers for spacecraft steering reduced the number of variables that had to be optimized. Together with the starting conditions of each phase, those variables were encoded on a string. The existing Evolutionary Algorithm of InTrance then solved the problem through concurrent optimization of the steering strategies and initial conditions of each mission phase.

Newly implemented features as well as the entire framework were verified before application for preliminary mission analysis. This was achieved in several validation stages. This means

validation was carried out starting with simple tests of elementary elements or components of methodology, such as the Evolutionary Algorithm. The successful component tests were followed by recalculation of single-phase non-heliocentric transfers from literature and comparison with the published results. InTrance was therefore used to recalculate several low-thrust orbit transfers about the Earth and other celestial bodies. The results obtained by InTrance showed in all cases comparable or even superior results. The next stage was the validation of the multiphase framework. Two examples of low-thrust transfers with each having two flight phases were chosen for that purpose: a SMART-1-like solar-electric transfer from an Earth-bound orbit into an orbit about the Moon and the Dawn scenario of an interplanetary double rendezvous. In the first case, the transfer trajectory was split into two phases. The first one lead from an orbit about the Earth to a handover point near the Moon's sphere of influence. From there, the second phase lead to the final orbit about the Moon. To achieve a flight time-optimal transfer, InTrance therefore had to optimize the steering strategy of both flight legs, the handover point, and the propellant masses. The final validation was the recalculation of the Dawn mission, excluding the gravity assist at Mars, which is part of the real mission design. This simplification can be done, as it was included for mission margin reasons and not due to orbit energy requirements. InTrance found basically the same results using the original mission parameters, i.e., power, masses, dates. In the case of that transfer, a strictly quantitative assessment is difficult because of the neglected gravity assist and due to intended purpose of InTrance. Intended for preliminary mission analysis, features such as duty cycles, i.e., times reserved other activities than for thruster operation, are not modeled. The real Dawn trajectory however must and does include such times, causing the respective effect on the transfer. A final aspect should be highlighted in the context of this validation case. While the real transfer was developed and optimized by a team of flight dynamics and optimization experts, the trajectory presented in this thesis was calculated by the improved version of InTrance. The author only had to provide a configuration with mission describing parameters, e.g., masses, launch windows, propulsion system technology, etc.

The developed methodology was finally applied to a challenging multiphase low-thrust mission scenario. This mission design, called Interstellar Heliopause Probe (IHP), should go to a distance of 200 AU in less than 25 years. Its transfer trajectory consisted of two flight phases and included a change of the propulsion system as well as gravity assist at Jupiter. While the first transfer leg until the flyby at Jupiter was achieved with a solar-electric propulsion system, the second flight leg after the gravity assist until end of mission was carried out using a nuclear-thermal-electric propulsion system. The simple steering strategy of always thrusting along the current velocity vector was chosen for the second flight leg. InTrance was configured to optimize the entire mission design, i.e., launch date, the launcher-provided hyperbolic excess velocity vector at orbit injection, the needed propellant mass, the steering strategy until burnout of the first propulsion stage and the optimal gravity assist at Jupiter. The resulting transfer lasted 23.8 yrs, well below the set limit of 25 yrs. This was achievable only through including the gravity assist at Jupiter. It must be mentioned that InTrance cannot use such maneuvers in a robust manner usable for mission analysis. For IHP, this was possible only by configuring InTrance for a very close flyby at Jupiter and accounting for its third-body perturbation during the heliocentric transfer. Only the masses of such large objects can influence the flight path of spacecraft already at very large distances. This explains, why planets like Mars cannot be used for gravity assists with the current implementation.

The results obtained in the course of the validation and preliminary mission analysis work show that the developed method Multiphase Evolutionary Neurocontrol globally optimizes a multitude of different heliocentric and non-heliocentric, multiphase low-thrust transfers. It does not require an initial guess solution and, based on the user-provided configuration, optimizes a mission design as a whole, i.e., including departure dates, handover points, and

propellant masses, if required. In doing this, Multiphase Evolutionary Neurocontrol eliminates the deficiencies of other optimization methods. Despite this improvement, a few mission types remain difficult to optimize or cannot be optimized yet with this method. The reason is the missing robust support of gravity assists within InTrance. Therefore mission designs requiring such flybys, especially at smaller planets like the Earth or Mars, to achieve the necessary velocity increment, generally cannot be optimized with the current implementation of Multiphase Evolutionary Neurocontrol. Exception to this statement are flybys at very large celestial objects, as has been demonstrated with the IHP mission design.

7.2 Recommendations for Further Work

This thesis showed the successful extension of Evolutionary Neurocontrol to the optimization of multiphase low-thrust transfers. However, a few questions remain unanswered and not all of its capabilities are yet explored. Section 7.2.1 therefore recommends topics for further fundamental research in the fields of Artificial Neural Networks and Evolutionary Algorithms, while section 7.2.2 gives more practical recommendations to improve the application of Evolutionary Neurocontrol to the optimization of low-thrust spacecraft trajectories.

7.2.1 Fundamental Research

An interesting research topic is the search for the answer to the question of the optimal NC input data set. The NCs within ENC and MENC were provided with a set of astrodynamical state information of the spacecraft and the target. Although the NCs solved the respective problems, it is unclear if another, maybe smaller, input data set would have solved the problem as well. Most likely, the optimum data set has no or only little physical meaning and is a yet unknown combination of problem parameters. Starting with a particular transfer problem, ENC could be useful for the determination of such a minimum data set for that transfer. Subsequent analysis of these input parameters could provide more knowledge into the nature of this abstract input data set and which physical input parameters have the most influence on the output. The extension from one transfer type to further mission types, e.g., flyby and rendezvous transfers to varying targets, also with different inclinations, could lead to an input data set appropriate for the generic description of transfer problems.

Similar to the search for the optimal input data set is the search for an ANN that can solve any low-thrust transfer of a particular type. Taking a flight-time-minimal RV-type transfer as an example and assuming the transfer is feasible, such an ANN would be capable of steering any low-thrust spacecraft from one planet to another planet without further training. Such an ANN would necessarily have generalized best and “understood” the underlying nature of the transfer problem. InTrance can help on such investigations with its support for multiple trajectory integrations with a single NC.

Trying to find for the smallest ANN capable of achieving the optimization task is another recommendation for further fundamental research on this AI element. An approach to achieve this might be the concurrent training of two separate NCs in addition to the one of a regular trajectory optimization with ENC. The first NC thereby features an additional neuron in its hidden layer than the regular NC, and the other NC has one less. As a training data set is available from the current best solution, back-propagation should allow parameter optimization of the additional NCs. The ANN with more neurons can then be used if the current one does not achieve the boundary constraints. The smaller one is consequently taken if it matches the boundary constraints and achieves or supersedes the objective function value of the original ANN. This approach requires feed-forward ANNs with differentiable neuron

transfer functions. Following the sketched approach, an on-line optimization of not only the ANN parameters but also its structure seems promising.

Besides research related to ANNs that reside within the NC, also the training algorithm, the GA, offers further research topics. The current algorithm is stable and robust. However, it is not clear whether further adaption to natural evolution might improve its behavior even more, i.e., reduce runtime through early and assured convergence to a the global optimum. Natural evolution of many life forms relies on diploid chromosomes but most GA implement only haploid genome structures. Using diploid chromosomes and deciding upon reproduction on which chromosome half to pass on to the offspring might help to keep up genetic diversity and thus improve convergence behavior.

7.2.2 Application-oriented

This section describes several recommendations related to the application of ENC to the analysis of space missions that involve low-thrust-propelled spacecraft. Further research can address several topics, such as usability, robustness, features, performance and reliability, and only a few can be listed here.

First, the suboptimal accuracy inherent to global optimization methods such as EAs might be alleviated through a combination with local optimizing techniques. For example, a solution obtained through InTrance could serve as initial guess solution to a local optimizer, which then achieves final accuracy. There may be however practical limits to this approach because some control vector histories are difficult to optimize for analytical methods. Instantaneous changes of parameter values, for example, are difficult to model for methods that require derivatives.

InTrance is constrained to the optimization of predefined mission scenarios, i.e., a user must specify target, trajectory type, dates, and spacecraft's capabilities. Automatic optimization of multiphase missions for which only the targets but not their sequence is known is currently not possible. Combining InTrance with a heuristic method that solves that combinatorial problem beforehand would be an approach to find global-optimal solutions to such problems. This would allow the increase of scientific return of a mission as the spacecraft, while on its way to the primary target, could conduct as many asteroid flybys as possible without spending significant additional propellant.

Within the work for this thesis only mission-specific objective functions were used, i.e., they were valid for the entire mission and not phase-specific. Dropping this assumption would allow, for example, to optimize one mission phase for minimum transfer time and another one for minimum propellant consumption. The assessment of the final fitness of a solution candidate is however not easy and often strongly mission dependent.

The intended application of InTrance is the support of mission analysis on Phase-A study level, and its current capabilities are sufficient for that purpose. Nevertheless, the extension with operational constraints would enhance its usefulness even more. For example, spacecraft on deep space missions must regularly communicate with its ground segment. This is not a problem for a chemical missions. For low-thrust-propelled spacecraft, however, with its long thrust phases, it certainly has an effect, as the spacecraft may have to be turned towards the Earth. During that communication time, it cannot point the thrust vector in the direction required for optimal trajectory change. This is expressed in a so-called duty cycle, which is the relative fraction of a day during which thrusting is allowed. This cannot be accounted for with InTrance. The effect resulting from such limitations on various mission characteristics, such as total mission duration, can thus not yet be determined.

The spacecraft model implemented in InTrance is designed for phase-A level study support. Complex spacecraft models are not supported. Improved models of the different spacecraft

subsystems would be beneficial for higher simulation accuracy. The implemented ion engine model, for example, lacks the feature of a maximum total impulse. In other words, every engine can operate as long as it is supplied with propellant, which is impossible in reality. Restricting an engine's spent impulse creates new constraints on the optimization because the load of thrust generation should be evenly shared among the engines of a propulsion stage. Using different engine types, with different I_{sp} -values, offers further optimization possibilities. An intelligent scheduling algorithm, for example, could select the engine(s) to use according to the current astrodynamics environment, the available electrical power, and the spent total impulse. A spacecraft with such intelligent propulsion subsystem control could therefore utilize the most efficient engine type when required electrical power is available and the more powerful but propellant-inefficient ones at times of limited electrical power.

Appendix A

Coordinate Frames

A.1 Inertial Cartesian Frame

The Cartesian reference frame $\mathcal{J} : (\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ is right-handed positively defined through its orthogonal unit vectors \mathbf{e}_x , \mathbf{e}_y , and \mathbf{e}_z . A vector \mathbf{r} expressed in cartesian frame coordinates is therefore described with its projections along these base vectors. The XY-plane of the inertial heliocentric cartesian reference frame $I : (\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ is defined by the Earth's mean orbit plane and the x-axis by the vernal equinox direction of epoch J2000 [58]

$$\begin{aligned}\mathbf{r} &= r\mathbf{e}_r = r[(\mathbf{e}_r\mathbf{e}_x)\mathbf{e}_x + (\mathbf{e}_r\mathbf{e}_y)\mathbf{e}_y + (\mathbf{e}_r\mathbf{e}_z)\mathbf{e}_z] \\ &= r \cos(\mathbf{e}_r, \mathbf{e}_x) \mathbf{e}_x + r \cos(\mathbf{e}_r, \mathbf{e}_y) \mathbf{e}_y + r \cos(\mathbf{e}_r, \mathbf{e}_z) \mathbf{e}_z \\ &= x\mathbf{e}_x + y\mathbf{e}_y + z\mathbf{e}_z = \begin{pmatrix} x \\ y \\ z \end{pmatrix}.\end{aligned}\tag{A.1}$$

The base vectors are invariant w.r.t. time and therefore their time derivatives are zero

$$\frac{d\mathbf{e}_x}{dt} = \frac{d\mathbf{e}_y}{dt} = \frac{d\mathbf{e}_z}{dt} = 0.\tag{A.2}$$

The velocity vector as the positions vector's first time derivative is computed as

$$\begin{aligned}\mathbf{v} = \dot{\mathbf{r}} &= \frac{d\mathbf{r}}{dt} = \frac{d(r\mathbf{e}_r)}{dt} = \frac{d(x\mathbf{e}_x + y\mathbf{e}_y + z\mathbf{e}_z)}{dt} = \frac{d(x\mathbf{e}_x)}{dt} + \frac{d(y\mathbf{e}_y)}{dt} + \frac{d(z\mathbf{e}_z)}{dt} \\ &= \left(\frac{dx}{dt}\mathbf{e}_x + x\frac{d\mathbf{e}_x}{dt}\right) + \left(\frac{dy}{dt}\mathbf{e}_y + y\frac{d\mathbf{e}_y}{dt}\right) + \left(\frac{dz}{dt}\mathbf{e}_z + z\frac{d\mathbf{e}_z}{dt}\right) \\ &= \dot{x}\mathbf{e}_x + \dot{y}\mathbf{e}_y + \dot{z}\mathbf{e}_z = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix}.\end{aligned}\tag{A.3}$$

The acceleration vector, i.e., the positions vector's second derivative is computed to

$$\mathbf{a} = \dot{\mathbf{v}} = \frac{d\mathbf{v}}{dt} = \ddot{\mathbf{r}} = \frac{d^2\mathbf{r}}{dt^2} = \ddot{x}\mathbf{e}_x + \ddot{y}\mathbf{e}_y + \ddot{z}\mathbf{e}_z = \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix}.\tag{A.4}$$

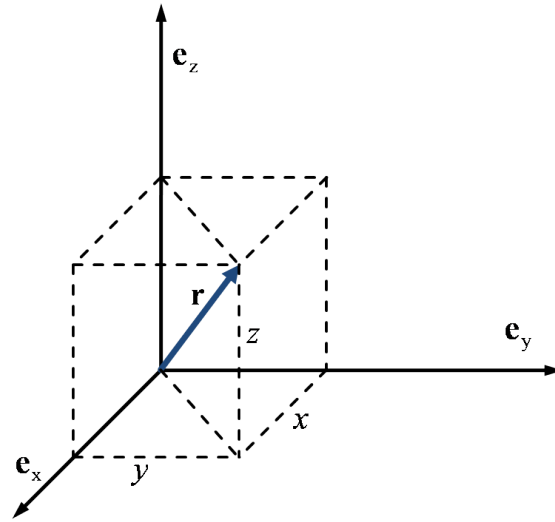


Figure A.1: The Cartesian Reference Frame \mathcal{J} . It is defined through its base vectors \mathbf{e}_x , \mathbf{e}_y , and \mathbf{e}_z . A vector \mathbf{r} is thus the vector sum of these base vectors multiplied with the components of \mathbf{r} .

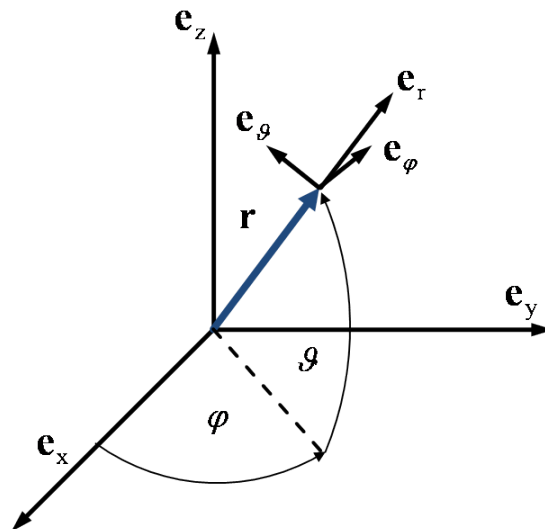


Figure A.2: The Polar Reference Frame $\mathcal{P} : (\mathbf{e}_r, \mathbf{e}_\varphi, \mathbf{e}_\vartheta)$ is defined through its base vectors \mathbf{e}_r , \mathbf{e}_φ and \mathbf{e}_ϑ . A vector \mathbf{r} is expressed with the vector length or magnitude r , the azimuth angle $\varphi \in (0, 2\pi]$, and the elevation angle $\vartheta \in (-\frac{\pi}{2}, +\frac{\pi}{2})$.

A.2 Polar Reference Frame

The right-handed, orthogonal polar reference frame $\mathcal{P} : (\mathbf{e}_r, \mathbf{e}_\varphi, \mathbf{e}_\vartheta)$ is defined with the orthonormal base vectors \mathbf{e}_r , \mathbf{e}_φ , and \mathbf{e}_ϑ , which can be described in cartesian frame coordinates

$$\mathbf{e}_r = \begin{pmatrix} \cos \varphi \cos \vartheta \\ \sin \varphi \cos \vartheta \\ \sin \vartheta \end{pmatrix} \quad (\text{A.5})$$

$$\mathbf{e}_\varphi = \begin{pmatrix} -\sin \varphi \\ \cos \varphi \\ 0 \end{pmatrix} \quad (\text{A.6})$$

$$\mathbf{e}_\vartheta = \begin{pmatrix} -\cos \varphi \sin \vartheta \\ -\sin \varphi \sin \vartheta \\ \cos \vartheta \end{pmatrix}. \quad (\text{A.7})$$

A vector \mathbf{r} is defined $\mathbf{r} = r\mathbf{e}_r$. The derivatives of the base vector are

$$\dot{\mathbf{e}}_r = \frac{d\mathbf{e}_r}{dt} = \begin{pmatrix} -\dot{\varphi} \sin \varphi \cos \vartheta - \dot{\vartheta} \cos \varphi \sin \vartheta \\ \dot{\varphi} \cos \varphi \cos \vartheta - \dot{\vartheta} \sin \varphi \sin \vartheta \\ \dot{\vartheta} \cos \vartheta \end{pmatrix} = \dot{\varphi} \cos \vartheta \mathbf{e}_\varphi + \dot{\vartheta} \mathbf{e}_\vartheta \quad (\text{A.8})$$

$$\dot{\mathbf{e}}_\varphi = \frac{d\mathbf{e}_\varphi}{dt} = \begin{pmatrix} -\dot{\varphi} \cos \varphi \\ -\dot{\varphi} \sin \varphi \\ 0 \end{pmatrix} = -\dot{\varphi} (\cos \vartheta \mathbf{e}_r - \sin \vartheta \mathbf{e}_\vartheta) \quad (\text{A.9})$$

$$\dot{\mathbf{e}}_\vartheta = \frac{d\mathbf{e}_\vartheta}{dt} = \begin{pmatrix} \dot{\varphi} \sin \varphi \sin \vartheta - \dot{\vartheta} \cos \varphi \cos \vartheta \\ -\dot{\varphi} \cos \varphi \sin \vartheta - \dot{\vartheta} \sin \varphi \cos \vartheta \\ -\dot{\vartheta} \sin \vartheta \end{pmatrix} = -\dot{\varphi} \sin \vartheta \mathbf{e}_\varphi - \dot{\vartheta} \mathbf{e}_r. \quad (\text{A.10})$$

The velocity vector \mathbf{v} is

$$\begin{aligned} \mathbf{v} = \dot{\mathbf{r}} &= \frac{d(r\mathbf{e}_r)}{dt} = \dot{r}\mathbf{e}_r + r\dot{\mathbf{e}}_r = \dot{r}\mathbf{e}_r + r(\dot{\varphi} \cos \vartheta \mathbf{e}_\varphi + \dot{\vartheta} \mathbf{e}_\vartheta) \\ &= \dot{r}\mathbf{e}_r + r\dot{\varphi} \cos \vartheta \mathbf{e}_\varphi + r\dot{\vartheta} \mathbf{e}_\vartheta \\ &= \begin{pmatrix} \dot{r} \cos \varphi \cos \vartheta - r\dot{\varphi} \sin \varphi - r\dot{\vartheta} \cos \varphi \sin \vartheta \\ \dot{r} \cos \varphi \sin \vartheta + r\dot{\varphi} \cos \varphi - r\dot{\vartheta} \sin \varphi \sin \vartheta \\ \dot{r} \sin \vartheta + r\dot{\vartheta} \cos \vartheta \end{pmatrix}. \end{aligned} \quad (\text{A.11})$$

The acceleration \mathbf{a} is derived from \mathbf{r} and \mathbf{v} with (A.8)- (A.10):

$$\begin{aligned} \mathbf{a} = \dot{\mathbf{v}} = \ddot{\mathbf{r}} &= \frac{d}{dt} (\dot{r}\mathbf{e}_r + r\dot{\varphi} \cos \vartheta \mathbf{e}_\varphi + r\dot{\vartheta} \mathbf{e}_\vartheta) \\ &= (\ddot{r} - r\dot{\vartheta}^2 - r\dot{\varphi}^2 \cos^2 \vartheta) \mathbf{e}_r \\ &+ (2\dot{r}\dot{\varphi} \cos \vartheta + r\ddot{\varphi} \cos \vartheta - 2r\dot{\varphi}\dot{\vartheta} \sin \vartheta) \mathbf{e}_\varphi \\ &+ (2\dot{r}\dot{\vartheta} + r\ddot{\vartheta} + r\dot{\varphi}^2 \cos \vartheta \sin \vartheta) \mathbf{e}_\vartheta \end{aligned} \quad (\text{A.12})$$

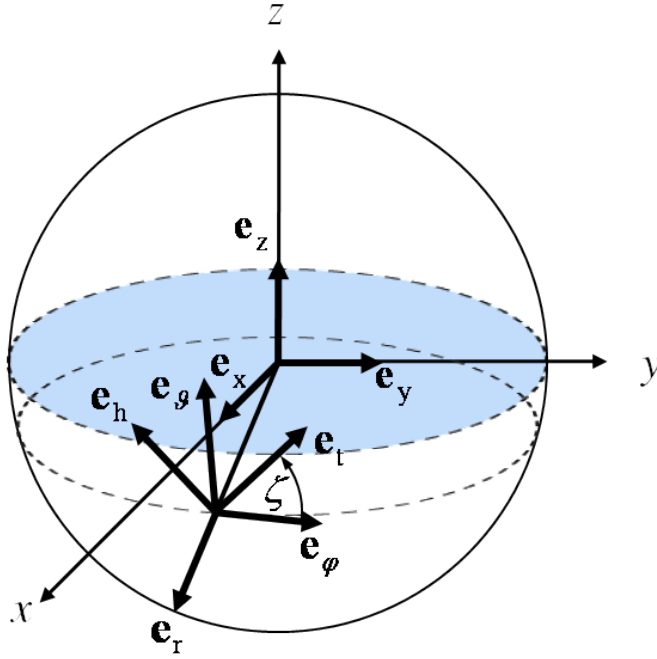


Figure A.3: The Orbit Reference Frame \mathcal{O} . The orbit angle ζ is between \mathbf{e}_t and \mathbf{e}_φ .

A.3 Orbit Reference Frame

The orbit reference frame $\mathcal{O} : (\mathbf{e}_r, \mathbf{e}_t, \mathbf{e}_h)$ offers easier description of a spacecraft's translational motion, as it has two unit vectors in the orbit plane. It derives from the \mathcal{P} -frame through a rotation about \mathbf{e}_r with the rotation angle ζ

$$\zeta = \arccos(\mathbf{e}_\varphi \mathbf{e}_t) = \arctan\left(\frac{v_\vartheta}{v_\varphi}\right) = \arctan\left(\frac{\dot{\vartheta}}{\dot{\varphi} \cos \vartheta}\right). \quad (\text{A.13})$$

The radial unit vector \mathbf{e}_r lies in the orbit plane and points to the spacecraft's current position.

$$\mathbf{e}_r = \frac{\mathbf{r}}{|\mathbf{r}|} = \begin{pmatrix} \cos \varphi \cos \vartheta \\ \sin \varphi \cos \vartheta \\ \sin \vartheta \end{pmatrix} \quad (\text{A.14})$$

The transversal unit vector \mathbf{e}_t also lies in the orbit plane and is perpendicular to \mathbf{e}_r .

$$\mathbf{e}_t = \mathbf{e}_h \times \mathbf{e}_r = \cos \zeta \mathbf{e}_\varphi + \sin \zeta \mathbf{e}_\vartheta = \begin{pmatrix} -\sin \varphi \cos \zeta - \cos \varphi \sin \vartheta \sin \zeta \\ \cos \varphi \cos \zeta - \sin \varphi \sin \vartheta \sin \zeta \\ \cos \vartheta \sin \zeta \end{pmatrix} \quad (\text{A.15})$$

The orbit-normal unit vector \mathbf{e}_h points in angular momentum direction and is perpendicular to \mathbf{e}_r and \mathbf{e}_t .

$$\mathbf{e}_h = \frac{\mathbf{r} \times \mathbf{v}}{|\mathbf{r} \times \mathbf{v}|} = -\sin \zeta \mathbf{e}_\varphi + \cos \zeta \mathbf{e}_\vartheta = \begin{pmatrix} \sin \varphi \sin \zeta - \cos \varphi \sin \vartheta \cos \zeta \\ -\cos \varphi \sin \zeta - \sin \varphi \sin \vartheta \cos \zeta \\ \cos \vartheta \cos \zeta \end{pmatrix} \quad (\text{A.16})$$

Appendix B

Orbital Elements

The size of an orbit in space, its shape, spatial orientation, and the in-orbit position of an object on that orbit must be known to allow prediction of the object's future position and velocity. A six-dimensional vector holding current position and velocity data is such a description. Supplemented with the time for which this information is valid, this vector describes the object's current state of motion and is therefore called state vector or state. A state vector comprising cartesian or polar position and velocity data is not the only suitable parameter set. Other sets exist, and they are called orbital elements, or Keplerian elements, or two-body elements. The following list of orbital elements is therefore not complete. Val-lado [91] also gives singularity-free parameter sets that are better suited for special problems, e.g., zero inclination or circular orbits.

Semimajor axis a

Unperturbed orbits follow the shape of conic sections. The distance between the two extreme points of an ellipse is called major axis and its half is called the semimajor axis a . The semimajor axis determines the size of ellipse orbits and hyperbolic orbits; it is not defined for parabolic orbits. One can either use the distances of the two extreme points from the focal point F , the apoapsis r_a and the periapsis r_p , or the current distance r , the velocity v , and gravitational parameter μ of the central body to calculate a

$$a = \frac{r_a + r_p}{2} = \left(\frac{2}{r} - \frac{v^2}{\mu} \right)^{-1}. \quad (\text{B.1})$$

Eccentricity e

The eccentricity e is the shape parameter, and for closed orbits it is defined as the ratio of the difference of r_a and r_p and the corresponding sum

$$e = \frac{r_a - r_p}{r_a + r_p}. \quad (\text{B.2})$$

Elliptic orbits have eccentricity values between zero and one. Parabolic orbits have an eccentricity of one, and for hyperbola-shaped orbits e is greater than one.

The infinite apoapsis distance in case of parabolic or hyperbolic orbits render (B.2) unsuitable for the calculation of e . With the mass-specific orbit energy E and the specific angular momentum h , the equation valid for all orbits is:

$$e = \sqrt{1 + \frac{2Eh^2}{\mu^2}}. \quad (\text{B.3})$$

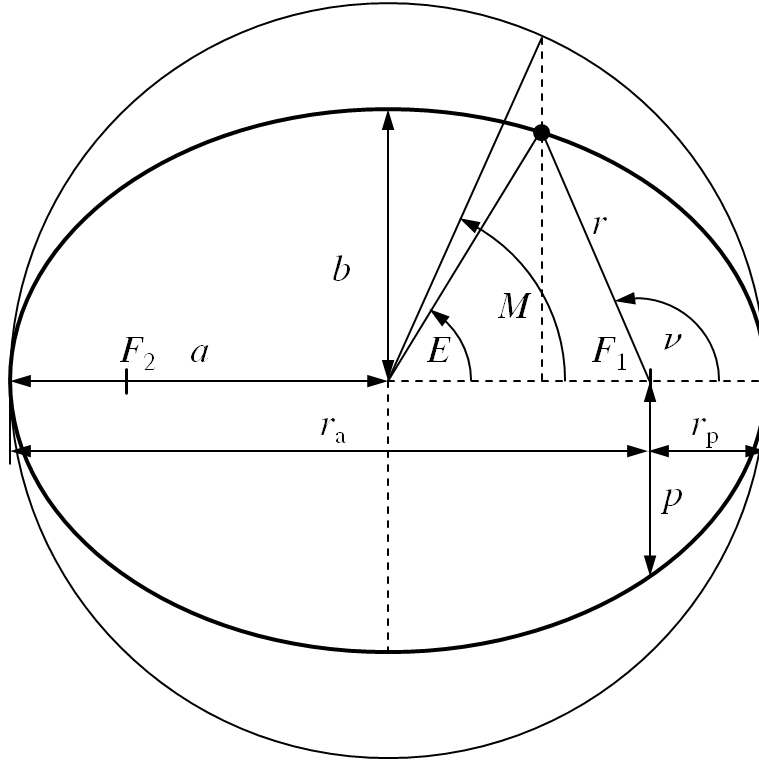


Figure B.1: Elliptical Orbit and orbit plane Keplerian Elements.

The eccentricity also equals the norm of the eccentricity vector, which points to the orbit's pericenter

$$\mathbf{e} = \frac{(v^2 - \frac{\mu}{r}) \mathbf{r} - (\mathbf{r} \cdot \mathbf{v}) \mathbf{v}}{\mu}, \quad (\text{B.4})$$

$$e = \|\mathbf{e}\|. \quad (\text{B.5})$$

Inclination i

The angle between the specific angular momentum vector $\mathbf{h} = \mathbf{r} \times \mathbf{v}$ and the reference frame's third unit vector \mathbf{e}_z is called inclination angle, or inclination i

$$i = \arccos\left(\frac{\mathbf{h} \cdot \mathbf{e}_z}{h}\right). \quad (\text{B.6})$$

The inclination angle is constrained to $0 \leq i \leq \pi$ and orbits with $i > 0$ are called inclined orbits. Orbits with $0 \leq i < \pi/2$ are called direct or prograde, and orbits with $\pi/2 < i \leq \pi$ are retrograde. Orbits whose inclination is zero or π are in the reference system plane spanned by \mathbf{e}_x and \mathbf{e}_y . These orbits are called equatorial if the Earth is the central body and ecliptical if they are Sun-centered. A special orbit type is the one with $i = \pi/2$. A spacecraft moving along this orbit crosses both poles of the central body. That gave this type of orbit the name polar orbits.

Right ascension of ascending node Ω

The right ascension of ascending node is the angle between the unit vector \mathbf{e}_x and the vector from the reference system's origin to the point on the reference system plane spanned by \mathbf{e}_x

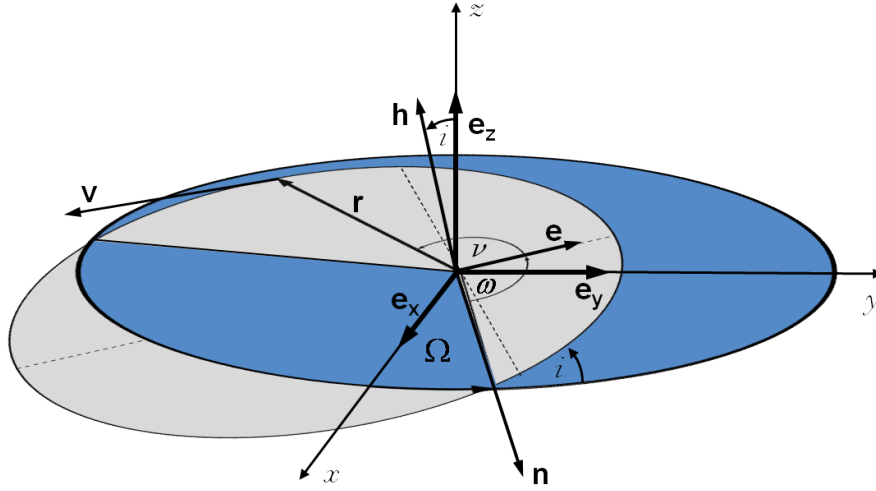


Figure B.2: Elliptical Orbit in Space

and \mathbf{e}_y where the orbit crosses that plane from south to north. The second such point is called descending node because an object on the orbit would enter the southern hemisphere at this point coming from the northern hemisphere. Both points exist only for inclined orbits, i.e., $i > 0$. The line that connects both points is called line of nodes, and the vector \mathbf{n} , pointing from the center of the coordinate system to the ascending node, is called node vector

$$\mathbf{n} = \mathbf{e}_z \times \mathbf{h}. \quad (\text{B.7})$$

Using the node vector, the right ascension of the ascending node is

$$\begin{aligned} \tilde{\Omega} &= \arccos\left(\frac{\mathbf{e}_x \cdot \mathbf{n}}{|\mathbf{n}|}\right) \\ \Omega &= \begin{cases} \tilde{\Omega} & (\mathbf{n} \cdot \mathbf{e}_y) \geq 0 \\ 2\pi - \tilde{\Omega} & (\mathbf{n} \cdot \mathbf{e}_y) < 0 \end{cases}. \end{aligned} \quad (\text{B.8})$$

Argument of pericenter ω

The argument of pericenter is measured from the ascending node along the orbit to the point closest to the central body, and it calculates from node vector \mathbf{n} and eccentricity vector \mathbf{e} .

$$\begin{aligned} \tilde{\omega} &= \arccos\left(\frac{\mathbf{n} \cdot \mathbf{e}}{|\mathbf{n}| |\mathbf{e}|}\right) \\ \omega &= \begin{cases} \tilde{\omega} & (\mathbf{e} \cdot \mathbf{e}_z) \geq 0 \\ 2\pi - \tilde{\omega} & (\mathbf{e} \cdot \mathbf{e}_z) < 0 \end{cases} \end{aligned} \quad (\text{B.9})$$

Mean anomaly M

The mean anomaly M is the angle between the pericenter and the fictitious orbit point at which an object would be if it moved with the mean motion n .

$$n = \frac{2\pi}{T} = \sqrt{\frac{\gamma(m_{\text{CB}} + m_{\text{SC}})}{|a|^3}} \quad (\text{B.10})$$

The orbit period T is the time span that an object on the orbit needs for a complete revolution, and m_{CB} and m_{SC} are the masses of the central body and the spacecraft. The mean anomaly is

$$M \in [0, 2\pi) = (t - t_p) n \quad (\text{B.11})$$

with the current time t and the periapsis passage t_p , which is the point in time when the object was at the orbit's periapsis. To calculate ν from M one needs the the eccentric anomaly E or, if $e > 1$, the hyperbolic anomaly H . Both can be iteratively computed from M .

$$M = \begin{cases} E - e \sin E, e < 1 \\ \sinh H - H, e > 1 \end{cases} \quad (\text{B.12})$$

$$\nu = \begin{cases} 2 \arctan \left(\sqrt{\frac{1+e}{1-e}} \tan \left(\frac{E}{2} \right) \right), e < 1 \\ 2 \arctan \left(\sqrt{\frac{e+1}{e-1}} \tanh \left(\frac{H}{2} \right) \right), e > 1 \end{cases} \quad (\text{B.13})$$

True anomaly ν

Although not being a Keplerian element, the orbital element true anomaly $\nu \in [0, 2\pi)$ is also used to describe a body's position on its orbit. It is the angle between the ascending node vector \mathbf{n} and the position vector \mathbf{r} .

$$\nu = \arccos \left(\frac{\mathbf{n} \cdot \mathbf{r}}{|\mathbf{n}| |\mathbf{r}|} \right) \quad (\text{B.14})$$

Appendix C

Optical SRP Force Model

Three components sum to the total SRP force $\mathbf{F}_{\text{SRP,optical}}$ of the optical force model: $\mathbf{F}_{\text{SRP,a}}$ due to absorption of sunlight photons, $\mathbf{F}_{\text{SRP,r}}$ due to the fraction that is reflected, and $\mathbf{F}_{\text{SRP,e}}$ due to thermal irradiation of absorbed energy.

$$\mathbf{F}_{\text{SRP,optical}} = \mathbf{F}_{\text{SRP,a}} + \mathbf{F}_{\text{SRP,r}} + \mathbf{F}_{\text{SRP,e}} \quad (\text{C.1})$$

The absorption force component is calculated identical to that of the ideal SRP force model with the solar radiation pressure, p_{SRP} , the sail area, A , and the sail pitch angle, β .

$$\mathbf{F}_{\text{SRP,a}} = p_{\text{SRP}} A \cos \beta \mathbf{e}_l \quad (\text{C.2})$$

$\mathbf{F}_{\text{SRP,a}}$ points into sunlight direction, and the relation $\mathbf{e}_l = \sin \beta \mathbf{e}_t + \cos \beta \mathbf{e}_n$ allows expressing (C.2) with a normal and a transversal force component.

$$\mathbf{F}_{\text{SRP,a}} = p_{\text{SRP}} A \cos \beta (\sin \beta \mathbf{e}_t + \cos \beta \mathbf{e}_n) \quad (\text{C.3})$$

Reflection, however, is different than in the ideal SRP force model. Not all sunlight photons are reflected, and the reflected ones are not all reflected specularly, i.e., in \mathbf{e}_r -direction.

$$\mathbf{F}_{\text{SRP,r}} = \mathbf{F}_{\text{SRP,rs}} + \mathbf{F}_{\text{SRP,rd}} \quad (\text{C.4})$$

Like the reflection force component in (C.4) divides into two components, also the reflection coefficient, $\rho = \rho_s + \rho_d$, is the sum of a specular reflection coefficient, ρ_s , and a diffuse reflection coefficient, ρ_d . The specular component of (C.4) points in \mathbf{e}_r -direction

$$\mathbf{F}_{\text{SRP,rs}} = -\rho_s p_{\text{SRP}} A \cos \beta \mathbf{e}_r \quad (\text{C.5})$$

and with the relation $\mathbf{e}_r = \sin \beta \mathbf{e}_t - \cos \beta \mathbf{e}_n$ it can be dissolved into normal and transversal direction.

$$\mathbf{F}_{\text{SRP,rs}} = -\rho_s p_{\text{SRP}} A \cos \beta (\sin \beta \mathbf{e}_t - \cos \beta \mathbf{e}_n) \quad (\text{C.6})$$

The photons that are not specularly reflected, leave the front side of the solar sail in all directions and therefore the resulting force direction is \mathbf{e}_n . An ideal surface would appear with constant brightness from all viewing angles. A real surface, however, does not, and the non-Lambertian coefficient B_f describes the deviation of the sail front side from an ideal surface.

$$\mathbf{F}_{\text{SRP,rd}} = \rho_d B_f p_{\text{SRP}} A \cos \beta \mathbf{e}_n \quad (\text{C.7})$$

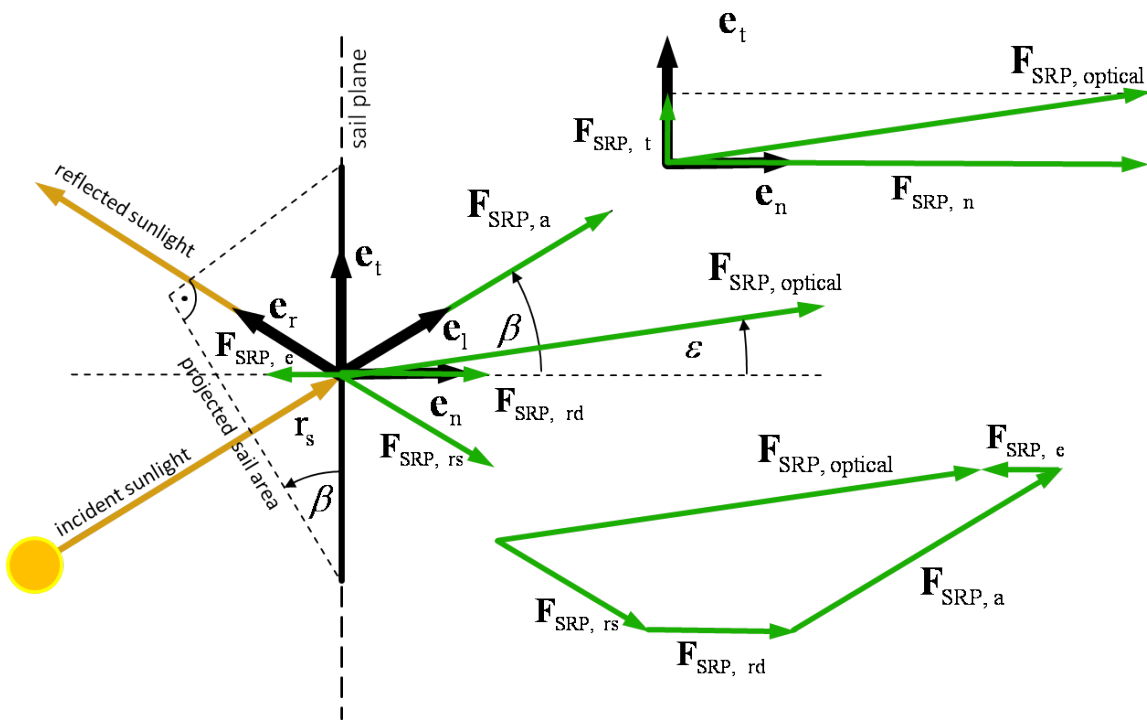


Figure C.1: Components of Optical SRP Force. The vector $\mathbf{F}_{\text{SRP, optical}}$ of the optical SRP force model consists of components resulting from different mechanisms: from absorption, $\mathbf{F}_{\text{SRP, a}}$; from specular reflection, $\mathbf{F}_{\text{SRP, rs}}$; from diffuse reflection, $\mathbf{F}_{\text{SRP, rd}}$; and from irradiation of thermal energy, $\mathbf{F}_{\text{SRP, e}}$. Resolving these vectors along the solar sail's normal unit vector, \mathbf{e}_n , and the transversal unit vector, \mathbf{e}_t , allows decomposition of the total force vector into the normal component $\mathbf{F}_{\text{SRP, optical, n}}$ and the transversal component $\mathbf{F}_{\text{SRP, optical, t}}$. The sail pitch angle, β , is defined between the direction of incident sunlight, \mathbf{e}_i , and the sail normal direction, \mathbf{e}_n . The centerline angle, ϵ , is defined between the sail normal direction and the direction of the SRP force.

The total reflection force is therefore

$$\mathbf{F}_{\text{SRP},r} = p_{\text{SRP}} A \cos \beta [(\rho_d B_f + \rho_s) \mathbf{e}_n - \rho_d \sin \beta \mathbf{e}_t]. \quad (\text{C.8})$$

A body having an absolute temperature $T > 0$ and an emission coefficient $\epsilon > 0$ irradiates thermal energy, and that relation is the Stefan–Boltzmann law, which, under the assumption of uniform temperature throughout the body, gives the energy resulting energy flux.

$$W = \epsilon \sigma T^4 \quad (\text{C.9})$$

The emissivity $\epsilon \in [0, 1]$ tells if the body is a so-called black body ($\epsilon = 1$), or if it does not irradiate energy ($\epsilon = 0$). A body having an emissivity coefficient between zero and one is called grey body.

If we assume a solar sail as a flat plate, the dominating surfaces from which irradiation takes place are the front side and the back side. Allowing both sides to have non-Lambertian character, the resulting force from irradiation on the front side is calculated with the speed of light c and the sail film surface area A .

$$\mathbf{F}_{e,f} = \frac{\sigma \epsilon_f B_f A T^4}{c} \mathbf{e}_n \quad (\text{C.10})$$

$$\mathbf{F}_{e,b} = -\frac{\sigma \epsilon_b B_b A T^4}{c} \mathbf{e}_n \quad (\text{C.11})$$

$$\mathbf{F}_{\text{SRP},e} = \mathbf{F}_{e,f} + \mathbf{F}_{e,b} = \frac{\sigma A T^4}{c} (\epsilon_f B_f - \epsilon_b B_b) \mathbf{e}_n \quad (\text{C.12})$$

If the sail is in thermal equilibrium, the incoming energy flux is equal to the irradiated thermal energy per time.

$$E_{\text{in}} = E_{\text{out}} = E_{\text{out},f} + E_{\text{out},b} \quad (\text{C.13})$$

The thermal input per time can be obtained from the solar radiation flux S times the projected sail area $A \cos \beta$ and the absorption factor α . The thermal output is known from (C.9) and the sail area A .

$$\alpha S A \cos \beta = \sigma A T^4 (\epsilon_f + \epsilon_b) \quad (\text{C.14})$$

Using the relation $S = c p_{\text{SRP}}$ allows resolving for the equilibrium temperature.

$$T = \sqrt[4]{\frac{\alpha c p_{\text{SRP}} \cos \beta}{\sigma (\epsilon_f + \epsilon_b)}} \quad (\text{C.15})$$

With this temperature the force component due to emission (C.12) is

$$\mathbf{F}_{\text{SRP},e} = \alpha p_{\text{SRP}} A \cos \beta \left(\frac{B_f \epsilon_f - B_b \epsilon_b}{\epsilon_b + \epsilon_f} \right) \mathbf{e}_n. \quad (\text{C.16})$$

Adding the force vectors and resolving in sail normal and sail transversal direction gives the two resulting forces which again sum up to the vector $\mathbf{F}_{\text{SRP},\text{optical}}$.

$$\mathbf{F}_{\text{SRP},\text{optical},n} = p_{\text{SRP}} A \cos \beta \left[(1 + \rho_s) \cos \beta + \alpha \left(\frac{B_f \epsilon_f - B_b \epsilon_b}{\epsilon_b + \epsilon_f} \right) + B_f \rho_s \right] \mathbf{e}_n \quad (\text{C.17})$$

$$\mathbf{F}_{\text{SRP},\text{optical},t} = p_{\text{SRP}} A \cos \beta \sin \beta (1 - \rho_s) \mathbf{e}_t \quad (\text{C.18})$$

Appendix D

InTrance Configuration Files

D.1 Earth-Moon Configuration

D.1.1 InTrance Configuration

COMMAND	= optimize
SIM_PARAM_FILE_1	= EarthMoonSIM1.sim
SIM_PARAM_FILE_2	= EarthMoonSIM2.sim
EA_PARAM_FILE	= coldstart.eap
COLDSTART	= yes
SIM_DATA_FILE	= coldstart.csv
TRAJ_DATA_FILE	= coldstart.dat
GESOP_FILE	= coldstart.gesop.txt
VRML_FILE	= coldstart.wrl
CTRL_FILE	= coldstart.ctr
BEST_CHROM_FILE	= coldstart.eac
REPORT_FILE	= coldstart.rep

D.1.2 Evolutionary Algorithm Configuration

SEARCH_SPACE_HYPERCUBE_SIZE	= 1.0
HYPERCUBE_START_SIZE	= 0.125
HYPERCUBE_SHRINKING_FACTOR	= 0.5
POPULATION_SIZE	= 75
SEARCH_SCAN_EPOCHS	= 10
FITNESS_FUNCTION_TYPE	= J_AND
CHROMOSOME_MUTATION_PROBABILITY	= 0.93
SELECTION_PRESSURE_ON_TIME	= 0.03
HYPERCUBE_UPPER_LIMIT	= 1.0E-3
IL_POP_CONV_FBC_MET	= 1.0E-4
IL_POP_CONV_FBC_NOT_MET	= 1.0E-2
IL_EA_CONV_FBC_MET	= 1.0E-3
IL_EA_CONV_FBC_NOT_MET	= 5.0E-3
FBC_MET_FITNESS	= 0
CHROMOSOME_INIT_MODEL	= RANDOM

D.1.3 Neurocontroller Configuration of Phase 1

```
HIDDEN_LAYERS           = 1
NEURONS_IN_HIDDEN_LAYER1 = 12
NC_OUTPUT                = direct
TRANSFER_FUNCTION       = sigmoid
```

D.1.4 Neurocontroller Configuration of Phase 2

```
HIDDEN_LAYERS           = 1
NEURONS_IN_HIDDEN_LAYER1 = 12
NC_OUTPUT                = direct
TRANSFER_FUNCTION       = sigmoid
```

D.1.5 Spacecraft Configuration of Phase 1

```
SC_TYPE                  = NSTAR
SC_DRYMASS               = 250.0
MIN_PROP_MASS            = 11
MAX_PROP_MASS            = 12
THROTTLE_TYPE           = variable
SOLAR_ARRAY_CHAR_POWER  = 15kW
POWER_VARIATION_EXPONENT = 1.7
N_THRUSTERS              = 1
```

D.1.6 Spacecraft Configuration of Phase 2

```
SC_TYPE                  = NSTAR
SC_DRYMASS               = 250.0
MIN_PROP_MASS            = 16kg
MAX_PROP_MASS            = 17kg
THROTTLE_TYPE           = variable
SOLAR_ARRAY_CHAR_POWER  = 15kW
POWER_VARIATION_EXPONENT = 1.7
N_THRUSTERS              = 1
```

D.1.7 Simulation Configuration of Phase 1

```
INTEGRATION_INTERVAL    = 60day
FLIGHT_TIME_MIN         = 30day
INTEGRATION_STEPS       = 500
MIN_OUTPUT_POINTS       = 1
DYN_INTEGRATION_INTERVAL = no
MET_MAX                 = 100day
DISTURBING_BODIES       = "disturbing_bodies.sim"
MIN_SOLAR_DISTANCE      = 6900km
MODIFY_INIT_PARAMETERS  = no
MODIFY_LAUNCH_DATE      = yes
MODIFY_INIT_PROP_MASS   = yes
```

MODIFY_INIT_VINF	= no
USE_DSSC	= yes
DSSC_STEP_ANGLE_CONTROL	= yes
DSSC_MAX_STEP_ANGLE	= 5DEG
DSSC_MAX_STEP_SIZE	= 5day
DSSC_MIN_STEP_SIZE	= 60min
DSSC_STEP_DISTANCE_CONTROL	= NO
DSSC_MAX_STEP_DISTANCE	= 50000km
DSSC_APPROACH_CONTROL	= no
DSSC_APPROACH_STEP_SIZE_FACTOR	= 0.0
SIM_START_TIME_MIN	= 55099
SIM_START_TIME_MAX	= 55104
ARRIVAL_DATE_MIN	= 0
ARRIVAL_DATE_MAX	= 0
INITIAL_STATE	= orbit
INITIAL_CENTRAL_BODY	= Earth
INITIAL_FRAME_EQUATORIAL	= false
INITIAL_SEMIMAJOR_AXIS	= 100E3km
INITIAL_ECCENTRICITY	= 0.716227925
INITIAL_INCLINATION	= 7.0deg
INITIAL_LONGITUDE_OF_NODE	= 178.0
INITIAL_ARGUMENT_OF_PERICENTER	= 177.935176deg
INITIAL_MEAN_ANOMALY	= 3.53deg
TARGET_STATE	= body capture
TARGET_BODY_NAME	= Moon
OPTIMIZATION_GOAL	= minimum transfer time
ACCURACY_FITNESS_FRACTION	= 0.0
INTEGRATOR	= RK54F
MAX_RELATIVE_ERROR	= 1.0E-8
MAX_ABSOLUTE_ERROR	= 1.0E-8
USE_ITGR_STOPPER	= no
SC_CONF	= EarthMoonSC1.scf
NAV_TYPE	= ANN
NAV_ANN_CONF	= EarthMoonNC1.ncf
STEERING_DYN_UNIT_CALC	= yes
STEERING_USE_MASS_PROPELLANT	= no
STEERING_USE_RANGE	= yes
STEERING_USE_RANGE_RATE	= yes
STEERING_USE_ACC_THRUST_MAX	= no
STEERING_USE_ACC_THRUST_MAX_DRY	= no
STEERING_USE_STEP_SIZE	= no
STEERING_USE_TIME_UNTIL_PERI_SC	= no
STEERING_USE_TIME_UNTIL_PERI_TGT	= no
STEERING_USE_ABS_CART_POS_X	= yes
STEERING_USE_ABS_CART_POS_Y	= yes
STEERING_USE_ABS_CART_POS_Z	= yes
STEERING_USE_ABS_CART_VEL_X	= yes
STEERING_USE_ABS_CART_VEL_Y	= yes
STEERING_USE_ABS_CART_VEL_Z	= yes

STEERING_USE_ABS_POLAR_POS_R	= no
STEERING_USE_ABS_POLAR_POS_AZI	= no
STEERING_USE_ABS_POLAR_POS_ELE	= no
STEERING_USE_ABS_POLAR_VEL_R	= no
STEERING_USE_ABS_POLAR_VEL_AZI	= no
STEERING_USE_ABS_POLAR_VEL_ELE	= no
STEERING_USE_TGT_CART_POS_X	= no
STEERING_USE_TGT_CART_POS_Y	= no
STEERING_USE_TGT_CART_POS_Z	= no
STEERING_USE_TGT_CART_VEL_X	= no
STEERING_USE_TGT_CART_VEL_Y	= no
STEERING_USE_TGT_CART_VEL_Z	= no
STEERING_USE_TGT_CART_POS_REL_X	= yes
STEERING_USE_TGT_CART_POS_REL_Y	= yes
STEERING_USE_TGT_CART_POS_REL_Z	= yes
STEERING_USE_TGT_CART_VEL_REL_X	= yes
STEERING_USE_TGT_CART_VEL_REL_Y	= yes
STEERING_USE_TGT_CART_VEL_REL_Z	= yes

D.1.8 Simulation Configuration of Phase 2

INDEPENDENT_FLIGHT_PHASE	= no
TRANSITION_THRESHOLD_MASS_PROP	= 0.01
TRANSITION_THRESHOLD_STATE	= 0.05 //0.09
SIM_START_TIME_MIN	= 55147
SIM_START_TIME_MAX	= 55153.2
INTEGRATION_INTERVAL	= 20day
FLIGHT_TIME_MIN	= 5day
MET_MAX	= 100day
INTEGRATION_STEPS	= 300
MIN_OUTPUT_POINTS	= 1
DYN_INTEGRATION_INTERVAL	= no
MIN_SOLAR_DISTANCE	= 1738km
DISTURBING_BODIES	= "disturbing_bodies.sim"
OPTIMIZATION_GOAL	= minimum transfer time
ALWAYS_USE_OBJ_FCN	= yes
ACCURACY_FITNESS_FRACTION	= 0.0
INTEGRATOR	= RK54F
MAX_RELATIVE_ERROR	= 1.0E-8
MAX_ABSOLUTE_ERROR	= 1.0E-8
USE_ITGR_STOPPER	= no
MODIFY_INIT_PARAMETERS	= no
MODIFY_LAUNCH_DATE	= yes
MODIFY_INIT_PROP_MASS	= yes
MODIFY_INIT_VINF	= no
MODIFY_INIT_DISTANCE	= yes
MODIFY_INIT_AZI	= yes
MODIFY_INIT_ELE	= yes

```

MODIFY_INIT_VEL           = yes
MODIFY_INIT_OFA          = yes
MODIFY_INIT_FPA          = yes

USE_DSSC                  = yes
DSSC_STEP_ANGLE_CONTROL  = yes
DSSC_MAX_STEP_ANGLE      = 5DEG
DSSC_MAX_STEP_SIZE       = 1day
DSSC_MIN_STEP_SIZE       = 10min
DSSC_STEP_DISTANCE_CONTROL = NO
DSSC_MAX_STEP_DISTANCE   = 2000km
DSSC_APPROACH_CONTROL    = no
DSSC_APPROACH_STEP_SIZE_FACTOR = 0.0

INITIAL_STATE             = vector
INITIAL_BODY_NAME        = Moon
INITIAL_STATE_EQUATORIAL = no
INITIAL_CENTRAL_BODY     = Moon
INITIAL_DISTANCE_MIN     = 55440km
INITIAL_DISTANCE_MAX     = 56540km
INITIAL_AZIMUTH_MIN      = -33deg
INITIAL_AZIMUTH_MAX      = -30deg
INITIAL_ELEVATION_MIN    = 52deg
INITIAL_ELEVATION_MAX    = 55deg
INITIAL_VELOCITY_MIN     = +236m/s
INITIAL_VELOCITY_MAX     = +248m/s
INITIAL_ORBIT_ANGLE_MIN  = -85deg
INITIAL_ORBIT_ANGLE_MAX  = -82deg
INITIAL_FPA_MIN          = -22deg
INITIAL_FPA_MAX          = -20deg

TARGET_STATE              = orbital elements
TARGET_BODY_NAME          = moon
TARGET_SEMIMAJOR_AXIS    = 20E3km // = 262km orbit height
TARGET_INCLINATION        = 40.0DEG //84.0DEG // almost polar
TARGET_ECCENTRICITY       = 0.0 // circular target orbit

TARGET_DIST_MAX_FINAL     = 2000km
TARGET_DIST_MAX_INIT     = 4000km
TARGET_DIST_MAX_SHRINK    = 0.9
TARGET_DIST_MAX_DECREASE = 100m
TARGET_DIST_MAX_REDUCTION_USE_MAX = FALSE

TARGET_RELVEL_MAX_FINAL  = 100m/s
TARGET_RELVEL_MAX_INIT  = 200m/s
TARGET_RELVEL_MAX_SHRINK = 0.9
TARGET_RELVEL_MAX_DECREASE = 1m/s
TARGET_RELVEL_MAX_REDUCTION_USE_MAX = FALSE

```

```

SC_CONF = p2.scp

NAV_TYPE = ANN
NAV_ANN_CONF = p2.ncp
STEERING_DYN_UNIT_CALC = yes
STEERING_USE_MASS_PROPELLANT = no
STEERING_USE_RANGE = no
STEERING_USE_RANGE_RATE = no
STEERING_USE_ACC_THRUST_MAX = no
STEERING_USE_ACC_THRUST_MAX_DRY = no
STEERING_USE_STEP_SIZE = no
STEERING_USE_TIME_UNTIL_PERI_SC = no
STEERING_USE_TIME_UNTIL_PERI_TGT = no
STEERING_USE_ABS_CART_POS_X = yes
STEERING_USE_ABS_CART_POS_Y = yes
STEERING_USE_ABS_CART_POS_Z = yes
STEERING_USE_ABS_CART_VEL_X = yes
STEERING_USE_ABS_CART_VEL_Y = yes
STEERING_USE_ABS_CART_VEL_Z = yes
STEERING_USE_ABS_POLAR_POS_R = no
STEERING_USE_ABS_POLAR_POS_AZI = yes
STEERING_USE_ABS_POLAR_POS_ELE = yes
STEERING_USE_ABS_POLAR_VEL_R = yes
STEERING_USE_ABS_POLAR_VEL_AZI = yes
STEERING_USE_ABS_POLAR_VEL_ELE = yes
STEERING_USE_TGT_CART_POS_X = no
STEERING_USE_TGT_CART_POS_Y = no
STEERING_USE_TGT_CART_POS_Z = no
STEERING_USE_TGT_CART_VEL_X = no
STEERING_USE_TGT_CART_VEL_Y = no
STEERING_USE_TGT_CART_VEL_Z = no
STEERING_USE_TGT_CART_POS_REL_X = no
STEERING_USE_TGT_CART_POS_REL_Y = no
STEERING_USE_TGT_CART_POS_REL_Z = no
STEERING_USE_TGT_CART_VEL_REL_X = no
STEERING_USE_TGT_CART_VEL_REL_Y = no
STEERING_USE_TGT_CART_VEL_REL_Z = no

```

D.2 Dawn Configuration

D.2.1 InTrance Configuration

```

COMMAND = optimize
COLDSTART = yes
NO_OF_EVAL_OBJECTS = 4
ASTEROID_DATA_FILE1 = c:\ELEMENTS.NUMBR
SIM_PARAM_FILE_1 = Vesta.sim
SIM_PARAM_FILE_2 = Ceres.sim
EA_PARAM_FILE = coldstart.eap

```

```
SIM_DATA_FILE      = coldstart.csv
TRAJ_DATA_FILE     = coldstart.dat
GESOP_FILE        = coldstart.gesop.txt
VRML_FILE         = coldstart.wrl
CTRL_FILE         = coldstart.ctr
BEST_CHROM_FILE   = coldstart.eac
REPORT_FILE       = coldstart.rep
```

D.2.2 Evolutionary Algorithm Configuration

```
SEARCH_SPACE_HYPERCUBE_SIZE = 1.0
HYPERCUBE_START_SIZE       = 1.0
HYPERCUBE_SHRINKING_FACTOR = 0.6
POPULATION_SIZE            = 50
POPULATION_SIZE_SSS       = 50
SEARCH_SCAN_EPOCHS        = 10
FITNESS_FUNCTION_TYPE      = J_AND
CHROMOSOME_MUTATION_PROBABILITY = 0.2
GENOM_MUTATION_PROBABILITY = 0.0
SELECTION_PRESSURE_ON_TIME = 0.03
HYPERCUBE_UPPER_LIMIT     = 1.0E-3
IL_POP_CONV_FBC_MET       = 1.0E-5
IL_POP_CONV_FBC_NOT_MET   = 1.0E-2
IL_EA_CONV_FBC_MET        = 1.0E-3
IL_EA_CONV_FBC_NOT_MET    = 1.0E-3
CHROMOSOME_INIT_MODEL     = RANDOM
FBC_MET_FITNESS           = 0.0
```

D.2.3 Neurocontroller Configuration of Phase 1

```
NC_OUTPUT          = direct
TRANSFER_FUNCTION  = sigmoid
HIDDEN_LAYERS      = 1
NEURONS_IN_HIDDEN_LAYER1 = 35
```

D.2.4 Neurocontroller Configuration of Phase 2

```
NC_OUTPUT          = direct
TRANSFER_FUNCTION  = sigmoid
HIDDEN_LAYERS      = 1
NEURONS_IN_HIDDEN_LAYER1 = 35
```

D.2.5 Spacecraft Configuration of Phase 1

```
SC_TYPE           = NSTAR
SC_DRYMASS        = 793
MIN_PROP_MASS     = 100
MAX_PROP_MASS     = 305
SOLAR_ARRAY_CHAR_POWER = 9.8kW
```

POWER_VARIATION_EXPONENT = 1.7
 N_THRUSTERS = 1
 THROTTLE_TYPE = variable

D.2.6 Spacecraft Configuration of Phase 2

SC_TYPE = NSTAR
 SC_DRYMASS = 793
 MIN_PROP_MASS = 50
 MAX_PROP_MASS = 120
 SOLAR_ARRAY_CHAR_POWER = 9.8kW
 POWER_VARIATION_EXPONENT = 1.7
 N_THRUSTERS = 1
 THROTTLE_TYPE = variable

D.2.7 Simulation Configuration of Phase 1

INTEGRATION_INTERVAL = 1600day
 FLIGHT_TIME_MIN = 1200day
 INTEGRATION_STEPS = 400
 MIN_OUTPUT_POINTS = 50
 DYN_INTEGRATION_INTERVAL = yes
 MODIFY_INIT_PARAMETERS = yes
 MODIFY_LAUNCH_DATE = no
 MODIFY_INIT_PROP_MASS = no
 MODIFY_INIT_VINF = yes
 USE_DSSC = no
 DSSC_STEP_ANGLE_CONTROL = yes
 DSSC_MAX_STEP_ANGLE = 6DEG
 DSSC_MAX_STEP_SIZE = 60DAY
 DSSC_MIN_STEP_SIZE = 1DAY
 DSSC_STEP_DISTANCE_CONTROL = NO
 DSSC_MAX_STEP_DISTANCE = 0.50AU
 DSSC_APPROACH_CONTROL = yes
 DSSC_APPROACH_STEP_SIZE_FACTOR = 0.0
 SIM_START_TIME_MIN = 54200
 SIM_START_TIME_MAX = 54400
 ARRIVAL_DATE_MIN = 55805
 ARRIVAL_DATE_MAX = 55805
 INITIAL_STATE = body
 INITIAL_BODY_NAME = EARTH
 INITIAL_VINF_MIN = 3.362km/s
 INITIAL_VINF_MAX = 3.362km/s
 INITIAL_VINF_AZIMUTH_MIN = -10deg
 INITIAL_VINF_AZIMUTH_MAX = +00deg
 INITIAL_VINF_ELEV_MIN = -46deg
 INITIAL_VINF_ELEV_MAX = -41deg

```

TARGET_STATE                = body rendezvous
TARGET_BODY_NAME            = Vesta
TARGET_DIST_MAX_FINAL       = 1.0E6km
TARGET_DIST_MAX_INIT        = 1.0E6km
TARGET_DIST_MAX_SHRINK      = 0.95
TARGET_DIST_MAX_DECREASE    = 1.0E4km
TARGET_DIST_MAX_REDUCTION_USE_MAX = no
TARGET_RELVEL_MAX_FINAL     = 500m/s
TARGET_RELVEL_MAX_INIT      = 500m/s
TARGET_RELVEL_MAX_SHRINK    = 0.95
TARGET_RELVEL_MAX_DECREASE  = 1m/s
TARGET_RELVEL_MAX_REDUCTION_USE_MAX = no
OPTIMIZATION_GOAL          = minimum transfer time
ACCURACY_FITNESS_FRACTION  = 0.5
MIN_SOLAR_DISTANCE         = 0.2AU
INTEGRATOR                  = RK54F
MAX_RELATIVE_ERROR          = 1.0E-8
MAX_ABSOLUTE_ERROR          = 1.0E-8
USE_ITGR_STOPPER           = yes
SC_CONF                     = Vesta.scp
NAV_TYPE                    = ANN
NAV_ANN_CONF                = Vesta.ncp
STEERING_DYN_UNIT_CALC      = yes
STEERING_USE_RANGE          = yes
STEERING_USE_RANGE_RATE     = yes
STEERING_USE_ACC_THRUST_MAX = no
STEERING_USE_ACC_THRUST_MAX_DRY = no
STEERING_USE_STEP_SIZE      = no
STEERING_USE_TIME_UNTIL_PERI_SC = no
STEERING_USE_TIME_UNTIL_PERI_TGT = no
STEERING_USE_ABS_CART_POS_X = yes
STEERING_USE_ABS_CART_POS_Y = yes
STEERING_USE_ABS_CART_POS_Z = yes
STEERING_USE_ABS_CART_VEL_X = yes
STEERING_USE_ABS_CART_VEL_Y = yes
STEERING_USE_ABS_CART_VEL_Z = yes
STEERING_USE_ABS_POLAR_POS_R = no
STEERING_USE_ABS_POLAR_POS_AZI = no
STEERING_USE_ABS_POLAR_POS_ELE = no
STEERING_USE_ABS_POLAR_VEL_R = no
STEERING_USE_ABS_POLAR_VEL_AZI = no
STEERING_USE_ABS_POLAR_VEL_ELE = no
STEERING_USE_TGT_CART_POS_X = no
STEERING_USE_TGT_CART_POS_Y = no
STEERING_USE_TGT_CART_POS_Z = no
STEERING_USE_TGT_CART_VEL_X = no
STEERING_USE_TGT_CART_VEL_Y = no
STEERING_USE_TGT_CART_VEL_Z = no
STEERING_USE_TGT_CART_POS_REL_X = yes

```

STEERING_USE_TGT_CART_POS_REL_Y = yes
 STEERING_USE_TGT_CART_POS_REL_Z = yes
 STEERING_USE_TGT_CART_VEL_REL_X = yes
 STEERING_USE_TGT_CART_VEL_REL_Y = yes
 STEERING_USE_TGT_CART_VEL_REL_Z = yes

D.2.8 Simulation Configuration of Phase 2

INDEPENDENT_FLIGHT_PHASE = no
 TRANSITION_THRESHOLD_MASS_PROP = 0.1
 SIM_START_TIME_MIN = 56000
 SIM_START_TIME_MAX = 56100
 MET_MAX = 9JYR
 INTEGRATION_INTERVAL = 1100day
 FLIGHT_TIME_MIN = 700
 INTEGRATION_STEPS = 110
 MIN_OUTPUT_POINTS = 1
 DYN_INTEGRATION_INTERVAL = yes
 MODIFY_INIT_PARAMETERS = yes
 MODIFY_LAUNCH_DATE = yes
 MODIFY_INIT_PROP_MASS = no
 USE_DSSC = no
 DSSC_STEP_ANGLE_CONTROL = YES
 DSSC_MAX_STEP_ANGLE = 5DEG
 DSSC_MAX_STEP_SIZE = 10DAY
 DSSC_MIN_STEP_SIZE = 1DAY
 DSSC_STEP_DISTANCE_CONTROL = NO
 DSSC_MAX_STEP_DISTANCE = 0.50AU
 DSSC_APPROACH_CONTROL = yes
 DSSC_APPROACH_STEP_SIZE_FACTOR = 0.0
 DWELL_TIME_MIN = 100day
 DWELL_TIME_MAX = 1.0JYR
 INITIAL_STATE = body
 INITIAL_BODY_NAME = Vesta
 ARRIVAL_DATE_MIN = 57054
 ARRIVAL_DATE_MAX = 57054
 TARGET_STATE = body rendezvous
 TARGET_BODY_NAME = Ceres
 TGT_PROX_STATE_THRESHOLD = 0.005
 TGT_PROX_STATE_THRESHOLD_FINAL = 0.001
 TARGET_DIST_MAX_FINAL = 1.0E6km
 TARGET_DIST_MAX_INIT = 1.0E6km
 TARGET_DIST_MAX_SHRINK = 0.8
 TARGET_DIST_MAX_DECREASE = 1.0E5km
 TARGET_DIST_MAX_REDUCTION_USE_MAX = no
 TARGET_RELVEL_MAX_FINAL = 500m/s
 TARGET_RELVEL_MAX_INIT = 500m/s
 TARGET_RELVEL_MAX_SHRINK = 0.9
 TARGET_RELVEL_MAX_DECREASE = 1m/s

```

TARGET_RELVEL_MAX_REDUCTION_USE_MAX = no
OPTIMIZATION_GOAL                    = maximum dwell time
ACCURACY_FITNESS_FRACTION            = 0.01
MIN_SOLAR_DISTANCE                   = 0.2AU
INTEGRATOR                           = RK54F
MAX_RELATIVE_ERROR                   = 1.0E-6
MAX_ABSOLUTE_ERROR                   = 1.0E-6
USE_ITGR_STOPPER                     = yes
SC_CONF                               = Ceres.scp
NAV_TYPE                              = ANN
NAV_ANN_CONF                          = Ceres.ncp
STEERING_DYN_UNIT_CALC                = yes
STEERING_USE_RANGE                   = yes
STEERING_USE_RANGE_RATE               = yes
STEERING_USE_ACC_THRUST_MAX          = no
STEERING_USE_ACC_THRUST_MAX_DRY      = no
STEERING_USE_STEP_SIZE               = no
STEERING_USE_TIME_UNTIL_PERI_SC      = no
STEERING_USE_TIME_UNTIL_PERI_TGT     = no
STEERING_USE_ABS_CART_POS_X          = yes
STEERING_USE_ABS_CART_POS_Y          = yes
STEERING_USE_ABS_CART_POS_Z          = yes
STEERING_USE_ABS_CART_VEL_X          = yes
STEERING_USE_ABS_CART_VEL_Y          = yes
STEERING_USE_ABS_CART_VEL_Z          = yes
STEERING_USE_ABS_POLAR_POS_R         = no
STEERING_USE_ABS_POLAR_POS_AZI       = no
STEERING_USE_ABS_POLAR_POS_ELE       = no
STEERING_USE_ABS_POLAR_VEL_R         = no
STEERING_USE_ABS_POLAR_VEL_AZI       = no
STEERING_USE_ABS_POLAR_VEL_ELE       = no
STEERING_USE_TGT_CART_POS_X          = no
STEERING_USE_TGT_CART_POS_Y          = no
STEERING_USE_TGT_CART_POS_Z          = no
STEERING_USE_TGT_CART_VEL_X          = no
STEERING_USE_TGT_CART_VEL_Y          = no
STEERING_USE_TGT_CART_VEL_Z          = no
STEERING_USE_TGT_CART_POS_REL_X      = yes
STEERING_USE_TGT_CART_POS_REL_Y      = yes
STEERING_USE_TGT_CART_POS_REL_Z      = yes
STEERING_USE_TGT_CART_VEL_REL_X      = yes
STEERING_USE_TGT_CART_VEL_REL_Y      = yes
STEERING_USE_TGT_CART_VEL_REL_Z      = yes

```

D.3 IHP Configuration

D.3.1 InTrance Configuration

```

ASTEROID_DATA_FILE1 = C:\ELEMENTS.UNNUM

```

```

COMMAND           = optimize
SIM_PARAM_FILE_1  = SEP.sim
SIM_PARAM_FILE_2  = REP.sim
EA_PARAM_FILE     = coldstart.eap
COLDSTART         = yes
SIM_DATA_FILE     = coldstart.csv
TRAJ_DATA_FILE    = coldstart.dat
GESOP_FILE        = coldstart.gesop.txt
VRML_FILE         = coldstart.wrl
BEST_CHROM_FILE   = coldstart.eac
REPORT_FILE       = coldstart.rep
NO_OF_EVAL_OBJECTS = 4

```

D.3.2 Evolutionary Algorithm Configuration

```

SEARCH_SPACE_HYPERCUBE_SIZE = 1.0
HYPERCUBE_START_SIZE       = 2.0E-1
HYPERCUBE_SHRINKING_FACTOR = 90E-2
POPULATION_SIZE            = 25
SEARCH_SCAN_EPOCHS        = 15
CHROMOSOME_MUTATION_PROBABILITY = 0.94
GENOM_MUTATION_PROBABILITY = 0.02
HYPERCUBE_UPPER_LIMIT     = 1.0E-4
IL_POP_CONV_FBC_MET       = 1.0E-6
IL_POP_CONV_FBC_NOT_MET   = 1.0E-5
IL_EA_CONV_FBC_MET        = 1.0E-5
IL_EA_CONV_FBC_NOT_MET    = 1.0E-5

```

D.3.3 Neurocontroller Configuration of Phase 1

```

HIDDEN_LAYERS           = 1
NEURONS_IN_HIDDEN_LAYER1 = 3

```

D.3.4 Neurocontroller Configuration of Phase 2

```

HIDDEN_LAYERS           = 1
NEURONS_IN_HIDDEN_LAYER1 = 5

```

D.3.5 Spacecraft Configuration of Phase 1

```

SC_TYPE           = RIT
RIT_MODEL         = SIMPLE
PAYLOAD_MASS     = 497.8kg
BUS_MASS          = 0.0kg
THRUSTER_MASS    = 37.3kg
STRUCTURE_MASS_RATIO = 0.25
TANK_MASS_2_PROPELLANT_RATIO = 0.06
MIN_PROP_MASS    = 200kg
MAX_PROP_MASS    = 450kg

```

```

SOLAR_ARRAY_CHAR_POWER_MIN = 53.0kW
SOLAR_ARRAY_CHAR_POWER_MAX = 53.0kW
POWER_SPECIFIC_MASS        = 5.0kg/kW
THROTTLE_TYPE              = variable
POWER_VARIATION_EXPONENT   = 1.6
MAX_THRUSTER_POWER        = 13.592E3
OPT_THRUSTER_POWER        = 13.592E3
MIN_THRUSTER_POWER        = 8.835E3
THRUST_TO_POWER_RATIO     = 19.78E-6
MASSFLOW_TO_POWER_RATIO   = 0.2724E-9
N_THRUSTERS                = 6

```

D.3.6 Spacecraft Configuration of Phase 2

```

SC_TYPE                    = NEP
SC_NAME                    = "IHP"
PAYLOAD_MASS              = 188.0kg
BUS_MASS                   = 0.0kg
PROP_SYS_MASS             = 92.5kg // 16.5kg (prop sys) + 76kg (RTG)
STRUCTURE_MASS            = 51.0kg
MIN_PROP_MASS             = 154.0kg
MAX_PROP_MASS            = 154.0kg
TANK_MASS_2_PROPELLANT_RATIO = 0.08
THROTTLE_TYPE            = bang-bang
MAX_THRUST                = 20.9E-3
SPECIFIC_IMPULSE          = 3810s
NEP_DECAY_CONSTANT        = -2.196450873E-5 // = -ln(2)/T_half_life

```

D.3.7 Simulation Configuration of Phase 1

```

SIM_START_TIME_MIN        = 58235.0
SIM_START_TIME_MAX        = 58265.0
MIN_SOLAR_DISTANCE        = 0.7AU
INTEGRATION_INTERVAL     = 3JYR
FLIGHT_TIME_MIN           = 600day
INTEGRATION_STEPS         = 3000
MIN_OUTPUT_POINTS         = 100
DISTURBING_BODIES        = disturbance.sim
INITIAL_STATE             = body
INITIAL_CENTRAL_BODY      = sun
INITIAL_BODY_NAME         = earth
INITIAL_VINF_MIN          = 6.0KM/S
INITIAL_VINF_MAX          = 6.0KM/S
INITIAL_VINF_AZIMUTH_MIN  = 0DEG
INITIAL_VINF_AZIMUTH_MAX  = 90DEG
INITIAL_VINF_ELEV_MIN     = 0DEG
INITIAL_VINF_ELEV_MAX     = 0DEG
LAUNCHER_MAX_C3_CAPACITY  = 7200kg

```

```

LAUNCHER_C3_EXPONENT           = 0.032
TARGET_STATE                   = gravity assist
TARGET_BODY_NAME               = Jupiter
TARGET_DIST_MAX_FINAL         = 20E6km
TARGET_DIST_MAX_INIT          = 20E6km
TARGET_DIST_MAX_SHRINK        = 0.9
TARGET_DIST_MAX_DECREASE      = 0.1AU
TARGET_DIST_MAX_REDUCTION_USE_MAX = FALSE
OPTIMIZATION_GOAL             = minimum transfer time
INTEGRATOR                    = RK54F
MAX_RELATIVE_ERROR            = 1E-6
MAX_ABSOLUTE_ERROR            = 1E-6
USE_ITGR_STOPPER              = NO
USE_DSSC                      = YES
DSSC_STEP_ANGLE_CONTROL      = YES
DSSC_MAX_STEP_ANGLE          = 1.0deg
DSSC_MAX_STEP_SIZE           = 10DAY
DSSC_MIN_STEP_SIZE           = 0.5min
SC_CONF                       = SEP.scp
NAV_TYPE                      = ANN
NAV_ANN_CONF                  = SEP.ncp
STEERING_DYN_UNIT_CALC        = TRUE
STEERING_USE_ACC_THRUST_MAX   = TRUE
STEERING_USE_ACC_THRUST_MAX_DRY = TRUE
STEERING_USE_STEP_SIZE        = TRUE
STEERING_USE_TIME_UNTIL_PERI_SC = TRUE
STEERING_USE_RANGE            = TRUE
STEERING_USE_RANGE_RATE       = TRUE
STEERING_USE_TIME_UNTIL_PERI_TGT = TRUE
STEERING_USE_ABS_CART_POS_X    = TRUE
STEERING_USE_ABS_CART_POS_Y    = TRUE
STEERING_USE_ABS_CART_POS_Z    = TRUE
STEERING_USE_ABS_CART_VEL_X    = TRUE
STEERING_USE_ABS_CART_VEL_Y    = TRUE
STEERING_USE_ABS_CART_VEL_Z    = TRUE
STEERING_USE_ABS_POLAR_POS_R    = TRUE
STEERING_USE_ABS_POLAR_POS_AZI = TRUE
STEERING_USE_ABS_POLAR_POS_ELE = TRUE
STEERING_USE_ABS_POLAR_VEL_R    = TRUE
STEERING_USE_ABS_POLAR_VEL_AZI = TRUE
STEERING_USE_ABS_POLAR_VEL_ELE = TRUE

```

D.3.8 Simulation Configuration of Phase 2

```

INDEPENDENT_FLIGHT_PHASE      = FALSE
SIM_START_TIME_MIN           = 0
SIM_START_TIME_MAX           = 0
INTEGRATION_INTERVAL         = 30JYR
FLIGHT_TIME_MIN              = 20JYR

```

```
MIN_SOLAR_DISTANCE           = 0.7
INTEGRATION_STEPS           = 6000
MIN_OUTPUT_POINTS           = 250
DYN_INTEGRATION_INTERVAL    = yes
INITIAL_STATE                = body
INITIAL_BODY_NAME           = earth
INITIAL_VINF_MIN            = 0.0
INITIAL_VINF_MAX            = 0.0
INITIAL_VINF_AZIMUTH_MIN    = 0.0
INITIAL_VINF_AZIMUTH_MAX    = 0.0
INITIAL_VINF_ELEV_MIN       = 0.0
INITIAL_VINF_ELEV_MAX       = 0.0
TARGET_STATE                 = body orbit flyby
TARGET_BODY_NAME            = 2009IHP05
TARGET_DIST_MAX_FINAL       = 5AU
TARGET_DIST_MAX_INIT        = 5AU
TARGET_DIST_MAX_SHRINK      = 0.9
TARGET_DIST_MAX_DECREASE    = 0.1AU
TARGET_DIST_MAX_REDUCTION_USE_MAX = FALSE
OPTIMIZATION_GOAL           = minimum transfer time
ACCURACY_FITNESS_FRACTION   = 0.01
INTEGRATOR                   = RKF54
MAX_RELATIVE_ERROR          = 1e-6
MAX_ABSOLUTE_ERROR          = 1e-6
USE_ITGR_STOPPER            = YES
USE_DSSC                     = YES
DSSC_STEP_DISTANCE_CONTROL  = YES
DSSC_MAX_STEP_DISTANCE      = 0.1AU
DSSC_STEP_ANGLE_CONTROL     = YES
DSSC_MAX_STEP_ANGLE         = 1.0deg
DSSC_MAX_STEP_SIZE          = 9DAY
DSSC_MIN_STEP_SIZE          = 10min
SC_CONF                      = REP.scp
NAV_TYPE                     = ANN
NAV_ANN_CONF                 = REP.ncp
STEERING_DYN_UNIT_CALC      = TRUE
STEERING_USE_ACC_THRUST_MAX = TRUE
STEERING_USE_ACC_THRUST_MAX_DRY = TRUE
STEERING_USE_STEP_SIZE      = TRUE
STEERING_USE_TIME_UNTIL_PERI_SC = TRUE
STEERING_USE_RANGE          = FALSE
STEERING_USE_RANGE_RATE     = FALSE
STEERING_USE_TIME_UNTIL_PERI_TGT = FALSE
STEERING_USE_ABS_CART_POS_X = TRUE
STEERING_USE_ABS_CART_POS_Y = TRUE
STEERING_USE_ABS_CART_POS_Z = TRUE
STEERING_USE_ABS_CART_VEL_X = TRUE
STEERING_USE_ABS_CART_VEL_Y = TRUE
STEERING_USE_ABS_CART_VEL_Z = TRUE
```

```
STEERING_USE_ABS_POLAR_POS_R      = TRUE
STEERING_USE_ABS_POLAR_POS_AZI     = TRUE
STEERING_USE_ABS_POLAR_POS_ELE     = TRUE
STEERING_USE_ABS_POLAR_VEL_R       = TRUE
STEERING_USE_ABS_POLAR_VEL_AZI     = TRUE
STEERING_USE_ABS_POLAR_VEL_ELE     = TRUE
```


Bibliography

- [1] A. J. ABRAHAM, D. B. SPENCER, AND T.J. HART. **Preliminary 2-D Optimization of Low-Thrust, Geocentric-to-Halo-Orbit Transfers via Particle Swarm Optimization.** In *Proceedings of the 24th Space Flight Mechanics Conference*, Santa Fe, New Mexico, U.S.A., January 2014. AAS/AIAA.
- [2] K. ALEMANY AND R. D. BRAUN. **Survey of Global Optimization Methods for Low-Thrust, Multiple Asteroid Tour Missions.** In M. R. AKELLA, J. W. GEARHART, R. H. BISHOP, AND A. J. TREDER, editors, *Proceedings of the AAS/AIAA 17th Space Flight Mechanics Meetings held Jan. 28 - Feb. 1, 2007, Sedona, AZ*, **127** of *Advances in the Astronautical Sciences*, P.O. Box 28130, San Diego, California 92198, 2007. AAS/AIAA, Univelt, Incorporated.
- [3] B. P. ANCAROLA. **Ariane 5 Performance Optimisation for Interplanetary Missions.** In *Collection of Papers from the AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Monterey, California, 5-8 August 2002*, Reston, VA, 2002. AAS/AIAA, AIAA.
- [4] M.A. BARUCCI, M. FULCHIGNONI, S. FORNASIER, E. DOTTO, P. VERNAZZA, M. BIRLAN, R.P. BINZEL, J. CARVANO, F. MERLIN, C. BARBIERI, AND I. BELSKAYA. **Asteroid target selection for the new Rosetta mission baseline.** *Astronomy & Astrophysics*, **430**(1):313–317, January 2005.
- [5] R. R. BATE, D. D. MUELLER, AND J. E. WHITE. *Fundamentals of Astrodynamics*. Dover Publications, Inc., 180 Varick Street, New York, N.Y. 10014, first edition, 1971.
- [6] R. H. BATTIN. *An introduction to the mathematics and methods of astrodynamics*. AIAA, New York, N.Y., first edition, 1987.
- [7] J. K. BEATTY, C. C. PETERSEN, AND A. (EDS.) CHAIKIN. *The New Solar System*. Cambridge University Press, Cambridge, New York, NY, fourth edition, 1999.
- [8] J. T. BETTS. **Using sparse nonlinear programming to compute low thrust orbit transfers.** *The Journal of the Astronautical Sciences*, **41**:349–371, 1993.
- [9] J. T. BETTS. **Survey of Numerical Methods for Trajectory Optimization.** *Journal of Guidance, Control, and Dynamics*, **21**(2):193–207, March 1998.
- [10] J. T. BETTS AND S. O. ERB. **Optimal Low Thrust Trajectories to the Moon.** *SIAM Journal of Applied Dynamical Systems*, **2**(2):144–170, 2003.
- [11] C. BRAMANTI, D. IZZO, T. SAMARAE, R. WALKER, AND D. FEARN. **Very high delta-V missions to the edge of the solar system and beyond enabled by the dual-stage 4-grid ion thruster concept.** *Acta Astronautica*, **64**:735–744, April–May 2009.
- [12] D. BÜCHE, G. GUIDATI, AND P. STOLL. **Automated Design Optimization Of Compressor Blades For Stationary, Large-Scale Turbomachinery.** In *Proceedings of the ASME Turbo Expo 2003, collocated with the 2003 International Joint Power Generation Conference, held June 16-19 2003, Atlanta, GA*, Fairfield,, September 2003. American Society of Mechanical Engineers, ASME.

- [13] B. DACHWALD. *Low-Thrust Trajectory Optimization and Interplanetary Mission Analysis Using Evolutionary Neurocontrol*. PhD thesis, University of the Armed Forces Germany, 2003.
- [14] B. DACHWALD, A. OHNDORF, AND B. WIE. **Solar Sail Trajectory Optimization for the Solar Polar Imager (SPI) Mission**. In *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference held August 21-24 2006, Keystone, Colorado*. AAS/AIAA, AIAA, 2006.
- [15] B. DACHWALD, W. SEBOLDT, HORST W. LÖB, AND K.-H. SCHARTNER. **A Comparison of SEP and NEP for a Main Belt Asteroid Sample Return Mission**. In *Proceedings of the 7th International Symposium on Launcher Technologies*, Barcelona, Spain, April 2007.
- [16] C. DARWIN. *On the Origin of Species by Means of Natural Selection*. John Murray, London, first edition, 1859.
- [17] K. A. DE JONG. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, 1975.
- [18] K. A. DE JONG. **Learning with Genetic Algorithms: An Overview**. *Machine Learning*, **3**:121–138, 1988.
- [19] K. A. DE JONG. *Evolutionary Computation: A Unified Approach*. MIT Press, Cambridge, MA, first edition, 2006.
- [20] U. DERZ AND W. SEBOLDT. **Mars Sample Return Architecture Utilizing Low Thrust Propulsion**. *Acta Astronautica*, **77**:83–96, August–September 2012.
- [21] D. DOODY. *Deep Space Craft: An Overview of Interplanetary Flight*. Praxis Publishing Ltd, Chichester, UK, first edition, 2009.
- [22] M. C. ECKSTEIN. **Optimal station keeping by electric propulsion with thrust operation constraints**. *Celestial Mechanics and Dynamical Astronomy*, **21**:129–147, 1980. 10.1007/BF01230889.
- [23] T. N. EDELBAUM. **Optimum Power-Limited Orbit Transfer in Strong Gravity Field**. *AIAA Journal*, **3**(5):921–925, May 1963.
- [24] Electric Rocket Propulsion Society. *Proceedings of the 32nd IEPC held September 11-15 2011, Kurhaus, Wiesbaden, Germany*, 2011.
- [25] N. S. FATEMI, S. SHARMA, O. BUITRAGO, J. CRISMAN, P. R. SHARPS, R. BLOK, M. KROON, JALINK CEES, R. HARRIS, P. STELLA, AND S. DISTEFANO. **Performance of high-efficiency advanced triple-junction solar panels for the LILT Mission Dawn**. In *Conference Record of the Thirty-first IEEE Photovoltaic Specialists Conference*, pages 618–621, Coronado Springs Resort, Lake Buena Vista, FL, January 3-7 2005. IEEE, IEEE Press Piscataway, NJ, USA.
- [26] E. FEHLBERG. **Classical Fifth-, Sixth-, Seventh-, and Eighth-Order Runge-Kutta Formulas with Step Size Control**. Technical Report TR R287, George C. Marshal Space Flight Center, Huntsville, Ala., 1968.
- [27] R. L. FIEHLER, D. MCNUTT JR. **Radioisotope electric propulsion (REP): A near-term approach to nuclear propulsion**. *Journal of Spacecraft and Rockets*, **43**(6):1239–1247, 2006.
- [28] D. B. FOGEL. **An Introduction to Simulated Evolutionary Optimization**. *Transactions on Neural Networks*, **5**(1):3–14, January 1994.
- [29] B. H. FOING, G. D. RACCA, A. MARINI, E. EVRARD, L. STAGNARO, M. ALMEIDA, D. KOSCHNY, D. FREW, J. ZENDER, J. HEATHER, M. GRANDE, H.U. HUOVELIN, J. KELLER, A. NATHUES, J. L. JOSSET, A. MALKKI, W. SCHMIDT, G. NOCI, R. BIRKL, L. TESS, Z. SODNIK, AND P. MCMANAMON. **SMART-1 mission to the Moon: Status, first results and goals**. *Advances in Space Research*, **37**(1):6–13, 2006. The Moon and Near-Earth Objects.

- [30] B. H. FOING AND G. R. RACCA. **The ESA SMART-1 mission to the Moon with solar electric propulsion.** *Advances in Space Research*, **23**(11):1865–1870, 1999. The Moon and Mars.
- [31] P. FORTESCUE, J. STARK, AND G. SWINERD. *Spacecraft Systems Engineering*. John Wiley & Sons, Chichester, West Sussex, PO19 8SQ, England, third edition, 2003.
- [32] S. GEFFROY AND R. EPELOY. **Optimal Low-Thrust Transfers with Constraints – Generalization of Averaging Techniques.** *Acta Astronautica*, **41**(3):133–149, August 1997.
- [33] I. GERDES. **Application of Genetic Algorithms to the problem of free routing for aircraft.** In *Proc. of the IEEE World Congress on Computational Intelligence: First IEEE Conference on Evolutionary Computation*, **2**, pages 536–541, Orlando, FL, June 1994. IEEE.
- [34] I. GERDES. **Application of Genetic Algorithms for solving problems related to free routing for aircraft.** In J. BIETHAN AND V. NISSEN, editors, *Evolutionary Algorithms in Management Applications*, pages 328–340. Springer, Secaucus, NJ, USA, 1995.
- [35] D. M. GOEBEL AND I. KATZ. *Fundamentals of Electric Propulsion: Ion and Hall Thrusters*. John Wiley & Sons, Hoboken, NJ, 2008.
- [36] Y. GUO AND R. W. FARQUHAR. **New Horizons Mission Design.** *Space Science Reviews*, **140**:49–74, October 2008.
- [37] I. HADAR AND A. GANY. **Augmentation of low power hydrazine thrusters.** *Acta Astronautica*, **32**(4):275–281, April 1994.
- [38] D. P. HAMILTON AND J. A. BRUNS. **Orbital Stability Zones about Asteroids: II. The Destabilizing Effect of Eccentric Orbits and of Solar Radiation.** *Icarus*, **96**:43–64, March 1992.
- [39] J. W. HARTMANN. *Low-Thrust Trajectory Optimization Using Stochastic Optimization Techniques*. Master’s thesis, University of Illinois at Urbana-Champaign, 1996.
- [40] A. L. HERMAN AND B. A. CONWAY. **Optimal, Low-Thrust, Earth-Moon Orbit Transfer.** *Journal of Guidance, Control, and Dynamics*, **21**(1):131–147, January 1998.
- [41] W. HOHMANN. *Die Erreichbarkeit der Himmelskörper*. Oldenburg Verlag, München, 1925.
- [42] T. E. HULL, W. H. ENRIGHT, AND A. E. SEDGWICK. **Comparing Numerical Methods For Ordinary Differential Equations.** *SIAM Journal on Numerical Analysis*, **9**(4):603–637, Dec 1972.
- [43] IEEE. *Proceedings of the First IEEE Conference on Evolutionary Computation*, Orlando, FL, USA, 1994. IEEE Press Piscataway, NJ, USA.
- [44] IEEE. *Proceedings of the IEEE Congress on Evolutionary Computation 2009*, Trondheim, Norway, May 2009. IEEE Press Piscataway, NJ, USA.
- [45] L. D. JAFFE AND C. V. IVIE. **Science Aspects of a Mission Beyond the Planets.** *Icarus*, **39**(3):486–494, 1979.
- [46] R. G. JAHN. *Physics of Electric Propulsion*. McGraw-Hill, New York, 1968.
- [47] E. R. KANDEL, J. H. SCHWARTZ, AND T. M. JESSEL. *Principles of Neural Science*. McGraw-Hill Professional, New York, fourth edition, 2000.
- [48] J. KAWAGUCHI, A. FUJIWARA, AND T. UESUGI. **Hayabusa–Its technology and science accomplishment summary and Hayabusa-2.** *Acta Astronautica*, **62**(10–11):639–647, May 2008.

- [49] K. KRISHNAKUMAR AND D. E. GOLDBERG. **Control System Optimization Using Genetic Algorithms**. *Journal of Guidance, Control, and Dynamics*, **15**(3):735–740, 1992.
- [50] P. C. LIEWER, J. A. MEWALDT, R. A. AND AYON, C. GARNER, S. GAVIT, AND R. A. WALLACE. **Interstellar Probe Using a Solar Sail: Conceptual Design and Technological Challenges**. In K. SCHERER, H. FICHTNER, H. JÖRG FAHR, AND E. MARSCH, editors, *The Outer Heliosphere: The Next Frontiers*, **11** of *COSPAR Colloquia Series*, pages 411–420, Amsterdam, 2001. Pergamon Press.
- [51] H. W. LÖB, K. H. SCHATNER, B. DACHWALD, A. OHNDORF, AND W. SEBOLDT. **Heliopause Probe-SEP Option**. Technical report, DLR, 2011.
- [52] H. W. LÖB, K.-H. SCHATNER, B. DACHWALD, A. OHNDORF, AND W. SEBOLDT. **Solar/Radioisotope Electric Propulsion Combination for a Mission to the Heliopause**. In *Proceedings of the 32nd IEPC held September 11-15 2011, Kurhaus, Wiesbaden, Germany* [24].
- [53] A. E. LYNQVI, M. L. VAN DEN BERG, AND P. FALKNER. **Study Overview of the Interstellar Heliopause Probe**. Technical Report SCI-A/2006/114/IHP, ESA, Noordwijk, The Netherlands, April 2007.
- [54] K. E. MATHIAS AND L. D. WHITLEY. **Initial Performance Comparisons for the Delta Coding Algorithm**. In *Proceedings of the First IEEE Conference on Evolutionary Computation* [43], pages 433–438.
- [55] C. R. MCINNES. *Solar Sailing: Technology, Dynamics, and Mission Applications*. Springer, Berlin, Heidelberg, New York, first edition, Second printing 2004.
- [56] R. L. MCNUTT JR., R. F. WIMMER-SCHWEINGRUBER, AND THE INTERNATIONAL INTERSTELLAR PROBE TEAM. **Enabling Interstellar Probe**. *Acta Astronautica*, **68**(7-8):790–801, April 2011.
- [57] O. MONTENBRUCK AND E. GILL. *Satellite Orbits: Models, Methods and Applications*. Springer, Berlin, Heidelberg, New York, third edition, 2005.
- [58] C. D. MURRAY AND S. F. DERMOTT. *Solar System Dynamics*. Cambridge University Press, Cambridge, New York, Melbourne, 1999.
- [59] NASA, 4800 Oak Grove Drive, Pasadena, California 91109. *Solar System Dynamics Group Website*, February 2009.
- [60] R. M. NELSON, M. D. RAYMAN, AND H. A. WEAVER. **The Deep Space 1 encounter with Comet 19P/Borrelly**. *Icarus*, **167**(1):1–3, 2004. Special Issue on DS1/Comet Borrelly.
- [61] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *National Institute of Standards and Technology Website*. <http://physics.nist.gov/cuu/index.html>.
- [62] A. OHNDORF AND B. DACHWALD. **INTRANCE - A Tool for Multi-objective Multi-phase Low-thrust Trajectory Optimization**. In *4th International Conference on Astrodynamical Tools and Techniques held May 3-6 2010 at the European Astronomy Centre, Madrid, Spain*, 2010.
- [63] A. OHNDORF, B. DACHWALD, AND E. GILL. **Optimization of Low-thrust Earth-Moon Transfers Using Evolutionary Neurocontrol**. In *Proceedings of the IEEE Congress on Evolutionary Computation 2009* [44], pages 358–364.
- [64] A. OHNDORF, B. DACHWALD, H. W. SEBOLDT, W. AND LÖB, AND K.-H. SCHATNER. **Flight Times to the Heliopause Using a Combination of Solar and Radioisotope Electric Propulsion**. In *Proceedings of the 32nd IEPC held September 11-15 2011, Kurhaus, Wiesbaden, Germany* [24].

- [65] M. J. PATTERSON AND S. W. BENSON. **NEXT Ion Propulsion System Development Status and Capabilities**. In *Proceedings of the 2007 NASA Science and Technology Conference*, Adelphi, MD, June 2007.
- [66] M. J. PATTERSON, F. ELLIOTT, S. P. MALLONE, G. C. SOULAS, D. GOEBEL, AND A. SENGUPTA. **Herakles Thruster Development for the Prometheus JIMO Mission**. In *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Tucson, AZ, USA, July 2005. AIAA/ASME/SAE/ASEE.
- [67] A. E. PETROPOULOS. **Simple Control Laws for Low-Thrust Orbit Transfers**. In *Proceedings of the 2003 Astrodynamics Specialists Conference*, Big Sky Resort, Big Sky, MN, USA, August 2003. AAS/AIAA.
- [68] A. E. PETROPOULOS. **Low-Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanism for Coasting**. In *Proceedings of the AIAA/AAS Astrodynamics Specialists Conference and Exhibit*, Providence, Rhode Island, Aug 16-19 2004.
- [69] A. E. PETROPOULOS. **Refinement to the Q-law for low-thrust orbit transfers**. In *Proceedings of the 15th Space Flight Mechanics Conference*, Copper Mountain, Colorado, U.S.A, January 2005. AAS/AIAA.
- [70] M. D. RAYMAN, T. C. FRASCHETTI, C. A. RAYMOND, AND C. T. RUSSEL. **Dawn: A Mission in Development for Exploration of Main Belt Asteroids Vesta and Ceres**. *Acta Astronautica*, **58**(11):605–616, June 2006.
- [71] M. D. RAYMAN, T. C. FRASCHETTI, C. A. RAYMOND, AND C. T. RUSSEL. **Coupling of System Resource Margins Through The Use of Electric Propulsion: Implications in Preparing for the Dawn Mission to Ceres and Vesta**. *Acta Astronautica*, **60**:930–938, May 2007.
- [72] M. D. RAYMAN AND R. A. MASE. **DAWN’S Exploration of Vesta**. In *63rd International Astronautical Congress, October 1-5 2012, Naples, Italy*, 2010.
- [73] M. D. RAYMAN AND R. A. MASE. **The Second Year of Dawn Mission Operations: Mars Gravity Assist and Onward to Vesta**. *Acta Astronautica*, **67**(3-4):483–488, August 2010.
- [74] M. D. RAYMAN AND K. C. PATEL. **The Dawn Project’s Transition to Mission Operations: On Its Way to Rendezvous with (4) Vesta and (1) Ceres**. *Acta Astronautica*, **66**(1-2):230–238, January 2010.
- [75] R. D. REED AND R. J. MARKS. *Neural Smoothing: Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press, Cambridge, MA, first edition, 1998.
- [76] H. H. ROSENBROCK. **An Automatic Method for finding the Greatest or Least Value of a Function**. *The Computer Journal*, **3**(3):175–184, 1960.
- [77] D. E. RUMELHART, G. E. HINTON, AND R. J. WILLIAMS. **Learning internal representations by back-propagating errors**. *Nature*, **232**:533–536, 1986.
- [78] C.T. RUSSEL, M.A. BARUCCI, R.P. BINZEL, M.T. CAPRIA, U. CHRISTENSEN, A. CORADINI, M.C. DE SANCTIS, W.C. FELDMAN, R. JAUMANN, H.U. KELLER, A.S. KONOPLIV, T.B. MCCORD, L.A. MCFADDEN, K.D. MCKEEGAN, H.Y. MCSWEEN, C.A. RAYMOND, H. SIERKS, D.E. SMITH, T. SPOHN, M.V. SYKES, F. VILAS, AND M.T. ZUBER. **Exploring the asteroid belt with ion propulsion: Dawn mission history, status and plans**. *Advances in Space Research*, **40**:193–201, 2007.
- [79] C. SAUER. **A Comparison of Solar Sail and Ion Drive Trajectories for a Halley’s Comet Rendezvous Mission**. In *Proceedings of AIAA/AAS Astrodynamics Conference*, Jackson, USA, September 1977.

- [80] G. R. SCHMIDT, D. H. MANZELLA, H. KAMHAWI, T. KREMIC, S. R. OLESON, J. W. DANKANICH, AND L. A. DUDZINSKI. **Radioisotope electric propulsion (REP): A near-term approach to nuclear propulsion.** *Acta Astronautica*, **66**(3–4):501–507, February 2010.
- [81] N. N. SCHRAUDOLPH AND R. K. BELEW. **Dynamic Parameter Encoding for Genetic Algorithms.** *Machine Learning*, **9**(1):9–21, July 1992.
- [82] S. C. SHAPIRO, editor. *Encyclopedia of Artificial Intelligence.* John Wiley & Sons, New York, N.Y., second edition, 1992.
- [83] W. M. SPEARS, K. A. DE JONG, T. BÄCK, D. B. VOGEL, AND H. DE GARIS. **An Overview of Evolutionary Computation.** In *Proceedings of the 1993 European Conference on Machine Learning*, pages 442–459, Vienna, Austria, Apr 5-7 1993.
- [84] E. STUHLINGER. *Ion Propulsion for Space Flight.* McGraw-Hill, New York, 1964.
- [85] B. N. TAYLOR AND ED. THOMPSON, A. **The International System of Units.** Technical Report Spec. Pub. 330, National Institute of Standards and Technology, Gaithersburg, MD, 2008.
- [86] R. C. TAYLOR, S. TAPIA, AND E. F. TEDESCO. **The rotation period and pole orientation of asteroid 4 Vesta.** *Icarus*, **62**:298–304, May 1985.
- [87] W. T. THOMSON. *Introduction to Space Dynamics.* Dover Publications, Mineola, N.Y., second edition, 1986.
- [88] L. TSINAS AND B. DACHWALD. **A Combined Neural and Genetic Learning Algorithm.** In *Proceedings of the First IEEE Conference on Evolutionary Computation* [43].
- [89] M. J. L. TURNER. *Rocket and Spacecraft Propulsion.* Springer, Berlin, Heidelberg, New York, third edition, 2009.
- [90] M. URSELMANN, G. SAND, AND S. ENGEL. **A Memec Algorithm for Global Optimization in Chemical Process Synthesis.** In *Proceedings of the IEEE Congress on Evolutionary Computation 2009* [44], pages 1721–1728.
- [91] D. A. VALLADO. *Fundamental of Astrodynamics and Applications.* Microcosm Press and Springer, Hawthorne, CA and New York, third edition, 2007.
- [92] G. VULPETTI, L. JOHNSON, AND G. L. MATLOFF. *Solar Sails: A Novel Approach to Interplanetary Travel.* Copernicus Books, New York, first edition, 2008.
- [93] R. WALKER, C. BRAMANTI, O. SUTHERLAND, R. BOSWELL, C. CHARLES, D. FEARN, J. DEL AMO, P. FRIGOT, AND M. ORLANDI. **Initial Experiments on a Dual-Stage 4-Grid Ion Thruster for Very High Specific Impulse and Power.** In *Proceedings of the 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Sacramento, CA, July 9-12 2006.
- [94] G. J. WHIFFEN. **Optimal Low-Thrust Orbital Transfers Around a Rotating Nonspherical Body.** In *Proceedings of the 14th Space Flight Mechanics Conference*, Maui, Hawaii, U.S.A, February 2004. AAS/AIAA.
- [95] G. J. WHIFFEN AND J. A. SIMS. **Application of a Novel Optimal Control Algorithm To Low-Thrust Trajectory Optimization.** In *Proceedings of the 2001 Space Flight Mechanics Conference*, Santa Barbara, CA, February 2001. AAS/AIAA.
- [96] D. WHITLEY. **An overview of evolutionary algorithms: practical issues and common pitfalls.** *Information and Software Technology*, **43**(14):817–831, December 2001.

-
- [97] L. D. WHITLEY AND K. E. MATHIAS. **Changing Representations During Search: A comparative Study of Delta Coding.** *Evolutionary Computation*, **2**(3):249–278, December 1994.
- [98] L. D. WHITLEY, K. E. MATHIAS, AND P. A. FITZHORN. **Delta Coding: An Iterative Search Strategy for Genetic Algorithms.** In RICHARD K. BELEW AND LASHON B. BOOKER, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, San Diego, CA, USA, July 1991. Morgan Kaufmann.
- [99] S. N. WILLIAMS AND V. COVERSTONE-CARROLL. **Mars Missions Using Solar Electric Propulsion.** *Journal of Spacecraft and Rockets*, **37**(1):71–77, January 2000.
- [100] R. F. WIMMER-SCHWEINGRUBER, R. L. MCNUTT JR., N. A. SCHWADRON, P. C. FRISCH, M. GRUNTMAN, P. WURZ, AND E. VALTONEN. **Interstellar heliospheric probe/heliospheric boundary explorer mission – a mission to the outermost boundaries of the solar system.** *Experimental Astronomy*, **24**:9–46, May 2009.
- [101] G. P. ZANK. **INTERACTION OF THE SOLAR WIND WITH THE LOCAL INTERSTELLAR MEDIUM: A THEORETICAL PERSPECTIVE.** *Space Science Reviews*, **89**:413–688, July 1999.

Index

- ΔV , 15
- acceleration
 - characteristic, 17
 - norm, 9
- agent, 43
- AI, 42, 43
 - general, 42
 - strong, 42
- Algorithm
 - Genetic, 49
- algorithm
 - evolutionary, 49
- angle
 - centerline, 21
 - inclination, 134
 - pitch, 18
 - thrust cone, 21
- annaeling
 - simulated, 39
- anomaly
 - eccentric, 136
 - hyperbolic, 136
 - mean, 135
 - true, 136
- apoapsis, 133
- approximation
 - patched conics, 33
 - patched two-body, 33
 - patched-conic, 33
- arcjet, 11
- argument
 - pericenter, 135
- ascension
 - right
 - ascending node, 134
- ASRG, 14
- axion, 44
- axis
 - semimajor, 133
- Back-Propagation, 49
- body
 - black, 139
 - grey, 139
- cell
 - nerve, 44
- chromosome, 50
- coding
 - delta, 86
 - floating-point, 87
 - real, 87
- coefficient
 - absorption, 20
 - reflection, 20
 - transmission, 20
- coldstart, 89
- Computing
 - Evolutionary, 49
- constant
 - decay, 15
 - solar, 16
- control
 - step size
 - dynamic, 84
- crossover, 51
- curve
 - optimality
 - pareto, 25
- DC, 92
- Deep Space 1, 9, 12, 28
- dendrite, 44
- deviation, 79
 - distance, 80
 - velocity, 80
- diploid, 51
- diversity
 - genetic, 51
- DNA, 51
- DOF, 71
- Dose
 - Radiation

- Solar, 25
- DS4G, 7
- DSSC, 83
- eccentricity, 133
- effect
 - photovoltaic, 13
- efficiency
 - PPU, 10
- elements
 - Keplerian, 133
 - orbital, 133
 - two-body, 133
- emissivity, 139
- ENC, 41, 65
- encoding
 - dynamic parameter, 85
- end of mission, 14
- energy
 - excess
 - hyperbolic, 78
 - orbit
 - specific, 133
- engine
 - ion, 11, 14
- EOM, 14, 71
 - cartesian, 72
 - polar, 72
- epoch, 23, 87
- EPS, 10
- equations
 - differential
 - ordinary, 74
 - motion, 58
 - of motion, 71
 - planetary
 - Lagrange, 35
- errors
 - square
 - sum of, 48
- evaluation, 43, 54
- evolution
 - differential, 39
 - natural, 49
- factor
 - weight, 45
 - weighting, 26
- FB, 44
- fitness, 52
- flight path, 23
- flow
 - mass
 - propellant, 10
- flux
 - radiation
 - solar, 16
- flyby, 30
- force
 - SRP, 16
 - thrust, 9
- frame
 - reference
 - orbit, 132
 - polar, 132
- freedom
 - degree of, 71
- function
 - control, 30
 - network, 47
 - shadow, 76
- gene, 51
- generator
 - Brayton, 14
 - Stirling, 14
 - thermoelectric
 - radioisotope, 14
- genome, 50
- genotype, 51
- GEO, 103
- GPHS, 15
- half-life, 14
- halftime, 15
- haploid, 51
- Hayabusa, 9
- HET, 12
- history
 - control vector
 - optimal, 31
- hot spot, 17
- impulse
 - mission
 - characteristic, 7
 - specific, 2, 7
- inclination, 134
- increment
 - velocity, 9
- individual, 49
- inheritance, 51
- intelligence
 - artificial, 41

- IPS, 10
- JIMO, 14
- JPL, 103
- Lambert solver, 28
- launch
 - piggy-back, 78
- law
 - steering
 - law, 35
- learning
 - machine, 41, 43
 - reinforcement, 43
- LH2, 9
- line of nodes, 135
- loading
 - assembly
 - sail, 17
 - sail, 17
 - sail assembly, 17
 - sailcraft, 17
- LOX, 7
- maneuver
 - correction
 - midcourse, 26
- mapping
 - associative, 43
- mass
 - assembly
 - sail, 17
 - burnout, 9
 - dry, 9
 - launch, 9
 - wet, 9
- mass flow
 - propellant, 9
- MCM, 26
- MENC, 41, 65, 99
- method
 - direct, 36
 - indirect, 36
 - optimization
 - trajectory, 36
 - trajectory optimization
 - global, 36, 38
 - local, 36
- ML, 43
- MLP, 46
- model
 - force
 - ideal, 18
 - simple, 18
 - solar sail, 18
 - standard, 18
 - standard SRP, 20
 - SRP force
 - simple, 19
- momentum
 - angular
 - specific, 133
- motion
 - mean, 135
- MPDT, 13, 14
- mutation, 51, 54
 - uniform, 92
 - fast, 92
- Mystic, 103
- networks
 - neural
 - natural, 44
- Neurocontrol
 - Evolutionary
 - Multiphase, 55
- neurocontroller, 56
- neuron, 44
 - motor, 44
 - relay, 44
 - sensory, 44
- neurotransmitter, 44
- NPGS, 29
- NSTAR, 12, 60
- number
 - lightness, 17
- objective
 - optimization, 24
- objectives
 - optimization, 24
- ODE, 74
- optimality, 24
- optimization
 - multi-objective, 25
 - particle swarm, 39
 - single-objective, 25
 - trajectory, 23
- orbit
 - direct, 134
 - ecliptical, 134
 - inclined, 134
 - polar, 134
 - prograde, 134

- retrograde, 134
- PAEHT, 11
- panel
 - solar, 13
- parameter
 - efficiency
 - sail, 19
- Pareto, Vilfredo, 25
- pareto-optimal, 25
- patched conics, 33
- perceptron
 - multilayer, 46
- performance metric
 - solar sail, 17
- periapsis, 133
- period
 - orbit, 136
- phenotype, 51
- policy, 43
- pool
 - genome, 50
- population, 49
- power
 - beam, 10
- power generation
 - electric
 - nuclear, 13
- PPT, 12
- PPU, 10
- pressure
 - radiation
 - pressure, 16
 - solar radiation, 16
- principle
 - sailing
 - solar, 16
- problem
 - optimization
 - multiphase low-thrust, 26
 - reinforcement
 - delayed, 43
 - immediate, 43
 - value
 - boundary, 37
- Programming
 - Genetic, 49
- programming
 - dynamic, 39
 - genetic, 39
 - nonlinear, 36
- propulsion
 - electric, 7
 - nuclear, 9, 13
 - radioisotope, 9, 14, 15
 - solar, 9, 13
 - electrodynamic, 11, 12
 - electrostatic, 11
 - electrothermal, 11
 - low-thrust, 7
 - system
 - ion, 10
- proximity, 79
 - flyby, 80
 - rendezvous, 80
- PU-238, 14
- PV, 13
- Q-law, 36
- recombination, 51, 54
- reinforcement, 43
- rendezvous, 30
- REP, 15, 116
- reproduction, 54
 - one-at-a-time, 91
 - steady-state, 91
- resistojet, 11
- reward, 43
- RF, 11
- RPS, 14
- RTG, 14
- Runge-Kutta, 74
- RV, 44
- sail
 - solar, 15
- scan
 - search space, 89
 - space
 - search, 87
- selection, 52
 - roulette wheel, 53
 - tournament, 53, 90
 - binary, 90
 - truncation, 53
- SEP, 116
- shooting
 - multiple, 39
- SMART-1, 9
- SOI, 41, 65, 67
- solar cell, 13
- solar sail, 7, 15

- solution
 - interim, 86
 - partial, 86
- soma, 44
- species, 49
- Sphere
 - Hill, 80
 - of Influence, 80
- SRP, 16
- state, 133
 - astrodynamic, 23
- stay-time, 82
- strategies
 - evolutionary, 49
- strategy, 43
 - steering, 28
 - direct, 57
 - indirect, 57
- superindividual, 90
- synapse, 44
- system
 - equations
 - differential, 58
 - nervous, 44
 - power
 - radioisotope, 14
 - propulsion
 - electric, 10
- technology
 - propulsion
 - electric, 11
- theory
 - control
 - optimal, 30
- thruster
 - Hall effect, 12
 - ion
 - radiofrequency, 12
 - pulsed plasma, 12
- time
 - dwell, 82
- time-of-flight, 23
- tournament
 - binary, 53
- training, 48
- trajectory, 23
 - high-thrust, 26
 - low-thrust, 26, 28
 - Multiphase, 32
 - single-phase, 32
- TRL, 12
- unit
 - processing
 - power, 10
- variable
 - state, 71
- vector
 - base
 - Polar, 131
 - eccentricity, 134
 - state, 71, 133
 - astrodynamic, 23
 - cartesian, 72
 - polar, 72
 - unit
 - solar sail, 16
 - sunlight direction, 16
- velocity
 - excess
 - hyperbolic, 78
- violation
 - mass, 83
 - state, 83
- warmstart, 89
- wind
 - solar, 16
- Xenon, 11

Curriculum Vitae

Born on the 20th of April 1978 in Finsterwalde, Germany

Education:

1984 - 1992	Oberschule, Klingenthal, Germany
1992 - 1996	Gymnasium Klingenthal, Germany, Abitur
1997 - 2000	Aerospace Engineering (Luft- und Raumfahrttechnik), Diploma Study, Universität der Bundeswehr München, Neubiberg, Germany

Profession:

1996 - 2008	Technical Officer in the German Air Force
07/1996 - 08/1996	Basic training, Bayreuth, Germany
09/1996 - 06/1997	Air Force Academy, Fürstenfeldbruck, Germany
01/2001 - 03/2002	Technical Officer (Fighter Airplanes), Fighter Wing 72/Flying Training Center F-4F, Rheine, Germany
04/2002 - 06/2005	Aircraft maintenance planning and funding planning, German Air Force Materiel Command, Cologne, Germany
07/2005 - 06/2008	Configuration management Eurofighter EF2000, Weapon System Command of the German Air Force, Cologne, Germany
since 2008	Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR, German Aerospace Center), Space Operations and Astronaut Training / German Space Operations Center (GSOC); Project Manager, Flight Director and System Engineer in several satellite projects, e.g., TerraSAR-X, TanDEM-X, EnMAP, PRISMA, DEOS, and PAZ

List of Publications

1. Ohndorf, A., Entwurf, Modellierung und Implementierung einer universellen Weltrepräsentation zum Einsatz in KI-Systemen, diploma thesis, Universität der Bundeswehr München, Neubiberg, 2000
2. Dachwald, B., Ohndorf, A., 1st ACT global trajectory optimisation competition: Results found at DLR, Acta Astronautica 61 (2007) 742-752
3. Ohndorf, A., Dachwald, B., Gill, E. Optimization of Low-Thrust Earth-Moon Transfers Using Evolutionary Neurocontrol, 2009 IEEE Congress on Evolutionary Computation (Conference proceedings), 18.-21. May 2009, Trondheim, Norway
4. Ohndorf, A., Eberle, S., Uhlig, T., Faller, R., Schmidhuber, M., Ballweg, R., Mission Operations, In: Spacecraft Operations, Springer. pp 35-89, ISBN 978-3-7091-1802-3
5. Eberle, S., Ohndorf, A., Faller, R., On-Orbit Servicing Mission Operations at German Space Operation Center, Space Operations: Exploration, Scientific Utilization, and Technology Development, 01/2011: pages 235-249, American Institute of Aeronautics and Astronautics., ISBN: 978-1-60086-817-7
6. Konstantinidis, K., Flores Martinez, C., Dachwald, B., Ohndorf, A., Dykta, P., Bowitz, P., Rudolph, M., Digel, I., Kowalski, J., Voigt, K., Förstner, R., A lander mission to probe subglacial water on Saturn's moon Enceladus for life, Acta Astronautica 106 (2015), 63-89, October 2014
7. Guanter, L., Kaufmann, K., Foerstner, S., Rogass, C., Chabrillat, T., Hollstein, A., Rossner, G., Chlebek, C., Straif, C., Fischer, S., Schrader, S., Storch, T., Heiden, U., Mueller, A., Bachmann, M., Mühle, H., Müller, R., Habermeyer, M., Ohndorf, A., Hill, J., Buddenbaum, H., Hoster, P., van der Linden, S., Leitao, P., Rabe, A., Doerffer, R., Krasemann, H., Xi, H., Mauser, W., Hank, T., Locherer, M., Rast, M., Staenz, K., Sang, B., The Environmental Mapping and Analysis Program (EnMAP) spaceborne imaging spectroscopy mission for Earth observation, Remote Sensing 7(7):8830, July 2015
8. Ohndorf, A., Dachwald, B., Seboldt, W., Optimierung der Lageregelung von Raumfahrzeugen mit Niedrigschubantrieb mittels evolutionärer neuronaler Regler, DGLR-Kongress, Friedrichshafen, Germany, 2005, DGLR Paper DGLR-2005-204
9. Dachwald, B., Ohndorf, A., Solar Sail Trajectory Optimization for the Solar Polar Imager (SPI) Mission, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, USA, August 2006, AIAA Paper 2006-6177
10. Seboldt, W., Dachwald, B., Streppel, J., Ohndorf, A., Löb, H., Schartner, K.-H., Lander Mission to Europa With Solarelectric Propulsion, 7th International Symposium on Launcher Technologies, Barcelona, Spain, April 2007

11. Löb, H., Schartner, K.-H., Dachwald, B., Ohndorf, A., Seboldt, W., Design of a SEP-Spacecraft for Deep Space Missions with Very Large Delta-V-Requirements, 5th International Space Propulsion Conference, 2008, Heraklion, Greece
12. Spurmann, J., Ohndorf, A., Dachwald, B., Seboldt, W., Löb, H., Schartner, K.-H. Interplanetary Trajectory Optimization for a SEP Mission to Saturn, 60th International Astronautical Congress 2009, 12.-16. October 2009, Daejeon, Korea
13. Eberle, S., Ohndorf, A., Faller, R., On-Orbit Servicing Mission Operations at GSOC, SpaceOps 2010 Conference, 25.-30. April 2010, Huntsville, Alabama
14. Löb, H., Schartner, K.-H., Dachwald, B., Ohndorf, A., Untersuchung einer Mission zur Heliopause mit solarelektrischem Antrieb, DLR study 50 RS 0901, 2010
15. Ohndorf, A., Dachwald, B., InTrance: A Tool for Multi-Objective Multi-Phase Low-Thrust Trajectory Optimization, 4th International Conference on Astrodynamics Tools and Techniques, 2010, Madrid, Spain
16. Sellmaier, F., Plura, M., Stelzer, M., Ohndorf, A., Müller, H., Landzettel, K., Mission Operation Concepts For Robotic Missions, 62nd International Astronautical Congress, 03-07 October 2011, Cape Town, South Africa
17. Derz, U., Ohndorf, A., Bischof, B., Mission Analysis of Robotic Low Thrust Missions to the Martian Moons Deimos And Phobos, 62nd International Astronautical Congress, 03-07 October 2011, Cape Town, South Africa
18. Borggräfe, A., Ohndorf, A., Dachwald, B., Seboldt, W., Analysis of Interplanetary Solar Sail Trajectories with Attitude Dynamics, Advances in the Astronautical Sciences 145, January 2012
19. Gnat, M., Falcone, R., Hauke, A., Ohndorf, A., Eberle, S., Technical and operational investigations of the real-time communication for robotic missions, 13th International Conference on Space Operations 2014, 59 May 2014, Pasadena, California
20. Ohndorf, A., Eberle, S., Faller, R., Gnat, M., Ground Segment Design for On-Orbit Servicing Missions at GSOC, 13th International Conference on Space Operations 2014, 59 May 2014, Pasadena, California
21. Maurer, E., Kahle, R., Mrowka, F., Morfill, G., Ohndorf, A., Zimmermann, S., Operational aspects of the TanDEM-X Science Phase, 14th International Conference on Space Operations, 16-20. May 2016, Daejeon, Korea