UNIVERSITY OF CALIFORNIA,
IRVINE

Rapid Onboard Trajectory Design for Autonomous Spacecraft in Multibody Systems

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mechanical and Aerospace Engineering

by

Eric Michael Trumbauer

Dissertation Committee:
Professor Kenneth D. Mease, Chair
Professor Athanasios Sideris
Professor Faryar Jabbari

2015

DEDICATION

To Damara, my family, and my friends for their love and laughter.

The world is big and I want to have a good look at it before it gets dark.

- John Muir

When the going gets weird, the weird turn pro.

- Hunter S. Thompson

TABLE OF CONTENTS

SECTION V: Conclusion

Appendices

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AFRL	Air Force Research Laboratory
BCBF	Body Centered, Body Fixed frame
BR4BP	Bicircular Restricted Four Body Problem
CCS	Compressed Column Storage
CP	Convex Problem
CR3BP	Circular Restricted Three Body Problem
CRMFCQ	Constant Rank Mangasarian-Fromovitz Constraint Qualification
DRO	Distant Retrograde Orbit
FSW	Flight Software
GN&C	Guidance, Navigation, and Control
GSFC	Goddard Space Flight Center
I/O	Input/Output
JPL	Jet Propulsion Laboratory
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independence Constraint Qualification
MFCQ	Mangasarian-Fromovitz Constraint Qualification
NASA	National Aeronautics and Space Administration
NEA	Near Earth Asteroid
LSC-B	Lower Semi-Continuous in the sense of Berge
LSC-H	Lower Semi-Continuous in the sense of Hausdorff
NLP	Non-Linear Program
QCQP	Quadratically Constraint Quadratic Problem
SOCP	Second Order Cone Problem
SCP	Sequential Convex Programming
SQP	Sequential Quadratic Programming
STM	State Transition Matrix
USC-B	Upper Semi-Continuous in the sense of Berge
USC-H	Upper Semi-Continuous in the sense of Hausdorff

NOTATION

(By chapter in which notation is first used, or when a prior usage is replaced)

Chapter 3

μ	Mass parameter of the CR3BP
x, y, z	Position coordinates in the CR3BP rotating frame
$\dot{x}, \dot{y}, \dot{z}$	Velocity coordinates in the CR3BP rotating frame
r_1	Distance from the center of mass of the larger massive body to the spacecraft
r_2	Distance from the center of mass of the smaller massive body to the spacecraft
E	CR3BP energy
Ω	Domain
r, θ, ϕ, α	Partition coordinates
R_i	Region of the domain partition
n_i	Node of a directed graph corresponding to R_i
C	Finite covering of the domain
f	Poincaré map
G	Directed graph
$s(i)$	Set of indices of those regions intersecting the image of the flow of R_i
d	Maximum diameter of regions within a partition
ΔV	Impulsive maneuver
t	Coasting time
Φ	State Transition Matrix
$\bar{\sigma}$	Maximum Singular Value
λ, μ	Eigenvalues

Chapter 4

P	Periodic boundary conditions within the directed graph
p	Number of nodes within P
n,m,q	Nodes of the directed graph
$c(n,n')$	Cost in the directed graph from the current to successor node
$g(n)$	Path cost to from start node to node n
F	Frontier set
b	Branching factor
C^*	True optimum path cost
ε	Minimum link cost in the directed graph
V_G	Number of vertices in the directed graph G
E_G	Number of edges in the directed graph G
$h_E(n)$	Heuristic estimate of the remaining cost from the node n to the goal node
$h_T(n)$	True remaining cost from the node n to the goal node
f	Best first search selection function

Chapter 5

$J( )$	Cost function
h_j	Equality constraint
g_k	Inequality constraint
n	Dimension of the domain

m	Number of constraints
N	Number of ballistic arcs in the transfer being optimized
i	Index of the ith patch point between arcs
(x_i, v_i)	Initial state of the arc beginning at patch point i
(x^-_i, v^-_i)	Terminal state of the arc ending at patch point i
t_i	Coasting time of the arc ending at patch point i
$\varphi_t(\ )$	System flow for a given state for over time t
X	Combined state of all transfer arc initial conditions and coasting times
ε	Fuel weighting constant
R_F	Radius of the forbidden region
K_k	A second order cone
$x_i^{ref}, v_i^{ref}, t_i^{ref}$	Variable values for the reference value taken from the prior iteration
$\tilde{x}_i, \tilde{v}_i, \tilde{t}_i$	Variations from the reference values so that $x_i = x_i^{ref} + \tilde{x}_i$, etc.
$\Phi_{i+1,i}$	STM for the reference arc beginning at patch point i and ending at periapsis
$A_{i+1,i}$	Top left quadrant of $\Phi_{i+1,i}$
$B_{i+1,i}$	Top right quadrant of $\Phi_{i+1,i}$
$C_{i+1,i}$	Bottom left quadrant of $\Phi_{i+1,i}$
$D_{i+1,i}$	Bottom right quadrant of $\Phi_{i+1,i}$
Φ_i^p	STM for the reference arc beginning at patch point i and ending at periapsis
A_i^p	Top left quadrant of Φ_i^p
B_i^p	Top right quadrant of Φ_i^p
C_i^p	Bottom left quadrant of Φ_i^p
D_i^p	Bottom right quadrant of Φ_i^p
$\tilde{J}(\ )$	Cost function of the CP in terms of variation variables
$\tilde{h}_j(\ )$	Equality constraint of the CP in terms of variation variables
$\tilde{g}_k(\ )$	Inequality constraint of the CP in terms of variation variables
σ_i	Epigraph transformation variables
λ, μ	Lagrange multipliers

Chapter 6

$\tilde{X}^*$	Optimal solution of the CP subproblem
α	Step length in direction of CP solution
δv_i	Initial velocity adjustment to the arc beginning at patch point i for recorrection

Chapter 7

$\kappa(\ ,\ )$	Discontinuity function in terms of step length and velocity correction
U_i, V_i	Open sets in the Implicit Function Theorem
γ_i	Function defining velocity corrections in terms of step length
$c_i(\ )$	Increase to the uncorrected cost due to the velocity correction δv_i

Chapter 8

Ξ	Set of points in the domain satisfying the KKT conditions of the NLP
Γ	Point to set mapping
$M(\ )$	Feasible set map
$S(\ )$	Solution set map

ACKNOWLEDGMENTS

My sincerest gratitude to my advisors, Dr. Kenneth D. Mease and Dr. Benjamin Villac. To Benjamin for eagerly taking on a new student with no engineering background, for encouraging exploration, for his good humor, and for truly being a fount of knowledge in spaceflight. To Professor Mease for taking me on as a student on short notice and jumping in to a research project years in development, bringing new perspectives, and encouraging leadership. I am honored to be their student.

At UC Irvine, thank you to my committee members for their time and feedback with this research. Thanks as well to the members of the Mechanical and Aerospace Engineering and the Computer Science departments for their classes that aided and inspired many part of this research. Thank you to Louise Yeager for years of always having exactly the answer I needed before I knew I needed it.

At Jet Propulsion Laboratory, special thanks to my NSTRF mentor and Autonomous Systems Deputy Division Manager MiMi Aung. Being at JPL was the goal of going to graduate school, and MiMi opened every door there for me. Her efforts ensured that rather than just studying spaceflight in the abstract that I would be able to meet with mission designers and flight software engineers and see projects development and missions as they happened. Special thanks as well to to Steve Broschart for his years of input, inspiration, and support, effectively taking on yet another NSTRF student. Thank you to Lloyd Manglapus and Kim Gostelow of the Flight Software Section (349), for their patience and to whom all credit is given for the transition of this project from yet another trajectory design algorithm to a program implemented on the RAD750 flight processor. Thank you to the Guidance, Navigation, and Control (344) and Mission Design and Navigation (392) sections for hosting me these past few years, in particular to Jordi Casoliva, Roby Wilson, Rodney Anderson, Jules Lee, Andrew Johnson, and Nikolas Trawny for sharing their time, insight, or cubicle. Thanks as well to Fernando Peralta for a view into the Voyager mission, and to Lois Berumen for joyfully bringing order to chaos.

At Goddard Space Flight Center, thank you to Dave Folta and Brent Barbee of the Navigation and Mission Design branch for their invitation to GSFC and their enthusiastic introduction to a very interesting application.

Thank you to the NASA Office of the Chief Technologist and everyone on the NASA Space Technology Research Fellowship team for their amazing program that provided the generous support for this research along with the opportunity to spend each Summer at a NASA facility (grant NNX11AM78H). Thanks to Dr. Michael Gazarik, former NASA Director of Space Technology, for his role in initiating this program and selecting members of its inaugural class. It truly has been an extraordinary experience.

To Dr. Navid Nakhjiri and Dr. Channing Chow, for brotherhood.

CURRICULUM VITAE

Eric Trumbauer

EDUCATION

University of California, Irvine. Irvine, CA

Ph.D., Mechanical and Aerospace Engineering **March 2015**

- Advisors: Kenneth D. Mease (1/2014 – present), Benjamin Villac (12/2010-1/2014)
- Thesis: Rapid Onboard Trajectory Design for Autonomous Spacecraft in Multibody Systems

M.S., Mechanical and Aerospace Engineering **June 2012**

- Advisor: Benjamin Villac

University of California, Berkeley. Berkeley, CA

B.A., Mathematics **May 2004**

HONORS AND AWARDS

NASA Space Technology Research Fellowship **August 2011 - Present**

Honors, Department of Mathematics **May 2004**

Graduated with Honors in General Scholarship **May 2004**

ACADEMIC AND PROFESSIONAL EXPERIENCE

Jet Propulsion Laboratory. Pasadena, CA

NASA Space Technology Research Fellow **July 2012- August 2014**
Autonomous trajectory redesign research, Autonomous Systems Division 34 and Mission Design and Navigation Section 392 (Mentor: MiMi Aung).

University of California, Irvine. Irvine, CA

NASA Space Technology Research Fellow **August 2011-Present**
Project research, Grant NNX11AM78H: Automated Trajectory Design Using Resonant Dynamics. UCI Flight Dynamics and Control Lab (Advisors: Benjamin Villac, Kenneth Mease).

Bank of America. Tustin, CA

Business Systems Analyst, LaSalle Global Trust Services **2007-2009**
Expand, modify, and maintain the RMBS external reporting system with a focus on the calculation engine. Requirements, testing, create and modify SQL stored procedures, user acceptance testing.

Pacific Life. Foothill Ranch, CA.

Systems Analyst, Life IT – Product Admin **2005-2007**
Business, technical, and testing analysis at all stages of the Software Development Life Cycle.

PAPERS

Trumbauer, E., and Villac, B., "Heuristic Search Based Framework for Onboard Trajectory Redesign," AIAA Journal of Guidance, Control, and Dynamics. Vol 37.1. January 2014. http://arc.aiaa.org/doi/abs/10.2514/1.61236

Trumbauer, E., Villac, B., Mease, K.D., "In-Flight Trajectory Redesign Using Sequential Convex Programming," 2014 SIAM Conference on Optimization, San Diego, CA, May 2014. [Poster Session]

Trumbauer, E., and Villac, B., "Autonomous Trajectory Redesign for Phobos Orbital Operations," 2014 AAS/AIAA Spaceflight Mechanics Conference, Santa Fe, NM, January 2014.

Trumbauer, E., and Villac, B., "Sequential Convex Programming for Impulsive Transfer Optimization in Multibody Systems," 2013 AAS/AIAA Spaceflight Mechanics Conference, Kauai, HI, February 2013.

Trumbauer, E., and Villac, B., "Search and Representation Strategies for Automated Trajectory Design," 2012 AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, MN, August 2012.

Trumbauer, E., and Villac, B., "Expanding Transfer Representations in Symbolic Dynamics for Automated Trajectory Design," 2012 AAS/AIAA Spaceflight Mechanics Conference, Charleston, SC, January 2012.

Trumbauer E., Villac B., "An Analysis of Multiple Revolution Third Body Driven Plane Change Maneuvers", 2011 AAS/AIAA Astrodynamics Specialist Conference, Girdwood, AK, August 1-4, 2011.

ABSTRACT OF THE DISSERTATION

Rapid Onboard Trajectory Design for Autonomous Spacecraft in Multibody Systems

by

Eric Trumbauer

Doctor of Philosophy in Mechanical and Aerospace Engineering

University of California, Irvine, 2015

Professor Kenneth D. Mease, Chair

This research develops automated, on-board trajectory planning algorithms in order to support current and new mission concepts. These include orbiter missions to Phobos or Deimos, Outer Planet Moon orbiters, and robotic and crewed missions to small bodies. The challenges stem from the limited on-board computing resources which restrict full trajectory optimization with guaranteed convergence in complex dynamical environments. The approach taken consists of leveraging pre-mission computations to create a large database of pre-computed orbits and arcs. Such a database is used to generate a discrete representation of the dynamics in the form of a directed graph, which acts to index these arcs. This allows the use of graph search algorithms on-board in order to provide good approximate solutions to the path planning problem. Coupled with robust differential correction and optimization techniques, this enables the determination of an efficient path between any boundary conditions with very little time and computing effort. Furthermore, the optimization methods developed here based on sequential convex programming are shown to have provable convergence properties, as well as generating feasible major iterates in case of a system interrupt – a key requirement for on-board application. The outcome of this project is thus the development of an algorithmic framework which allows the deployment of this approach in a variety of specific mission contexts. Test cases related to missions of interest to

NASA and JPL such as a Phobos orbiter and a Near Earth Asteroid interceptor are demonstrated, including the results of an implementation on the RAD750 flight processor. This method fills a gap in the toolbox being developed to create fully autonomous space exploration systems.

Chapter 1

Introduction

1.1 Motivation

Spacecraft autonomy has been recognized as a key technology needed for further progress in space exploration. Within this broad area, automated guidance and trajectory design plays an important role. NASA's Space Technology Roadmap section on *Robotics, Tele-Robotics and Autonomous Systems: Autonomy* highlights this when noting that "onboard maneuver planning and execution monitoring will increase the vehicle agility, enabling new mission capabilities and reducing costs."[NRC12] This ability will impact a wide variety of mission types, including manned deep space exploration, robotic science missions, and planetary landers.[Cang12] In a Jet Propulsion Laboratory survey on technologies needed for the next generation of small body and planetary missions, it was noted that in cases of extremely perturbed gravitational environments that course adjustments may be required on the timescale of minutes, [Ried09] which even for nearby Mars rules out on-the-ground intervention due to the round trip communication times.

Even with recent advances in autonomous guidance, navigation, and control with systems like AutoNAV and AutoGNC, [RiedAN] mission risks in these settings due to maneuver execution errors, missed maneuvers, temporary loss of power, divergence due to system safing, change of goals, or other off-nominal situations will require a level of replanning beyond standard control methods. This research develops a method to enable such automated maneuver planning, in particular the ability to perform rapid onboard trajectory redesigns in dynamically complex environments.

Within these mission types, several specific targets exist which would benefit from autonomous trajectory design capabilities. For example, planetary moons of interest such as Phobos, Deimos, Europa, and Enceladus have highly perturbed dynamics and short instability timescales, creating a need for fast action in off nominal conditions.[Wall12, Sche01] In these cases, the features of multibody dynamics provide both opportunities for efficient transfers as well as risks of impact and escape. Irregularly shaped bodies such as Golevka and Itokawa represent another class of examples, combining irregular gravity fields, short timescales, and a strong perturbing influence of the sun which cannot be ignored.[Vill08, Mond10] The challenges these systems pose to unmanned autonomous spacecraft are clear. However, even in the NASA Technology Roadmap section *Human Exploration Destination Systems: Mission Operations and Safety*, keys to success include "on-board autonomous systems" and "on-board intelligent software for situational awareness" to face communication delays or disruptions.[NRC12]

In particular these systems show there is a need for a rapid trajectory design system which directly takes the structure of the complex multibody gravity environment into account, yet does not use the full machinery of ground based methods. Advancing the Roadmap area of *Modeling, Simulation, Information Technology and Processing,* such a system must not only include both initial guess generation and optimization in an integrated platform, it must do it in less time and

with fewer resources than on-the-ground trajectory design tool. Existing onboard computing systems such as the RAD750 are very limited in both processing power and available memory, which combined with additional requirements of onboard systems adds strong limitations to the types of methods that may be considered. As a result, the integration of the trajectory redesign software with the flight computing architecture is a necessary component for implementing an actual system, posing a challenge for algorithm development.

1.2 Problem Statement

1.2.1 Problem Definition

This research focuses on providing a solution to the following scenario in systems with complex dynamical environments: an off-nominal situation such as those outlined above exists such that existing closed-loop guidance techniques are unable to provide a feasible transfer to a given goal.

The problem to be solved then is to rapidly provide a new transfer from the current state to the desired goal. Furthermore, the transfer design system is constrained by the limitations of flight hardware including low processing power and available memory, and the ability to tolerate system interrupts during long computations. Also, as to be expected of onboard computations, methods with guaranteed convergence results are needed.

1.2.2 Assumptions

Although true onboard implementation will require system integration effort with command, navigation, attitude control, and propulsion systems, this dissertation assumes that the necessary system information has been made available to the transfer design system as is shown in Figure 1.1. Thus, that a specific target has been identified, thrust limitations are known, and that an accurate model and orbit determination have been provided at execution. Due to the long

coasting arcs between maneuvers, it is also assumed that the attitude control system has the control authority to create any necessary orientation needed for the (possibly directionally constrained) maneuver. Lastly, as is made clear in the title, an impulsive model of chemical propulsion is used.

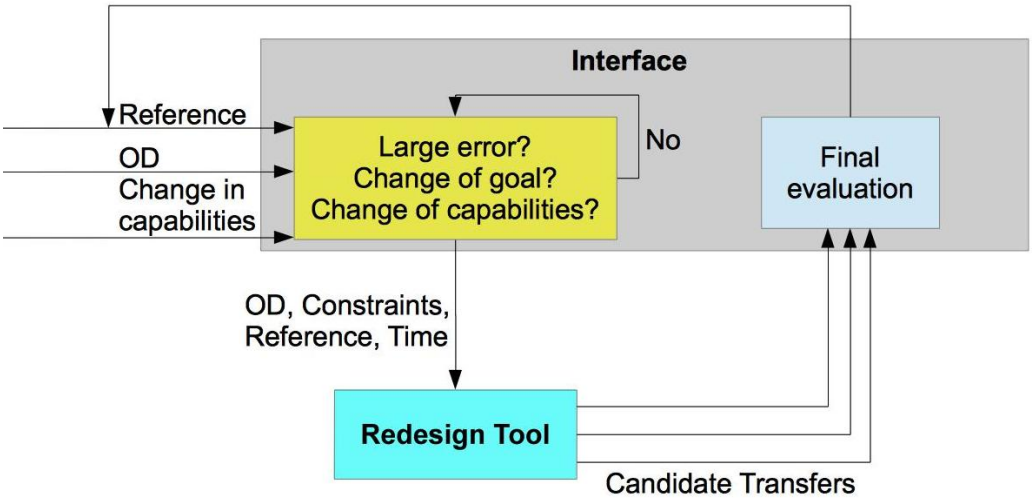


Figure 1.1: Relation of the transfer redesign tool with a satellite system interface. Errors, changes of goals, or changes in capabilities in other satellite systems signal the need for transfer redesign.

1.3 Overview of the Proposed Solution Method

The final continuous transfers generated by this method are created by linking precomputed arcs together using differential correction and then performing a rapid local optimization. The process of initial guess generation revolves around where these arcs come from, and how to *describe, store, select, order, and assemble* them. The method can be divided into two primary sections: the offline / mission planning phases, and the onboard phases. The mission planning steps are relatively expensive computationally and involve creating, cataloging, and efficiently storing the transfer arcs and boundary conditions, as well as describing the relationship between the discretized dynamics and fuel estimates.

On the other hand, the onboard process quickly leverages the work done on the ground and stored in memory to quickly generate a transfer using search, differential correction, and optimization methods. The onboard and offline portions are complementary images of each other – the pre-mission stages translate and compress the continuous system dynamics into a discrete representation, while the onboard portion searches and selects discrete information and reconstructs a continuous trajectory. Due to the nature of the onboard application, the local optimization process is quite specialized as well. This includes requirements that major iterations be feasible due to system interrupts despite nonlinear and dynamic constraints, as well as convergence properties of the approximate subproblems as well as the iterative process as a whole.

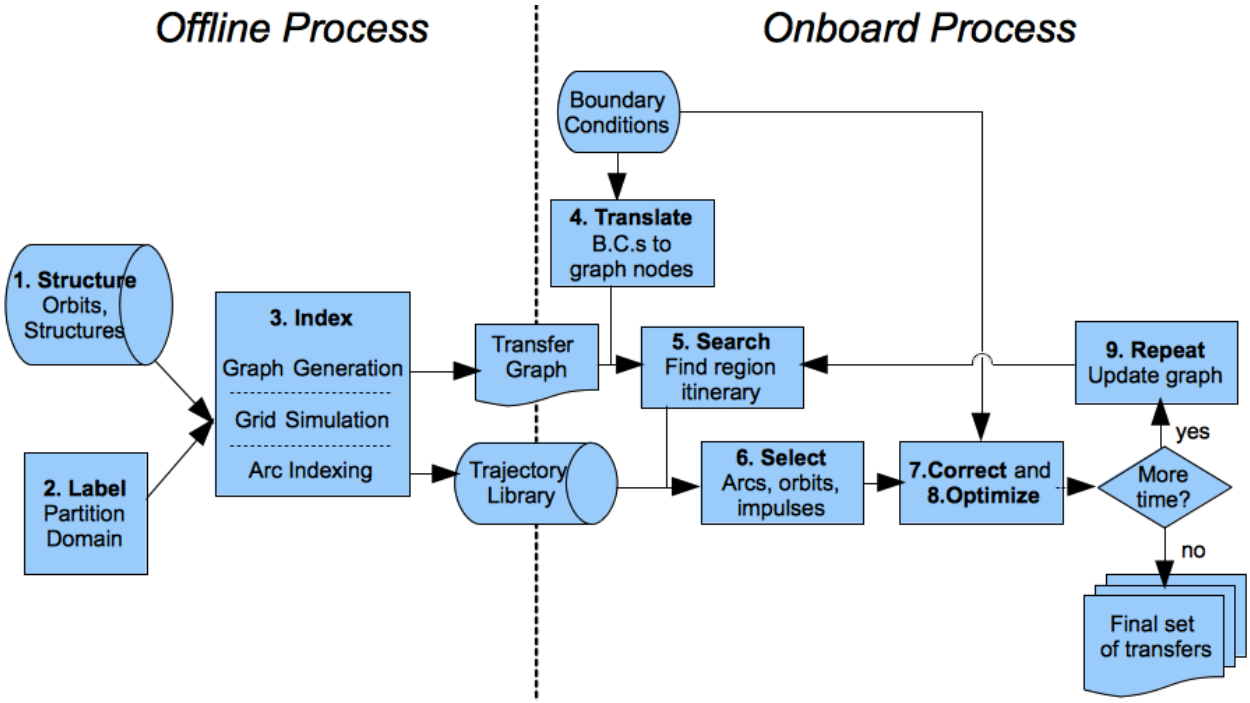


Figure 1.2: Flow diagram of the steps both on the ground and onboard for the transfer design algorithm.

Before describing each component individually, laying out the overall sequence and providing keywords for each step of the proposed method is helpful to place each element in

context. Figure 1.2 illustrates the connections of the phases listed here.

The Offline / Mission Planning Process

1. STRUCTURE: Combine database of periodic orbits (the key boundary conditions), invariant manifolds, and other efficient transfers that brute force techniques would miss without using a computationally prohibitive resolution.

2. LABEL: Decide on a physically meaningful partition of the domain.

3. INDEX: Create a symbolic/graph representation of dynamics, adding transitions if a linking trajectory is found among or simulations. This approximates the continuous system in a format that discrete search algorithms can use.

The Online Process

4. TRANSLATE the boundary conditions into start and end nodes in the search graph.

5. SEARCH for the optimal path within the graph, creating an itinerary of regions.

6. SELECT appropriate arcs for each region to region transition.

7. CORRECT the arcs into continuous path.

8. OPTIMIZE the corrected transfer to reduce fuel cost.

9. REPEAT steps 5-8 to create more options for evaluation.

1.3.1 Step 1 - Offline: Catalogue Relevant Dynamical Features

The process begins by considering what types of trajectories to explicitly include in a multibody trajectory design program. Their main application is in providing a set of high priority arcs for selection as transitions between regions and as boundary conditions. First and foremost, periodic and quasi-periodic orbits serve as likely boundary conditions for the types of transfer problems of interest. Thus periodic orbits must be available as options when specifying initial and final conditions. Other important structural elements such as libration points, invariant manifolds, and collision trajectories play a role shaping the system dynamics. By explicitly including these

trajectories into the system, the goal is to avoid the combinatorial problems that arise from trying to capture these features solely through cell mapping or mass integrations. These elements are then supplemented by a reasonably sized grid based integration to capture a variety of simpler transitions.

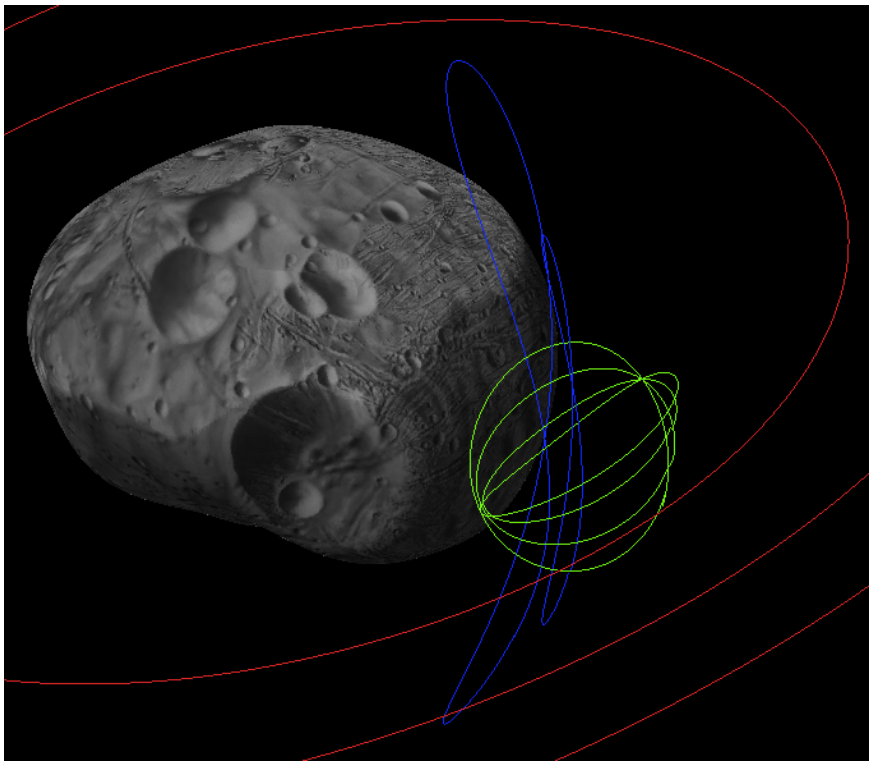


Figure 1.3: Example Halo (green), Vertical Lyapunov (blue) and Distant Retrograde Orbits (red) in the Mars/Phobos system perturbed under the influence of the highly irregular gravity of Phobos.

1.3.2 Step 2 - Offline: Partition the Domain

This system depends on building transfers using sequences of trajectories – some but not all consisting of elements of periodic orbits, manifolds, and other structures just mentioned. The boundary conditions and transfer arcs must now be represented in such a way as to utilize the speed of discrete search algorithms. This is a necessary step to ensure the right orbits and arcs are selected and ordered in the proper sequence to solve the transfer design problem. Given that our approach is to differentially correct and optimize a set of arcs into a continuous transfer, we

implicitly require that one arc begins near where another ends. As some sort of discretization of the dynamics is needed, this naturally leads to the concept of partitioning the domain into relatively compact regions, and a condition to discretize time. Depending on the application, this may be done using hypercubes defined using an appropriate coordinate system whether Cartesian, orbital elements, or as is done here:, energy, spherical coordinates for position, and flight path angle. See Figure 1.5 for an example of the projection on to position space of a portion of such regions.

1.3.3 Step 3 – Offline: Create a Discrete System Representation

With a partition chosen for the domain, the next step is to classify trajectories in relation to the discretization – namely within which regions of the surfaces of section do each trajectory arc start and end. Using a directed graph is an ideal way to represent which transitions are possible, and provides the structure on which discrete search algorithms are run. A graph that represents and approximates a continuous dynamical system is referred to as the *symbolic image* of the system.

In brief, the construction process begins by creating a node/vertex for every region in the partitioned domain. A set of ballistic arcs is assembled through integrating a large grid of initial conditions *and* combining these with the arcs from the database of structural elements. If an arc or an impulse exists which begins in Region R_i and ends in Region R_j, a directed edge from Node n_i to Node n_j is added. The arcs or impulses associated with each transition are then labelled with their starting and ending region or node. Such a graph indicates whether or not the underlying system has a natural transition between two specified regions. This is shown conceptually in Figure 1.4. In reality these graphs have tens to hundreds of thousands of nodes, but are quite sparse resulting in reasonable search times and memory usage

The relationship between graph transitions and the original problem can now be defined. Suppose a path is found in the graph which leads from the node associated with an initial region to

the node associated with a target region. *This corresponds to a sequence of regions, each connected to its successor via ballistic arcs or an impulse, linking the regions containing the boundary conditions.* As a result, the directed graph and arcs/impulses associated with transitions within that graph provide the foundation for the online processes and need to be stored onboard. This includes: (1) The directed graph, which provides a meaningful representation of the system that can be used by a discrete search process. (2) For ballistic transitions, one or more of the arcs that led to the creation of a directed edge in the graph must be stored and labelled as transitioning between the regions associated with the graph nodes. These elements allow the spacecraft to create the skeleton of a transfer quickly.

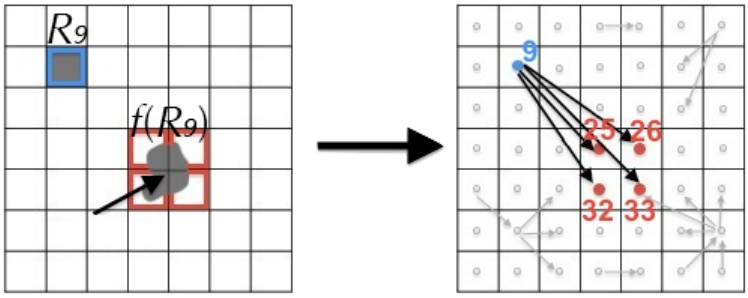


Figure 1.4: The intersection of the image of the system flow of one region with others leads to the creation of directed links in the graph representation.

1.3.4 Step 4 – Onboard: State Boundary Conditions as Graph Nodes

The previous steps create a discrete/symbolic representation of the dynamics and impulses. Boundary conditions, however, are unlikely to be provided as simply a region in a computer discretization and thus simply a node on a graph. Thus in order to provide start and goal nodes for the search process, it is necessary to translate a variety of types of boundary conditions into symbols. As a result of this process, states are translated into single nodes and periodic orbits are translated into a periodic sequence of nodes within the graph. One or more search problems are then defined, one for each possible pair of initial and target nodes.

1.3.5 Step 5 – Onboard: Search for the Transfer Itinerary

By this step, a weighted directed graph is available which approximates the problem structure, and boundary conditions have been translated into nodes on this graph. Together, a search problem is defined. While several different search algorithms have been considered, the selected A* heuristic search algorithm is used to find the optimal path *within the graph* for each run of the search. It possesses provable completeness and optimality properties, and has been observed to be sufficiently fast for the application tested using simple estimates (heuristic function) of the remaining cost based on the graph structure. The sequence of transitions in the resulting path tells us which types of arcs and impulses are needed in what order to build the desired transfer.

1.3.6. Step 6 – Onboard: Select Arcs to Create an Initial Transfer Guess

As stated previously, a transition in the graph exists only if there is a stored coasting arc or impulsive maneuver between the regions represented by the vertices. Thus for every transition in the path found by the search process, we have at least one arc or impulse which links the corresponding regions. Joining this sequence of arcs together provides a good approximation of a transfer linking the two boundary conditions, with infeasibilities such as discontinuities limited by the partition resolution and cost estimated by the link costs. An example of such an initial guess in Figure 1.5, where for visual clarity regions with radii beyond those needed for this visualization are not shown.

1.3.7 Step 7 - Onboard: Differential Correction to Establish Feasibility

The optimization is required to maintain feasible major iterates in case of a system interrupt, and as a result a process must take place where the initial guess is differentially corrected into a continuous, feasible, but non-optimal transfer. Due to constraint based graph pruning, some of these constraints are already satisfied. Others however must be dealt with and so a penalty based approach using sequential Second Order Cone Problems is used. Such a process is used to

transform the discontinuous arcs from the initial guess in Figure 1.5 into the feasible but un-optimized transfer in Figure 1.6.

1.3.8 Step 8 - Onboard: Locally Optimize Transfer ΔV

Once a feasible guess has been created, it remains to rapidly optimize the transfer until the fuel usage is within acceptable levels. The optimization process developed here synthesizes elements of Sequential Convex Programming [Boyd08, Caso13] and the Two Level Differential Corrector [Marc07] in order to meet the requirements of the application. It is proven that the additional structure provided by using solutions of convex sub-problems in addition to a recorrection process is sufficient to guarantee that successive iterates both remain feasible and have lower cost. This is essential for an onboard process where the possibility of a system interrupt do not allow for methods where only the final converged solution is guaranteed to be feasible. In addition to this benefit, the wider set of allowable costs and constraints mean that convex approximations within the iterative process are able to more accurately represent the impulsive transfer problem, while themselves still able to be solved rapidly and within a calculable finite number of iterations using interior point methods.

1.3.9 Step 9 – Onboard Create Additional Transfers with Graph Feedback

After the correction and optimization procedures are complete, the costs may be higher than expected or a mission constraint may not be able to be satisfied with the search itinerary. If this is the case – or if there is simply time permitting to evaluate other potential options – it may be beneficial to create other transfer types. After the correction and optimization procedure is complete, the directed edge for which the true cost is the highest may either have its edge weight adjusted, or the edge temporarily removed. Once this adjustment for costly transitions is made, the search process can be rerun. Unless the path is still graph-optimal with a weight adjustment, this will result in a new itinerary and a qualitatively different transfer.

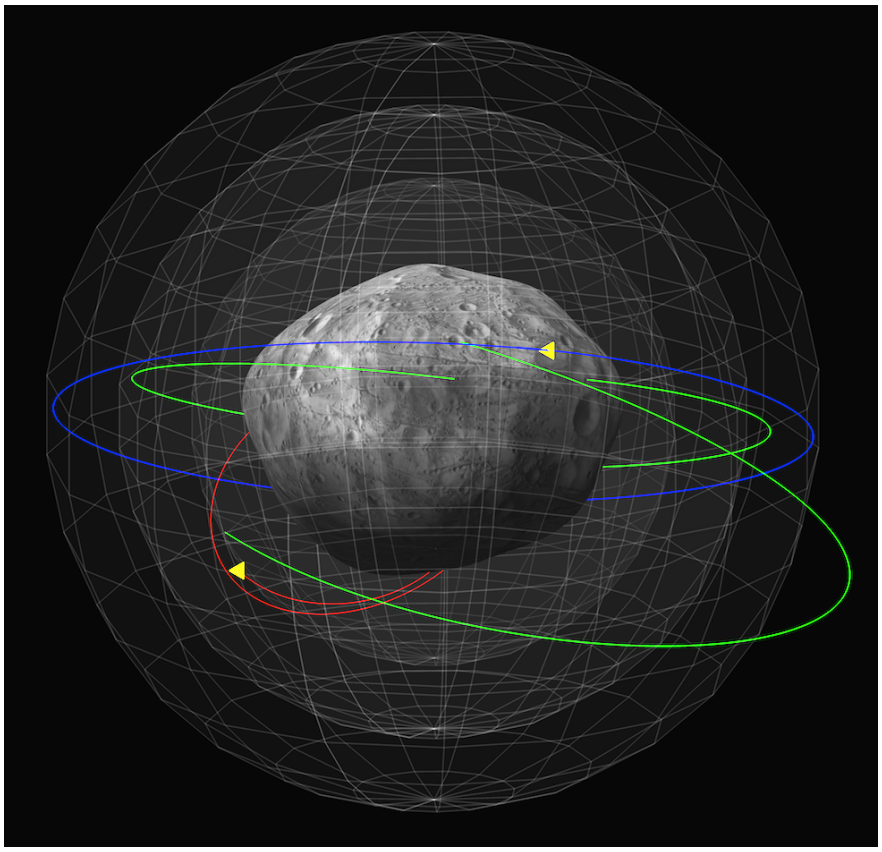


Figure 1.5: A sequence of regions is selected by the search process, and ballistic arcs (green) and impulses are found linking these regions, approximating a transfer between the initial condition (red, divergence from Halo) to a stable abort orbit (blue)

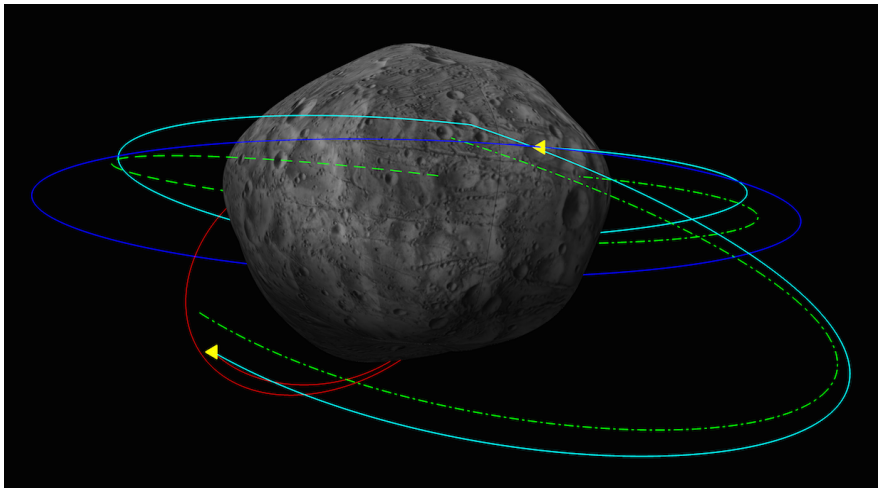


Figure 1.6: The discontinuous arcs (dashed green) are corrected into a continuous, if inefficient, transfer (teal) between the two boundary conditions.

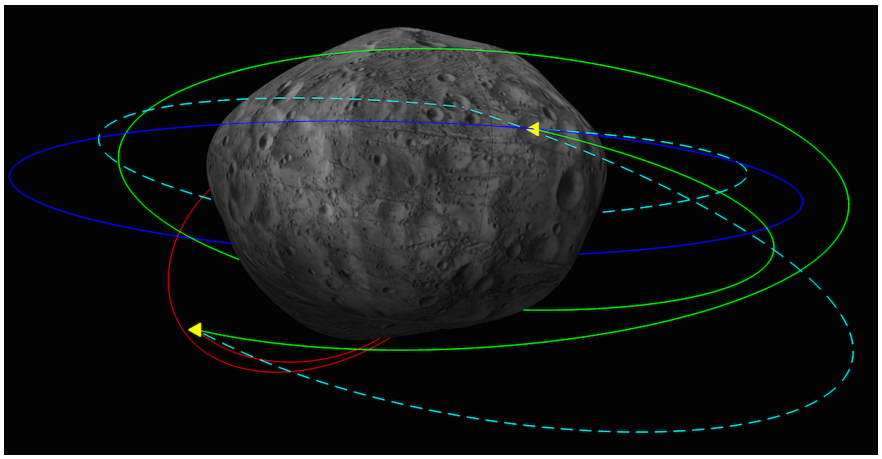


Figure 1.7: The feasible initial transfer (dashed teal) is rapidly optimized, reducing fuel while satisfying constraints for the final transfer (green).

1.4 Contributions

The research contributions vary by the stages of the transfer design process. The first stage is the use of directed graph representations of the impulsive transfer problem. Although using the solution of a discrete approximation as an initial guess for a continuous domain is not a new concept, it required several insights for implementation in this setting. For example, the domain partitioning method used has many essential properties. The resulting graph structure of energy layers is used to create a simple but very effective heuristic function to increase search speed by an order of magnitude or more. The number of dimensions is reduced, leading to smaller graphs, and the velocity range definitions lead to simple calculations of impulse cost estimates. Another contribution during the graph construction is the combination of precomputed periodic orbits and other structural elements along with a mass integration to model the system flow capture both the precise and mundane aspects of the dynamics without requiring a prohibitively fine division of the domain.

During the onboard process, contributions include an analysis of the directed graph structure that is used to generate an effective and admissible heuristic via the process of rule

relaxation. The results of the path finding algorithm are then related back to elements saved during the discretization process in order to assemble an initial guess. Major contributions in this research involve the synthesis of elements of existing impulsive transfer optimization methods and a full analysis of the feasibility and convergence properties of the resulting iterative method. There are several significant proofs on this theme. The first is that this iterative approach generates feasible major iterates for both nonlinear equality and inequality constraints well as being a global descent method, as required in case of system interrupts. While the persistence of Slater's condition for the convex subproblems throughout an iterative process is assumed by many in the field, due to its importance in proving this result – as well as for the efficacy of interior point solvers – an entire appendix is dedicated to showing this fact for this application. Also, while existing differential correction procedures assume the invertibility of certain State Transition Matrix submatrices, this assumption is not valid in general and modifications are discussed in case this submatrix is ill-conditioned or even truly singular. Next it is shown that the conditions of Zangwill's Convergence Theorem are satisfied so that the optimization procedure is not just a descent method but will converge to points satisfying the Karush-Kuhn-Tucker conditions of the original problem. Neither the minimum fuel problem nor its convex approximation are strictly convex, and so it cannot be assumed that each subproblem's solution set consists of a single point. As a result, rather than eliding over this fact, this convergence result is shown using the apparatus of point-to-set mappings.

There are also contributions within the application of this method to two very different cases: onboard orbital operations around Phobos in the multibody and highly irregular dynamics of that moon, and a Near Earth Asteroid interceptor concept initial investigation. The case of Phobos demonstrates the ability to design transfers quickly in a complex environment. For the NEA application, the importance is in showing how the system developed as an onboard replanner

with specified target states can be significantly but quickly modified to design a large set of flyby transfers involving both “low energy” libration point orbit dynamics and high energy target bodies.

Lastly, aside from developing an integrated transfer design process around several of the demands of onboard application in general, an additional contribution is made in describing the algorithmic adaptations made to run this method on a current RAD750 based flight software platform.

1.5 Outline of the Dissertation

The dissertation is divided into several sections based on major themes of the trajectory design process. After this introduction follows Section II: Initial Guesses Generation for Impulsive Transfers in Multibody Systems. Within it, Chapter 2 provides background on existing methods both classical and more recent that have played a role in mission design, as well as describing their limitations for use in an autonomous spacecraft trajectory design tool. Chapter 3 lays the foundation for the discretized system approximation that is developed in this research for use by rapid heuristic search algorithms. Thus it describes the on-the-ground work done to partition the state space, construction of a weighted directed graph to represent the system dynamics, and a database of ballistic arcs indexed by the graph transitions. Chapter 4 defines the initial guess generation process where spacecraft boundary conditions are translated into graph nodes, heuristic search algorithms quickly find a path within the graph, and how this itinerary is used to assemble the coasting arcs and impulses that constitute the initial approximation of the desired transfer.

Section III: Local Optimization of Impulsive Transfers describes the process by which these initial guesses are made into feasible, fuel efficient transfers. Before describing the types of iterative processes that are required for such complex systems, Chapter 5 described convex optimization problems and important subcategories subcategories including Second Order Cone Problems.

Progress in solution algorithms for these problems has lead to methods with provable convergence, a deterministic limit on iterations, and short calculation times. Additionally, the wider set of allowable constraints than quadratic problems will be shown to more naturally capture the cost and constraints of the minimum fuel problem, but play a key theoretical role. The impulsive transfer NLP is stated and after a sequence of steps it is shown how convex approximations may be derived for use in an iterative solution method. Chapter 6 describes how the existing approaches of Sequential Convex Programming and the Two Level Differential Corrector used in the spaceflight community may be synthesized into a single iterative solution method. Chapter 7 justifies the assumptions made in the iterative process design and proves that not only is the process a descent method, but that when starting from a feasible guess that the major iterates of the optimization process will themselves be feasible. This is a key requirement in case of a system interrupt during optimization. Chapter 8 extends this result and shows that not only is the procedure a descent method but that it satisfies Zangwill's Convergence Theorem. After the problem constraints have been discussed throughout this section, Chapter 9 describes a related process by which the initial transfer approximation provided by the heuristic search process is transformed into an initial guess for the optimization process that is feasible for the given set of constraints.

Section IV: Applications discusses two quite different uses for the transfer design method developed in the prior sections. Chapter 10 focuses on the role such a tool may play in Phobos orbital operations. After describing the mission scenarios and needs relative to a Jet Propulsion Laboratory operations plan, the chapter shows how the general method is applied to the Mars/Phobos system and presents results that demonstrate the ability to quickly generate efficient transfers in this sensitive gravitational environment. Chapter 11 describes the transition of the Phobos application from standard development platforms to implementation on a RAD750 flight hardware testbed so that the algorithm can be tested in a more onboard-like environment. The

chapter focuses less on the porting process and more on the ways in which the goal of onboard implementation has shaped the algorithm design from the beginning, and those challenges which necessitated major changes in order to run within the flight software environment. Chapter 12 turns to a different application, that of generating both flyby and rendezvous transfers in the Sun/Earth/Moon system between libration point orbits to different Near-Earth Asteroids as part of an initial feasibility investigation at Goddard Space Flight Center.

After the applications follows the conclusion along with appendices describing constraint qualifications needed for interior point solvers, additional constraints omitted from the main text, and dynamical models used throughout the dissertation.

Chapter 2

Standard Methods for Initial Guess Generation

2.1 Introduction

Henri Poincaré proved a central fact of transfer design for satellites under the influence of more than a single massive body: that there is only a single integral of motion, and so no general analytical solution method exists. When forces due to additional massive bodies, irregular gravity fields, solar radiation pressure, atmospheric drag, finite thrust, and others are included, hopes of finding an efficient transfer in such a model in a single step are abandoned. As a result, transfers are designed in relatively simple models and then differentially corrected using shooting type methods or otherwise locally optimized in the new dynamics using iterative numerical methods.

The choice of model and design strategy has evolved both with time and priorities. Earlier methods such as patched conics switch between two body models and rely on high velocities for a good approximation. Three-body and patched three-body "low energy" transfers came later and use the dynamical structure of low relative velocity motion to find very efficient transfers.

However, unlike in the two-body problem, periodic orbits only sparsely populate the domain, as do the invariant manifolds that are a key part of these efficient design strategies. This may be acceptable for mission design, but it is not in off-nominal situations where a spacecraft may no longer be able to directly utilize these dynamical structures.

Chapters 3 and 4 will discuss an approach to take advantage of these features when possible, but be able to generate an initial guess without them if need be. This chapter will summarize some of the existing methods, highlighting those elements that may still be used in an onboard redesign scenario in a complex gravitational environment, and those shortcomings that prevent their use in the application. It begins, however, by introducing the impulsive thrust model for satellites using chemical propulsion.

2.2 The Impulsive Assumption

For chemical propulsion systems that produce large amounts of thrust over short periods, it is commonly assumed for simplicity that during the duration of the maneuver, the change to the state due to the maneuver is much greater than that due to other forces over the short time period. In such cases, maneuvers may be modeled as instantaneous changes to the spacecraft velocity, referred to as a ΔV. An impulsive transfer is then an alternating sequence of ΔVs and unpowered flight, moving the spacecraft from one boundary condition to another while satisfying the system constraints. This is shown conceptually in Figure 2.1. This is in contrast with "low thrust" propulsion systems such as ionic propulsion or solar sails, which require very different, optimal control based solution methods.

The impulsive assumption not only simplifies modeling the dynamics, it also allows us to convert the amount of fuel used into a function of the ΔVs themselves. Assume a spacecraft has an initial mass at the time of the transfer m_0. Minimizing the fuel used is equivalent to maximizing the

final mass m_f. In order to get to the desired form of the Rocket Equation, this is first transformed to the equivalent form of minimizing m_0/m_f, and since the log function is monotonic, this then is equivalent to minimizing $\log(m_0/m_f)$. Under this impulsive assumption, this is related to the sum of the ΔV magnitudes by the Rocket Equation: [Curt13]

$$\log \frac{m_o}{m_f} = \frac{1}{c_e} \sum_i \|\Delta V_i\|$$

where c_e is the effective exhaust velocity of the engine. Since c_e is assumed to be constant, a minimum fuel impulsive transfer problem is seen to be equivalent to minimizing the cost function $J = \sum \|\Delta V\|$ for a feasible transfer. This problem is precisely stated as a NLP in Chapter 5.5, but for now it is these general concepts that are needed to describe the structure and goals of the transfer design methods that follow. Even in cases where there is a different objective, the total $\sum \|\Delta V\|$ gives a spacecraft independent measure of the fuel used by an impulsive transfer.

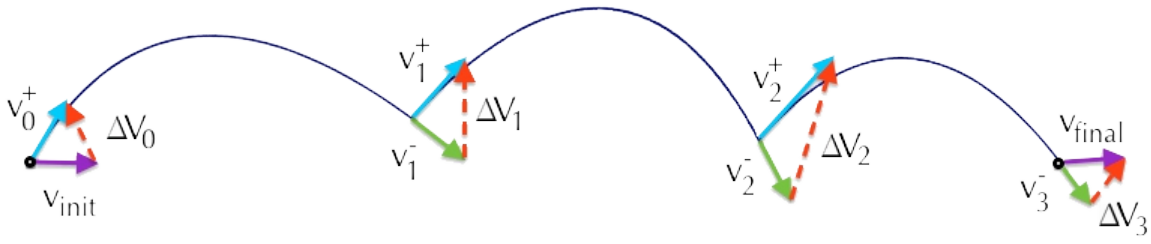


Figure 2.1: Diagram of an Impulsive Transfer

2.3 Two-Body Model Based Methods

Classical orbital transfers and perturbation theory are based on the essentially Keplerian trajectory structure that arises in regions where the gravity of a single body dominates other forces present. When these spheres of influence are large compared to the body radius and relative velocities are high, patching two body model based trajectories together create good initial

guesses. As we will see these assumptions are not valid for many scenarios of interest.

2.3.1 Transfers in the Restricted Two-Body Problem

The restricted two-body problem assumes a massless spacecraft moves under the influence of a single massive body treated as a point mass. Since the time of Kepler it has been known that the resulting motion is defined by the periodic motion of ellipses for non-escaping trajectories, and later hyperbolae for most escaping trajectories, and parabolae at the transition between the two. Under this assumption there are five integrals of motion that define each periodic orbit, known as orbital elements. In classical maneuver design, transfers usually are expressed as transitions between orbits rather than between states since if time is free then conditions on the orbits will naturally coast to any boundary states of interest. Such transfers include the Hohmann transfer between concentric circular orbits, escape maneuvers, in-plane rotations, single maneuvers and bi-elliptic plane changes, among others. [Vall01, Curt13] Depending on the magnitude of the quantity to be desired, it is also known which variant – such as single maneuver vs. bi-elliptic plane changes – is the most efficient. On the other hand, should a fixed time transfer between two states be needed, Lambert's Method provides transfer orbits linking the positions of the boundary states, from which the most efficient may be chosen for the given velocities. [Curt13]

2.3.2 Multibody Transfers Using Patched Conics

Patched conics describe a transfer design approach where the domain is divided into *spheres of influence* around each massive body inside which only the gravity of that body is considered. Within each portion of the domain, the well established two-body problem transfers may be used and there are standard processes to patch the two. [Curt13] For example, consider a transfer from a low Earth orbit to a similar orbit at the Moon. The resulting transfer switches from an ellipse in the Earth centered two-body problem to a hyperbola in the Moon centered two-body problem (since the spacecraft is below the escape velocity of the Earth but above that of the Moon)

at the boundary of the sphere of influence, with the hyperbola intersecting the desired final orbit so that an insertion maneuver may be made. This initial guess is then corrected to include the more complete gravity model and additional perturbations such as Solar Radiation Pressure. [Wils98]

2.3.3 Limitations for this Application

The radius of the sphere of influence of of a smaller body orbiting a larger body – the region within which a two-body approximation is valid – is given by $a(m/M)^{2/5}$ where here *a* is the semi-major axis of the orbit, *m* is the small of the smaller body and *M* is the mass of the larger body. [Curt13] Note that for many large bodies in the solar system this value is quite large and so orbiters of these bodies can use two-body problem approximations as good initial guesses for more robust models. For example, using the above formula shows that the radius of the sphere of influence of the Moon is 38 times larger than the average radius of the Moon, 45 times for Mercury, and 170 times for Mars. For orbiters in such cases, the gravity of other bodies and additional forces may be treated as perturbations using the Lagrange Planetary Equations of the Gauss Variational Equations. [Vall01] On the other hand, for targets of future interest in the outer solar system, we have spheres of influence 6.23 times the average radius of Europa, 4.5 times for Io, and 1.93 times for Enceladus. For the moons of Mars the situation is even worse with a sphere of influence radius 8.22km with a maximum radius of 7.8km for Deimos, and for Phobos the sphere of influence is *inside* of the moon with a radius of 7.24km for the sphere of influence but an average radius of 11.2667km for Phobos itself. For these moons with both small spheres of influence combined with the low relative velocities of orbiters, the patched two-body approximation is not well suited even as a method to generate a coarse initial guess.

Another issue is that patched conic methods cannot capture libration point dynamics, regardless of the system. So, even for bodies with relatively large spheres of influence such as the Earth and the Moon, for missions such as ISEE-3, Genesis, Artemis, and the James Webb Space

Telescope, this approach could not be used for either mission design or guidance purposes. [Lo01]

In particular, for the two rather different applications of a Phobos orbiter redesign system and a Libration orbit based NEA interceptor discussed in Chapters 10 and 12, patched conics cannot be used for either onboard or initial design purposes for these cases.

2.4 Three-Body Model Based Methods

For cases where the influence of multiple bodies cannot be ignored or treated as small perturbations, it may be necessary to move to three-body (or more) body problem models. This is also the case when features like libration point orbits and their manifolds are needed to enable a wider set of mission scenarios. While in many cases more efficient transfers may be found with multi-body based techniques, it comes at the expense simplicity, speed, analytical methods, and generality of the methods across the problem domain.

2.4.1 Transfers in the Restricted Three-Body Problem

The restricted three-body problem considers the motion of a massless spacecraft (or low enough in mass that the motion of the massive bodies being orbited are not affected) under the gravitational influence of two other bodies. Since the spacecraft does not affect their orbits and other celestial bodies are not considered, the two bodies move in elliptical orbits around their barycenter. If these orbits are circular, this approximation is the Circular Restricted Three-body Problem (CR3BP), otherwise the result is the Elliptical Restricted Three-body Problem (ER3BP). In the rotating frame with angular frequency equal to that of the bodies, the equations of the CR3BP become time invariant. Please see Appendix C for the equations of motion.

Additionally, even though the energy (or the equivalent Jacobi constant) is the only integral of motion for the system, a rich structure emerges in the rotating frame dynamics. Lagrange showed the existence of five equilibrium points in the CR3BP, and later it was shown that various

periodic orbit families exist around these so called libration points. We focus now on the unstable orbit families near the libration points closest to the smaller massive body – such as a moon orbiting a planet or an asteroid orbiting the Sun. Recall that the patched conic methods were not well suited to cases of low relative velocity compared to the smaller body. In these so called "low energy" cases, it was shown that for coplanar motion that the stable and unstable invariant manifolds of planar libration point orbits act as gateways between the energy barriers separating the regions near the larger body, near the smaller body, and exterior to both. [Conl68, McGe69] Partial generalizations of this result to the spatial case exist and have been sufficient in practice. [Koon08, Lo01, Davi12]

As a result, even though numerical methods must be used in the CR3BP, the search space may be significantly reduced by utilizing these structures. The invariant manifolds are propagated (by applying a small perturbation to a state on a periodic orbit in the direction of the stable and unstable eigenvectors of the monodromy matrix) and their intersections analyzed. For example, by finding an intersection in position between the unstable manifold of an initial orbit and a stable manifold of a target orbit, an efficient transfer between the two may be found. In some cases these manifolds intersection in full state so that a "free" transfer exists between them: a heteroclinic connection. In other cases, by either targeting or avoiding the intersections of the interiors of these manifolds, different regional transit or ballistic capture trajectories may be found. [Davi12, Koon08] An example of this is shown in Figure 2.2 in the Jupiter/Sun system for a surface of section defined by a specified x coordinate in the planar problem. Figure 2.2a shows the intersection of the unstable manifold of an L2 planar libration point orbit with the section for up to a few revolutions. Figure 2.2b shows the intersections of the stable manifolds of a L1 planar libration point orbit of the same energy with the same surface of section. Figure 2.2c shows the interior of the intersections of these sets, which due to the manifold properties are exactly those

trajectories that transit from the region exterior to Jupiter's orbit (labeled X) to the region inside of Jupiter's orbit (labeled S for Sun), making fewer than three revolutions around Jupiter. This set is parameterized by one divided by the transfer time and represents a significant reduction of the search space. Points on the boundaries either asymptotically approach or diverge from one or both of the orbits. However by selecting the representation function maximum, the minimum time free transit at this energy is found and is shown in Figure 2.2d.

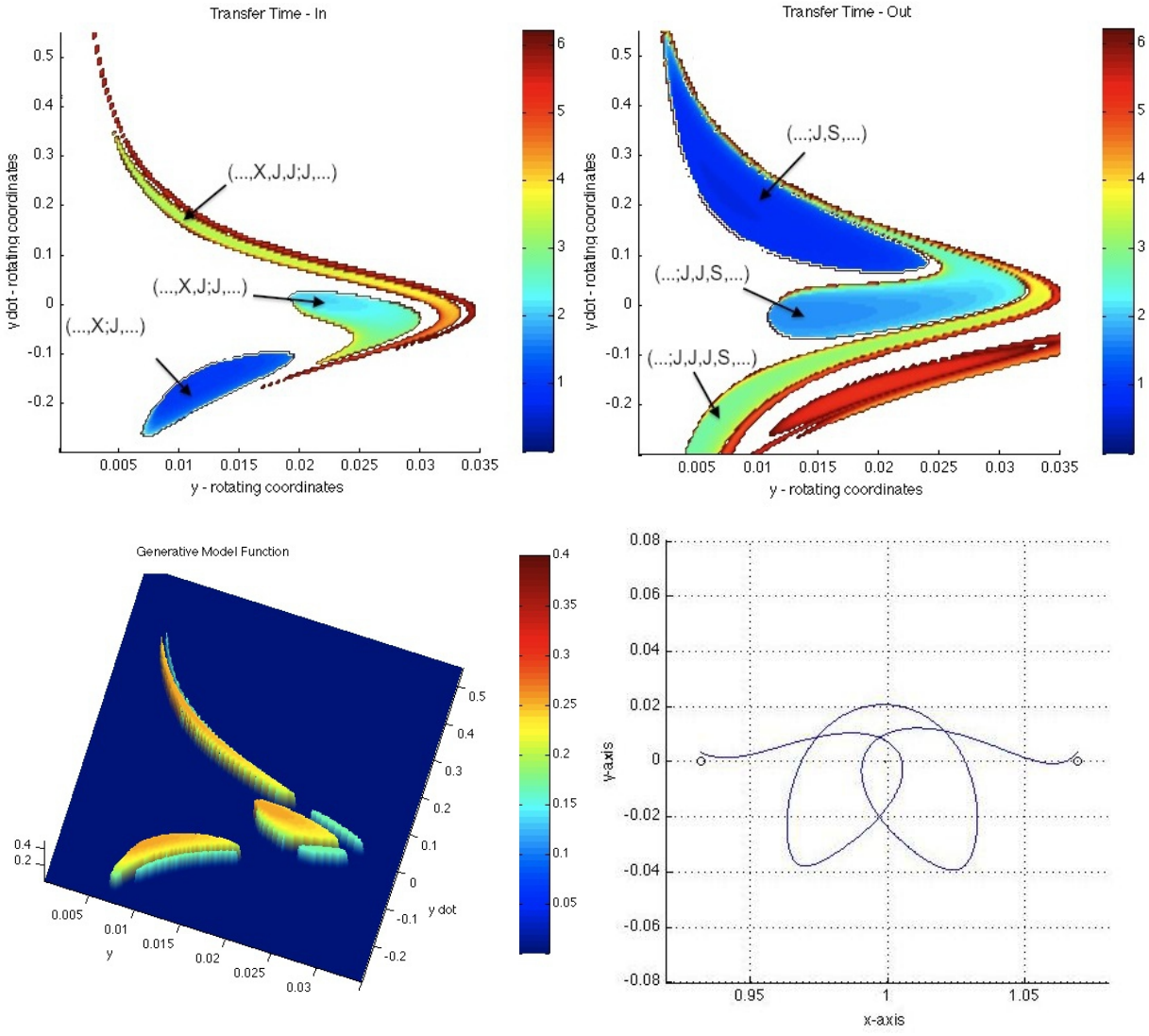


Figure 2.2: a) L2 libration orbit unstable manifold intersections. b) L1 libration orbit stable manifold intersection. c) Intersection of these sets, parameterized by 1 / transit time. d) Fastest free transit.

Methods to generate heteroclinic connections between orbits or specify transit properties via manifold intersection are well developed and the process somewhat automated. They do not however specify more local characteristics such as altitude at closest approach, inclination, or stability. Dynamical maps involve a mass integration of initial conditions in order to identify sets of trajectories with desirable properties such as utilizing the third body influence to create efficient plane change maneuvers, [Trum11] switching between prograde and retrograde motion, [Trum12a] or stable transfers in case of missed maneuvers, [Nakh13] among others. An example of a dynamical map is shown in Figure 2.3 for changes to inclination for a polar orbit in the Europa/Jupiter system. It shows the change in inclination towards either prograde or retrograde motion at the next closest approach after an apoapsis raising maneuver, with black lines representing trajectories with the closest approach distance unchanged. Points on these lines may be used to construct two maneuver, third-body driven plane change maneuvers with the resulting change specified by the map values. Progress is being made in identifying target states with desirable properties by speeding up the map generation process, [Suro15, Nakh13] but this is oriented towards the related but different goal of identifying a target area for the case of a single coasting arc rather than solving a two point boundary value problem with multiple maneuvers. Other classes of methods that discretize the domain in one way or another and analyze connections between these regions are discussed in Chapter 3. One notable search based method involves using a Greedy Search algorithm to find a sequence of orbits that overlap in position taking the spacecraft from orbit to orbit using impulsive maneuvers. [Tsir13]

2.4.2 Limitations for this Application

The various methods described above are very useful for mission design, but not necessarily for onboard trajectory redesign to a designated target orbit or state. This is due to several reasons. The first of these is that the application is simply different in that no target has

been specified but rather a trajectory is sought that meets some other set of goals. [Suro15, Conw07] In the case of map based methods, aside from the time and memory limitations (see Chapter 11) of onboard computing platforms relative to the time and memory that most map generation and use requires, the author is not aware of a well defined automated method to use the multiple maps that would be required for multiple impulse transfers.

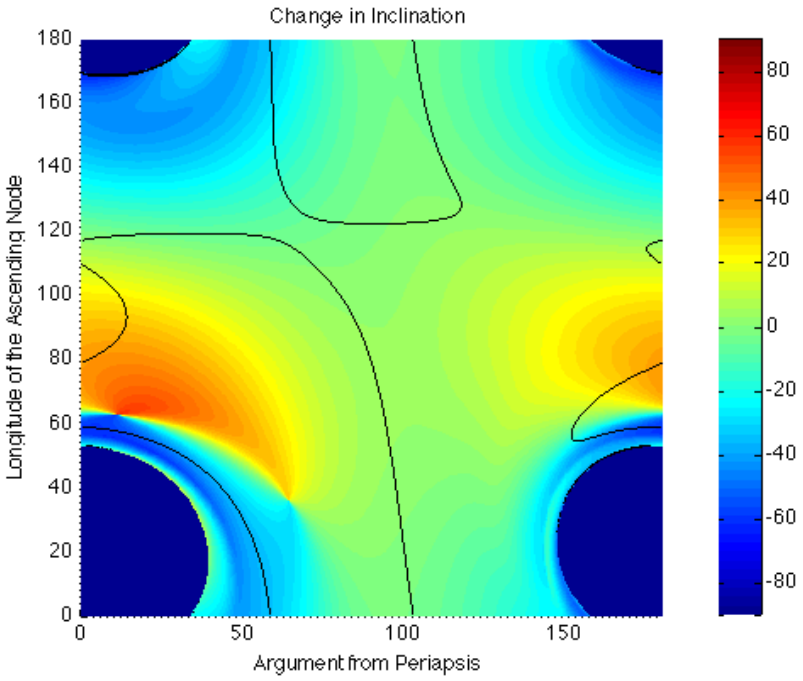


Figure 2.3: Dynamical map showing third body driven changes to inclination after an apoapsis raising maneuver for a polar orbit.

The largest issue with the methods involving linking orbits or manifolds is that while these approaches are suitable for mission design, they are problematic for off-nominal conditions. In short, if a spacecraft is no longer on a periodic orbit or invariant manifold that can be used to reach the target – assuming a situation exists as described in the introduction where redesign is needed rather than station-keeping for some reference – then these methods may be ill equipped. Alternately, the target may not be on one of these structures as was the case for some of the

asteroids to be intercepted in Chapter 12. Even if it is desired to use these structures for most of the transfer, a targeting problem from the off-nominal state to the structure would still need to be solved.

Returning to the restrictions of flight systems, it is also the case that while adaptations to these types of methods may prove suitable in the future or even currently on current desktop platforms, there has been little progress on demonstrating these newer types of automated design methods on flight hardware and software.

2.5 Summary

Impulsive transfers in multibody systems include long coasting arcs and hence accurate approximations of the dominant forces of the dynamics. In many systems of interest, two-body problem based approximations are insufficient to capture such forces and utilize the structures present in three or more body systems. However, standard methods used to generate initial guesses for transfer in multibody systems have several shortcomings. The popular invariant manifold based methods do not provide sufficient coverage of the entire domain, particularly for off-nominal conditions. Dynamical map based methods provide information for a wide set of conditions, but in addition to being costly to compute there is no systematic method to process and join these maps for transfers with maneuvers between the boundary conditions. Even more detailed methods for four or more bodies – for example by patching three-body models or calculating large Poincaré sections to utilize resonant kicks from a fourth body – inherit these issues as well aside from introducing their own complications for an automated system.

Chapter 3

Directed Graph Approximation of the Impulsive Transfer Problem

3.1 Introduction

Within astrodynamics, the two point boundary value problem has no analytic solution for any system more complex than the Two Body Problem. Moreover, while some analytic approximations of the coasting dynamics do exist for limited certain cases [Rich80, Hira06] – and of course a variety of polynomial approximations can be derived for most dynamical systems over short timesteps – these do not provide sufficient state space coverage or time span. On the other hand, comprehensive exploration of the continuous domain using numerical integration is not feasible onboard given the time and computation limitations of spacecraft.

In such situations it is common to turn to a discrete approximation of the system in order to provide a high level description of a transfer. This is because graph and tree search algorithms are

quite fast and have well understood time and memory complexity. The search method and use of search results to construct a transfer will be discussed in the next chapter. This chapter will focus instead on the creation of the discrete approximation of the problem on which the search algorithm is run. After discussing approaches used in similar application, Periapsis Poincaré sections are introduced as ideal domains for dividing the domain into regions. Construction of a directed graph representing the impulsive transfer problem is then described, combining the coasting dynamics and timestep defined by the Poincaré map with impulsive maneuvers. Finally, the resulting structure that influences search times is discussed, leading into the search methods themselves described in the next chapter.

3.2 Related Work

Several different approaches have made steps towards the goal of automated trajectory design using discrete representations and have created a foundation for our design. The structure centered approach of Koon, Lo, Marsden, and Ross [Koon08] started the the use of symbolic dynamics to find allowable itineraries within a class of low energy, resonance crossing trajectories. This itinerary information was then used to find regions of phase space containing the desired trajectories through a related sequence of intersections defined by invariant manifolds associated with libration point orbits. In that case, larger scale features of the CR3BP were of interest and the regions specified consisted of either the entire region near the primary mass, the region near the secondary mass, or the exterior region. What is more difficult to specify without a large amount of user intervention with this symbolic approach are local properties such as radius of closest approach, inclination, direction of motion, etc. As these quantities are necessary elements of mission requirements at the target body, a system is needed which is as adept at approaching a Lyapunov orbit as it is at simply lowering the satellite altitude.

Another class of method are "cell based" decompositions of the domain. In addition to earlier work introducing this concept to the spaceflight community, [Dell06] Dellnitz, et al. used an adaptive cell partition based discretization method to verify transport phenomena. [Dell05] This showed the effect that dynamically important or sensitive areas have on the choice of discretization. Though these methods can address more general constraints, there are limitations to implementing them onboard a spacecraft. The focus on capturing sensitive, long term transport dynamical features implicitly through smaller and smaller refinements led to a discretized system too large for use on a spacecraft, with thousands to tens of thousands of cells for the planar problem with two fewer dimensions. Other partition based method in different domains include triangulation based motion planning for ground robots, [Lava06] and hexagonal decomposition of the sphere of satellite pointing directions. [Kjel12] In these cases, the underlying connection structure is relatively simple since regions only connect with other *adjacent* regions, and any errors arising may be easily corrected for by the large control authority of a ground or air vehicle. These two characteristics certainly do necessarily not hold for an orbiter coasting with any duration.

Concurrent with this research, a graph based mission design approach has been developed involving transfers constructed by linking elements within a large set of periodic orbits. [Tsir12] In that approach, the graph is not an approximation of the dynamics in general, but rather each node consists of one of the available orbits and a connection exists if there is a low cost transition between orbits. This allows for the ability to create desirable "orbit hopping" type transfers as long as the boundary conditions both are periodic orbits in the database, along with the connections between them. While quite useful for mission design in some cases, the orbit-to-orbit restriction, the amount of orbits then required to cover general transfers, and the use of non-heuristic search are likely not intended for the onboard redesign problem described in this paper and earlier work. [Trum12b] Additionally, for cases such as Phobos and Deimos where the massive bodies extend

almost to the libration points, most orbit families will be impacting, restricting the use of such a method.

3.3 Partitioning the Domain

The trajectory design method developed here depends on building transfers using sequences of trajectories – some but not all consisting of elements of periodic orbits, manifolds, and other structures just mentioned. The boundary conditions and transfer arcs must now be represented in such a way as to utilize the speed of discrete search algorithms. This is a necessary step to ensure the right orbits and arcs are selected and ordered in the proper sequence to solve the transfer design problem. Given that our approach is to differentially correct a set of arcs into a continuous transfer, we implicitly require that one arc begins near where another ends. As some sort of discretization of the dynamics is needed, this naturally leads to the concept of partitioning the domain into relatively compact regions, and a condition to discretize time so that the concept of beginning and end have a meaning.

The first step in discretizing the problem is indeed to discretize time. As closest approaches are often used in expressing many mission requirements – and since there is previous analysis done on the subject to verify results [Vill09, Davi12] – our time step for each trajectory in the example model is the time from periapsis to periapsis. This creates a well defined surface of section, reduces the dimension by one, and allows the use of periapsis Poincaré maps to represent and visualize the dynamics. Additionally, this forces the operational benefit of forcing coasting phases between impulses. In exchange for these benefits, true optimality is lost even though several important maneuver types are optimal at apsis conditions. Different time steps will be considered in future research to reduce this liability, but it should be understood that future mentions of optimums will refer to optimality *within* the system constraints.

3.3.1 Periapsis Poincaré Maps

A Poincaré map is a method to associate a discrete time system to a continuous dynamical system. This is done by choosing a two *surfaces of section* and noting the mapping of points from one surface to to the other after evolving under the underlying continuous dynamics. It is often that a single surface is chosen so that a return mapping is created. There are many options for selecting such surfaces, such as the popular choice of $x=1-\mu, y>0$ for analyzing homoclinic and heteroclinic connections of libration point orbits. [Koon08] For this work, the Periapsis Poincaré map will be used. This map is created by analyzing trajectories as they make their closest approach to the moon or asteroid being orbited. In the Circular Restricted Three-Body Problem where the body being orbited is centered at $(1-\mu,0,0)$, if $r_2=\|(x,y,z)-(1-\mu,0,0)\|$ then the periapsis condition may be expressed as $\dot{r}_2=0, \ddot{r}_2>0$.

The work of [Vill09] introduced this concept and showed that the conditions for a proper surface of section are indeed satisfied, and that various alternatives such as apoapsis (a relative maximum of the distance) to apoapsis, or apoapsis to periapsis, etc. also create valid surfaces of section. The Periapsis Poincaré map has benefits over these other options. Primarily, this is due to the fact that mission requirements are generally not expressed relative to the state on a half hyperplane such as $x=1-\mu, y>0$. In comparison, many requirements do relate to states at flybys or viewing requirements at closest approach. Therefore having arcs and impulses tied to closest approach more closely aligns with the boundary conditions that will be provided.

Use of a Poincaré map in the context of this project provides a discrete time step that is tied to the system dynamics rather than a constant step for every initial condition. Additionally, it serves to reduce the dimension of a discretization by one dimension. In the case where each Poincaré map is associated with a constant energy as it is here, the number of dimensions needed to specify the state is reduced by a total of two. The remaining coordinates used in this application are

described further in the next section.

3.3.2 Coordinate Partition

Once a surface of section is defined, the next step of the approach is to partition the Poincaré section into a finite set of regions, and then to describe transitions between these regions. A simple way of doing this involves dividing the domain into a grid of rectangles, cubes, or hypercubes in phase space due to ease of defining each set, as was pursued in cell mapping approaches. [Dell05] In robotic path planning, triangulation is a popular choice. [LaVa06]. In our setting, squares and triangles are not 'natural' shapes either for reflecting the dynamics or describing boundary conditions. A division based on orbital elements might be a better choice; approximate radius, argument of periapsis, longitude, and energy are much more natural for defining orbits near the secondary body. Divisions along values of these natural properties will not lead to cubical regions; including radius leads to shells around the body being part of the region boundaries, for example. However, such an approach is more a generalization of a cellular approach rather than standing in opposition. Suppose an alternative set of parameters such as spherical coordinates or orbital elements is used to define orbit properties we desire to use to partition the domain. Assuming this set of "coordinates" is able to completely describe the state, there exists what is essentially a change of coordinates:

$$\phi : [a_1, b_1] \times [a_2, b_2] \times \ldots [a_n, b_n] \subset \mathbb{R}^n \rightarrow \Omega \subset \mathbb{R}^n$$
$$\phi : (q_1, \ldots, q_n) \rightarrow (x, y, z, \dot{x}, \dot{y}, \dot{z})$$

By considering the image of a small hypercube in the space of parameters under this function, it is seen that this function warps such a cube into a connected set bounded by some parameter values. In the examples presented in this paper, spherical coordinates are used to define position, and the CR3BP energy E (such that $J=-2E$ is the Jacobi constant) is used to define the velocity magnitude. The velocity direction is determined by the alternate form of the apsis

condition that $\langle \vec{r}_2, \vec{v} \rangle = 0$ and by the angle α of the velocity in the plane normal to the position vector, taken relative to the *xy*-plane. This is shown in Figure 3.1. Thus for each energy level *J*, cells defined by ranges of *r*, θ, φ, and α values define the partition used to divide the periapsis Poincaré section within the domain near the orbited body, defined by the radius of the Hill's Sphere or otherwise. Nonuniform interval values may be chosen to have smaller, more precisely defined regions near areas of interest. This includes altitudes corresponding to mapping orbits, or apses of orbit families such as libration point orbits.

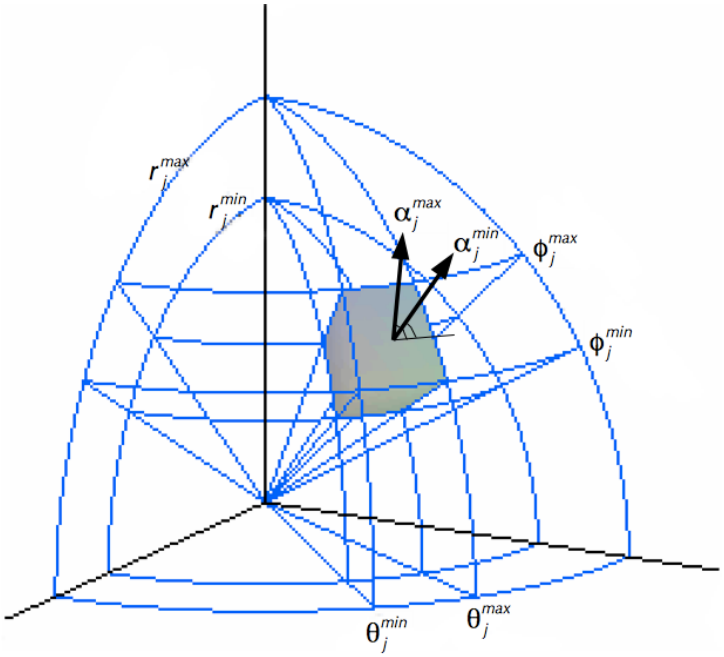


Figure 3.1: Defining a region within the Periapsis Poincaré section using spherical coordinates for position and velocity angle.

The region size within the partition depends on external factors to a large extent. Smaller regions lead to a larger graph, taking both more time and memory, both of which may be limited by spacecraft computer constraints. On the other hand, as was shown by the Weak Shadowing Theorem, smaller regions lead to a decrease in the discontinuities that the differential correction process must fix. The development of a strong correction and optimization process has facilitated the creation of larger and larger regions.

3.4 Directed Graph Construction

With a partition chosen for the domain, the next step is to classify trajectories in relation to the discretization – namely within which regions of the surfaces of section do each trajectory arc start and end. In the present case, this is the location of a trajectory at each periapsis. Using a directed graph is an ideal way to represent which transitions are possible without focusing on excessive point-to-point details. Not only does this record this information, it is also the structure over which discrete search algorithms run. A graph that represents and approximates a continuous dynamical system is referred to as the *symbolic image* of the system.[Osip06]

3.4.1 The Symbolic Image of Osipenko

After briefly giving an overview of the process to create a directed graph approximation of the dynamics in this opening paragraph, the formalism of Osipenko [Osip06] will be used to rigorously describe the process. The first step is to partition of the Poincaré map as described in the last section. A list of these regions is created so that a graph node is associated with each region of the partition. A large set of initial conditions spread throughout the domain is then integrated to sample the system flow, making sure multiple trajectories per region are integrated. If an arc from this set of conditions, the database of key orbits, or an impulse exists which begins in a region R_i and ends in a region R_j, a directed edge from Node n_i to Node n_j is added. This is shown conceptually in Figure 3.2. The arcs or impulses associated with each transition are then labelled with their starting and ending region or node. As will be discussed later, the link weight can correspond to fuel cost for an impulse, or some estimate of control cost, uncertainty, or time of flight for ballistic connections. Such a graph, called a *symbolic image*, then approximates the system of interest for both coasting flight and impulsive maneuvers.

In the formalism of Osipenko [Osip06], this requires that for a compact domain $\Omega \subset \mathbb{R}^m$ we have a finite covering of Ω by closed sets $C = \{R_1, R_2, \ldots, R_{max}\}$. We have this in our choice of

closed domain bounded between libration points – excluding the secondary body interior – and the partition chosen. For each region R_i consider its image $f(R_i)$ where here f is the function taking a point to the intersection of its trajectory with the surface of section. In this case, f is the Poincaré map taking an initial state to its next periapsis. Let $s(i)=\{j : R_j \cap f(R_i) \neq \varnothing\}$ be the set of labels or *symbols* associated with regions intersecting $f(R_i)$. These images are approximated by integrating a fine grid of initial conditions and by including orbits, manifolds, etc. in dynamically sensitive areas that a grid of points may miss. By taking this combined approach, an unmanageable number of integrations may be avoided.

Let G be a graph whose nodes correspond to the regions of the partition. Two vertices i and j are connected by a directed edge if and only if $j \in s(i)$, though only one edge is added regardless of the number or measure of points in the intersection. This is illustrated in Figure 3.2. Osipenko calls a graph created in such a way the *symbolic image* of f with respect to the covering C. Thus if a link exists in G, *there exists at least one ballistic arc in our collection* which joins the two regions.

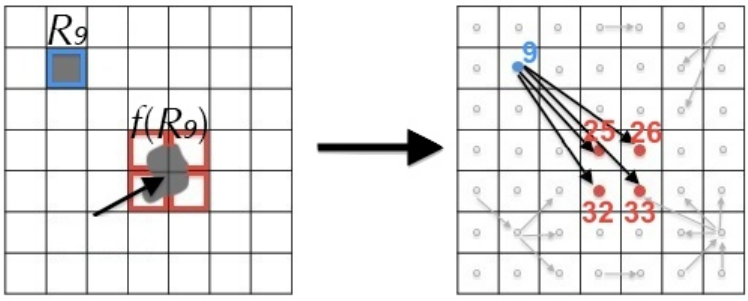


Figure 3.2: The intersection of the image of one region with others leads to the creation of directed arrows in the graph representation.

In this context, a *path* is a sequence of nodes $\{z_k\}$ for which $z_{k+1} \in s(z_k)$. An ε-*orbit* is a sequence of points $x_k \in \Omega \subset \mathbb{R}^M, k \in \mathbb{N}$ that, for a given ε>0, satisfies $|f(x_k) - x_{k+1}| < \epsilon$ for all k.

One of the results of the *Weak Shadowing Theorem* [Osip06] states that for a path $\{z_k\}$ on a symbolic image *G,* then for all ε>*d* where *d* is the maximum diameter of the sets in the partition, there exists a sequence of points $\{x_k : x_k \in M(z_k)\}$ such that $\{x_k\}$ is an ε-orbit in the original domain. In other words, for any path there is an ε-orbit with the largest discontinuity bounded from above by the maximum cell diameter.

This formalism supports the fact that by creating a graphical representation of the dynamics using transition maps and the region-labeled database trajectories, a ballistic transition arc will exist for each graph link. Furthermore, for any path on the graph generated by a search process, there will exist a sequence of arcs starting at x_k and ending at $f(x_k)$ with gaps between the endpoints of each arc no more than the size of the region they are meeting in, and no more than the maximum *d*. This confirms the physical intuition of the construction method, and implies the ability to construct a complete transfer approximated by the arcs. The gap between endpoints is precisely why differential correction is needed for any set of arcs selected via graph search.

The relationship between a path in such a graph and a transfer in the original problem can now be defined. Suppose a path is found in the graph which leads from the node associated with an initial region to the node associated with a target region. *This corresponds to a sequence of regions, each connected to its successor via ballistic arcs or an impulse, linking the regions containing the boundary conditions.* As a result, the directed graph and arcs/impulses associated with transitions within that graph provide the foundation for the online processes and need to be stored onboard. This includes: (1) The symbolic image, which provides a meaningful representation of the system that can be used by a discrete search process. (2) For ballistic transitions, one or more of the arcs that led to the creation of a directed edge in the graph must be stored and labelled as transitioning between the regions associated with the graph nodes. These elements allow the spacecraft to create the skeleton of a transfer quickly.

3.4.2 Additional Structural Elements

The process begins by considering what types of key trajectories are necessary to explicitly include in a multibody trajectory design program aside from a general sampling of arcs in the domain. The role of such trajectories is to provide a set of high priority arcs for selection as transitions between regions and as boundary conditions. Of primary importance are periodic orbits, which play multiple roles within the proposed method. First and foremost, periodic and quasi-periodic orbits serve as the most likely boundary conditions for the types of transfer problems of interest. Thus periodic orbits must be available as options when specifying initial and final conditions. The specific orbits of interest depend on the application, and with specific examples provides in Chapter 10.3.1. Aside from their use as boundary conditions, periodic orbits and related features act as gateways for efficient transport across resonances [Koon08, Dell05] and have been effectively used in other correction based transfer methods. [Sukh04]

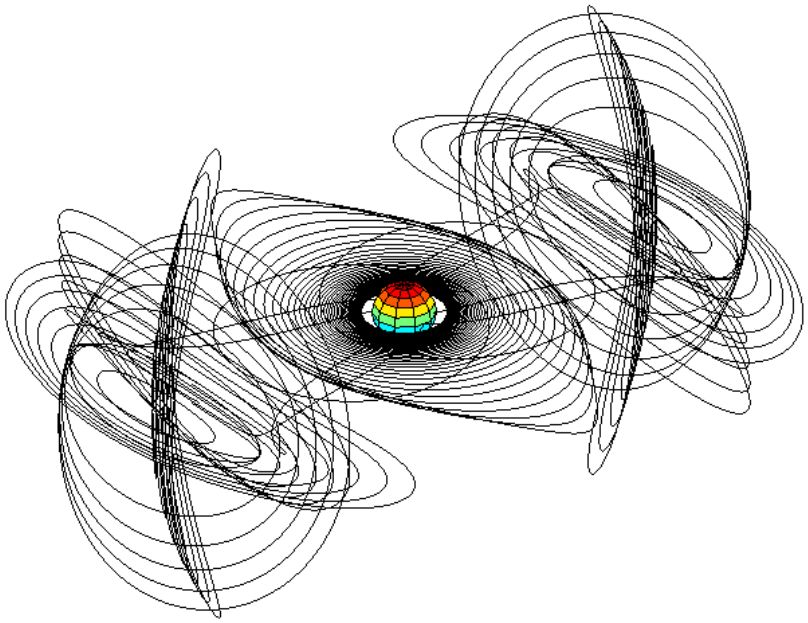


Figure 3.3: Example periodic orbit families that should be explicitly included in the list of arcs used in graph creation and available later to the arc selection process.

Other important structural elements such as libration points, invariant manifolds, and collision trajectories play a role shaping the system dynamics. By explicitly including these trajectories into the system, the goal is to avoid the combinatorial problems that arise from trying to capture these features solely through cell mapping. Orbits may either be obtained by using numerical continuation such as the tool AUTO, [Doed97] or differentially corrected from a library in a closely related model. [Marc07] Various methods exist for calculating invariant manifolds [Koon08, Vill04] and plane change maneuvers. [Vill09, Trum11]

3.4.3 Impulsive Connections

Thus far, the focus has been on arcs subject to motion in the spacecraft's dynamic environment. Next, impulsive maneuvers need to be incorporated into this directed graph framework. An impulse is an instantaneous change to the spacecraft velocity, which does not change the spacecraft position. Therefore for any region, impulsive connections may exist with all other regions with the same position. These may be divided into two cases: regions on the same Poincaré section with the same energy and velocity magnitude but different velocity directions, and regions on different Poincaré sections which will have different velocity magnitudes and possibly different velocity direction as well.

The number of such connections is bounded by the product of the number of velocity directions and the number of Poincaré sections. In practice this is lower for several reasons. First is that there will be a maximum amount of impulse allowed for a single maneuver, Δv_{max}. If no impulse exists between two regions with magnitude less than this number, then no impulsive link should be created. The same may be applied in the case there is a limit to the thrust angle, which in turn limits the set of velocity directions that can result from an allowable impulse. Finally, there are cases where the positions within a region have potential energy greater than the energy of a Poincaré section. As such regions of state space are inaccessible, no links to or from such regions

should be created. Such examples of graph pruning save both memory and time spent during the search process.

Link costs are simply estimates of the ΔV magnitude required for an impulse between these two regions of state space. As the ballistic arcs that are to be joined by such an impulse are not known ahead of time, use the average ΔV magnitude joining states in these two regions. The necessary integrations needed to perform this calculation are simplified by the use of Periapsis Poincaré sections. To begin, since the states satisfy the apsis condition, the angle between them is determined by the difference $\alpha_k - \alpha_j = \alpha(\vec{v}_k) - \alpha(\vec{v}_j)$ where α is defined in 3.3.2 above and is one of the parameters used to define the partition. Next, since the energy is fixed for each section, the velocity magnitude is a function of the potential. Also, since impulses are between states with the same position, this integration is only needed across a single set of positions. For each position and two velocities, the Law of Cosines gives the magnitude of the ΔV as:

$$\|\vec{v}_k - \vec{v}_j\| = \sqrt{\|\vec{v}_k\|^2 + \|\vec{v}_j\|^2 - 2\|\vec{v}_k\|\|\vec{v}_j\|\cos(\alpha_k - \alpha_j)} \quad \text{where}$$

$$\|\vec{v}_k\| = \sqrt{2E_k - 2U(\vec{x})}, \|\vec{v}_j\| = \sqrt{2E_j - 2U(\vec{x})}$$

These last equalities hold because in the CR3BP or Hills Model with or without higher order gravity, the energy in the restricted problem *E=-J/2* where *J* is the Jacobi constant, is given by

$E = \frac{1}{2}\|\vec{v}\|^2 + U(\vec{x})$ where U is the *augmented potential* including rotating frame effects. [Koon08, Vill03] Please see Appendix C for more detailed model description.

Potential for both spherical harmonic gravity and the CR3BP can be expressed in either Cartesian or polar coordinates for position, thus for either type of cell definition the potential function and integration bounds are easily expressed. So, for regions with position defined by the polar coordinate range $[r_j^{min}, r_j^{max}] \times [\theta_j^{min}, \theta_j^{max}] \times [\phi_j^{min}, \phi_j^{max}]$ and velocity angles within ranges

$[\alpha_j^{min}, \alpha_j^{max}], [\alpha_k^{min}, \alpha_k^{max}]$, the $\|\Delta V\|$ between states in the two regions is given by the function

$$f(r,\theta,\phi,\alpha_j,\alpha_k)=\left[2(E_k+E_j)-4U(r,\theta,\phi)-4\sqrt{E_k-U(r,\theta,\phi)}\sqrt{E_j-U(r,\theta,\phi)}\cos(\alpha_k-\alpha_j)\right]^{1/2}$$

The average ΔV magnitude $\overline{\|\Delta V\|}=\frac{1}{\mu(R_j\times[\alpha_k^{min},\alpha_k^{max}])}\int_{R_j\times[a_k^{min},a_k^{max}]} f\,d\mu$ is then given by an iterated integral with simple integration bounds that can be calculated numerically in Matlab:

$$\frac{1}{\mu(R_j)(\alpha_k^{max}-\alpha_k^{min})}\int_{r_j^{min}}^{r_j^{max}}\int_{\theta_j^{min}}^{\theta_j^{max}}\int_{\phi_j^{min}}^{\phi_j^{max}}\int_{\alpha_j^{min}}^{\alpha_j^{max}}\int_{\alpha_k^{min}}^{\alpha_k^{max}} f(r,\theta,\phi,\alpha_j,\alpha_k)\,d\alpha_k\,d\alpha_j\,r^2\sin\phi\,d\phi\,d\theta\,dr$$

3.4.4 Ballistic Link Costs

As the graph search is to guide the overall process towards the lowest cost transfer within the system restrictions, a cost must be associated with each transition. This differs from the constant unit weights used in the theoretical treatments of Osipenko [Osip06] and the probability of transition used by Dellnitz [Dell05] as the focus here is in transfer design. Note that until differential correction is performed, we will not know the true cost for our problem, so we will populate the graph with *estimated costs* for the search process.

Costs for impulsive connections were described above as an average are an estimate of the ΔV required to transition from one region to another with a single impulse. Costs for ballistic arcs are more subtle and depend on a few factors. One of these is whether the final cost function is solely based on ΔV or is a weighted combination of fuel and transfer time. Another factor is that in order to prevent the search process from wandering around and to establish the search time complexity, there must be some nonzero minimum link cost ϵ.

In the case where the eventual cost is a combination of the fuel cost and transfer time, assigning the cost of ballistic arcs is very straightforward. Assuming the final cost function is of the form $J=\sum\|\Delta V\|+w\sum t$, then the cost of each ballistic transition is simply $w\cdot t_{arc}$ where t_{arc} is the coasting time of the arc associated with the ballistic transition. Should multiple arcs be

associated with each transition, as with impulsive connections above either the minimum or average may be used. In Chapter 5, among other constraints a minimum and maximum coasting time constraint is included in the problem. This is due to operational constraints relating to engine firing frequency as well as time for orbit determination after maneuvers. As a result of the existence of this constraint, a minimum link cost $w \cdot t_{min}$ is established for ballistic arcs. This is the approach that has been taken in the examples.

The case where the cost is entirely dependent is more complicated. Setting the cost for ballistic links to 0 is not an option. Setting them to a constant nonzero value ϵ is possible, but this is not very physical and so the question of what the "cost" of an arc is terms of units of ΔV will result in an ad hoc design decision. Using a weighted total transfer time would be more physically relevant, but still not connected to the fuel-only cost. Limited to only information about the ballistic transition itself, instability provides a link to total fuel expenditures. This is due to the fact that higher instability leads to greater magnitudes of trajectory correction maneuvers (TCMs) as small errors in navigation or maneuver execution lead to larger errors from the designed trajectory. Another appealing feature aside from smaller and less frequent TCMs is that the more stable option is better from a mission safety viewpoint.

For the stored ballistic arc linking regions R_i, R_j, let Φ be the corresponding State Transition Matrix (STM). Consider the operator norm $\|\Phi\| = \sup_{\vec{v}} \frac{\|\Phi \vec{v}\|}{\|\vec{v}\|}$ of the STM, which measures how large a small perturbation from the reference may become after bring propagated by the dynamics for the reference coasting time. It is also equal to the largest singular value of the STM, $\bar{\sigma}(\Phi)$. While other measures of stability exist, [Sche01, Vill08, Froe97] this choice has the benefit of being easy to calculate and using only the information of the ballistic arc that will actually be used in the transfer construction process. Additionally, as the design models used to

build the graph are Hamiltonian, [Koon08, Vill03] $\bar{\sigma}(\Phi)\geq 1$. This is because $\bar{\sigma}(\Phi)\geq|\lambda|$ for all eigenvalues of Φ. Since the system is Hamiltonian, if λ is an eigenvalue of Φ then $\mu=1/\lambda$ is as well, [Koon08] therefore the existence of any λ with $|\lambda|<1$ implies the existence of another eigenvalue μ, $|\mu|>1$. The exact function which assigns a link cost to the measure of instability is application specific and will depend on the navigation and guidance capabilities of the spacecraft. However, whatever form this function has – which may even be multiplying by a constant – its value will increase with increasing instability. Thus its value when evaluated at $\bar{\sigma}(\Phi)=1$ provides the needed lower bound the directed graph link costs must possess.

3.5 Resulting Graph Structure

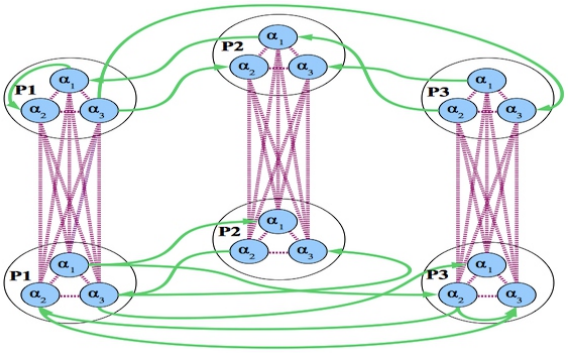


Figure 3.4: Conceptual model showing graph links from ballistic arcs and impulses.

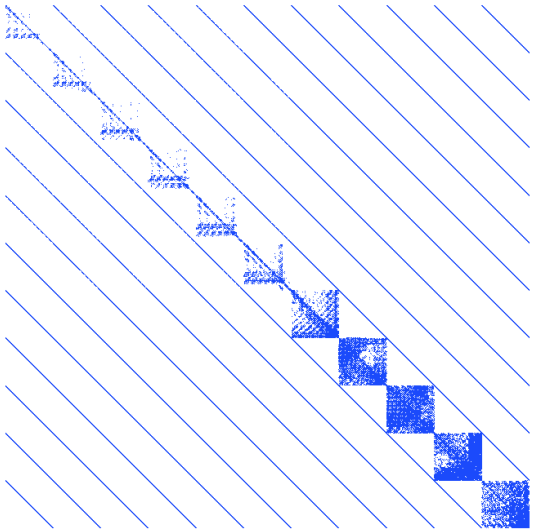


Figure 3.5: Sparse graph representation of the actual Mars / Phobos model.

As will be discussed in Chapter 4, several aspects of the structure result in a fast search process, so the structure should be made explicit here. To begin, recall that the initial conditions and regions are defined on Poincaré sections of a specified energy (the graph and arc model being conservative). Thus the graph can be viewed as a set of layers with a layer per energy level. Ballistic connections then only exist within individual layers. Impulsive maneuvers may exist

between any regions with the same position. This results in connections both within layers due to changing direction without changing energy, and between layers due to changes in velocity (magnitude and possibly direction) that change the energy. This is shown conceptually in Figure 3.4. The green curved, solid arrows represent ballistic connections, while the purple dashed arrows represent impulsive maneuvers. Within each layer, the clusters of nodes represent states with the same position but different velocities. As will be discussed in Chapter 4, this layered structure may be used to generate a simple yet effective heuristic function.

In reality, this graph is quite large. For the Phobos application discussed in Chapter 10, each Poincaré section is divided into 6,300 regions, and there are 11 energy levels/sections considered for these scenarios. This leads to a total of 69,300 total graph nodes. (These values are not random but are based on the needs of the scenarios expected to be faced. Please see Chapter 10.4.1) A dense graph of this size could potentially contain billions of connections, which would be far to large to work with. Thankfully, the structure described above is quite sparse, since only matching positions may be linked impulsively, and on average a given region only flows into several other regions ballistically. To be precise, the graph is sparse with only .07% of all possible connections being created. This is visualized in Figure 3.5, with a dot added where a connection exists. Note that as the smallest dot size is 1 pixel, this looks more dense than it actually is. With the connection cost stored as a real number, this results in 9MB of memory used in Matlab using the *sparse* datatype.

3.6 Summary

This chapter describes the development of a discrete representation of the impulsive transfer problem. Periapsis Poincaré maps were used to define physically meaningful timesteps and reduce the dimension of the discretization. Both the coasting dynamics that dominate these

systems and the possibility of impulsive maneuvers are modeled in this approach, with link costs determined in such a way as to provide a good initial guess to a correction and local optimization process for the minimum fuel or minimum fuel and transfer time problem.

Chapter 4

Initial Guess Generation via Heuristic Search

4.1 Introduction

The previous chapter discussed how to construct a directed graph approximation of the dynamics from a set of ballistic arcs and impulsive changes in velocity. However, the importance of these arcs does not end at that stage, nor is finding a path in the directed graph a goal in and of itself. By construction, each ballistic transition in the graph has at least one arc associated with it. For a given pair of boundary states, consider a path in the graph between the region nodes that contain these states. Such a path found may be used to select a set of arcs that link together the regions associated with the path nodes. The discontinuity in position between these arcs is limited by the region size, as is the difference in position between the first and last arcs with the boundary states. Furthermore, the difference in velocity at the arc endpoints is approximately equal to the link cost of the impulsive transition. In this way, the graph search results are used to provide an initial guess to a differential correction and local optimization process. This part of the transfer design process may be summarized as follows:

Given:

- Directed graph as described in Chapter 3.
- Set of arcs containing one or more arc for each ballistic transition.
- Two boundary condition states or periodic orbits.

Process:

1. For each boundary condition, associate one or more graph nodes.
2. Perform path finding search algorithm on the directed graph.
3. For ballistic transitions in path, select a ballistic arc.
4. Provide these arcs, in order, and boundary conditions to differential corrector.

The remainder of this chapter will elaborate on the steps of this process, with an emphasis on the search process itself.

4.2 From Boundary Condition to Graph Nodes

Boundary conditions requirements are precise enough that they are unlikely to be provided as simply a region within a partition. Thus in order to provide start and goal nodes for the search process, it is necessary to translate a variety of types of boundary conditions into one or more appropriate graph nodes.

4.2.1 Boundary Conditions as States

One possibility is when a specific set of orbital elements or state is required. In this case, a necessary step is to determine which region to associate with this state. Recall that each periapsis Poincare section has a specified energy and that the apsis condition holds. Different procedures must then exist in cases when the boundary conditions are on the surface of section, and when they are not.

For states on the surface of section, the method is simple. Regions have been defined here by a range of values in the coordinates r, θ, ϕ, and α. Once the state or orbital elements are transformed into these coordinates, a sequence of inequalities may be evaluated programmatically. Other types of partitions would lead to other methods such as oct-trees [Nakh13] or a set of function evaluations. [Trum12a]

For conditions off the Poincaré section, a little more work is required. If a state is not at apsis, two methods should be employed and compared. The first is that the initial condition may be propagated forward and its next apsis (should it exist without escape or impact) taken as the initial condition used for the search. For the final target condition, the same process is applied but integrating backwards in time. The trajectory to the surface of section is then appended to the set of arcs found by the search process. The other approach is to use the *projection* of the states on to the surface of section to determine set membership. A maneuver is then added to account for the change in velocity between the boundary state and the projection on to the surface of section. If in addition the energy level is not among those included in the mission planning calculations, an impulse is added to put it on the section closest in energy. In any of these cases these changes are done before the optimization procedure so that the effects of these accommodations may be minimized.

4.2.2. Boundary Conditions as Periodic Orbits

Another likely option is that a periodic orbit will be supplied as a boundary condition, particularly for the target. Ideally these orbits would be selected from the database of periodic orbits and thus already be listed in terms of their regions at each time step. Otherwise the process used to determine region membership for single points should be run at each apse or other time step of the orbit to find the periodic sequence of regions that the orbit defines. Regardless of which type of timestep was used, a periodic orbit has a periodic path in a graph representation. [Osip06]

For each search run, one node must be specified as the goal node and one as the start node. Therefore if one or both of the boundary conditions are periodic orbits, one node must be picked from each orbit per run, and so the search process is executed for each pair of nodes defining start and goal nodes, time permitting. For example, Figure 4.1 shows two transfers consisting of a different number of arcs and impulses resulting from different search paths which target the two periapses of the target Distant Retrograde orbit (DRO).

Suppose then that the periodic symbolic boundary conditions P_1 and P_2 *within the directed graph* consist of p_1 and p_2 nodes each, where for single state boundary conditions these 'orbits' are simply one node. That is, $P_1=\{n_1, n_2,\ldots, n_{p1}\}$ and $P_2=\{m_1, m_2,\ldots, m_{p2}\}$. From these sets of nodes, p_1p_2 number of pairs of start and goal nodes may be created, and transfers constructed for each, time allowing. Once a path in the graph is found, the remaining nodes in the periodic graph orbits are appended to the beginning and end of the sequence. For example, suppose for start node n_1 and goal node m_1 a path $n_1, q_1, \ldots, q_k, m_1$ is found. In order to select both the arcs of the periodic orbits *and* the transfer joining them, the final node sequence is:

$$n_1, n_2, \ldots, n_{p1}, n_1, q_1, \ldots, q_k, m_1, m_2, \ldots, m_{p2}, m_1.$$

This modified path, which captures the periodic nature of the boundary conditions, is then what is used in arc selection and differential correction.

Using this discretization scheme multiple itineraries may need to be evaluated in order to find the optimal transfer, though the first option considered may provide a feasible transfer in the case of severe time constraints. On the other hand, a benefit of this procedure is that it can create multiple transfer options between the two orbits in the case the point to point paths are different for each search. As providing multiple transfer options has been identified as a key requirement for an onboard system in case there are operational constraints not captured in the discretization, this is an important benefit. Additionally, as the resulting transfer problems are not dependent on one

another, they may be run in parallel if the computational architecture allows it.

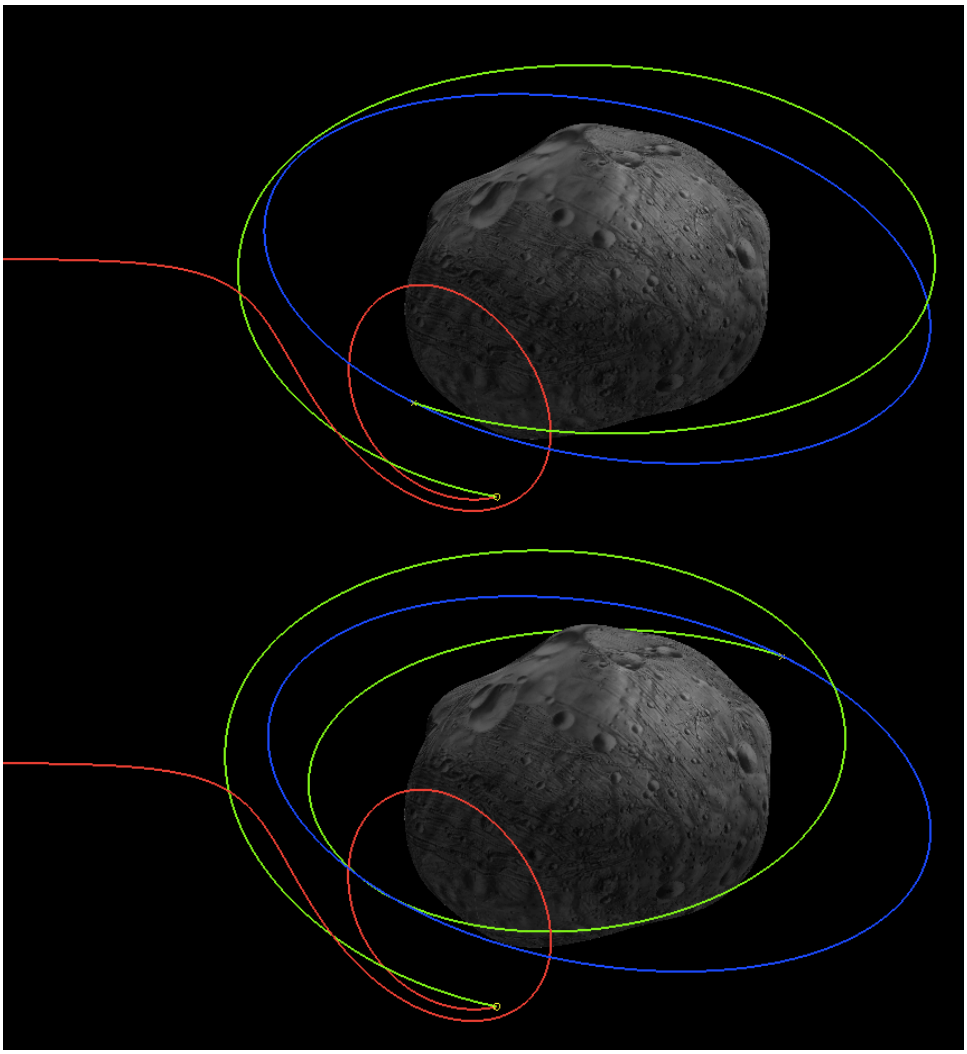


Figure 4.1: An uncontrolled initial condition (red) will lead to escape from the system. Two transfers (green) from this initial condition to different periapses of a Distant Retrograde Orbit (blue) are generated. Different search paths resulted in a different number of impulses.

4.3 Search Method

A central motivation of this approach is that discrete search algorithms are quite fast compared to a large scale optimization problem in a complex system. However, there is a wide variety of search algorithms types suited to different types of problems. Given the requirements of

our problem, the primary challenge then is to select the best method that produces results that are both relevant *and* fast. This section describes several of the distinctions between search families and assesses each option for suitability in solving the transfer problem as well as for performance.

One division between different search techniques is graph vs. tree search. While this can refer to the data structure the search is performed on, here we are distinguishing between allowing a node to appear on a path more than once. Graph search requires maintaining a list of nodes which have been visited and only considering those nodes which have not been visited yet. For a tree search (even on a directed graph), this is not done and so a path may visit one node multiple times. [Russ09]

Another division is between Uninformed and Heuristic Search methods. Uninformed methods are those in which no a priori knowledge of the system under consideration is used to steer the search towards the goal. This is in contrast to *heuristic search methods* which use estimates of the remaining cost to a goal state for different nodes under consideration to pick the best option to proceed. Heuristic search methods can be much faster than uninformed methods. However, in the limiting case where the estimates provide no useful knowledge, heuristic methods perform like uninformed methods. Because of this fact and because uninformed methods provide the backbone for heuristic methods, a detailed description of uninformed search is provided as well. These distinctions and their relevance to the transfer problem now follow.

4.3.1 Graph or Tree Search?

When both tree and graph searches are possible, graph searches are generally faster, though at the cost of memory to maintain the list of visited nodes. [Russ09] The difference between the two is that graph search methods maintain a list of visited nodes so that a single node is not repeated along a path, reducing the number of nodes that need to be evaluated. Due to the method of handling periodic boundary conditions in Chapter 4.2.2 above, both graph and tree

search are capable of being run in such a situation, despite the fact nodes are repeated in the periodic graph orbit. This is due to the fact the search itself is between pairs of nodes with the periodic orbit being *appended* to the resulting path found *after* the search process but before the necessary arcs are selected.

The choice between the two methods is then a product of the computational environment. If sufficient memory is available to maintain a list of visited nodes, then the potential increase in speed (and a worst case complexity result as will be described in the next section on Dijkstra's Algorithm) may make graph search preferable. If the memory is not available, then the tree search should be used. In the applications tested thus far – even on a memory limited RAD750 flight system at Jet Propulsion Laboratory – graph search has been used with no issues.

4.3.2 Uninformed Search Methods

The algorithms chosen for comparison to heuristic algorithms are those standard in the case where search optimality is a concern. *Uniform Cost Search* is the tree search method and *Dijkstra's Algorithm* the graph search method of a fairly simple principle related to Dynamic Programming. [Russ09] More complete algorithm descriptions can be found, [Russ09] but briefly:

1. Beginning from the start node, examine cost $c(n,n')$ from the current node n to each *successor* – a node connected by a directed link (see Chapter 3) from the current node – n'.
2. For each node considered, take the *path cost* $g(n')$ to be the total cost from the start node to n' by summing the link costs on the current path linking the two.
3. The list of nodes that are successors of all the nodes considered is called the *frontier, F.* Set the current node as the node on the frontier with the lowest cost. i.e. n_{new}=arg min { $g(n')$: $n' \in F$ }
4. Add successors of the current node to the frontier, calculate their path cost. *[Difference: As Dijkstra's Algorithm is a graph search, any node previously expanded cannot be added to*

the frontier again.]

5. If a goal state has been reached and its path cost is less than the path cost for the other nodes on the frontier, the process terminates. i.e. terminate if $n=n_{goal}$ and $g(n) \leq \min \{ g(n') : n' \in F \}$

These particular methods have some well known properties. With a minimum step cost (which exists by construction as shown in Chapter 3) both methods are *complete* – a path to the goal state will be found if one exists – as well as optimal within the context of the search on the graph. Without taking the finite size of the graph into account yet, analyzing each step of the search process shows that the time and memory complexity is exponential, given by $O(b^{C^*/\varepsilon})$ where b is the branching factor, C^* the true optimum cost for the case under consideration and ε the minimum step cost. Dijkstra's Algorithm will provide a path at least as fast, but for finite graphs has also been shown to be bounded by $O(E_G + V_g \log V_G)$ in efficient implementations using Fibonacci Heaps (though these can be difficult to implement in practice), [Fred87] where E_G is the number of edges on the graph and V_G the number of vertices. While this cap is useful, the exponential formula more clearly illustrates the role the connection density in the graph for coasting and impulses – combined here as the branching factor – and itinerary length have on search time.

4.3.3 Search Heuristics

Heuristic search methods can be seen as extensions of the previous methods as both classes involve sorting the options under consideration using functions of cost in order to best choose the next step. However, heuristic search methods use *partial knowledge of the future cost remaining* to the goal for each option in order to improve its decision making. A *heuristic* is simply an easy to calculate estimate of this remaining cost to the goal derived from system knowledge, with the stipulation it is zero at a goal. If these estimates are good, by choosing nodes with lower

heuristic values *in addition to* the current cost, the search will be steered towards the solution. This is what is needed to speed up the uninformed search algorithms above. A few concepts are needed first. The first is *admissibility.* Let $h_E(n)$, $h_T(n)$ be the estimated and actual costs remaining to reach the goal state, respectively. A heuristic function h_E is admissible if $h_E(n) \leq h_T(n) \forall n \in G$, meaning it never overestimates the remaining cost. A slightly stronger condition is *consistency,* which is essentially the triangle inequality: For any node n and for all of its successors n', $h_E(n) \leq h_E(n') + c(n, n')$ where the last term is the true cost from n to n'.[Russ09]

Consider the structure of the graph resulting from the construction process described in Chapter 3.5. First note that the graph is very sparse, which greatly reduces the branching factor and thus the number of nodes that need to be considered at each step of the search algorithm. Next, the graph consists of layers representing Poincare sections with a specified energy for each layer. While ballistic connections exist within each layer, only impulsive connections exist between layers. For each node, the majority of connections are impulsive connections. So if the search process can be steered towards the energy level of the goal, this would greatly reduce needless exploration and speed up the search process. Between any two layers, there will be a minimum connection cost. Consider the global minimum ΔV to switch between each pair of energy levels. For an increase in energy we should sum the minimum ΔV to move up between each pair of adjacent layers, as the velocity will be higher at each step, leading to a lower ΔV needed to increase the energy at each step. [Trum14a] This results in multiple maneuvers being cheaper than a single maneuver for the case of increasing energy. For a decrease it is calculated as one step. The adjustment is needed for consistency. These values may be stored in a small square matrix with the number of rows and columns equaling the number of layers in the graph. For any node, let $h_E(n)$ be this minimum ΔV magnitude between the energy layer containing node n and the layer containing the goal node. This heuristic steers the search process towards trajectories with the same energy as

the goal. In the case the start and goal are at different energies, this is necessary to avoid spending too much time searching for a ballistic transfer that does not exist. By construction this heuristic is admissible and consistent. Detailed results vary with application (see Chapter 10), however this simple heuristic results in at least a 10x speed up.

4.3.4 Heuristic Search Methods

Now that the heuristic functions have been discussed, heuristic based *informed search methods* can be placed in the same framework used to describe Uniform Cost Search and Dijkstra's Algorithm. Each step is the same as in the uninformed case above, except the step where the node with the lowest path cost on the frontier is selected. It is a feature of "Best First" methods – of which both Dijkstra's Algorithm and informed search methods are examples – that they use an evaluation function f to select the next, "best" node according to the function chosen. In the uninformed search strategies, $f(n)=g(n)$ where $g(n)$is the path cost to that node. Two such heuristic search strategies are *Greedy Best First* and *A* Search*. Both follow the same framework above, but with $f(n)=h_E(n)$ for Greedy Best and $f(n)=g(n)+h_E(n)$ for A* Search where $h_E(n)$ is the heuristic estimate.

A* Search, or variations thereof, is generally the preferred strategy in the case where we have access to the true cost. This is because it systematically moves towards the optimum solution when the heuristic is consistent and is also complete. The short proof of optimality is standard, [Russ09] but will be repeated here to illustrate the importance of admissibility and consistency. To show this, it is sufficient to show that the evaluation function value at the goal node on a suboptimal path is greater than a node n on the optimal path:

$$
\begin{aligned}
& f(G_{not\,optimal}) = g(G_{not\,optimal}) && \text{since } h(G_{n.o.})=0 \\
& > g(G_{opt}) && \text{since suboptimal} \\
& = g(G_{opt}) + h_E(G_{opt}) && \text{since } h(G_{opt})=0 \\
& = g(n) + c(n', G_{opt}) + h_E(G_{opt}) && \\
& \geq g(n) + h_E(n) = f(n) && \text{by consistency}
\end{aligned}
$$

As performance is heuristic and problem dependent, there is no general formula for the decrease in time complexity.

4.4 Initial Guess Generation using the Search Results

After the search algorithm has successfully been run, the result is a sequence of nodes called either a *path* or *itinerary*. Each node in the graph corresponds to a small region of state space in the domain of the spacecraft. Each transition corresponds either to an impulsive connection between regions, or a ballistic connection. For each ballistic connection in the graph an arc corresponding to this connection is stored in an arc database, and is indexed by the starting and ending region. Therefore for every ballistic transition in the itinerary, an arc may be selected. Depending on whether or not the boundary conditions are on the surface of section, an additional arc may have been used to establish an initial condition on the section as described in 4.2.1.

Consider a sequence of such arcs. The beginning of the first arc would match in position to the region containing the initial condition on the Poincare section. The next arc would also begin in a region with position matching the region in which the first arc ends. This continues until the final arc ends in a region which matches in position to the terminal condition. Thus this sequence of arcs has the following properties:

Discontinuity in position between endpoints of successive arcs or the endpoints and the boundary states is limited by the diameter of the regions in position space. As the partition is user defined, this may be set as 'small' as is necessary for the correction process, at the expense of more regions.

The difference in velocities, i.e. the ΔV magnitude, between successive arc endpoints or arc endpoints and boundary conditions has been approximated by the impulsive graph connections, since the arcs begin and end in the state space regions the link cost calculation is

based on. The actual vector impulse is simply the difference between the beginning of the next arc or terminal condition, and the end of the previous arc or initial condition. Therefore no data need be stored for impulsive connections aside from the cost.

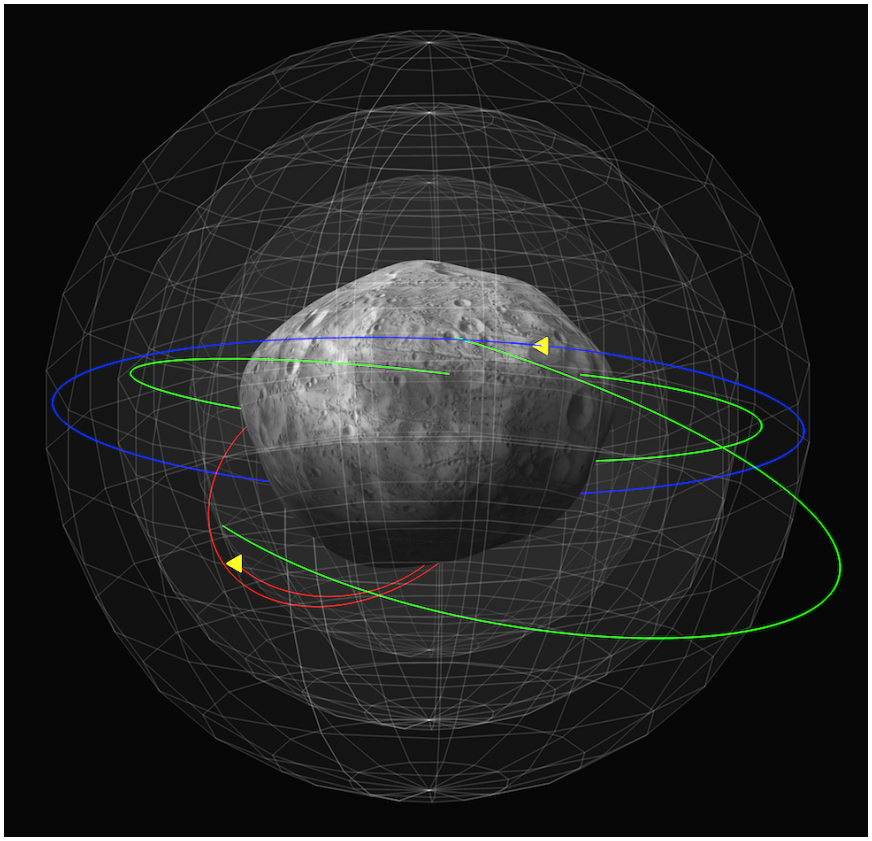


Figure 4.2: Transfer between boundary condition is approximated by a set of ballistic arcs. These arcs are passed on for differential correction and local optimization.

In Figure 4.2, the initial condition is marked with the small triangle on the trajectory in red, which left uncontrolled would lead to escape from a Halo orbit out of the Phobos system. The target condition is the triangle on a Distant Retrograde Orbit used as an abort target. The result of the search process performed on these conditions is used to select three ballistic arcs, shown in green. These arcs approximate the desired transfer and provide an initial guess in order to find it.

It is the states of initial conditions of these arcs and the corresponding coasting times, in

sequence, constitute the initial guess for the differential correction and optimization process described in the next several chapters.

4.5 Generating Additional Transfers with Graph Updates

After the correction and optimization procedures are complete, the costs may be higher than expected or a mission constraint may not be able to be satisfied with the search itinerary. If this is the case – or if there is simply time permitting to evaluate other potential options – it may be beneficial to create other transfer types. After the correction and optimization procedure is complete, the resulting ΔV at each impulse can be compared to the estimated ΔV held in the corresponding impulsive connection. The directed edge for which the true cost is the highest, or which is the greatest above the estimate, may either have its edge weight adjusted or the edge temporarily removed. Once this adjustment for costly transitions is made, the search process can be rerun. Unless the path is still graph-optimal with this adjustment, this will result in a new itinerary and a qualitatively different transfer. This process may be repeated as time allows.

4.6. Summary

To summarize the results of the above discussion, the proposed onboard portion of the transfer redesign algorithm consists of the following steps:

1. Translate boundary conditions into graph nodes. Periodic boundary conditions handled as above with multiple point to point searches performed, time permitting. For each point to point search:
2. Perform A* search on the directed graph which uses cost estimates as link weights.
3. Append periodic boundary condition elements if necessary to obtain complete itinerary.

4. Perform arc selection and differential correction process. Produces complete transfer. (Chapters 5-9)
5. Time permitting, modify graph by removing link associated with highest cost transition between regions in the transfer.
6. Return to Step 2 and iterate as desired.

Chapter 5

The Impulsive Transfer Optimization Problem and its Approximations

5.1 Introduction

The minimum fuel and minimum energy impulsive transfer problems are stated as nonlinear programs (NLP) whose variables are the initial conditions of ballistic arcs and the coasting time of each arc. Constraints include continuity in position, min/max coasting time per arc, min/max thrust per maneuver, and impact avoidance. Additional constraints whose derivations would impede the readability of this text, such as maximum thrust angle relative to the velocity, have been moved to Appendix B.

The goal of this chapter is to define the convex approximations that will be used in an iterative approximation process that is described in Chapter 6. This will be done in stages, linking sets of related problems. First, for any feasible reference trajectory, a set of corresponding NLPs more amenable to approximation and suitable for iterative processes is defined. It is then shown that a point of the original NLP satisfies the Karush-Kuhn-Tucker (KKT) conditions of that problem if

and only if it satisfies the KKT conditions of the iterative NLP defined using that point.

Next, as both of these NLPs are intractable in their current forms, natural Convex Problem (CP) approximations of the iterative NLPs are derived using submatrices of the State Transition Matrix (STM) of each arc. It is shown that the reference value is a KKT point of the CP precisely when it is a KKT point of the corresponding iterative NLP, and hence when a KKT point of the original problem. These convex problems are also shown to satisfy the form of the more restrictive class of Second Order Cone Problems (SOCP) for the minimum fuel problem, and a Quadratically Constrained Quadratic Program (QCQP) for the minimum energy problem. Lastly, a modification to these problems leading to better convergence properties but not affecting the KKT relationships with the NLPs is given. The use of these subproblems within an iterative process, and a comparison to other common iterative approaches is described in Chapter 6. Feasible major iterates and the global cost descent properties of this process are described in Chapter 7.

Before jumping directly into this sequence of problems and their CP/SOCP approximations, both Convex and Second Order Cone problems are described, including definitions, properties, and uses in related fields.

5.2 Convex Programming

5.2.1 Definition

A nonlinear program:

$$\textit{Find} \quad x \in \Omega \subseteq \mathbb{R}^r$$

$$\textit{that minimizes the cost function} \quad J(x)$$

$$\textit{subject to equality constraints} \quad h_j(x) = 0$$

$$\textit{and inequality constraints} \quad g_k(x) \leq 0$$

will be a convex program (CP) in the case that $\Omega \subseteq \mathbb{R}^r$ is a convex set, the cost function is convex, the equality constraints are linear (affine) and the inequality constraints are convex functions. These restrictions have the effect that the problem becomes one of minimizing a convex function over a convex feasible set.

5.2.2 Convergence and Complexity of Solutions

While complex in the details of implementation, Interior Point methods for CPs are based on a simple idea: the problem may be well approximated by a convex function combining the cost and barrier functions, thus guaranteeing Newton steps are always feasible. Such methods are guaranteed to converge to a solution,[Nest94, Boyd09] and due to convexity to attain the global minimum.

Ignoring any exploitable structure in the problem such as sparsity (which since our constraints act on pairs of patch points is not negligible), using Interior Point methods result in each step requiring at worst $O(max\{n^3, n^2 m, F\})$ operations in the limit, where n is the dimension of the problem, m the number of constraints, and F the cost of evaluating the first and second derivatives of the cost and constraint functions.[Boyd09, Peng09] For certain subclasses of CP to be discussed shortly, the number of such steps may also be bounded by the number of constraints, resulting in overall polynomial time convergence. It should be noted that the dimension of the impulsive transfer problem is not large. The approach taken here results in $7N$ variables, where N is the number of ballistic arcs, which number in the single digits.

5.2.3 Applications in Spaceflight

Within the spaceflight community, convex programming has been used within iterative methods, as will be the case for this application. One major difference is that these applications are optimal control problems, leading to a larger set of variables, but where dynamic nonlinearities and maintenance of continuity is less of a problem. A prominent example being tested on actual

test landers is the powered descent problem, where a sequence of SOCPs are solved.[Caso13, Acik13] That application is particularly interesting because theoretical results allowed the introduction of slack variables that create a convex domain containing all potential optimal solutions in the original, non-convex domain. In the orbital domain, sequences of convex problems have been used in the Model Predictive Control of satellite swarms. [Morg13] Another example is the use of iterative CPs to refine initial guesses for constrained satellite attitude optimal control problems.[Kjel12]

As embedded CP solvers become faster and more common,[Doma13,Chu13, Matt12] and as it is demonstrated that CP based methods have both convergence and feasible iterate properties desirable for onboard systems, they will likely expand within the spaceflight community.

5.2.4 Suitability for the Impulsive Transfer Problem

Convex problems were used in this work for multiple reasons. First is that they enjoy provable polynomial time convergence properties, particularly the SOCP/QCQP subclasses. Why not then Quadratic Programs (QP) which are faster still? One reason is that the cost as well as the inequality constraints of the iterative NLP of 5.5 naturally fall into the CP framework when using State Transition Matrix based approximations. A minor issue is that linear inequalities of QPs (descent of the Lagrangian is insufficient for feasible iterations, the constraints must be included in the subproblems) are less accurate than the quadratic and other conic constraints allowed in a CP/SOCP. More importantly, as will be seen in Chapter 7, linear inequalities on propagated variables in an approximate problem cannot be shown to hold in the NLP once necessary corrections are made to restore continuity, even with a line search. This then limits the use of Linear Programs as well.

5.3 Second Order Cone Problems

5.3.1 Definition

This is the form the CP approximation of the minimum fuel problem will take. A Second Order Cone Problem (SOCP) is defined as:

$$\textit{Find} \quad x \in \Omega \subseteq \mathbb{R}^r$$

$$\textit{that minimizes the linear cost function} \quad f^T x$$

$$\textit{subject to affine equality constraints} \quad F_j x = g_j$$

$$\textit{and inequality constraints} \quad \|A_k x + b_k\|_2 \leq c_k^T x + d_k$$

SOCPs are convex. This is because the cost is linear which means it is both convex and concave, the equality constraints are affine, and the inequality constraints require the affine function $x \rightarrow (A_k x + b_k, c_k^T x + d_k)$ to lie within the standard convex second order cone in $\mathbb{R}^{dim(b_k)+1}$, and the inverse image of a convex set under an affine function is convex.[Boyd09] The constraints are more general than those of QPs or QCQPs, since $c_k = 0$ results in a constraint equivalent to a quadratic constraint by simply squaring what is left, and if $A_k, b_k = 0$ the result is a linear inequality. The linear cost is not as prohibitive as it would seem at first glance, as various problems may be transformed into an SOCP with the introduction of extra variables, as seen for the minimum fuel problem in Section 5.7.3.

5.3.2 Convergence and Complexity of Solutions

Barrier / path following applied to SOCPs satisfy a self-concordance condition that bounds the number of Newton steps (themselves requiring $O(max\{n^3, n^2 m\})$ operations with no additional structure) required per outer iteration by a finite and computable number of steps. [Boyd09] Additionally, the number of such iterations may also be bounded by a finite and

computable number of steps. In particular, for a desired tolerance for the optimum value $\epsilon > 0$:

$$\#\text{ Newton steps} \leq \left(\sqrt{m}\log_2\left(\frac{m}{t^{(0)}\epsilon}\right)+1\right)\left(\frac{1}{2\gamma}+c_{nt}\right)$$

Where the constants γ, c_{nt} are functions of fixed backtracking parameters and Newton's method tolerances respectively, the constant $t^{(0)}$ determining the initial "strength" of the barrier function, and *m* the number of constraints.[Boyd09] Primal-Dual Interior-Point methods have been observed to have better performance in practice, and also have a finite number of steps on the order of $O\left(\sqrt{m}\log\left(\frac{m}{\epsilon}\right)\right)$. [Peng09] It should be noted that these oft-quoted results assume there is a strictly feasible point, i.e. Slater's condition holds. In Appendix A it is shown that this is the case for the convex subproblems defined in this chapter.

5.4 Quadratically Constrained Quadratic Problems

5.4.1 Definition

This is the form the minimum energy problem will take. A convex Quadratically Constraint Quadratic Problem (QCQP) is given by:

$$\textit{Find} \quad x \in \Omega \subseteq \mathbb{R}^r$$

$$\textit{that minimizes the cost function} \quad J(x) = \frac{1}{2}x^T P_0 x + q_0^T x + r_0, P_0 \geq 0$$

$$\textit{subject to affine equality constraints} \quad F_j x = g_j$$

$$\textit{and inequality constraints} \quad \frac{1}{2}x^T P_k x + q_k^T x + r_k \leq 0, P_k \geq 0$$

5.4.2 Convergence and Complexity of Solutions

The self-concordance condition that holds for SOCPs also holds for the more restrictive LP,

QP, and QCQP problems, thus the results for SOCPS holds for QCQPs using barrier methods. [Nemi01].

5.5 Original NLP Formulation for the Impulsive Transfer Problem

5.5.1 Variables and Notation

The goal is to minimize the total fuel cost for a set of *N* ballistic arcs, two given boundary states, *N-1* intermediate patch points, and *N+1* impulsive maneuvers. Let (x_0, v_0), … ,(x_i, v_i), … , (x_{N-1}, v_{N-1}), be the states at the beginning of each arc. Let $(\bar{x}_1, \bar{v}_1), \ldots, (\bar{x}_{i+1}, \bar{v}_{i+1}), \ldots, (\bar{x}_N, \bar{v}_N)$ be the states at the end of each ballistic arc. Let $t_1, \ldots, t_{i+1}, \ldots, t_N$ be the coasting times for each ballistic arc. These terms are related by the system flow φ_t in that $[\bar{x}_{i+1}\, \bar{v}_{i+1}]^T = \varphi_{t_{i+1}}([x_i\, v_i]^T)$. Finally, let $(x_{initial}, v_{initial})$, (x_{final}, v_{final}) be the given boundary conditions.

As the endpoints are determined by the system flow, take as variables the initial conditions of each arc and the coasting times. Thus the vector valued variable is:

$$X = \begin{bmatrix} x_0^T & v_0^T & t_1 & x_1^T & \cdots & x_{N-1}^T & v_{N-1}^T & t_N \end{bmatrix}^T.$$

For each arc there are seven variables: three for position, three for velocity, and one for coasting time. Thus there are 7**x***N* variables in total.

5.5.2 Cost Function

The fuel minimization problem is equivalent to minimizing the sum of the velocity change magnitudes,[Hugh03] with the added benefit that this is now a spacecraft mass independent problem. Similarly, the energy minimization problem is equivalent to minimizing the sum of the squares of the velocity magnitudes. The cost function for the original problem may be stated as:

$$\begin{aligned} J_{fuel} &= \|\Delta V_0\| + \ldots + \|\Delta V_N\| \\ &= \|v_0 - v_{initial}\| + \|v_1 - \bar{v}_1\| + \ldots + \|v_{N-1} - \bar{v}_{N-1}\| + \|v_{final} - \bar{v}_N\| \end{aligned}$$

or

$$J_{energy} = \|\Delta V_0\|^2 + \ldots + \|\Delta V_N\|^2$$
$$= \|v_0 - v_{initial}\|^2 + \|v_1 - \bar{v}_1\|^2 + \ldots + \|v_{N-1} - \bar{v}_{N-1}\|^2 + \|v_{final} - \bar{v}_N\|^2$$

In order to create a preference for shorter transfer times for transfers with similar ΔV expenditures, these cost functions may be easily modified. The total transfer time $t_{total} = t_1 + \ldots + t_N$ may be added to the above costs with a small weighting constant ϵ so that $J_{combined} = J_{fuel} + \epsilon t_{total}$ or $J_{combined} = J_{energy} + \epsilon t_{total}$.

5.5.3 Equality Constraint - Continuity

This is the central constraint of the whole problem. The endpoint of arc *i*, $\bar{x}_{i+1}$, is defined by the system flow $[\bar{x}_{i+1}\ \bar{v}_{i+1}]^T = \varphi_{t_{i+1}}([x_i\ v_i]^T)$. The continuity constraint is defined simply by requiring that the endpoint position of arc *i* equal the starting position of arc *i*+1. That is, $\bar{x}_{i+1} = x_{i+1}$.

5.5.4 Inequality Constraints

Min/Max coasting times:

Here the coasting time is required to be between some minimum and maximum amounts. This is simply stated as $t_{min} \le t_{i+1} \le t_{max}$.

Min/Max impulse per maneuver:

There is assumed to be a minimum and maximum impulse magnitude that the spacecraft's propulsion system may apply. While the existence of a maximum is intuitive, for navigation purposes as well as thruster operation restrictions there is a minimum amount of time and minimum amount of thrust that may be produced by the engines in the finite burn reality. This translates to a minimum impulse magnitude in an impulsive model. Thus this creates a constraint

at each patch point that $\Delta V_{min} \leq \|v_{i+1} - v_{i+1}^-\| \leq \Delta V_{max}$.

Impact Avoidance:

This is a requirement that closest approaches are required to be a minimum distance from the surface or center of a massive body. For this work we take the simple center of mass formulation and so for the position of any of the closest approaches, x_{p_j}, $\|x_{p_j} - x_{CM}\| \geq R_F$.

Note that there is no implication that the x_{p_j} are themselves variables, and this choice is intentional. Adding extra patch points wherever a periapsis occurred mid-arc but requiring no maneuver leads to an overly constrained problem during the recorrection process described in Chapter 7, as well as causing potential complications in coasting time constraints and indeed regularity. While having an arc end at an apsis condition may be a desired optional addition (see Appendix B), this is not necessary to avoid impact. By considering a periapsis altitude following an initial condition *wherever it may be* rather than forcing the inner product the position and velocity of an arc endpoint to be 0, an inequality with a natural convex approximation may replace an equality of a non-convex function approximated by a linear equality.

Maximum Thrust Angle Relative to the Velocity:

This is developed further in Appendix B, but handles the case where the impulse direction is limited to be within a maximum angle γ_{max} of the line of velocity, thus to lie within a cone with half angle γ_{max} with vertex and opening direction aligned with the incoming velocity vector. This can be stated as $\Delta V_{i+1} = v_{i+1} - v_{i+1}^- \in K_{\gamma max}(v_{i+1}^-)$. It is listed here as optional since it is dependent on the type of spacecraft, number and orientation of thrusters, and attitude control capabilities.

5.5.5 NLP Statement

Find variables

$$X=\begin{bmatrix} x_0^T & v_0^T & t_1 & x_1^T & \cdots & x_{N-1}^T & v_{N-1}^T & t_N \end{bmatrix}^T.$$

that minimize the cost function

$$\begin{aligned} J_{fuel} &= \|\Delta V_0\| + \ldots + \|\Delta V_N\| \\ &= \|v_0 - v_{initial}\| + \|v_1 - \bar{v}_1\| + \ldots + \|v_{N-1} - \bar{v}_{N-1}\| + \|v_{final} - \bar{v}_N\| \end{aligned}$$

or

$$\begin{aligned} J_{energy} &= \|\Delta V_0\|^2 + \ldots + \|\Delta V_N\|^2 \\ &= \|v_0 - v_{initial}\|^2 + \|v_1 - \bar{v}_1\|^2 + \ldots + \|v_{N-1} - \bar{v}_{N-1}\|^2 + \|v_{final} - \bar{v}_N\|^2 \end{aligned}$$

or

$$J_{fuel,time} = J_{fuel} + \epsilon(t_1 + \ldots + t_N),\ J_{energy,time} = J_{energy} + \epsilon(t_1 + \ldots + t_N)$$

subject to equality constraints

$$\text{Continuity: } \begin{bmatrix} x_0 \\ \bar{x}_1 \\ \vdots \\ \bar{x}_{i+1} \\ \vdots \\ \bar{x}_N \end{bmatrix} = \begin{bmatrix} x_{initial} \\ x_1 \\ \vdots \\ x_{i+1} \\ \vdots \\ x_{final} \end{bmatrix}$$

and inequality constraints

$$\text{Minimum coasting time per arc: } -t_{i+1} \le -t_{min},\ i = 0, \ldots, N-1$$

$$\text{Maximum coasting time per arc: } t_{i+1} \le t_{max},\ i = 0, \ldots, N-1$$

$$\text{Minimum impulse per maneuver: } \Delta V_{min} \le \|v_{i+1} - \bar{v}_{i+1}\|$$

$$\text{Maximum Impulse per maneuver: } \|v_{i+1} - \bar{v}_{i+1}\| \le \Delta V_{max}$$

$$\text{Impact Avoidance: } \|x_{p_j} - x_{CM}\| \ge R_F,\ \forall\, p_j$$

$$\text{Thrust Direction Limit (optional, Appendix B): } v_{i+1} - \bar{v}_{i+1} \in K_{\gamma max}(\bar{v}_{i+1})$$

5.6 The Iterative NLP

Here an alternative NLP is stated different from the one above. While a solution of this new NLP is not necessarily a solution of the original NLP, it has the property that if used in an iterative process, the set of fixed points that satisfy the Karush-Kuhn-Tucker (KKT) conditions of the iterative problem and the set of points that satisfy the Karush-Kuhn-Tucker conditions of the original NLP are the same. This allows the goal to shift from finding KKT points of the original to finding KKT fixed points of the iterative NLP.

5.6.1 Replacement Constraints

While many of the constraints are of a form that is naturally approximated by convex functions, two of these constraints are clearly non-convex: the minimum impulse magnitude and impact avoidance.

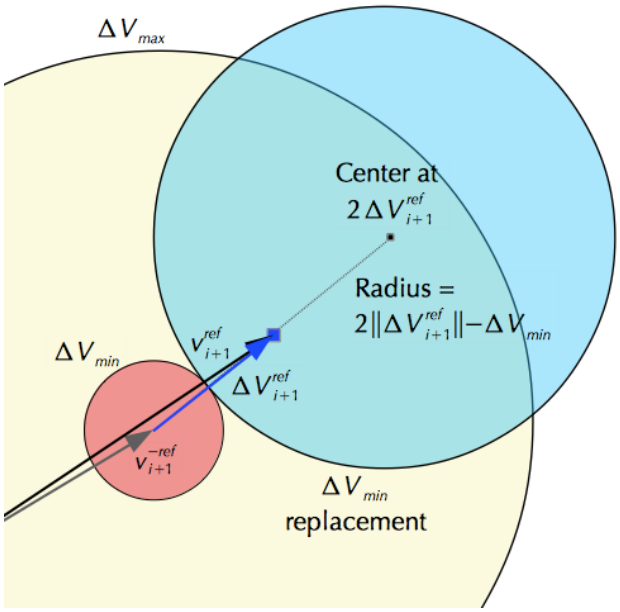


Figure 5.1: Comparison of the minimum impulse magnitude of the original NLP (red) and its replacement in the iterative NLP formulation (blue). The same idea is used to replace the impact avoidance.

For the impulse at a reference trajectory, ΔV_{i+1}^{ref}, consider a ball centered at the point $\zeta \Delta V_{i+1}^{ref}, \zeta > 1$ with radius $(\zeta \|\Delta V_{i+1}^{ref}\| - \Delta V_{min})$. Any potential ΔV_{i+1} within this ball – thus satisfying $\left\|\zeta \Delta V_{i+1}^{ref} - \Delta V_{i+1}\right\| \le \zeta \|\Delta V_{i+1}^{ref}\| - \Delta V_{min}$ – will satisfy $\|\Delta V_{i+1}\| \ge \Delta V_{min}$. With $\zeta = 1$, this is equivalent to using a ball centered at the reference maneuver with radius equal to the difference between the current magnitude and the minimum. This however, would limit the search space prohibitively as the magnitude approaching the minimum would reduce the ball radius to zero. As $\zeta \to \infty$, this approaches the use of a supporting hyperplane. Any $\zeta \ge 2$ ensures the radius of the ball is never less than the minimum impulse magnitude. In terms of the velocities themselves this may be written as $\left\|\zeta (v_{i+1}^{ref} - v_{i+1}^{-ref}) - (v_{i+1} - v_{i+1}^{-})\right\| \le \zeta \|v_{i+1}^{ref} - v_{i+1}^{-ref}\| - \Delta V_{min}$. Any v_i satisfying this will satisfy the original constraint, though the converse is not true.

In a similar manner for the impact avoidance, for $\eta > 1$ and a feasible reference, consider the constraints: $\left\|(\eta (x_{p_j}^{ref} - x_C) + x_C) - x_{p_j}\right\| \le \eta \|x_{p_j}^{ref} - x_C\| - R_F, \forall p_j$. The center of this ball is in the direction of $x_{p_i}^{ref} - x_C$ relative to the body center, but at a distance of $\eta \|x_{p_i}^{ref} - x_C\|$. By specifying the radius of this ball to be the distance from the center to the edge of the forbidden region, any point in this ball will satisfy the original constraint, though again the converse is not true.

5.6.2 Iterative NLP Statement

For a specified, feasible reference state X^{ref}, *find variables*

$$X = \begin{bmatrix} x_0^T & v_0^T & t_1 & x_1^T & \cdots & x_{N-1}^T & v_{N-1}^T & t_N \end{bmatrix}^T.$$

that minimize the cost function

$$\begin{aligned} J_{fuel} &= \|\Delta V_0\| + \ldots + \|\Delta V_N\| \\ &= \|v_0 - v_{initial}\| + \|v_1 - v_1^-\| + \ldots + \|v_{N-1} - v_{N-1}^-\| + \|v_{final} - v_N^-\| \end{aligned}$$

or

$$J_{energy} = \|\Delta V_0\|^2 + \ldots + \|\Delta V_N\|^2$$
$$= \|v_0 - v_{initial}\|^2 + \|v_1 - \bar{v}_1\|^2 + \ldots + \|v_{N-1} - \bar{v}_{N-1}\|^2 + \|v_{final} - \bar{v}_N\|^2$$

or

$$J_{fuel,time} = J_{fuel} + \epsilon(t_1 + \ldots + t_N), J_{energy,time} = J_{energy} + \epsilon(t_1 + \ldots + t_N)$$

subject to equality constraints

$$\text{Continuity: } \begin{bmatrix} x_0 \\ \bar{x}_1 \\ \vdots \\ \bar{x}_{i+1} \\ \vdots \\ \bar{x}_N \end{bmatrix} = \begin{bmatrix} x_{initial} \\ x_1 \\ \vdots \\ x_{i+1} \\ \vdots \\ x_{final} \end{bmatrix}$$

and inequality constraints

$$\text{Minimum coasting time: } -t_{i+1} \le -t_{min}, i = 0, \ldots, N-1$$

$$\text{Maximum coasting time: } t_{i+1} \le t_{max}, i = 0, \ldots, N-1$$

$$\text{Maximum Impulse per maneuver: } \|v_{i+1} - \bar{v}_{i+1}\| \le \Delta V_{max}$$

$$\text{Min impulse per maneuver: } \left\|\zeta\left(v_{i+1}^{ref} - v_{i+1}^{-ref}\right) - \left(v_{i+1} - \bar{v}_{i+1}\right)\right\| \le \zeta\|v_{i+1}^{ref} - v_{i+1}^{-ref}\| - \Delta V_{min}, i = 0, \ldots, N-1$$

$$\text{Impact avoidance: } \left\|\left(\eta\left(x_{p_j}^{ref} - x_C\right) + x_C\right) - x_{p_j}\right\| \le \eta\|x_{p_j}^{ref} - x_C\| - R_F, \forall p_j.$$

$$\text{Thrust Direction Limit (optional, Appendix B): } v_{i+1} - \bar{v}_{i+1} \in K_{\gamma max}(\bar{v}_{i+1})$$

5.6.3 Comparison of KKT Points

Here the components of the KKT conditions are compared between the original NLP and an iterative NLP defined in terms of a feasible point X^{ref}.

<u>Active constraints match:</u>

Consider the minimum impulse constraint function evaluated at X^{ref}:

$$\left\|\zeta\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)-\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)\right\|=(\zeta-1)\left\|v_{i+1}^{ref}-v_{i+1}^{-ref}\right\|=\zeta\left\|v_{i+1}^{ref}-v_{i+1}^{-ref}\right\|-\left\|v_{i+1}^{ref}-v_{i+1}^{-ref}\right\|$$

Clearly this can satisfy the constraint with equality, i.e.

$$\zeta\left\|v_{i+1}^{ref}-v_{i+1}^{-ref}\right\|-\left\|v_{i+1}^{ref}-v_{i+1}^{-ref}\right\|=\zeta\left\|v_{i+1}^{ref}-v_{i+1}^{-ref}\right\|-\Delta V_{min}$$

if and only if $\left\|v_{i+1}^{ref}-v_{i+1}^{-ref}\right\|=\Delta V_{min}$ and thus if and only if the original NLP is satisfied with equality.

Consider the impact avoidance constraint function evaluated at X^{ref}:

$$\left\|\left(\eta\left(x_{p_j}^{ref}-x_C\right)+x_C\right)-x_{p_j}^{ref}\right\|=\left\|(\eta-1)\,x_{p_j}^{ref}-(\eta-1)\,x_C\right\|=(\eta-1)\left\|x_{p_j}^{ref}-x_C\right\|=\eta\left\|x_{p_j}^{ref}-x_C\right\|-\left\|x_{p_j}^{ref}-x_C\right\|$$

This can satisfy the constraint with equality, i.e.

$$\eta\left\|x_{p_j}^{ref}-x_C\right\|-\left\|x_{p_j}^{ref}-x_C\right\|=\eta\|x_{p_j}^{ref}-x_C\|-R_F \quad \text{and so} \quad \left\|x_{p_j}^{ref}-x_C\right\|=R_F$$

if and only if the original NLP constraint is satisfied with equality.

All of the other constraints are identical. Thus the set of active constraints match.

<u>Constraint gradient directions match when active:</u>

First, the minimum impulse constraint. Begin with the constraint for the iterative NLP. $\left\|\zeta\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)-\left(v_{i+1}-v_{i+1}^{-}\right)\right\|=\left\|-\zeta\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)+\left(v_{i+1}-v_{i+1}^{-}\right)\right\|$ so the gradient of this constraint at the reference is:

$$\left[\frac{-\zeta\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)+\left(v_{i+1}-v_{i+1}^{-}\right)}{\left\|-\zeta\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)+\left(v_{i+1}-v_{i+1}^{-}\right)\right\|}\cdot\frac{\partial}{\partial X}\left(v_{i+1}-v_{i+1}^{-}\right)\right]_{X=X^{ref}}$$

$$=\frac{-\zeta\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)+\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)}{\left\|-\zeta\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)+\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)\right\|}\cdot\frac{\partial}{\partial X}\left(v_{i+1}-v_{i+1}^{-}\right)|_{X=X^{ref}}$$

$$=\frac{-\zeta\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)+\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)}{\left\|-\zeta\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)+\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)\right\|}\cdot\frac{\partial}{\partial X}\left(v_{i+1}-v_{i+1}^{-}\right)|_{X=X^{ref}}$$

$$=\frac{-(\zeta-1)\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)}{\left\|-(\zeta-1)\left(v_{i+1}^{ref}-v_{i+1}^{-ref}\right)\right\|}\cdot\frac{\partial}{\partial X}\left(v_{i+1}-v_{i+1}^{-}\right)|_{X=X^{ref}}$$

$$=-\frac{v_{i+1}^{ref}-v_{i+1}^{-ref}}{\left\|v_{i+1}^{ref}-v_{i+1}^{-ref}\right\|}\cdot\frac{\partial}{\partial X}\left(v_{i+1}-v_{i+1}^{-}\right)|_{X=X^{ref}}$$

Meanwhile, for the original NLP the gradient of $-\|v_{i+1}-v_{i+1}^{-}\|$ at the reference is indeed the same:

$$-\frac{v_{i+1}^{ref}-v_{i+1}^{-ref}}{\left\|v_{i+1}^{ref}-v_{i+1}^{-ref}\right\|}\cdot\frac{\partial}{\partial X}\left(v_{i+1}-v_{i+1}^{-}\right)|_{X=X^{ref}}$$

Next, the impact avoidance constraint. Begin with the constraint for the iterative NLP. $\left\|\eta\left(x_{p_j}^{ref}-x_C\right)+x_C-x_{p_j}\right\|=\left\|-\eta\left(x_{p_j}^{ref}-x_C\right)-x_C+x_{p_j}\right\|$ so the constraint gradient at the reference is:

$$\left[\frac{-\eta\left(x_{p_j}^{ref}-x_C\right)-x_C+x_{p_j}}{\left\|-\eta\left(x_{p_j}^{ref}-x_C\right)-x_C+x_{p_j}\right\|}\cdot\frac{\partial}{\partial X}\left(x_{p_j}\right)\right]_{X=X^{ref}}$$

$$=\frac{-\eta\left(x_{p_j}^{ref}-x_C\right)-x_C+x_{p_j}^{ref}}{\left\|-\eta\left(x_{p_j}^{ref}-x_C\right)-x_C+x_{p_j}^{ref}\right\|}\cdot\frac{\partial}{\partial X}\left(x_{p_j}\right)|_{X=X^{ref}}$$

$$=\frac{-(\eta-1)\left(x_{p_j}^{ref}-x_C\right)}{\left\|-(\eta-1)\left(x_{p_j}^{ref}-x_C\right)\right\|}\cdot\frac{\partial}{\partial X}\left(x_{p_j}\right)|_{X=X^{ref}}$$

$$=-\frac{x_{p_j}^{ref}-x_C}{\left\|x_{p_j}^{ref}-x_C\right\|}\cdot\frac{\partial}{\partial X}\left(x_{p_j}\right)|_{X=X^{ref}}$$

Meanwhile, for the original NLP the gradient of $-\|x_{p_j}-x_{CM}\|$ at the reference is indeed the same:

$$-\frac{x_{p_j}^{ref}-x_C}{\left\|x_{p_j}^{ref}-x_C\right\|}\cdot\frac{\partial}{\partial X}\left(x_{p_j}\right)|_{X=X^{ref}}$$

Comparison of KKT Conditions:

To summarize the above, the same set of constraints is active between the original NLP and the iterative NLP defined in terms of X^{ref}. Among those that are active, the gradients are identical. If the KKT conditions for the original NLP then there exist vectors of multipliers λ,μ such that

$$\nabla J(X^{ref})+\lambda^T\nabla h(X^{ref})+\mu^T\nabla g(X^{ref})=0$$
$$h(X^{ref})=0$$
$$g(X^{ref})\le 0$$
$$\mu\ge 0$$
$$\mu^T g(X^{ref})=0$$

X^{ref} is feasible in either formulation and thus $h(X^{ref})=0, g(X^{ref})\le 0$ if and only if $h_{iter}(X^{ref})=0, g_{iter}(X^{ref})\le 0$. At X^{ref} the same set of constraints are active and so $g_k(X^{ref})=0$ if and only if $g_{iter,k}(X^{ref})=0$ and thus $\mu^T g(X^{ref})=0$ if and only if $\mu^T g_{iter}(X^{ref})=0$. Lastly:

$$\nabla J(X^{ref})+\lambda^T\nabla h(X^{ref})+\mu^T\nabla g(X^{ref})=\nabla J_{iter}(X^{ref})+\lambda^T\nabla h_{iter}(X^{ref})+\mu^T\nabla g_{iter}(X^{ref})$$

since $\mu_k=0$ for inactive constraints and the gradients match for the active constraints. Thus:

$$\nabla J_{iter}(X^{ref})+\lambda^T\nabla h_{iter}(X^{ref})+\mu^T\nabla g_{iter}(X^{ref})=0$$
$$h_{iter}(X^{ref})=0$$
$$g_{iter}(X^{ref})\le 0$$
$$\mu\ge 0$$
$$\mu^T g_{iter}(X^{ref})=0$$

for the same multipliers as the original NLP. The implications are all if and only if statements, and so X^{ref} satisfies the KKT conditions for the iterative NLP defined in terms of X^{ref} if and only if satisfies the KKT conditions of the original NLP.

5.7 Convex Problem Approximation

5.7.1 Variables and Notation

In order to state the approximation of the Iterative NLP as a convex program (CP), a change of variables for the local CP approximation is needed. Let $x_i^{ref}, v_i^{ref}, t_{i+1}^{ref}$ define the initial state and coasting time of the arc beginning at patch point *i* of the previous (or initial) feasible trajectory in the iteration process. Let $\tilde{x}_i, \tilde{v}_i, \tilde{t}_{i+1}$ be variations from these references variables so

that $x_i = \tilde{x}_i + x_i^{ref}$, $v_i = \tilde{v}_i + v_i^{ref}$, $t_{i+1} = \tilde{t}_{i+1} + t_{i+1}^{ref}$. The convex approximation will solve for the change from previous iteration rather than the resulting variables themselves.

For each *reference* arc, let $\Phi_{i+1,i}$ be the 6x6 state transition matrix (STM) corresponding to the flow from starting to ending points defined by $[\bar{x}_{i+1}\ \bar{v}_{i+1}]^T = \varphi_{t_{i+1}}([x_i\ v_i]^T)$. Partition the STM into 3x3 submatrices so that:

$$\Phi_{i+1,i} = \begin{bmatrix} A_{i+1,i} & B_{i+1,i} \\ C_{i+1,i} & D_{i+1,i} \end{bmatrix}$$

From the properties of STMs, [Marc07] to first order, the result of propagating a small variation near the reference is given by:

$$\begin{bmatrix} \tilde{\bar{x}}_{i+1} \\ \tilde{\bar{v}}_{i+1} \end{bmatrix} \approx \begin{bmatrix} A_{i+1,i} & B_{i+1,i} \\ C_{i+1,i} & D_{i+1,i} \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \end{bmatrix} + \begin{bmatrix} v_{i+1}^{-ref} \\ a_{i+1}^{-ref} \end{bmatrix} \tilde{t}_{i+1}$$

where a_{i+1}^{-ref} is the acceleration calculated at $(x_{i+1}^{-ref}, v_{i+1}^{-ref})$. The constraints below use this approximation for the state change at the arc endpoint in terms of these small variation variables

$$\tilde{X} = \begin{bmatrix} \tilde{x}_0^T & \tilde{v}_0^T & \tilde{t}_1 & \tilde{x}_1^T & \cdots & \tilde{x}_{N-1}^T & \tilde{v}_{N-1}^T & \tilde{t}_N \end{bmatrix}^T.$$

5.7.2 Convex Cost Function

Recall the cost function for the NLP is given by

$$\begin{aligned} J_{fuel} &= \|\Delta V_0\| + \ldots + \|\Delta V_N\| \\ &= \|v_0 - v_{initial}\| + \|v_1 - \bar{v}_1\| + \ldots + \|v_{N-1} - \bar{v}_{N-1}\| + \|v_{final} - \bar{v}_N\| \end{aligned}$$

or

$$\begin{aligned} J_{energy} &= \|\Delta V_0\|^2 + \ldots + \|\Delta V_N\|^2 \\ &= \|v_0 - v_{initial}\|^2 + \|v_1 - \bar{v}_1\|^2 + \ldots + \|v_{N-1} - \bar{v}_{N-1}\|^2 + \|v_{final} - \bar{v}_N\|^2 \end{aligned}$$

or

$$J_{fuel,time} = J_{fuel} + \epsilon(t_1 + \ldots + t_N),\ J_{energy,time} = J_{energy} + \epsilon(t_1 + \ldots + t_N)$$

Each term must now be expressed in terms of the new variables $\tilde{x}_i, \tilde{v}_i, \tilde{t}_i$. The first term is a

special case as unlike the others this is a simple subtraction of vectors at the same position, without an integration:

$$\|v_0 - v_{initial}\| = \|v_0^{ref} + \tilde{v}_0 - v_{initial}\| = \|\tilde{v}_0 - (v_{initial} - v_0^{ref})\|$$

The other terms are of the form $\|v_{i+1} - v_{i+1}^-\|$. From the definition of the new variables, this means that $v_{i+1} = \tilde{v}_{i+1} + v_{i+1}^{ref}$. Let $\tilde{v}_{i+1}^-$ be such that $v_{i+1}^- = \tilde{v}_{i+1}^- + v_{i+1}^{-ref}$. To approximate $\tilde{v}_{i+1}^-$ in terms of the new variables, we will use the STM based approximation:

$$\begin{bmatrix} \tilde{x}_{i+1}^- \\ \tilde{v}_{i+1}^- \end{bmatrix} \approx \begin{bmatrix} A_{i+1,i} & B_{i+1,i} \\ C_{i+1,i} & D_{i+1,i} \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \end{bmatrix} + \begin{bmatrix} v_{i+1}^{-\,ref} \\ a_{i+1}^{-\,ref} \end{bmatrix} \tilde{t}_{i+1} \quad .$$

This gives:

$$v_{i+1} - v_{i+1}^- \approx v_{i+1}^{ref} + \tilde{v}_{i+1} - v_{i+1}^{-ref} - \begin{bmatrix} C_{i+1,i} & D_{i+1,i} \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \end{bmatrix} - a_{i+1}^{-\,ref} \tilde{t}_{i+1}$$

$$= \begin{bmatrix} -C_{i+1,i} & -D_{i+1,i} & -a_{i+1}^{-\,ref} & I_{3x3} \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \\ \tilde{t}_{i+1} \\ \tilde{v}_{i+1} \end{bmatrix} - (v_{i+1}^{-ref} - v_{i+1}^{ref})$$

We then have the cost function:

$$\tilde{J}_{fuel}(\tilde{X}) = \|F_0 \tilde{X} - e_0\| + \|F_1 \tilde{X} - e_1\| + \ldots + \|F_N \tilde{X} - e_N\|$$

or

$$\tilde{J}_{energy}(\tilde{X}) = \|F_0 \tilde{X} - e_0\|^2 + \|F_1 \tilde{X} - e_1\|^2 + \ldots + \|F_N \tilde{X} - e_N\|^2$$

where for the fixed boundary condition velocities

$$F_0 = \begin{bmatrix} 0_{3x3} & I_{3x3} & 0_{3x3} & \cdots & 0_{3x3} \end{bmatrix}, e_0 = v_{initial} - v_0^{ref}$$

$$F_N = \begin{bmatrix} 0_{3x3} & \cdots & 0_{3x3} & -C_{i+1,i} & -D_{i+1,i} & -a_{i+1}^{-\,ref} \end{bmatrix}, e_N = v_{N+1}^{-ref} - v_{final}$$

and for i+1=1, …, N

$$F_{i+1} = \begin{bmatrix} 0_{3x3} & \cdots & -C_{i+1,i} & -D_{i+1,i} & -a_{i+1}^{-\,ref} & 0_{3x3} & I_{3x3} & \cdots & 0_{3x3} \end{bmatrix}, e_{i+1} = v_{i+1}^{-ref} - v_{i+1}^{ref}$$

Each $F_i X - e_i$ is an affine transformation. Note also that any norm in $\mathbb{R}^n$ is convex. [Boyd09] Thus $\|F_i X - e_i\|$ is a composition of an affine function into a convex function; such a composition is always convex.[Boyd09] The minimum fuel cost function is then a nonnegative weighted sum (all weights being equal to +1) of convex functions, which preserves convexity. Note that neither the original cost nor this approximation is quadratic, but a sum-of-norms fits into the more general convex setting. For the minimum energy formulation, each $\|F_i X - e_i\|^2 = (F_i X - e_i)^T (F_i X - e_i)$ has Hessian $F_i^T F_i \geq 0$, and so is convex by the second order conditions on convexity. The total cost is then also a sum of convex functions, hence convex.

If total transfer time is a component of the cost, then:

$$\begin{aligned} \epsilon t_{total} &= \epsilon(t_1 + \ldots + t_N) \\ &= \epsilon(\tilde{t}_1 + \ldots + \tilde{t}_N) + \epsilon(t_1^{ref} + \ldots + t_N^{ref}) \\ &= F_t \tilde{X} - e_t \\ &= \left\| F_t \tilde{X} - e_t \right\| \end{aligned}$$

where

$$F_t = \begin{bmatrix} 0_{1x6} & \epsilon & \ldots & 0_{1x6} & \epsilon \end{bmatrix}, e_t = -\epsilon(t_1^{ref} + \ldots + t_N^{ref})$$

resulting in cost functions:

$$\tilde{J}_{fuel,time}(\tilde{X}) = \|F_0 \tilde{X} - e_0\| + \|F_1 \tilde{X} - e_1\| + \ldots + \|F_N \tilde{X} - e_N\| + \left\| F_t \tilde{X} - e_t \right\|$$

or

$$\tilde{J}_{energy,time}(\tilde{X}) = \|F_0 \tilde{X} - e_0\|^2 + \|F_1 \tilde{X} - e_1\|^2 + \ldots + \|F_N \tilde{X} - e_N\|^2 + F_t \tilde{X} - e_t$$

For the former, another composition of an affine function into a norm is added, so the result is still convex. For the latter, an affine function is added, which also preserves convexity.

5.7.3 Epigraph Transformation of the Minimum Fuel Cost

An addition step is required if the desired form of the approximation is that of a SOCP. These require a linear cost function, but allows inequalities defined by second order cones. By

introducing new variables $\sigma_0,\ldots,\sigma_N$, we can convert the approximate CP to an SOCP by replacing the cost function:

$$\tilde{J}(\tilde{X})=\|F_0\tilde{X}-e_0\|+\|F_1\tilde{X}-e_1\|+\ldots+\|F_N\tilde{X}-e_N\|$$

with

$$\tilde{J}_{SOCP}(\tilde{X},\sigma_0,\ldots,\sigma_N)=\sigma_0+\ldots+\sigma_N$$

and adding additional constraints

$$\|F_iX-e_i\|<\sigma_i,\ i=0,\ldots,N$$

which are all second order cones with c=0. This creates an equivalent problem and is known as an epigraph transformation.[Boyd09]

If total transfer time is included, then this becomes

$$\tilde{J}_{SOCP}(\tilde{X},\sigma_0,\ldots,\sigma_N,\sigma_t)=\sigma_0+\ldots+\sigma_N+\sigma_t$$

and adding additional constraints

$$\|F_iX-e_i\|<\sigma_i,\ i=0,\ldots,N$$
$$\|F_tX-e_t\|<\sigma_t$$

5.7.4 Linear/Affine Equality Constraints

Continuity:

In terms of the small variation variables and STM approximations, both sides of this equality constraint are approximated as follows for a feasible reference:

$$x_{i+1}^{-ref}+\begin{bmatrix}A_{i+1,i} & B_{i+1,i}\end{bmatrix}\begin{bmatrix}\tilde{x}_i\\ \tilde{v}_i\end{bmatrix}+v_{i+1}^{-ref}\tilde{t}_{i+1}\approx x_{i+1}^{-ref}+\tilde{x}_{i+1}^{-}=x_{i+1}^{-}=x_{i+1}=x_{i+1}^{ref}+\tilde{x}_{i+1}$$

By rearranging variables and recalling the continuity of the corrected reference trajectories for patch points we have:

$$\begin{bmatrix} A_{i+1,i} & B_{i+1,i} & v_{i+1}^{-ref} - I_{3x3} \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \\ \tilde{t}_{i+1} \\ \tilde{x}_{i+1} \end{bmatrix} = 0.$$

Two special cases remain to keep the initial and final positions fixed. For the initial point, this is done by simply adding the additional constraint $\tilde{x}_0 = 0$. For the final patch point it is necessary that changes result in no variation in the specified final position. Since the final position and velocity are not controls of the system, there is no need to include them as variables. However, the equivalent of a $\tilde{x}_N = 0$ constraint may be expressed as

$$\begin{bmatrix} A_{N,N-1} & B_{N,N-1} & v_N^{- ref} \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \\ \tilde{t}_{i+1} \end{bmatrix} = 0$$

Combining these cases results in the final, banded linear equality constraint:

$$\begin{bmatrix} I_{3x3} & 0_{3x3} & & & \cdots & & & 0_{3x3} \\ A_{1,0} & B_{1,0} & v_1^{- ref} & -I_{3x3} & 0_{3x3} & \cdots & & 0_{3x3} \\ \vdots & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ 0_{3x3} & \cdots & A_{i+1,i} & B_{i+1,i} & v_{i+1}^{- ref} & -I_{3x3} & \cdots & 0_{3x3} \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0_{3x3} & & \cdots & & 0_{3x3} & A_{N,N-1} & B_{N,N-1} & v_N^{-ref} \end{bmatrix} \begin{bmatrix} \tilde{x}_0 \\ \tilde{v}_0 \\ \tilde{t}_1 \\ \tilde{x}_1 \\ \vdots \\ \tilde{x}_{N-1} \\ \tilde{v}_{N-1} \\ \tilde{t}_N \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Each continuity constraint consists of three rows, and there is one such constraint per arc. Combined with the constraint on the position of the initial condition, this results in 3**x**(*N*+1) rows.

Figure X gives a visual representation of how this constraint in the variation variables related to continuity in the original problem: the predicted change to the endpoint (black) of arc i from varying the initial conditions from the reference (teal) and coasting time must be equal to the variation to the position (red) of the initial condition of arc i+1.

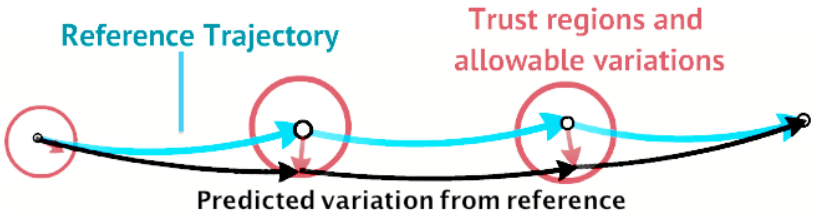


Figure 5.2: Requiring position variation of arc *i+1* to equal predicted endpoint effect of varying arc i.

5.7.5 Convex Inequality Constraints

Min/Max coasting times:

First, begin with a form as a second order cone constraint. The maximum coasting time constraint in the original problem $t_{i+1} \le t_{max}$ can be rewritten without approximation as $t_{i+1}^{ref} + \tilde{t}_{i+1} \le t_{max}$ and so $\tilde{t}_{i+1} \le t_{max} - t_{i+1}^{ref}$. For the minimum, the original constraint is $-t_{i+1} \le -t_{min}$ and so in terms of the new variables is $-\tilde{t}_{i+1} \le -t_{min} + t_{i+1}^{ref}$. These are affine inequalities, and so are not only convex but are also instances of second order cone constraints.

Next, as a quadratic inequality, reframe $t_{min} \le t_{i+1} \le t_{max}$ as $\left| t_{i+1} - \frac{t_{min} + t_{max}}{2} \right| \le \frac{t_{max} - t_{min}}{2}$, or rather $\left(t_{i+1} - \frac{t_{min} + t_{max}}{2} \right)^2 \le \left(\frac{t_{max} - t_{min}}{2} \right)^2$. Then with $t_{i+1} = t_{i+1}^{ref} + \tilde{t}_{i+1}$ the result is a quadratic inequality constraint $\left(\tilde{t}_{i+1} + \left(t_{i+1}^{ref} - \frac{t_{min} + t_{max}}{2} \right) \right)^2 \le \left(\frac{t_{max} - t_{min}}{2} \right)^2$.

Maximum Thrust Magnitude:

Using the small variation variables and STM approximation of the dynamics, this is approximated as:

$$\| v_i - \bar{v}_i \| \approx \| F_i \tilde{X} - e_i \| \le \Delta V_{max}$$

where F_i, e_i are defined in the description of the cost function. Formulated in this way we see that this results in a set of Second Order Cone constraints. An equivalent quadratic constraint comes from simply squaring this to get $\|F_i \tilde{X} - e_i\|^2 \le \Delta V_{max}^2$.

Minimum Thrust Magnitude:

In terms of the STM based approximations, this constraint may be derived as:

$$\left\|\zeta \Delta V_{i+1}^{ref} - \Delta V_{i+1}\right\| \le \zeta \|\Delta V_{i+1}^{ref}\| - \Delta V_{min}$$

$$\left\|\zeta (v_{i+1}^{ref} - v_{i+1}^{-ref}) - (v_{i+1} - v_{i+1}^{-})\right\| \le \zeta \|v_{i+1}^{ref} - v_{i+1}^{-ref}\| - \Delta V_{min}$$

$$\left\|\zeta (-e_{i+1}) - (F_{i+1}\tilde{X} - e_{i+1})\right\| \le \zeta \|-e_{i+1}\| - \Delta V_{min}$$

$$\left\|(1-\zeta) e_{i+1} - F_{i+1}\tilde{X}\right\| \le \zeta \|e_{i+1}\| - \Delta V_{min}$$

$$\left\|F_{i+1}\tilde{X} - (1-\zeta) e_{i+1}\right\| \le \zeta \|e_{i+1}\| - \Delta V_{min}$$

where F_i, e_i are defined in the description of the cost function. Formulated in this way we see that this results in a set of Second Order Cone constraints. An equivalent quadratic form is

$$\left\|F_{i+1}\tilde{X} - (1-\zeta) e_{i+1}\right\|^2 \le \left(\zeta \|e_{i+1}\| - \Delta V_{min}\right)^2.$$

Impact Avoidance:

Let $t_i^p < t_{i+1}$ be the time at which the arc beginning at x_i, v_i attains periapsis. Then using the notation $[x_i(t_p)\ \ v_i(t_p)]^T = \varphi_{t_i^p}([x_i\, v_i]^T)$ for the state at periapsis, the iterative NLP constraint is

$$\left\|(\eta(x_i^{ref}(t_i^{p,ref}) - x_C) + x_C) - x_i(t_i^p)\right\| \le \eta \|x_i^{ref}(t_i^{p,ref}) - x_C\| - R_F.$$

To first order, $x_i(t_i^p) = x_i^{ref}(t_i^{p,ref}) + A_i^p \tilde{x}_i + B_i^p \tilde{v}_i + v_i^{ref}(t_i^{p,ref})\tilde{t}_i^p$, where here A_i^p, B_i^p are the submatrices of the STM of the trajectory with initial condition (x_i^{ref}, v_i^{ref}) integrated forward only to $t_i^{p,ref}$. t_i^p is not itself a variable, so we will need a first order expression for it in terms of the

actual variables. That is, find T_i such that $t_i^p = t_i^{p,ref} + \tilde{t}_i^p \approx t_i^{p,ref} + T_i \tilde{X}$ so that $t_p \approx T_i \tilde{X}$ and thus

$$x_i(t_p) = x_i^{ref}(t_i^{p,ref}) + A_i^p \tilde{x}_i + B_i^p \tilde{v}_i + v_i^{ref}(t_i^{p,ref}) T_i \tilde{X}$$

Consider the function $F : \mathbb{R}^{6+1} \to \mathbb{R}, F(x(0), v(0), t) = \langle x(t) - x_C, v(t) \rangle$. At the reference, $F(x_i^{ref}, v_i^{ref}, t_p^{ref}) = 0$ since a closest approach satisfies the apsis condition. Checking the derivative with respect to the time variable:

$$\frac{\partial F}{\partial t} = \frac{\partial}{\partial t} \langle x(t) - x_C, v(t) \rangle = \frac{\partial}{\partial t} ((x(t) - x_C)^T v(t))$$

$$= \left(\frac{\partial}{\partial t} (x(t) - x_C) \right) v(t) + (x(t) - x_C) \left(\frac{\partial}{\partial t} v(t) \right)$$

$$= v^T(t) v(t) + (x(t) - x_C) a(t)$$

If $\Xi_i^{ref} = v_i^T(t_i^{p,ref}) v_i(t_i^{p,ref}) + (x_i(t_i^{p,ref}) - x_C)^T a_i(t_i^{p,ref}) = 0$ then the apsis condition has no first order change in t. If $\Xi_i^{ref} = v_i^T(t_i^{p,ref}) v_i(t_i^{p,ref}) + (x_i(t_i^{p,ref}) - x_C)^T a_i(t_i^{p,ref}) \neq 0$ then by Implicit Function Theorem there exists function $t_i^p(x_i, v_i)$ defined on an open set containing $t_i^{p,ref}$ s.t. $F(x_i, v_i, t_i^p(x_i, v_i)) = 0$. Additionally, the derivative results of the theorem state that with:

$$\frac{\partial t_i^p}{\partial x_i} = -\left(\frac{\partial F}{\partial t}(x_i^{ref}, v_i^{ref}, t_i^{p,ref}) \right)^{-1} \frac{\partial F}{\partial x_i}(x_i^{ref}, v_i^{ref}, t_i^{p,ref}) = -\frac{1}{\xi_i^{ref}} \frac{\partial F}{\partial x_i}(x_i^{ref}, v_i^{ref}, t_i^{p,ref})$$

$$\frac{\partial t_i^p}{\partial v_i} = -\left(\frac{\partial F}{\partial t}(x_i^{ref}, v_i^{ref}, t_i^{p,ref}) \right)^{-1} \frac{\partial F}{\partial v_i}(x_i^{ref}, v_i^{ref}, t_i^{p,ref}) = -\frac{1}{\xi_i^{ref}} \frac{\partial F}{\partial v_i}(x_i^{ref}, v_i^{ref}, t_i^{p,ref})$$

Since

$$\frac{\partial}{\partial x_i} \left((x(t) - x_C)^T v(t) \right) = \frac{\partial}{\partial x_i} \left((x(t) - x_C)^T \right) v(t) + (x(t) - x_C)^T \frac{\partial}{\partial x_i} (v(t))$$

$$= (v_i^{ref}(t_i^{p,ref}))^T A_i^p + (x_i^{ref}(t_i^{p,ref}) - x_C)^T C_i^p$$

$$\frac{\partial}{\partial v_i} \left((x(t) - x_C)^T v(t) \right) = \frac{\partial}{\partial v_i} \left((x(t) - x_C)^T \right) v(t) + (x(t) - x_C)^T \frac{\partial}{\partial v_i} (v(t))$$

$$= (v_i^{ref}(t_i^{p,ref}))^T B_i^p + (x_i^{ref}(t_i^{p,ref}) - x_C)^T D_i^p$$

we now have a first order approximation of the time to periapsis:

$$t_p \approx t_p^{ref},\ T_i = 0 \text{ if } \Xi_i^{ref} = 0,$$

$$t_p \approx t_p^{ref} + T_i \tilde{X} \quad \text{otherwise, with}$$

$$T_i = \frac{1}{\Xi_i^{ref}}\left[\ldots \quad 0 \quad (v_i^{ref}(t_i^{p,ref}))^T A_i^p + (x_i^{ref}(t_i^{p,ref}) - x_C)^T C_i^p \quad (v_i^{ref}(t_i^{p,ref}))^T B_i^p + (x_i^{ref}(t_i^{p,ref}) - x_C)^T D_i^p \quad 0 \quad \ldots\right]$$

where the nonzero terms multiply $\tilde{x}_i, \tilde{v}_i$.

Now all the terms are defined in the first order approximation:

$$x_i(t_p) = x_i^{ref}(t_i^{p,ref}) + A_i^p \tilde{x}_i + B_i^p \tilde{v}_i + v_i^{ref}(t_i^{p,ref}) T_i \tilde{X}$$

The iterative NLP constraint $\left\|(\eta(x_i^{ref}(t_i^{p,ref}) - x_C) + x_C) - x_i(t_i^p)\right\| \le \eta \|x_i^{ref}(t_i^{p,ref}) - x_C\| - R_F$ is then approximated by the Second Order Cone:

$$\left\|(\eta(x_i^{ref}(t_i^{p,ref}) - x_C) + x_C) - (x_i^{ref}(t_i^{p,ref}) + A_i^p \tilde{x}_i + B_i^p \tilde{v}_i + v_i^{ref}(t_i^{p,ref}) T_i \tilde{X})\right\|$$

$$= \left\|(\eta - 1)(x_i^{ref}(t_i^{p,ref}) - x_C) - (A_i^p \tilde{x}_i + B_i^p \tilde{v}_i + v_i^{ref}(t_i^{p,ref}) T_i \tilde{X})\right\| \le \eta \|x_i^{ref}(t_i^{p,ref}) - x_C\| - R_F$$

With no variables on the right side of the equation, as with the other constraints we may square both sides to get a quadratic constraint.

Trust Regions:

The accuracy of the state transition matrix based approximation depends on the variations being contained within a "small enough" domain, referred to as a Trust Region. This may be expressed as a set of inequality constraints on norms of the position, velocity, and time variations, which are in fact very simple Second Order Cone constraints:

$$\|\tilde{x}_i\| \le \xi_i,\ \|\tilde{v}_i\| \le \upsilon_i,\ |\tilde{t}_i| \le \tau_i$$

or quadratic constraints

$$\|\tilde{x}_i\|^2 \le \xi_i^2,\ \|\tilde{v}_i\|^2 \le \upsilon_i^2,\ \tilde{t}_i^2 \le \tau_i^2$$

On one hand, the $\xi_i, \upsilon_i, \tau_i$ values should be tied to the estimate accuracy of previous iterations and step size. However, for multibody systems with unstable dynamics, tying the trust region size to the instability of the reference trajectory is helpful when continuity is desired. As a result, take the values to be of the form of a tunable constant divided by the operator 2-norm of the STM submatrices for ξ_i, υ_i and the norm of the velocity for τ_i. To first order, this will keep the arc starting point variation small enough that it will keep arc endpoints from varying more than the tunable parameter. More formally:

$$\begin{aligned}\xi_i &= \xi / \|A_{i+1,i}\|_{2,} \\ \upsilon_i &= \upsilon / \|B_{i+1,i}\|_{2,} \\ \tau_i &= \tau / \|v_{i+1}^{-ref}\|_2\end{aligned}$$

5.7.6 General Convex Program

Find variables:

$$\tilde{X} = \begin{bmatrix} \tilde{x}_0^T & \tilde{v}_0^T & \tilde{t}_1 & \tilde{x}_1^T & \cdots & \tilde{x}_{N-1}^T & \tilde{v}_{N-1}^T & \tilde{t}_N \end{bmatrix}^T$$

That minimize the cost function:

$$\tilde{J}_{fuel}(\tilde{X}) = \|F_0\tilde{X} - e_0\| + \|F_1\tilde{X} - e_1\| + \ldots + \|F_N\tilde{X} - e_N\|$$

or

$$\tilde{J}_{energy}(\tilde{X}) = \|F_0\tilde{X} - e_0\|^2 + \|F_1\tilde{X} - e_1\|^2 + \ldots + \|F_N\tilde{X} - e_N\|^2$$

or

$$\tilde{J}_{fuel,time}(\tilde{X}) = \|F_0\tilde{X} - e_0\| + \|F_1\tilde{X} - e_1\| + \ldots + \|F_N\tilde{X} - e_N\| + \left\|F_t\tilde{X} - e_t\right\|$$

or

$$\tilde{J}_{energy,time}(\tilde{X}) = \|F_0\tilde{X} - e_0\|^2 + \|F_1\tilde{X} - e_1\|^2 + \ldots + \|F_N\tilde{X} - e_N\|^2 + F_t\tilde{X} - e_t$$

where

$$F_0 = \begin{bmatrix} 0_{3x3} & I_{3x3} & 0_{3x3} & \cdots & 0_{3x3} \end{bmatrix}, e_0 = v_{initial} - v_0^{ref}$$

$$F_N = \begin{bmatrix} 0_{3x3} & \cdots & 0_{3x3} & -C_{i+1,i} & -D_{i+1,i} & -\dot{a}_{i+1}^{ref} \end{bmatrix}, e_N = v_{N+1}^{-ref} - v_{final}$$

$$F_{i+1} = \begin{bmatrix} 0_{3x3} & \cdots & -C_{i+1,i} & -D_{i+1,i} & -\dot{a}_{i+1}^{ref} & 0_{3x3} & I_{3x3} & \cdots & 0_{3x3} \end{bmatrix}, e_{i+1} = v_{i+1}^{-ref} - v_{i+1}^{ref}$$

$$F_t = \begin{bmatrix} 0_{1x6} & \epsilon & \ldots & 0_{1x6} & \epsilon \end{bmatrix}, e_t = -\epsilon\left(t_1^{ref} + \ldots + t_N^{ref}\right)$$

Subject to the linear/affine equality constraints:

$$\text{Continuity: } \begin{bmatrix} I_{3x3} & 0_{3x3} & & & \cdots & & & 0_{3x3} \\ A_{1,0} & B_{1,0} & v_1^{-\text{ref}} & -I_{3x3} & 0_{3x3} & \cdots & & 0_{3x3} \\ \vdots & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ 0_{3x3} & \cdots & A_{i+1,i} & B_{i+1,i} & v_{i+1}^{-\text{ref}} & -I_{3x3} & \cdots & 0_{3x3} \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0_{3x3} & & \cdots & & 0_{3x3} & A_{N,N-1} & B_{N,N-1} & v_N^{-ref} \end{bmatrix} \begin{bmatrix} \tilde{x}_0 \\ \tilde{v}_0 \\ \tilde{t}_1 \\ \tilde{x}_1 \\ \vdots \\ \tilde{x}_{N-1} \\ \tilde{v}_{N-1} \\ \tilde{t}_N \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

and Convex / Second Order Cone Inequality Constraints:

Max coasting time: $\tilde{t}_{i+1} \le t_{max} - t_{i+1}^{ref}, i = 0, \ldots, N-1$

Min coasting time: $-\tilde{t}_{i+1} \le -t_{min} + t_{i+1}^{ref}, i = 0, \ldots, N-1$

Impact Avoidance:

$$\left\| (\eta - 1)(x_i^{ref}(t_i^{p,ref}) - x_C) - (A_i^p \tilde{x}_i + B_i^p \tilde{v}_i + v_i^{ref}(t_i^{p,ref}) T_i \tilde{X}) \right\| \le \eta \| x_i^{ref}(t_i^{p,ref}) - x_C \| - R_F$$

Max thrust magnitude: $\| F_i \tilde{X} - e_i \| \le \Delta V_{max}, i = 0, \ldots, N-1$

Minimum Thrust Magnitude: $\left\| F_{i+1} \tilde{X} - (1-\zeta) e_{i+1} \right\| \le \zeta \| e_{i+1} \| - \Delta V_{min}, \zeta > 1, i = 0, \ldots, N-1$

Trust Regions:
$$\begin{aligned} \| \tilde{x}_i \| &\le \xi / \| A_{i+1,i} \|_2, i = 0, \ldots, N-1 \\ \| \tilde{v}_i \| &\le \upsilon / \| B_{i+1,i} \|_2, i = 0, \ldots, N-1 \\ | \tilde{t}_i | &\le \tau / \| v_{i+1}^{-ref} \|_2, i = 0, \ldots, N-1 \end{aligned}$$

Thrust Direction Limit (optional): See Appendix B.

5.7.7 Minimum Fuel as a Second Order Cone Problem

Recall that due to the nature of the cost function and the fact the inequality constraints are all Second Order Cones, the above problem is transformed into a Second Order Cone Problem via the epigraph transformation:

Find variables:

$$\tilde{X}_{SOCP} = \begin{bmatrix} \tilde{x}_0^T & \tilde{v}_0^T & \tilde{t}_1 & \tilde{x}_1^T & \cdots & \tilde{x}_{N-1}^T & \tilde{v}_{N-1}^T & \tilde{t}_N & \sigma_0 & \sigma_1 & \ldots & \sigma_N \end{bmatrix}^T$$

That minimize the linear cost function:

$$\tilde{J}_{SOCP}(\tilde{X}_{SOCP})=\sigma_0+\ldots+\sigma_N$$

$$\tilde{J}_{SOCP}(\tilde{X},\sigma_0,\ldots,\sigma_N,\sigma_t)=\sigma_0+\ldots+\sigma_N+\sigma_t$$

and adding additional constraints

$$\|F_i X-e_i\|<\sigma_i, i=0,\ldots,N$$
$$\|F_t X-e_t\|<\sigma_t$$

Subject to the constraints of the convex problem above with the addition of:

$$\|F_0\tilde{X}-e_0\|\leq\sigma_0$$
$$\|F_1\tilde{X}-e_1\|\leq\sigma_1$$
$$\ldots$$
$$\|F_N\tilde{X}-e_N\|\leq\sigma_N$$

If this transfer time is included in a weighted combination, then this becomes

Find variables:

$$\tilde{X}_{SOCP}=\begin{bmatrix}\tilde{x}_0^T & \tilde{v}_0^T & \tilde{t}_1 & \tilde{x}_1^T & \cdots & \tilde{x}_{N-1}^T & \tilde{v}_{N-1}^T & \tilde{t}_N & \sigma_0 & \sigma_1 & \ldots & \sigma_N & \sigma_t\end{bmatrix}^T$$

That minimize the linear cost function:

$$\tilde{J}_{SOCP}(\tilde{X}_{SOCP})=\sigma_0+\ldots+\sigma_N+\sigma_t$$

Subject to the constraints of the convex problem above with the addition of:

$$\|F_i X-e_i\|<\sigma_i, i=0,\ldots,N$$
$$\|F_t X-e_t\|<\sigma_t$$

5.7.8 Minimum Energy as a Quadratically Constrained Quadratic Problem

Find variables:

$$\tilde{X}=\begin{bmatrix}\tilde{x}_0^T & \tilde{v}_0^T & \tilde{t}_1 & \tilde{x}_1^T & \cdots & \tilde{x}_{N-1}^T & \tilde{v}_{N-1}^T & \tilde{t}_N\end{bmatrix}^T$$

That minimize the cost function:

$$\tilde{J}_{energy}(\tilde{X})=\|F_0\tilde{X}-e_0\|^2+\|F_1\tilde{X}-e_1\|^2+\ldots+\|F_N\tilde{X}-e_N\|^2$$

$$\text{where}\quad F_0=\begin{bmatrix}0_{3x3} & I_{3x3} & 0_{3x3} & \cdots & 0_{3x3}\end{bmatrix}, e_0=v_{initial}-v_0^{ref}$$

$$F_N=\begin{bmatrix}0_{3x3} & \cdots & 0_{3x3} & -C_{i+1,i} & -D_{i+1,i} & -a_{i+1}^{\text{- ref}}\end{bmatrix}, e_N=v_{N+1}^{-ref}-v_{final}$$

$$F_{i+1}=\begin{bmatrix}0_{3x3} & \cdots & -C_{i+1,i} & -D_{i+1,i} & -a_{i+1}^{\text{- ref}} & 0_{3x3} & I_{3x3} & \cdots & 0_{3x3}\end{bmatrix}, e_{i+1}=v_{i+1}^{-ref}-v_{i+1}^{ref}$$

Subject to the linear/affine equality constraints:

$$\text{Continuity:}\quad \begin{bmatrix} I_{3x3} & 0_{3x3} & & & \cdots & & & 0_{3x3} \\ A_{1,0} & B_{1,0} & v_1^{\text{- ref}} & -I_{3x3} & 0_{3x3} & \cdots & & 0_{3x3} \\ \vdots & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ 0_{3x3} & \cdots & A_{i+1,i} & B_{i+1,i} & v_{i+1}^{\text{- ref}} & -I_{3x3} & \cdots & 0_{3x3} \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0_{3x3} & & \cdots & & 0_{3x3} & A_{N,N-1} & B_{N,N-1} & v_N^{-ref} \end{bmatrix} \begin{bmatrix} \tilde{x}_0 \\ \tilde{v}_0 \\ \tilde{t}_1 \\ \tilde{x}_1 \\ \vdots \\ \tilde{x}_{N-1} \\ \tilde{v}_{N-1} \\ \tilde{t}_N \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Fixed total transfer time (optional): $\sum_{i=0}^{N-1}\tilde{t}_{i+1}=t_{total}-\sum_{i=0}^{N-1}t_{i+1}^{ref}$.

and Convex Quadratic Inequality Constraints:

Coasting time: $\left(\tilde{t}_{i+1}+\left(t_{i+1}^{ref}-\frac{t_{min}+t_{max}}{2}\right)\right)^2\le\left(\frac{t_{max}+t_{min}}{2}\right)^2, i=0,\ldots,N-1$

Impact Avoidance:

$$\left\|(\eta-1)\left(x_i^{ref}\left(t_i^{p,ref}\right)-x_C\right)-\left(A_i^p\tilde{x}_i+B_i^p\tilde{v}_i+v_i^{ref}\left(t_i^{p,ref}\right)T_i\tilde{X}\right)\right\|^2\le\left(\eta\left\|x_i^{ref}\left(t_i^{p,ref}\right)-x_C\right\|-R_F\right)^2$$

Max thrust magnitude: $\|F_i\tilde{X}-e_i\|^2\le(\Delta V_{max})^2, i=0,\ldots,N-1$

Min Thrust Magnitude: $\left\|F_{i+1}\tilde{X}-(1-\zeta)e_{i+1}\right\|^2\le(\zeta\|e_{i+1}\|-\Delta V_{min})^2, \zeta>1, i=0,\ldots,N-1$

Trust Regions:
$$\|\tilde{x}_i\|^2\le(\xi/\|A_{i+1,i}\|_2)^2, i=0,\ldots,N-1$$
$$\|\tilde{v}_i\|^2\le(\upsilon/\|B_{i+1,i}\|_2)^2, i=0,\ldots,N-1$$
$$\tilde{t}_i^2\le(\tau/\|v_{i+1}^{-ref}\|_2)^2, i=0,\ldots,N-1$$

If total transfer time is included in a weighted combination, then the cost function becomes

$$\tilde{J}_{energy}(\tilde{X})=\|F_0\tilde{X}-e_0\|^2+\|F_1\tilde{X}-e_1\|^2+\ldots+\|F_N\tilde{X}-e_N\|^2+F_t\tilde{X}-e_t$$

$$F_t=\begin{bmatrix}0_{1x6} & \epsilon & \ldots & 0_{1x6} & \epsilon\end{bmatrix}, e_t=-\epsilon\left(t_1^{ref}+\ldots+t_N^{ref}\right)$$

5.7.9 Comparison of KKT Points

If X^{ref} is a fixed KKT point of the iterative NLP, then it satisfies the following conditions:

$$\nabla J_{iter}(X^{ref})+\lambda^T \nabla h_{iter}(X^{ref})+\mu^T \nabla g_{iter}(X^{ref})=0$$
$$h_{iter}(X^{ref})=0$$
$$g_{iter}(X^{ref})\leq 0$$
$$\mu\geq 0$$
$$\mu^T g_{iter}(X^{ref})=0$$

Since the convex approximations are at least first-order approximations *at the reference trajectory*, the function values and gradients of the unapproximated $J_{iter}, g_{iter}, h_{iter}$ are equal to those of their convex approximations at this point, i.e. $J_{iter}(X^{ref})=\tilde{J}(0), \nabla J_{iter}(X^{ref})=\nabla \tilde{J}(0),$ etc. (Note that using a quadratic form of the constraints will change the gradient magnitude but not the direction, so the result will still hold) Substitution of these quantities into the above equations for this case and noting that the reference trajectory is the origin of the convex approximation and so all of the trust region constraints are inactive gives:

$$\nabla \tilde{J}(0)+\lambda^T \nabla \tilde{h}(0)+\hat{\mu}^T \nabla \tilde{g}(0)=0$$
$$\tilde{h}(0)=0$$
$$\tilde{g}(0)\leq 0$$
$$\hat{\mu}\geq 0$$
$$\hat{\mu}^T \tilde{g}(0)=0$$

Thus satisfaction of the KKT conditions at the reference for the iterative NLP is equivalent to the satisfaction of the KKT condition at the reference for the approximate CP, with the same multipliers for all constraints other than the trust regions, which are inactive. Thus the following are equivalent:

X^{ref} is a KKT point of the original NLP

X^{ref} is a KKT point of the iterative NLP defined in terms of X^{ref}

$\tilde{X}=0$ is a KKT point of the CP defined in terms of X^{ref}

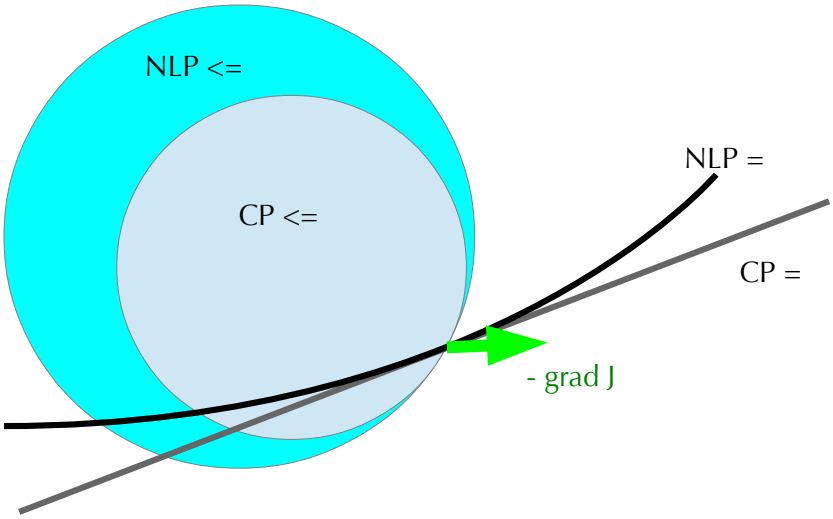


Figure 5.3: Despite approximation errors, since the cost and functions of the CP approximation match to first order, the KKT conditions are identical.

5.7.10 KKT Points vs. Fixed Points

For regular points, it is a standard result that an optimum will satisfy the KKT conditions. Convex problems have the unique property that a point that satisfies the KKT conditions is a global optimum of that problem. Thus if $\tilde{X}=0$ is a KKT point of the CP defined in terms of X^{ref}, which means it is a KKT point of the original NLP, it is also a global optimum of that CP.

5.8 Avoiding Degenerate Solutions for Feasible Iterations

5.8.1 Motivation

The descent proof in Chapter 7 relies on the fact that within the CP approximation, a nonzero change to the impulse vector direction or magnitude of an impulse constraint met with equality at the reference will result in a descent direction of the constraint. In such a case, if an optimal solution $\tilde{X}$ to the CP results in no change the the impulse yet changes the states of the arcs involved, this *may* result in $\tilde{X}$ not being a feasible direction.

5.8.2 Defining the Projection Matrix

Examining the impulse constraints, the change to the impulse from the reference is given by $F_i\tilde{X}$, as can be seen in the inequality constraints:

Max thrust magnitude: $\|F_{i+1}\tilde{X}-e_{i+1}\|\le\Delta V_{max}, i=0,\ldots,N-1$

Minimum Thrust Magnitude: $\left\|F_{i+1}\tilde{X}-(1-\zeta)e_{i+1}\right\|\le\zeta\|e_{i+1}\|-\Delta V_{min}, \zeta>1, i=0,\ldots,N-1$

We wish to identify the component of the change to the arcs states and coasting time which preserves continuity but results in no change to the impulse. These components satisfy:

$$\begin{bmatrix} C_{i+1,i} & D_{i+1,i} & a_{i+1}^{\text{ref}} & 0_{3x3} & -I_{3x3} \end{bmatrix}\begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \\ \tilde{t}_{i+1} \\ \tilde{x}_{i+1} \\ \tilde{v}_{i+1} \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \\ \tilde{t}_{i+1} \\ \tilde{x}_{i+1} \\ \tilde{v}_{i+1} \end{bmatrix}\neq 0.$$

Let $\hat{F}_{i+1}=\begin{bmatrix} C_{i+1,i} & D_{i+1,i} & a_{i+1}^{\text{ref}} & 0_{3x3} & -I_{3x3} \end{bmatrix}$. Since the null space of $\hat{F}_{i+1}$ is the orthogonal complement to the range of $\hat{F}_{i+1}^T$, if $\hat{Q}_{i+1}$ is the orthogonal projection of these variables onto the range of $\hat{F}_{i+1}^T$, then $\hat{P}_{i+1}=I-\hat{Q}_{i+1}$ will project onto its complement, hence the null space of $\hat{F}_{i+1}$. Since $\hat{F}_{i+1}^T$ is full rank, we may define the desired projection operator as $\hat{P}_{i+1}=I-\hat{F}_{i+1}^T(\hat{F}_{i+1}\hat{F}_{i+1}^T)^{-1}\hat{F}_{i+1}^T$. Let $P_{i+1}=\begin{bmatrix} \ldots & 0 & \hat{P}_{i+1} & 0 & \ldots \end{bmatrix}$ be the projection matrix padded with 0-matrices s.t.

$$P_{i+1}\tilde{X}=\hat{P}_{i+1}\begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \\ \tilde{t}_{i+1} \\ \tilde{x}_{i+1} \\ \tilde{v}_{i+1} \end{bmatrix}$$

Thus this projection operator projects $\tilde{X}$ to the null space of $\hat{F}_{i+1}$ and so F_{i+1}.

5.8.3 The Augmented Constraints

For both of the impulse constraints, the change to the current constraint function value is $F_{i+1}\tilde{X}$. We want a new constraint function that includes changes to the underlying variables that does not affect $F_{i+1}\tilde{X}$. Consider the following

$$\text{Max thrust magnitude: } \left\| \begin{bmatrix} F_{i+1} \\ P_{i+1} \end{bmatrix} \tilde{X} - \begin{bmatrix} e_{i+1} \\ 0 \end{bmatrix} \right\| \leq \Delta V_{max}$$

$$\text{Minimum Thrust Magnitude: } \left\| \begin{bmatrix} F_{i+1} \\ P_{i+1} \end{bmatrix} \tilde{X} - (1-\zeta) \begin{bmatrix} e_{i+1} \\ 0 \end{bmatrix} \right\| \leq \zeta \|e_{i+1}\| - \Delta V_{min}$$

These are clearly second order cone constraints, or may be squared if quadratic constraints are desired. The following analysis will use the maximum thrust magnitude as the example, though the same steps apply to the minimum thrust magnitude.

$$\left\| \begin{bmatrix} F_{i+1} \\ P_{i+1} \end{bmatrix} \tilde{X} - \begin{bmatrix} e_{i+1} \\ 0 \end{bmatrix} \right\| = \left\| \begin{bmatrix} F_{i+1}\tilde{X} - e_{i+1} \\ P_{i+1}\tilde{X} \end{bmatrix} \right\| = \left(\|F_{i+1}\tilde{X} - e_{i+1}\|^2 + \|P_{i+1}\tilde{X}\|^2 \right)^{1/2} < \Delta V_{max}$$

So, this new constraint is equivalent to the original with an additional term that penalizes changes to the variables with the undesirable behavior. It is immediately clear that any point that satisfies this constraint satisfies the original constraint, and that the reference is feasible in either constraint.

5.8.4 Equivalence of Fixed KKT Points

We will refer to the SOCP/QCQPs with the original constraints as SOCP1, QCQP1 and those with these new, augmented constraints as SOCP2, QCQP2. Consider the gradients of the constraints in at the reference $\tilde{X} = 0$.

$$\nabla \tilde{g}_{SOCP2}(0)=\frac{\begin{bmatrix} F_{i+1} \\ P_{i+1} \end{bmatrix}^T \begin{bmatrix} -e_{i+1} \\ 0 \end{bmatrix}}{\left\| \begin{bmatrix} -e_{i+1} \\ 0 \end{bmatrix} \right\|}=\frac{-F_{i+1}^T e_{i+1}}{\left\| -e_{i+1} \right\|}=\nabla \tilde{g}_{SOCP1}(0)$$

$$\nabla \tilde{g}_{QCQP2}(0)=2\begin{bmatrix} F_{i+1} \\ P_{i+1} \end{bmatrix}^T \begin{bmatrix} -e_{i+1} \\ 0 \end{bmatrix}=-2\,F_{i+1}^T(-e_{i+1})=\nabla \tilde{g}_{QCQP1}(0)$$

Thus constraints modified in this way have the exact same gradients as the original constraints at $\tilde{X}=0$. Thus $\tilde{X}=0$ is a KKT point of SOCP1/QCQP1 if and only if it is a KKT point of SOCP2/QCQP2. This can be shown for any other constraints modified in the same way, such as the minimum impulse constraint, which is almost identical. It may also be applied to the impact avoidance constraint, though of course the projection matrices will be different and the terms reliant on A_i^p, B_i^p, T_i terms. In Chapters 7 and 8, rather than include these bulkier terms that are only applied in very limited cases, the CP1 formulation will be used, but it can now be assumed without loss of generality that nonzero changes to $\begin{bmatrix} \tilde{x}_i^T & \tilde{v}_i^T & \tilde{t}_{i+1}^T & \tilde{x}_{i+1}^T & \tilde{v}_{i+1}^T \end{bmatrix}^T$ will result in a nonzero change in the vector magnitude or direction of the impulses or position involved in these constraints.

5.9 Summary

The result of this chapter may be summarized into the following proposition:

Proposition 5.9:

For the given impulsive transfer optimization NLP, at each feasible reference value X^{ref} there exist SOCP and QCQP approximations of the minimum fuel and minimum energy problems respectively such that the following conditions hold:

i) $X = X^{ref}$ (i.e. $\tilde{X} = 0$) satisfies the KKT conditions of the SOCP/QCQP if and only if $X = X^{ref}$ satisfies the KKT conditions of the original NLP.

ii) If $X = X^{ref}$ satisfies the KKT conditions of the original NLP, $\tilde{X} = 0$ solves the SOCP2/QCQP2.

iii) If $\tilde{X} = 0$ solves the SOCP/QCQP and is a regular point, then $X = X^{ref}$ satisfies the KKT conditions of the original NLP.

Chapter 6

The Iterative Process

6.1 Introduction

For an onboard application in an emergency situation, there is the threat of an interrupt or timeout during the calculation process. As a result, an algorithm should continue to produce a sequence of feasible iterations once feasibility is established. This way, a feasible if non-optimal transfer is available at each major iteration in case the process must be terminated. This means that many standard methods, even those with provable convergence properties or global descent of a Lagrangian based merit function, are not sufficient when applied to the impulsive trajectory design problem. This includes popular SQP methods such as those used in SNOPT,[Gill02] as well as what are referred to as Feasible SQP. [Lawr01] In the latter case this is due to the fact that feasible refers only to nonlinear *inequality* constraints – nonlinear equality constraints such as continuity are not guaranteed to be feasible. [Lawr96] The Two-Level Differential Corrector [Marc07] has the structure of restoring feasibility at each major iteration, but the Level V reduction step lacks enough structure to have produced a proof of cost descent after feasibility is restored. Additionally, its use of the psuedoinverse to define its minimization step means it is equality constraint focused [Marc07] and as such is not strictly appropriate for the constraints listed in Chapter 5.

Sequential Convex Programming (SCP) is another iterative local optimization method. [Boyd08] An early application came from optimal geometries in structural mechanics, but there have also been recent applications in spaceflight for optimal descent algorithms.[Caso13] It is similar to SQP, with the exception that the more general set of CPs (including SOCPs or QCQPs) are used as approximate subproblems during the iteration process. Unlike the SQP implementation in SNOPT, for example, it is often implemented without a line search step,[Morg13, Caso13, Augu12, Boyd08] though this is not always the case. [Zill04] Examples have shown very rapid progress towards the optimum relative to other iterative methods, with final local convergence rates more elusive.[Boyd08] For a scenario where quick progress to a reasonable trajectory is more important than finding the precise optimum, this is not necessarily an issue.

For the current application, in order to fulfill the requirement that each major iteration provide a feasible solution, it will be shown in Chapter 7 that the ability to line search is necessary in some cases. Conditions for which line search may be omitted for an iteration will also be provided in Chapter 7. Thus both a line search and trust region only iteration sequence will be provided in this chapter. More importantly though, a velocity adjustment step to restore continuity similar to that of the Two-Level Corrector will be included. Indeed, it is perhaps just as accurate to think of this process as a Two-Level Corrector with the ΔV minimization step replaced by a CP and line search than as a SCP method with a correction step.

6.2 General Sequential Convex Programming

6.2.1 Basic Algorithm

This is a description of the basic trust region based approach described in [Boyd08].

Begin SCP Iteration

Step 1: From a reference state X^{ref}, create an approximate CP:

Find change from the reference $\tilde{X}$

within the trust region, e.g. $\|\tilde{X}\| \le T$

that minimizes convex cost $\tilde{f} = \tilde{J}$

subject to convex inequality constraints $\tilde{g} \le 0$

and affine equality constraints $\tilde{h} = 0$

An example of this for this problem was provided in Chapter 5, but it not unique. Alternately, use the convex approximations to create an unconstrained CP using a penalty function.

Find change from the reference $\tilde{X}$

within the trust region, e.g. $\|\tilde{X}\| \le T$

that minimizes an augmented cost, e.g. $\tilde{f} = \tilde{J} + \frac{1}{2}\gamma_h \sum \tilde{h}_j^2(\tilde{X}) + \frac{1}{2}\gamma_g \sum (\tilde{g}_k^+)^2(\tilde{X})$

where $g_k^+(\tilde{X}) = max\{0, g_k(\tilde{X})\}$

The type of cost with penalty or merit function is not unique. This approach serves to reduce constraint violations as well as cost without including constraints explicitly.

Step 2: Solve the approximate CP. Let $\tilde{X}$ be the change from the reference.

Step 3: IF $f(X^{ref} + \tilde{X}) - f(X^{ref}) \ge \epsilon(\tilde{f}(\tilde{X}) - \tilde{f}(0)), 0 < \epsilon < 1$

The actual cost (or cost with penalty, Lagrangian, etc.) decrease is less than a minimum fraction of the expected decrease from the CP. Multiply trust region parameters by 0.5 or similar fraction and return to Step 1.

ELSE

Take $X^{ref} + \tilde{X}$ as the new reference value, return to Step 1.

End SCP Iteration

This loop may also terminate if $\|X\| < \text{step\_tol}$, KKT conditions are evaluated, a maximum number of iterations, etc., depending on the problem.

6.2.2 Suitability for this Application

While a CP approximation is less limited than a QP, the standard SCP method suffers from

the same issue as SQP in that it does not produce feasible iterations. This is true whether a full CP approximation with constraints is used or some sort of merit function with penalty. There is synergy between CP solutions and line search methods that can be exploited, since the first-order conditions on convexity imply the CP solution defines a descent direction of the cost. Line search alone cannot provide the feasible iterates needed, but as will be shown, the descent properties of CP solutions may be combined with a continuity correction step similar to the Two-Level Corrector below in order to make progress towards this goal.

6.3 Two-Level Differential Corrector

6.3.1 Method Description: Level I

This differential corrector consists of two distinct processes. The Level I process is used to alter a discontinuous trajectory through iterative velocity adjustments in order to establish or re-establish continuity. Using the notation of this work, when there is a discontinuity at a patch point, the position of the incoming arc x_{i+1}^- does not equal the position of the arc beginning at the patch point, x_{i+1}. The arc endpoint x_{i+1}^- is defined in terms of the arc's initial conditions x_i, v_i, t_{i+1}, but since only the velocity is to be adjusted, a change to v_i that results in a change of $(x_{i+1} - x_{i+1}^-)$ is sought. This cannot be solved directly, but recall that for small variations, we can use the STM to define

$$\begin{bmatrix} \delta x_{i+1}^- \\ \delta v_{i+1}^- \end{bmatrix} \approx \begin{bmatrix} A_{i+1,i} & B_{i+1,i} \\ C_{i+1,i} & D_{i+1,i} \end{bmatrix} \begin{bmatrix} \delta x_i \\ \delta v_i \end{bmatrix} + \begin{bmatrix} v_{i+1}^{-ref} \\ a_{i+1}^{-ref} \end{bmatrix} \delta t_{i+1}$$

or, since the position and coasting times are to remain fixed during this process, $\delta x_{i+1}^- \approx B_{i+1,i} \delta v_i$. Since a change in position equal to $(x_{i+1} - x_{i+1}^-)$ is desired, a first-order approximation of the necessary velocity adjustment is $\delta v_i = B_{i+1,i}^{-1}(x_{i+1} - x_{i+1}^-)$. As this result is

only approximate, the trajectory and the STMs are recalculated and the process repeated with the hopefully smaller gap. This iterative process should be familiar as part of most correctors and shooting method approaches to solving two point boundary value problems. [Sukh04] This does not converge to a continuous trajectory in every case, and an issue with the Two-Level corrector (at least for onboard use), is that the result of the Level II step below cannot be shown to only cause correctable discontinuities.

6.3.2 Method Description: Level II

The Level II step adjust the positions and coasting time in order to reduce the total ΔV. The derivations and their results are quite lengthy, but linear relationships between the ΔV_i at each patch point and changes to the relevant position and times are established.[Marc07] These differ from the relationships given in the derivation of the SOCP approximations in Chapter 5 because the continuity constraint is not treated as an extra constraint but rather substituted directly into the expressions used in the Level II step. To first-order, and after these substitutions:

$$\delta\Delta V_i = \frac{\partial \Delta V_i}{\partial x_{i-1}}\delta x_{i-1} + \frac{\partial \Delta V_i}{\partial t_i}\delta t_i + \frac{\partial \Delta V_i}{\partial x_i}\delta x_i + \frac{\partial \Delta V_i}{\partial t_{i+1}}\delta t_{i+1} + \frac{\partial \Delta V_i}{\partial x_{i+1}}\delta x_{i+1} + \frac{\partial \Delta V_i}{\partial t_{i+2}}\delta t_{i+2}$$

$$\delta\Delta V_i = \begin{bmatrix} \frac{\partial \Delta V_i}{\partial x_{i-1}} & \frac{\partial \Delta V_i}{\partial t_i} & \frac{\partial \Delta V_i}{\partial x_i} & \frac{\partial \Delta V_i}{\partial t_{i+1}} & \frac{\partial \Delta V_i}{\partial x_{i+1}} & \frac{\partial \Delta V_i}{\partial t_{i+2}} \end{bmatrix} \begin{bmatrix} \delta x_{i-1} \\ \delta t_i \\ \delta x_i \\ \delta t_{i+1} \\ \delta x_{i+1} \\ \delta t_{i+2} \end{bmatrix} = M\vec{b}$$

Other <u>equality</u> may be included as well after similar relationship are established, where (in the case they are currently satisfied:

$$\begin{bmatrix} \delta\Delta V_i \\ 0 \\ \vdots \end{bmatrix} = \begin{bmatrix} M \\ \frac{\partial h_j}{\partial \vec{b}} \\ \vdots \end{bmatrix} \vec{b} = \hat{M}\vec{b}$$

For some desired decrease $\delta\Delta V_i$ in the maneuver cost, the position and the coasting time changes are specified by the pseudo-inverse (i.e. min norm or least squares depending on of the system is under or overconstrained):

$$\vec{b}=(\hat{M}^T\hat{M})^{-1}\hat{M}^T\begin{bmatrix}\delta\Delta V_i\\0\\\vdots\end{bmatrix} \text{ or } \vec{b}=\hat{M}^T(\hat{M}\hat{M}^T)^{-1}\begin{bmatrix}\delta\Delta V_i\\0\\\vdots\end{bmatrix}$$

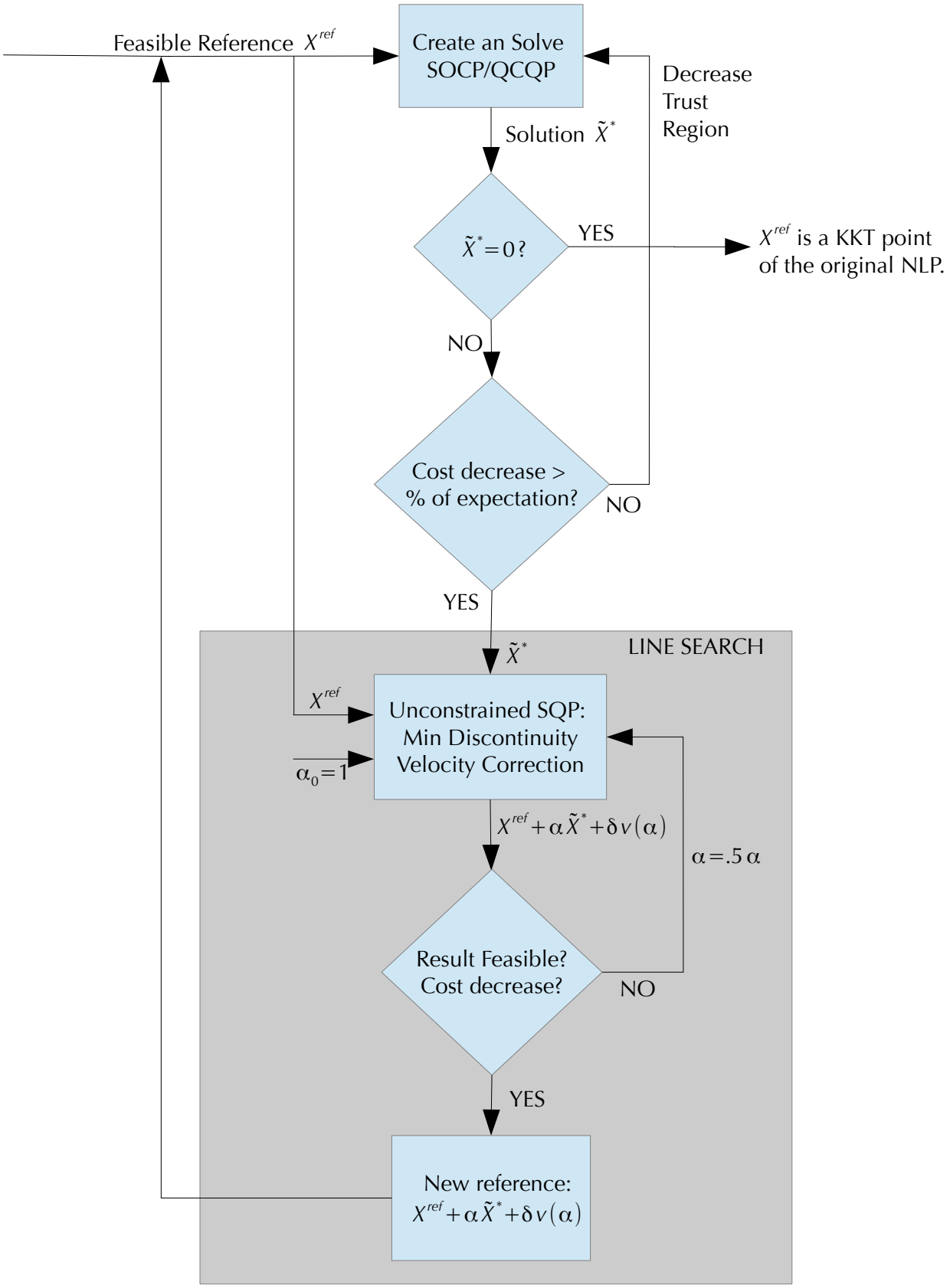
Due to approximation errors, the result will be discontinuous, thus Level I correction is performed to eliminate this.[Marc07] Then the process is repeated.

6.3.3 Suitability for this Application

In practice, when step sizes are modest, this method is successful and has been used in mission design, including the Genesis mission [Lo01] and for moving trajectories from basic to more complicated models [Koon08]. Compared to a standard SCP/SQP method, the step within each major iteration of correcting for continuity is an improvement when feasible iterates are required. This change plays a central role in the algorithm used here. On the other hand, the Level II step definition cannot be shown to lead to a correctable guess, indeed it has been observed not to in some instances. Additionally, inequality constraints such as those listed in Chapter 5 pose a problem. For more open ended mission design, the equality constraints such as beginning and ending energy levels, periodicity, declination, etc. play the key role. Including these for each patch point would indeed be difficult in the method developed here, but the optimization method here is to be applied when high level goals and target states satisfying them have been selected and the need is to simply get there from the current spacecraft state.

6.4 Proposed Iterative Method

6.4.1 Algorithm with Line Search


Feasible Reference X^{ref}
Create an Solve SOCP/QCQP
Decrease Trust Region
Solution $\tilde{X}^*$
$\tilde{X}^* = 0$?
YES
X^{ref} is a KKT point of the original NLP.
NO
Cost decrease > % of expectation?
NO
YES
LINE SEARCH
$\tilde{X}^*$
X^{ref}
Unconstrained SQP: Min Discontinuity Velocity Correction
$\alpha_0 = 1$
$X^{ref} + \alpha \tilde{X}^* + \delta v(\alpha)$
$\alpha = .5\,\alpha$
Result Feasible? Cost decrease?
NO
YES
New reference: $X^{ref} + \alpha \tilde{X}^* + \delta v(\alpha)$

This is a hybrid method where the step direction is defined by the solution to a convex problem, but where a correction step has been added in order to restore continuity and the feasibility of other constraints at the end of each major iteration. It is given in reference to the particular problem at hand. Additionally, it uses the fact that if X^{ref} is a KKT point of the original problem will be a KKT point of the SOCP/QCQP, and since KKT satisfaction implies optimality in convex problems,[Boyd09] the SOCP/QCQP will have a fixed point solution. The effectiveness of this algorithm in providing feasible iterates with lower cost will be described in Chapter 7.

BEGIN SCP Major Iteration Loop

Step SCP1: From a feasible reference trajectory, calculate the trajectories resulting from the specified initial conditions $x_i^{ref}, v_i^{ref}, t_{i+1}^{ref}$. Store the reference arc endpoint states $x_{i+1}^{-ref}, v_{i+1}^{-ref}$ as well as the corresponding State Transition Matrix submatrices $A_{i+1,i}, B_{i+1,i}, C_{i+1,i}, D_{i+1,i}$. Additionally, for any periapsis on the arc, store the corresponding state and coasting time to periapsis $x_i^{ref}(t_i^{p,ref}), v_i^{ref}(t_i^{p,ref}), t_i^{p,ref}$, and the corresponding STM submatrices $A_i^p, B_i^p, C_i^p, D_i^p$.

Step SCP2: Use these quantities and the current trust region parameters to define a SOCP (for the min fuel problem) or QCQP (for min energy) approximation as defined in Chapter 5.

Step SCP3: Solve the SOCP/QCQP. Since the reference is feasible, there exists at least one feasible point at $\tilde{X}=0$ and so a solution will exist.

Step SCP4:

IF $\tilde{X}=0$ or $\|\tilde{X}\|<$step_tol

STOP. RETURN X^{ref} as the final result of the optimization process.

ELSE IF $J(X^{ref}+\tilde{X})-J(X^{ref})\geq\epsilon(\tilde{J}(\tilde{X})-\tilde{J}(0)), 0<\epsilon<1$

(i.e. the decrease in the NLP cost is less than some fraction of the estimated decrease)

Multiply trust region parameters by 0.5. Return to Step SCP2.

ELSE

Multiply trust region parameters by 1.1.

BEGIN Line Search Loop

Step LS1: Take $\tilde{X}$ as the search direction for backstepping (or other) line search. Initialize search parameter $\alpha=1$.

Step LS2: For current α value, consider point defined by $X^{ref}+\alpha\tilde{X}$. With only the v_i as variables, attempt correction to restore continuity. That is, find velocity adjustments δv_i such that the trajectory defined by $x_i(\alpha)=x_i^{ref}+\alpha\tilde{x}_i$, $v_i(\alpha)=v_i^{ref}+\alpha\tilde{v}_i+\delta v_i(\alpha)$, $t_i(\alpha)=t_i^{ref}+\alpha\tilde{t}_i$ is continuous, if possible. This may be done via Level I correction or *unconstrained* SQP process with cost $\sum_{i=0}^{N}\|x_{i+1}(\alpha)-\bar{x}_{i+1}(\alpha)\|^2$.

Step LS3:

IF continuity is restored, i.e. $\exists\delta v_i(\alpha)$ s.t. $\sum_{i=0}^{N}\|x_{i+1}(\alpha)-\bar{x}_{i+1}(\alpha)\|^2=0$

IF $X^{ref}+\alpha\tilde{X}+\delta v$ is within the feasible range and the Armijo conditions (or other) are satisfied.

Take feasible, improved result as the new reference trajectory:

Set $\boldsymbol{X^{ref}=X^{ref}+\alpha\tilde{X}+\delta v}$. Go to Step SCP1.

ELSE

Set $\alpha=0.5\alpha$, return to step LS2.

ELSE

Set $\alpha=0.5\alpha$, return to step LS2.

Line search steps repeat until it ends with an improved, feasible result. Results of Chapter 7 proves this exists, but due to iteration limits or step size tolerance, if $\alpha <$step_tol multiply trust region parameters by 0.5. Return to Step SCP2.

END Line Search Loop

Repeat SCP iteration loop until next reference is found, step size is below tolerance and counted as a fixed point, or in case of interrupt.

END SCP Major Iteration Loop

Repeat outer SCP iterations loop until step size is below tolerance and counted as a fixed point, or in case of interrupt, or maximum iterations/calculation time is exceeded. Return current X^{ref}.

6.4.2 Variant Algorithm Without Line Search

As is shown in Section 7.4.2, when the reference trajectory is strictly feasible, the line search step may be eliminated and trust region size alone may be used to control the iteration process. For iterations whose reference trajectory satisfies these conditions, the number of steps could be reduced.

BEGIN SCP Major Iteration Loop

Step SCP1 (unchanged): From a feasible reference trajectory, calculate the trajectories resulting from the specified initial conditions $x_i^{ref}, v_i^{ref}, t_{i+1}^{ref}$. Store the reference arc endpoint states $x_{i+1}^{-ref}, v_{i+1}^{-ref}$ as well as the corresponding State Transition Matrix submatrices $A_{i+1,i}, B_{i+1,i}, C_{i+1,i}, D_{i+1,i}$. Additionally, for any periapsis on the arc, store the corresponding

state and coasting time to periapsis $x_i^{ref}(t_i^{p,ref}), v_i^{ref}(t_i^{p,ref}), t_i^{p,ref}$, and the corresponding STM submatrices $A_i^p, B_i^p, C_i^p, D_i^p$.

Step SCP2 (unchanged): Use these quantities and the current trust region parameters to define SOCP/QCQP approximation as defined in Chapter 5.

Step SCP3 (unchanged): Solve the SOCP/QCQP. Since the reference is feasible, there exists at least one feasible point at $\tilde{X}=0$ and so a solution will exist.

Step SCP4:

IF $\tilde{X}=0$ or $\|\tilde{X}\|<$step_tol

STOP. RETURN X^{ref} as the final result of the optimization process.

ELSE IF $J(X^{ref}+\tilde{X})\geq J(X^{ref})$

Multiply trust region parameters by 0.5. Return to Step SCP2.

ELSE

Multiply trust region parameters by 1.1. Consider point defined by $X^{ref}+\tilde{X}$. With only the v_i as variables, attempt correction to restore continuity. That is, find velocity adjustments δv_i such that the trajectory defined by $x_i=x_i^{ref}+\tilde{x}_i, \quad v_i(\alpha)=v_i^{ref}+\tilde{v}_i+\delta v_i, \quad t_i(\alpha)=t_i^{ref}+\tilde{t}_i$ is continuous, if possible. This may be done via Level I correction or *unconstrained* SQP process with cost $\sum_{i=0}^{N}\|x_{i+1}-\bar{x}_{i+1}\|^2$.

Step SCP5:

IF continuity is restored, i.e. $\exists \delta v_i$ s.t. $\sum_{i=0}^{N}\|x_{i+1}-\bar{x}_{i+1}\|^2=0$

IF $X^{ref} + \tilde{X} + \delta v$ is within the feasible range and Armijo conditions (or other) are satisfied.

Take feasible, improved result as the new reference trajectory:

Set $\boldsymbol{X^{ref}} = \boldsymbol{X^{ref}} + \tilde{\boldsymbol{X}} + \delta \boldsymbol{v}$. Go to Step SCP1.

ELSE

Multiply trust region parameters by 0.5. Return to Step SCP2.

ELSE

Multiply trust region parameters by 0.5. Return to Step SCP2.

Repeat SCP iteration loop until next reference is found, step size is below tolerance and counted as a fixed point, or in case of interrupt.

END SCP Major Iteration Loop

Repeat outer SCP iterations loop until step size is below tolerance and counted as a fixed point, or in case of interrupt, or maximum iterations/calculation time is exceeded; return current X^{ref}.

6.5 Use of SQP in the Correction Phases

There are two places within the method where convergence to a true minimum is needed: the initial correction from the arcs selected from the graph search itinerary to provide a feasible guess for the optimizer, and the continuity correction step within the line search portion of the above algorithm. Both of these problems may be described in terms of quadratic penalties on the infeasible constraints in *unconstrained* subproblems. In both cases the "cost" is known ahead of time and is 0. In such a case, the lack of feasible iterates of SQP that make it inappropriate for the fuel cost minimization is no longer an issue, while the provable convergence rates are of benefit. It might be the case that the initial guess is within the basin of convergence of a nonzero minimum,

a fact which another local method would not change. While it is guaranteed that the back stepping line search will eventually yield test points for which the minimum discontinuity is 0, this is not necessarily the case for every value of the line search parameter. If this is not the case, then the parameter is simply reduced as described in the algorithm. Similarly, if the first graph search based result is not near a feasible solution, the link corresponding to an infeasible constraint can be eliminated, resulting in another search and initial guess.

6.6 Summary

A trajectory optimization process has been proposed that blends the positive aspects of Sequential Convex Programming and the Two-Level Differential Corrector. As will be shown in Chapters 7 and 8, this approach combined with the structure of the SOCP/QCQP approximations derived in Chapter 5 creates a globally convergent method (to local KKT points) that produces feasible iterates. In providing feasible iterates with both nonlinear equality and inequality constraints, it is able to satisfy the requirement to provide a feasible result in case of an onboard system interrupt.

Chapter 7

Proof of Global Descent with Feasible Iterates

7.1 Introduction

In a flight environment, there is a possibility of a system interrupt triggered by various system requirements including time limitations. The optimization process is the final stage of the transfer design process, the most time consuming, and without a finite iteration limit even with the convergence results proved in this research. Therefore, it must be able to return a feasible transfer whenever called for, not just at final convergence. This chapter proves that from a feasible initial guess (Chapter 9), the major iterates of the process described in Chapter 6 will continue to be feasible. In addition, the process is shown to be a global descent method on the space of feasible transfers. These results are akin to those of [Miel70] and subsequent work, but in the context of impulsive transfer optimization (hence without the additional structure of variational methods for continuous controls) and with the high speed, wider constraint set, and finite iterations of SOCP defined step directions. Lastly, results on problem simplification using unconstrained recorrection and the elimination of the line search step are provided.

7.2 Theorem Statement and Assumptions

Theorem:

Let the cost and constraints considered be those defined in Chapter 5. Then given the assumptions A1, A2 below, if X^{ref} satisfies the Karush-Kuhn-Tucker (KKT) conditions of the original NLP, the algorithm defined in Chapter 6.4.1 will terminate as a fixed point of the SCP process. Otherwise, the next major iteration is also feasible transfer with a strictly lower cost.

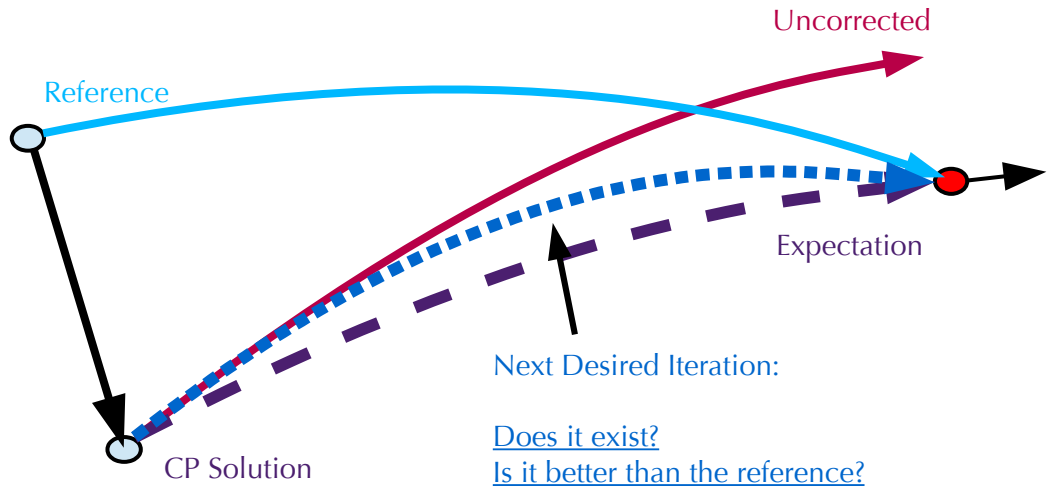


Figure 7.1: Conceptual diagram of the central question: Does a correction to the CP solution exist that yields a lower cost than the reference after eliminating approximation errors.

Assumption 1: Without loss of generality, it is assumed that the matrix $B_{i+1,i}$, the upper right quadrant of the State Transition Matrix, is invertible for each arc.[1]

Assumption 2: It is also assumed that the initial guess for the process satisfies one of the following, in decreasing order of strength of assumption: the Linear Independence Constraint Qualification (LICQ), the Mangasarian-Fromovitz Constraint Qualification (MFCQ), Slater's Condition, or the Constant Rank / Extended Mangasarian-Fromovitz Constraint Qualification

(CRMFCQ).[Krug14]

However, should even this mild assumption not hold, we could still state the following: If X^{ref} satisfies the Karush-Kuhn-Tucker (KKT) conditions of the original NLP, the algorithm defined in Chapter 6.1.4 will terminate as a fixed point of the SCP process. Otherwise, the next major iteration, *if a nonzero step is taken*, is also feasible transfer with a strictly lower cost.

7.3 Proof

7.3.1 Continuity Only

In order to more clearly illustrate the central principles of the proof, a simple case where continuity is the only constraint is given first. Assume we begin with a feasible initial guess, i.e. a position continuous but not locally optimal transfer between two given states, consisting of a set of *N* arcs and up to *N+1* impulses.

Step 1: X^{ref} is a KKT point of original NLP if and only if it is a fixed point of the CP.

As summarized in Section 5.9, X^{ref} is a KKT point of the original NLP if and only if $\tilde{X}=0$ is a KKT point of the CP defined in terms of X^{ref}. If $\tilde{X}=0$ is KKT point of the CP, satisfying the KKT conditions is sufficient for global optimality of the CP. [Boyd09 244] Additionally, if $\tilde{X}=0$ is a regular point, then if $\tilde{X}=0$ solves the CP it satisfies the KKT conditions. By assumption, the initial guess is a regular point, which as shown in Appendix A-Proposition A.4, implies $\tilde{X}=0$ is also a regular point for every iteration. Thus if X^{ref} is a KKT point of the original NLP, then the process will terminate if and only if $\tilde{X}=0$ is an optimum solution. Even without regularity of $\tilde{X}=0$, we could still state that if X^{ref} is a KKT point of the original NLP, $\tilde{X}=0$ is an optimum solution of the CP.

Step 2: Equivalence of Descent Directions

Since the CP cost function is a first order approximation of the original cost function at the reference trajectory, a descent direction of the CP cost function is a descent direction of the original cost function.

Step 3: CP Solution Defines a Descent Direction

Unless at a stationary point, any step taken by solving the CP decreases the CP cost function. Combining this with first order conditions on convex functions [Boyd09] results in:

$$\nabla \tilde{J}(X^{\text{ref}})\tilde{X} \leq \tilde{J}(X^{\text{ref}}+\tilde{X}) - \tilde{J}_C(X^{\text{ref}}) < 0.$$

Thus $\tilde{X}$ is a descent direction for the CP cost function, and so is a descent direction of the original cost function by Step 2.

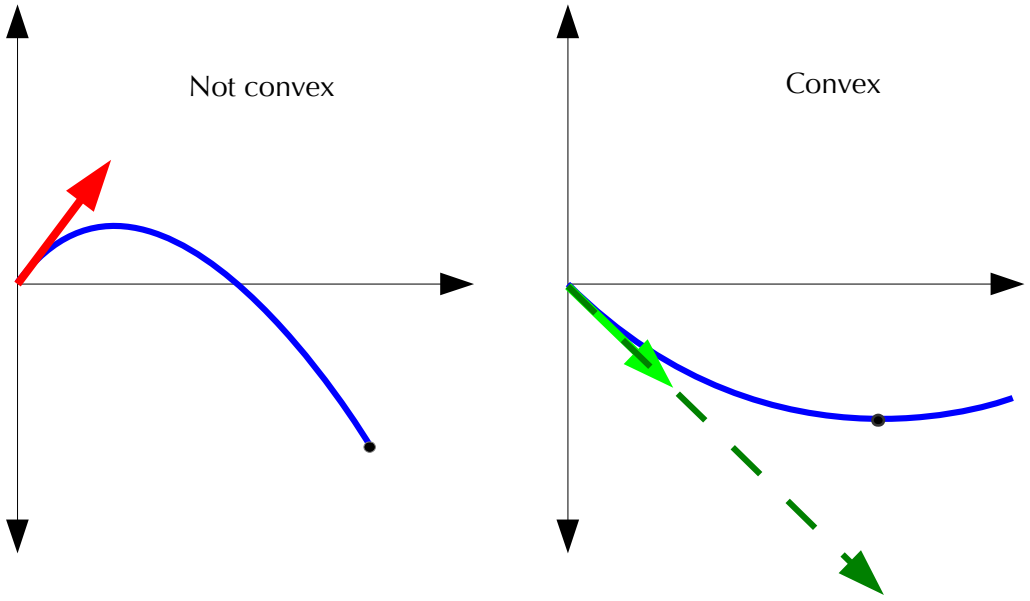


Figure 7.2: The first-order necessary conditions on convexity imply a decrease to the cost function in a specified direction makes that a descent direction of the function.

Step 4: Order of the Uncorrected Cost Decrease

Since $\tilde{X}$ is a descent direction of the original cost function, there exists a range of α s.t.

$J(X^{ref}+\alpha\tilde{X})<J(X^{ref})$ and this decrease is $O(\alpha)$ due to being a first order descent direction. Note that this cost is before correction has restored continuity, thus does not yet correspond to a feasible trajectory.

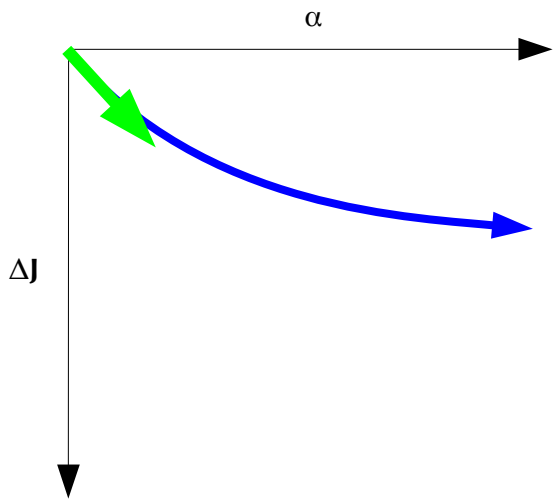


Figure 7.3: The CP solution defines a descent direction for the cost.

Step 5: Existence of Continuous Family Defined by CP Direction Plus Velocity Correction

For each patch point, between arcs *i* and *i+1*, let $\kappa:\mathbb{R}^4\to\mathbb{R}^3$ be given by:

$$\kappa_i(\alpha_i,\delta v_i)=x^-_{i+1}(\alpha_i,\delta v_i)-x_{i+1}(\alpha_i)=x(x_i^{ref}+\alpha_i\tilde{x}_i,v_i^{ref}+\alpha_i\tilde{v}_i+\delta v_i,t_{i+1}^{ref}+\alpha_i\tilde{t}_{i+1})-(x_{i+1}^{ref}+\alpha_i\tilde{x}_{i+1}).$$

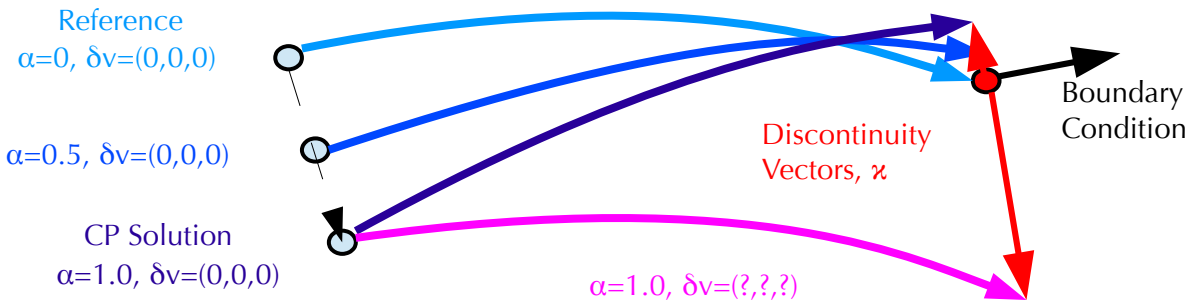


Figure 7.4: The discontinuity function, with inputs of the line search parameter in the CP solution direction and a velocity adjustment, and outputs the vector discontinuity.

This function defines the gap between arc endpoints, which means there is position continuity when the function value is the 0 vector. When starting from a continuous reference trajectory

$(\alpha_i, \delta v_i) = (0, \vec{0})$, and this function satisfies $\kappa_i(0,\vec{0}) = x(x_i^{ref}, v_i^{ref}, t_{i+1}^{ref}) - (x_{i+1}^{ref}) = x_{i+1}^{-ref} - (x_{i+1}^{ref}) = \vec{0}$.

Also, at $(\alpha_i, \delta v_i) = (0, \vec{0})$ we have $\frac{\partial \kappa_i}{\partial(\delta v_i)}(0,\vec{0}) = B_{i+1,1}$, which by assumption is invertible. Finally, note that $\varkappa$ is continuously differentiable everywhere in the domain.

Therefore the conditions of the Implicit Function Theorem are satisfied.[Lee03] Thus, there exist open sets $U_i \subset \mathbb{R}, V_i \subset \mathbb{R}^3$ such that $\alpha_i = 0 \in U_i, \delta v_i = \vec{0} \in V_i$ and a unique, continuously differentiable function $\gamma_i : U_i \to V_i$ such that:

$$\{(\alpha_i, \gamma_i(\alpha_i)) : \alpha_i \in U_i\} = \{(\alpha_i, \delta v_i) \in U_i \times V_i : \kappa_i(\alpha_i, \delta v_i) = 0\}.$$

This means that in a neighborhood of the reference trajectory for each α_i there exists a unique velocity correction $\delta v_i(\alpha_i) = \gamma_i(\alpha_i)$ such that taking a step defined by $\alpha \tilde{x}_i, \alpha \tilde{t}_{i+1}$ for position and coasting time and $\alpha \tilde{v}_i + \delta v_i(\alpha)$ for velocity results in a continuous trajectory.

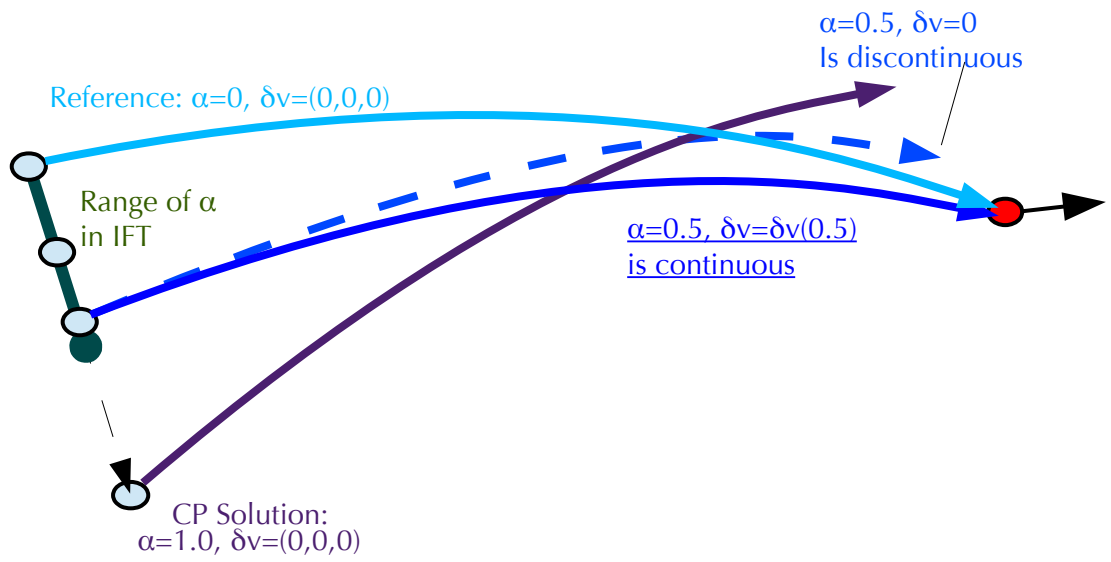


Figure 7.5: The Implicit Function Theorem guarantees there is a range of step size values where a velocity correction yields continuity.

Step 5: Determining the Order of the Correction

The Implicit Function Theorem also provides a formula for the derivative of γ_i:

$$\frac{\partial \gamma_i}{\partial \alpha_i}(0) = -\left(\frac{\partial \kappa_i}{\partial(\delta v_i)}(0,\vec{0})\right)^{-1} \frac{\partial \kappa_i}{\partial \alpha_i}(0,\vec{0}).$$

From the definition of κ, the second term is $\frac{\partial \kappa_i}{\partial \alpha_i}(0,\vec{0}) = \begin{bmatrix} A_{i+1,1} & B_{i+1,1} & v_{i+1}^{-ref} \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \\ \tilde{t}_{i+1} \end{bmatrix} - \tilde{x}_{i+1}$. However, since this is precisely the CP equality constraint, this quantity is the 0-vector, and so $\frac{\partial \gamma_i}{\partial \alpha_i}(0) = 0$.

Thus, the Implicit Function Theorem derivative result shows that the magnitude of the velocity correction vector is $\|\delta v(\alpha_i)\| = \|\gamma(\alpha_i)\| = o(\alpha_i)$.

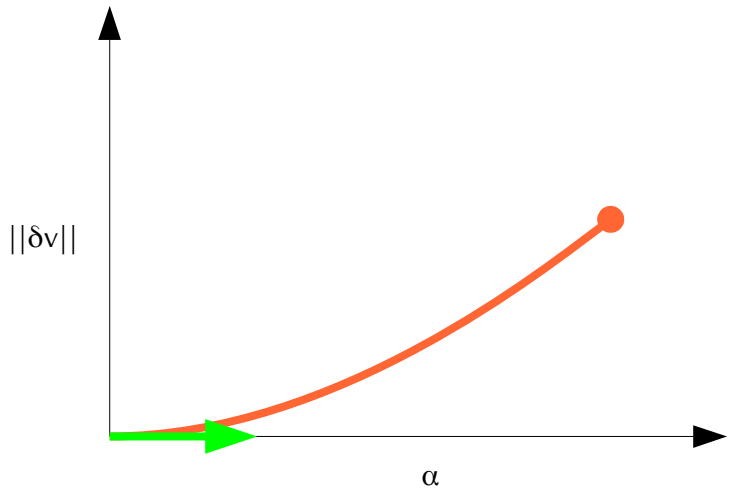


Figure 7.6: The derivative of the velocity adjustment w.r.t. the parameter is 0.

Step 7: Determining the Order of the Cost Function Increase due to Correction

It is probable that the correction terms δv_i will increase the cost function value above that of the uncorrected values at $X^{ref} + \bar{\alpha}\tilde{X}$. Note that each δv_i contributes to the cost at the beginning and end of arc *i*.

For the Minimum Fuel Problem:

At the beginning of an arc:

$$c_i^{beg}(\delta v_i(a_i)) = \|v_i^{ref} + \alpha_i \tilde{v}_i + \delta v_i(\alpha) - v_i^{-ref}\| - \|v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref}\|$$

$$\leq \|\delta v_i(\alpha_i)\| + \|v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref}\| - \|v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref}\| \quad \text{by the triangle inequality}$$

$$= \|\delta v_i(\alpha_i)\|$$

At the end of an arc:

$$c_i^{end}(\delta v_i(\alpha_i)) = \|v(x_i^{ref} + \alpha_i \tilde{x}_i, v_i^{ref} + \alpha_i \tilde{v}_i + \delta v_i(\alpha_i), t_{i+1}^{ref} + \alpha_i \tilde{t}_{i+1}) - v_{i+1}^{ref}\|$$
$$- \|v(x_i^{ref} + \alpha_i \tilde{x}_i, v_i^{ref} + \alpha_i \tilde{v}_i, t_{i+1}^{ref} + \alpha_i \tilde{t}_{i+1}) - v_{i+1}^{ref}\|$$

$$= \|v(x_i^{ref} + \alpha \tilde{x}_i, v_i^{ref} + \alpha \tilde{v}_i, t_{i+1}^{ref} + \alpha \tilde{t}_{i+1}) + D_{i+1,i}\delta v_i(\alpha_i) + o(\|\delta v(\alpha_i)\|) - v_{i+1}^{ref}\|$$
$$- \|v(x_i^{ref} + \alpha \tilde{x}_i, v_i^{ref} + \alpha_i \tilde{v}_i, t_{i+1}^{ref} + \alpha \tilde{t}_{i+1}) - v_{i+1}^{ref}\|$$

$$\leq \|D_{i+1,i}\delta v_i(\alpha_i)\| + o(\|\delta v(\alpha_i)\|) \quad \text{by the triangle inequality and canceling terms}$$

$$\leq \|D_{i+1,i}\|\|\delta v_i(\alpha_i)\| + o(\|\delta v(\alpha_i)\|) \quad \text{by the definition of the matrix operator norm.}$$

As a result, the total increase to the uncorrected cost for each correction term is:

$$c_i(\delta v_i(\alpha_i)) = c_i^{beg}(\delta v_i(\alpha_i)) + c_i^{end}(\delta v_i(\alpha_i))$$
$$\leq \|\delta v_i(\alpha_i)\| + \|D_{i+1,i}\|\|\delta v_i(\alpha_i)\| + o(\|\delta v(\alpha_i)\|)$$
$$= \left(1 + \|D_{i+1,i}\|\right)\|\delta v_i(\alpha_i)\| + o(\|\delta v(\alpha_i)\|)$$

Thus the total for N arcs is bounded from above by:

$$\sum c_i(\delta v_i(\alpha_i)) \leq N\left(1 + max_i(\|D_{i+1,i}\|)\right) max_i(\|\delta v_i(\alpha_i)\|) + o\left(max_i(\|\delta v(\alpha_i)\|)\right)$$

But since $\|\delta v(\alpha)\| = o(\alpha)$ for any arc from Step 5, including the maximum, and since the $N\left(1 + max_i(\|D_{i+1,i}\|)\right)$ term is a constant, we have that the total increase to the cost from the velocity corrections is $\sum c_i(\delta v_i(\alpha_i)) = o(\alpha)$.

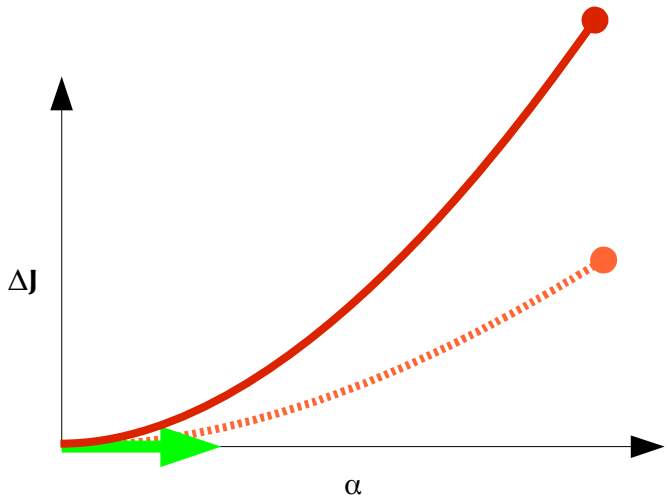


Figure 7.7: Derivative of the increase to the cost function w.r.t. the parameter is 0 since it is the same order as the velocity adjustment magnitude.

For the Minimum Energy Problem:

At the beginning of an arc:

$$c_i^{beg}(\delta v_i(a_i)) = \|v_i^{ref} + \alpha_i \tilde{v}_i + \delta v_i(\alpha) - v_i^{-ref}\|^2 - \|v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref}\|^2$$

$$= \langle v_i^{ref} + \alpha_i \tilde{v}_i + \delta v_i(\alpha) - v_i^{-ref}, v_i^{ref} + \alpha_i \tilde{v}_i + \delta v_i(\alpha) - v_i^{-ref} \rangle - \langle v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref}, v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref} \rangle$$

$$= \langle \delta v_i(\alpha), \delta v_i(\alpha) \rangle + 2\langle \delta v_i(\alpha), v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref} \rangle + \langle v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref}, v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref} \rangle$$
$$- \langle v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref}, v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref} \rangle$$

$$= \langle \delta v_i(\alpha), \delta v_i(\alpha) \rangle + 2\langle \delta v_i(\alpha), v_i^{ref} + \alpha_i \tilde{v}_i - v_i^{-ref} \rangle$$

$$= \langle \delta v_i(\alpha), \delta v_i(\alpha) \rangle + 2\langle \delta v_i(\alpha), \alpha_i \tilde{v}_i \rangle + 2\langle \delta v_i(\alpha), v_i^{ref} - v_i^{-ref} \rangle$$

$$\leq \|\delta v_i(\alpha)\|^2 + 2\|\delta v_i(\alpha)\|\|\alpha_i \tilde{v}_i\| + 2\|\delta v_i(\alpha)\|\|v_i^{ref} - v_i^{-ref}\|$$

which since the first two terms are $o(\|\delta v_i(\alpha)\|)$, and the last term is the norm of the correction multiplied by a constant, this whole term is $O(\|\delta v_i(\alpha)\|) = o(\alpha)$.

At the end of an arc:

$$c_i^{end}(\delta v_i(\alpha_i)) = \|v(x_i^{ref} + \alpha_i \tilde{x}_i, v_i^{ref} + \alpha_i \tilde{v}_i + \delta v_i(\alpha_i), t_{i+1}^{ref} + \alpha_i \tilde{t}_{i+1}) - v_{i+1}^{ref}\|^2$$

$$-\|v(x_i^{ref}+\alpha_i\tilde{x}_i, v_i^{ref}+\alpha_i\tilde{v}_i, t_{i+1}^{ref}+\alpha_i\tilde{t}_{i+1})-v_{i+1}^{ref}\|^2$$

$$=\|v(x_i^{ref}+\alpha\tilde{x}_i, v_i^{ref}+\alpha\tilde{v}_i, t_{i+1}^{ref}+\alpha\tilde{t}_{i+1})+D_{i+1,i}\delta v_i(\alpha_i)+o(\|\delta v(\alpha_i)\|)-v_{i+1}^{ref}\|^2$$
$$-\|v(x_i^{ref}+\alpha\tilde{x}_i, v_i^{ref}+\alpha_i\tilde{v}_i, t_{i+1}^{ref}+\alpha\tilde{t}_{i+1})-v_i^{ref}\|^2$$

which after applying the bi-linearity of the inner product and canceling like above

$$=\langle D_{i+1,i}\delta v_i(\alpha_i)+o(\|\delta v(\alpha_i)\|), D_{i+1,i}\delta v_i(\alpha_i)+o(\|\delta v(\alpha_i)\|)\rangle$$
$$+2\langle D_{i+1,i}\delta v_i(\alpha_i)+o(\|\delta v(\alpha_i)\|), v(x_i^{ref}+\alpha\tilde{x}_i, v_i^{ref}+\alpha_i\tilde{v}_i, t_{i+1}^{ref}+\alpha\tilde{t}_{i+1})-v_i^{ref}\rangle$$

$$=\langle D_{i+1,i}\delta v_i(\alpha_i), D_{i+1,i}\delta v_i(\alpha_i)\rangle+2\langle D_{i+1,i}\delta v_i(\alpha_i), v(x_i^{ref}+\alpha\tilde{x}_i, v_i^{ref}+\alpha_i\tilde{v}_i, t_{i+1}^{ref}+\alpha\tilde{t}_{i+1})-v_i^{ref}\rangle$$
$$+o(\|\delta v(\alpha_i)\|)$$

$$\le\|D_{i+1,i}\|^2\|\delta v_i(\alpha_i)\|^2+2\langle D_{i+1,i}\delta v_i(\alpha_i), v(x_i^{ref}+\alpha\tilde{x}_i, v_i^{ref}+\alpha_i\tilde{v}_i, t_{i+1}^{ref}+\alpha\tilde{t}_{i+1})-v_i^{ref}\rangle+o(\|\delta v(\alpha_i)\|)$$

$$=2\langle D_{i+1,i}\delta v_i(\alpha_i), v(x_i^{ref}+\alpha\tilde{x}_i, v_i^{ref}+\alpha_i\tilde{v}_i, t_{i+1}^{ref}+\alpha\tilde{t}_{i+1})-v_i^{ref}\rangle+o(\|\delta v(\alpha_i)\|)$$

$$=2\langle D_{i+1,i}\delta v_i(\alpha_i), v(x_i^{ref}+\alpha\tilde{x}_i, v_i^{ref}+\alpha_i\tilde{v}_i, t_{i+1}^{ref}+\alpha\tilde{t}_{i+1})-v_i^{-ref}+v_i^{-ref}-v_i^{ref}\rangle+o(\|\delta v(\alpha_i)\|)$$

$$=2\langle D_{i+1,i}\delta v_i(\alpha_i), v(x_i^{ref}+\alpha\tilde{x}_i, v_i^{ref}+\alpha_i\tilde{v}_i, t_{i+1}^{ref}+\alpha\tilde{t}_{i+1})-v_i^{-ref}\rangle$$
$$+2\langle D_{i+1,i}\delta v_i(\alpha_i), v_i^{-ref}-v_i^{ref}\rangle+o(\|\delta v(\alpha_i)\|)$$

$$\le 2\|D_{i+1,i}\delta v_i(\alpha_i)\|\|v(x_i^{ref}+\alpha\tilde{x}_i, v_i^{ref}+\alpha_i\tilde{v}_i, t_{i+1}^{ref}+\alpha\tilde{t}_{i+1})-v_i^{-ref}\|$$
$$+2\|D_{i+1,i}\delta v_i(\alpha_i)\|\|v_i^{-ref}-v_i^{ref}\|+o(\|\delta v(\alpha_i)\|)$$

$$\le 2\|D_{i+1,i}\|\|\delta v_i(\alpha_i)\|\|v(x_i^{ref}+\alpha\tilde{x}_i, v_i^{ref}+\alpha_i\tilde{v}_i, t_{i+1}^{ref}+\alpha\tilde{t}_{i+1})-v_i^{-ref}\|$$
$$+2\|D_{i+1,i}\|\|\delta v_i(\alpha_i)\|\|v_i^{-ref}-v_i^{ref}\|+o(\|\delta v(\alpha_i)\|)$$

$$\le O(\|\delta v_i(\alpha_i)\|)O(\alpha)+O(\|\delta v(\alpha_i)\|)+o(\|\delta v(\alpha_i)\|)$$

$$\le O(\|\delta v(\alpha_i)\|)$$

Thus the increases to the cost due to each correction term is $O(\|\delta v_i(\alpha)\|)=o(\alpha)$. As was the case for the min fuel problem, the sum of $o(\alpha)$ terms is still $o(\alpha)$.

Step 8: Change to the Cost After Correction

From Step 4 the decrease in the cost function is $O(\alpha)$ and from Step 7 the increase to

this value to restore continuity is $o(\alpha)$. Thus there exists some $\bar{\alpha}>0$ such that if $\alpha_i<\bar{\alpha}, i=1,\ldots,N,$ then $J(X^{\text{ref}})-J(X^{\text{ref}}+\alpha\tilde{X})>\sum c_i(\delta v_i(\alpha_i))$ and so:

$$J(X^{\text{ref}})>J(X^{\text{ref}}+\alpha\tilde{X})+\sum c_i(\delta v_i(\alpha_i))=J(X^{\text{ref}}+\alpha\tilde{X}+\vec{\delta}v)$$

This means that the corrected transfer has strictly lower cost than the previous iteration after continuity has been restored. Since a range of solutions exists, a backstepping or other inexact line search method may be performed [Luen08], terminating when the continuity correction succeeds and the resulting cost is decreased sufficiently.

Therefore, at every outer iteration, this process leads to a feasible transfer with a decrease in the cost for any non-optimal but feasible initial guess. Since the result of each step is itself is feasible, this result shows the iterative process leads to global descent with feasible iterates.

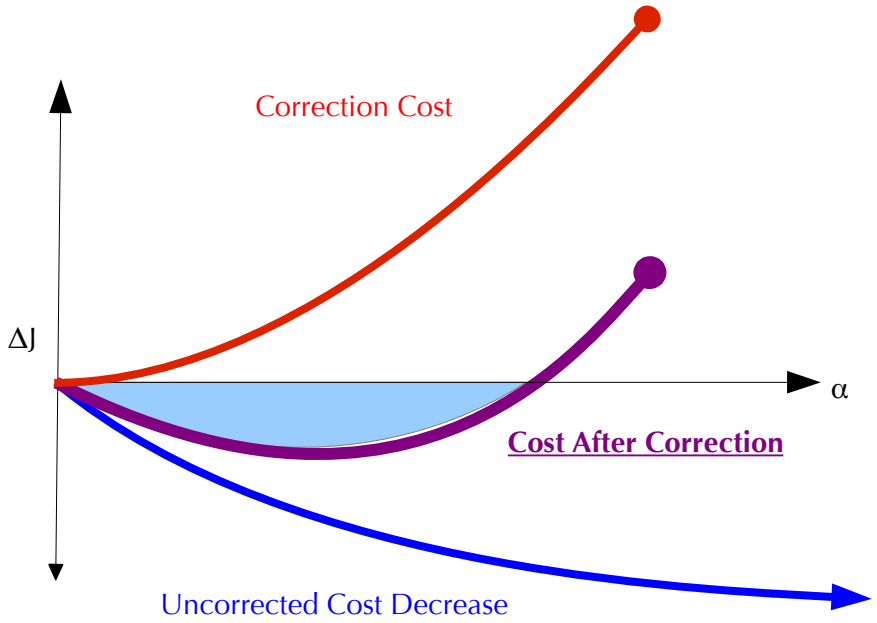


Figure 7.8: The zero derivative to the correction cost and the negative derivative of the cost decrease before correction w.r.t. the search parameter imply the total cost after correction has strictly negative derivative w.r.t. the search parameter. Thus an open solution set exists.

Note that if the cost is a weighted sum of fuel/energy terms and total transfer time, then since the correction is only to velocity and not the coasting times, the transfer time terms will be unaffected by the correction process.

7.3.2 Additional Constraints

The above proof considered the case where the only constraint is continuity. The additional constraints must be shown to satisfy the above method.

Projection and Trust Region Constraints:

These are helper constraints for the CP only which have no corresponding constraints in the NLP that need to hold.

Time Constraints:

Min/max coasting time, and fixed total transfer time constraints are satisfied automatically. This is because these constraints are on the variables themselves, so no approximation is involved. Thus if they are feasible in the CP they are feasible in the NLP. They are also unchanged in the velocity correction process. Due to their convex nature, the inequalities are satisfied for any value of alpha from 0 to 1, due to the definition of a convex set. For the fixed total transfer time, as the constraint is linear, then clearly any value of alpha from 0 to 1 as well.

Maximum Delta V Per Maneuver:

Recall this constraint is approximated by

$$\left\| \begin{bmatrix} \dots & -C_{i+1,i} & -D_{i+1,i} & -a_{i+1}^{\text{ref}} & 0_{3x3} & I_{3x3} & \dots \end{bmatrix} \tilde{X} + \left(v_{i+1}^{\text{ref}} - v_{i+1}^{\text{- ref}} \right) \right\| \leq \Delta V_{max}.$$

Note that though this constraint can made quadratic by squaring both sides, here it is in the format of a Second Order Cone constraint. Regardless, it must be determined that there is a range of feasible solutions in the original problem when moving in the direction of the CP solution.

If the reference solution satisfied the inequality constraint strictly, then there is some open

ball with radius *r* around the reference where the constraint is still strictly satisfied. Then for line search parameter in the range $0<\alpha_i<min(\bar{\alpha},r)$ there will be a decrease to the cost after restoring continuity and while strictly satisfying this constraint. This is true for any inequality constraint that is strictly satisfied at the reference.

Turn now to the case where the constraint is met with equality at the reference. In this case, the CP constraint function value must satisfy $\tilde{g}(X^{\text{ref}}+\tilde{X})\leq\tilde{g}(X^{\text{ref}})=0$. The augmented constraints of Section 5.8 guarantee that a nonzero change in the variables involved in the constraint will lead to a nonzero change in the impulse. In its quadratic form, the constraint is strictly convex w.r.t. changes in the impulse, thus the direction defined by the CP solution is a descent direction since $\nabla\tilde{g}(X^{\text{ref}})\tilde{X}<\tilde{g}(X^{\text{ref}}+\tilde{X})-\tilde{g}(X^{\text{ref}})\leq 0$ by the FONC for strict convexity. In the Second Order Cone form above, any decrease in the constraint function implies the direction of the CP solution is a descent direction since $\nabla\tilde{g}(X^{\text{ref}})\tilde{X}\leq\tilde{g}(X^{\text{ref}}+\tilde{X})-\tilde{g}(X^{\text{ref}})<0$. The last case is when the CP solution satisfies $\tilde{g}(X^{\text{ref}}+\tilde{X})=\tilde{g}(X^{\text{ref}})=0$. For convex but not strictly convex functions, it cannot be deduced that the CP solution defines a descent direction as only $\nabla\tilde{g}(X^{\text{ref}})\tilde{X}\leq\tilde{g}(X^{\text{ref}}+\tilde{X})=0$ is guaranteed. However, a further analysis of this particular constraint shows that the CP solution defines descent direction even in this case. Consider that if both the reference and the CP solution meet the CP maximum norm constraint, both points distinct points residing on the boundary of a closed ball. The segment connecting these two points is a chord which is interior to the ball everywhere but the endpoints. Thus while $\tilde{g}(X^{\text{ref}}+\tilde{X})=\tilde{g}(X^{\text{ref}})$, it must be the case that $\tilde{g}(X^{\text{ref}}+\frac{1}{2}\tilde{X})\leq\tilde{g}(X^{\text{ref}})$. From this, $\nabla\tilde{g}(X^{\text{ref}})\frac{1}{2}\tilde{X}<\tilde{g}(X^{\text{ref}}+\frac{1}{2}\tilde{X})-\tilde{g}(X^{\text{ref}})<0$ and so indeed the CP solution does still define a descent direction.

When the constraint was met with equality at the reference, unless the variables involved

are unchanged, the CP solution defines a descent direction of the original constraint function. Thus, like with the cost, the decrease to the original constraint function g when moving in the CP solution direction is $O(\alpha_i)$. As was shown in the cost function analysis, the increase in single maneuver magnitude due to the velocity correction is $o(\alpha_i)$. Thus there is some $\beta_i>0$ for each maneuver such that moving in the CP direction results in a decrease to the constraint function of the original problem. Take $\bar{\beta}=min\{\beta_i\}$. Then there exists a range of α values such that if $0<\alpha_i<min\{\bar{\alpha},\bar{\beta}\}, i=1,\ldots,N,$ then both:

$$J(X^{\text{ref}})>J(X^{\text{ref}}+\alpha\tilde{X}+\vec{\delta}v)$$
$$g_{\Delta V_{max}}(X^{\text{ref}})>g_{\Delta V_{max}}(X^{\text{ref}}+\alpha\tilde{X}+\vec{\delta}v)$$

Thus there is a range of step sizes where both the cost and constraint function are reduced when continuity is reestablished, i.e. there is descent of the cost function while preserving feasibility.

Minimum Delta V per Maneuver

The analysis is identical to the above, both being ball constraints on the velocities.

Impact Avoidance

As with the maximum impulse magnitude constraint, if the reference satisfies the constraint strictly, then there is some open ball with radius r around the reference where the constraint is still strictly satisfied in the NLP. Then for line search parameter in the range $0<\alpha_i<min(\bar{\alpha},r)$ there will be a decrease to the cost after restoring continuity and while strictly satisfying this constraint. Also, just like the maximum impulse constraint, it can be shown that any nonzero step from the reference value is a descent direction for the constraint function. The same conclusion will hold if the change to the distance to the center of mass due to correction is $o(\alpha_i)$.

Since $\|\delta v(\alpha_i)\|=\|\gamma(\alpha_i)\|=o(\alpha_i)$ it suffices to show that the change to the distance to the center of mass due to the velocity correction, let's call it $q_i(\delta v_i(\alpha_i)),$ is of the same order as

$\|\delta v(\alpha_i)\|$.

$$
\begin{aligned}
q_i(\delta v_i(\alpha_i)) = &\|x(x_i^{\text{ref}} + \alpha_i \tilde{x}_i, v_i^{\text{ref}} + \alpha_i \tilde{v}_i + \delta v_i(\alpha_i), t_i^p) - x_i^{ref}(t_i^{p,ref})\| \\
&- \|x(x_i^{\text{ref}} + \alpha_i \tilde{x}_i, v_i^{\text{ref}} + \alpha_i \tilde{v}_i, t_i^p) - x_i^{ref}(t_i^{p,ref})\| \\
&= \|x(x_i^{\text{ref}} + \alpha \tilde{x}_i, v_i^{\text{ref}} + \alpha \tilde{v}_i, t_i^p) + B_i^p \delta v_i(\alpha_i) + o(\|\delta v(\alpha_i)\|) - x_i^{ref}(t_i^{p,ref})\| \\
&- \|x(x_i^{\text{ref}} + \alpha \tilde{x}_i, v_i^{\text{ref}} + \alpha_i \tilde{v}_i, t_i^p) - x_i^{ref}(t_i^{p,ref})\| \\
&\leq \|x(x_i^{\text{ref}} + \alpha \tilde{x}_i, v_i^{\text{ref}} + \alpha \tilde{v}_i, t_i^p) - x_i^{ref}(t_i^{p,ref})\| + \|B_i^p \delta v_i(\alpha_i) + o(\|\delta v(\alpha_i)\|)\| \\
&- \|x(x_i^{\text{ref}} + \alpha \tilde{x}_i, v_i^{\text{ref}} + \alpha_i \tilde{v}_i, t_i^p) - x_i^{ref}(t_i^{p,ref})\| \\
&= \|B_i^p \delta v_i(\alpha_i)\| + o(\|\delta v(\alpha_i)\|) \\
&\leq \|B_i^p\| \|\delta v_i(\alpha_i)\| + o(\|\delta v(\alpha_i)\|) \quad \text{by the definition of the matrix operator norm.}
\end{aligned}
$$

Thus we have that $q_i(\delta v_i(\alpha_i))$ is bounded by a constant multiplied by $\|\delta v(\alpha_i)\|$, and so they are the same order, which has been shown to be $o(\alpha_i)$. Since the convex subproblem solution is a descent direction of the constraint function in this case, the function value decreases $O(\alpha_i)$, but can increase due the correction term at $o(\alpha_i)$. Thus there is a range of search values where the constraint function decreases in the NLP. Intersecting this open set of values with those needed for the other constraints and cost descent gives an open set where all are satisfied.

Maximum Thrust Angle:

See Appendix B.

Summary

Additional constraints have been shown to either be unaffected by the velocity correction, or for each g whose terms change from the reference, there exists an α_g such that $\alpha<\alpha_g$ guarantees the constraint holds after correction. Since there are a finite number of constraints and one such value for the cost function, for $0<\alpha<\alpha_{desc} = min\{\alpha_{g1}, \alpha_{g2}, \ldots, \bar{\alpha}\}$ the post-correction transfer is

feasible and has lower cost.

7.4 Additional Algorithmic Implications

7.4.1 Simplification of Subproblems Solved for Better Performance

Without the knowledge gained from the investigations involved in this project, the initial structure for each major iteration was solving a CP with total ΔV as the cost to determine the search direction, and then a sequence of constrained CPs with discontinuity as the cost at each candidate of the line search to reestablish feasibility. An important detail in the above proof from the Implicit Function Theorem states that within the domain the theorem applies there is a *unique* velocity correction (or velocity and time if adding an apsis constraint) that reestablishes continuity, which by the proof will additionally satisfy the other constraints if $\alpha \leq \bar{\alpha}$. This means that any correction that yields continuity within this range will automatically satisfy the other constraints regardless if they are included in that problem or not. Thus, rather than needing a sequence of constrained CP problems to reestablish continuity, these may be unconstrained, aside from the beneficial addition of some sort of trust region.

In order to solve faster subproblems, by replacing the sum of discontinuities with the sum of squares, and using box constraints to define trust regions, the problem of reducing $J_{cont} = \|x_1 - \bar{x}_1\|^2 + \ldots + \|x_N - \bar{x}_N\|^2$ to zero may be solved by a sequence of problems with quadratic cost

$$\tilde{J}_{cont} = \sum_{i=0}^{N-1} \left\| \left(x_{i+1}^{\text{ref}} + \tilde{x}_{i+1} \right) - \left(x_{i+1}^{-ref} + \begin{bmatrix} A_{i+1,1} & B_{i+1,1} & v_{i+1}^{-ref} \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \\ \tilde{t}_{i+1} \end{bmatrix} \right) \right\|^2$$

$$= \sum_{i=0}^{N-1} \left\| \left(x_{i+1}^{ref} - x_{i+1}^{-ref} \right) - \begin{bmatrix} A_{i+1,1} & B_{i+1,1} & v_{i+1}^{-ref} & I \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \\ \tilde{t}_{i+1} \\ \tilde{x}_{i+1} \end{bmatrix} \right\|^2$$

and no constraints other than trust region linear inequality constraints:

$$-x_{max} \le \tilde{x}_{i(j)} \le x_{max}$$
$$-v_{max} \le \tilde{v}_{i(j)} \le v_{max}$$
$$-t_{max} \le \tilde{t}_{i(j)} \le t_{max}$$

As a result, for each major iteration a single SOCP/QCQP is used to establish a search direction followed by unconstrained SQP to reestablish continuity during the line search process may be used.

7.4.2 Conditions for Elimination of the Line Search

It may be desirable to eliminate the line search portion of the algorithm and rely solely on reducing the trust region if restoring continuity fails or the true cost is nor improved. To motivate why this may be possible, consider that in the above line search algorithm, for approximation solution $\tilde{X}^*$ an α^* is found such that $X = X^{ref} + \alpha^* \tilde{X}^* + \delta v(\alpha^*)$ yields a feasible trajectory with decreased cost. Suppose instead of solving $\text{CP}_{\tilde{X}}$ – the problem that yields $\tilde{X}^*$ as a solution – a new problem $\text{CP}_{\tilde{Y}}$ is solved that is identical to $\text{CP}_{\tilde{X}}$ but with the additional constraint that $\|\tilde{X}\|_2 \le \alpha^*$. If this were the case, $\alpha^* \tilde{X}^*$ provides a feasible point, but not necessarily the optimal solution, $\tilde{Y}^*$. If $\tilde{Y}^* = \alpha^* \tilde{X}^*$, then it is known by the results of the theorem that there is a feasible solution with decreased cost $X = X^{ref} + \tilde{Y}^* + \delta v(\alpha^*)$. Otherwise $\tilde{Y}^*$ is a solution to $\text{CP}_{\tilde{Y}}$ that also satisfies all of the approximate constraints as does $\alpha^* \tilde{X}^*$, but with a lower approximate cost. One would hope that conditions may exist where $X = X^{ref} + \tilde{Y}^* + \delta v_Y(1)$ provides a feasible

solution with decreased cost without a need for line search to be performed.

Let $\tilde{X}(\beta)$ be the solution(s) to the problem CP_β, where CP_β is the same as the standard CP defined in Chapter 5, with the change that the trust region constraints are all multiplied by β. Thus $CP_{\beta=1}$ means no change to the problem definition, and $CP_{\beta=0}$ means that $\tilde{X}=0$ is the only allowable solution. Assume there is some neighborhood of 0 such that $\tilde{X}(\beta)$ is a well-defined, C^1 function. [It would not be continuously differentiable in general even if well defined, as the reduced trust region causes certain constraints to become inactive there may be kinks in the path. But here it is assumed that there is some small neighborhood where finite number of activation changes no longer occurs.]

For each patch point, between arcs *i* and *i+1*, let $\hat{\kappa}_i(\beta,\delta v_i):\mathbb{R}^4\to\mathbb{R}^3$ be defined by:

$$\hat{\kappa}_i(\beta,\delta v_i)=x(x_i^{\text{ref}}+\tilde{x}_i(\beta),v_i^{\text{ref}}+\tilde{v}_i(\beta)+\delta v_i,t_{i+1}^{\text{ref}}+\tilde{t}_{i+1}(\beta))-(x_{i+1}^{\text{ref}}+\tilde{x}_{i+1}(\beta))$$

where $\tilde{x}_i(\beta)$, etc. are the components of $\tilde{X}(\beta)$. As before, these functions output the discontinuities in the original dynamics. The velocity correction term is also the same. The difference is that each term includes as inputs solutions to a scaled CP rather than a scaled solution to a single CP.

These functions also have the property that from the continuous reference trajectory defined by $(\beta,\delta v_i)=(0,\vec{0})$, we have $\hat{\kappa}_i(0,\vec{0})=x(x_i^{\text{ref}},v_i^{\text{ref}},t_{i+1}^{\text{ref}})-(x_{i+1}^{\text{ref}})=x_{i+1}^{\text{ref}}-(x_{i+1}^{\text{ref}})=\vec{0}$. Next, note that they also still satisfy $\frac{\partial\hat{\kappa}_i}{\partial(\delta v_i)}(0,\vec{0})=B_{i+1,1}$, which is assumed to be invertible. Since it is assumed here that we are within a neighborhood where $\tilde{X}(\beta)$ is a well-defined C^1 function, the $\hat{\kappa}_i$ are C^1 as well. Thus the Implicit Function Theorem may be invoked again and so there is a range of β and function $\delta v_i(\beta)$ such that $\hat{\kappa}_i(\beta,\delta v_i(\beta))=\vec{0}$. Additionally, we have the

derivative expression $\frac{\partial \delta v_i}{\partial \beta}(0) = -\left(\frac{\partial \hat{\kappa}_i}{\partial(\delta v_i)}(0,\vec{0})\right)^{-1} \frac{\partial \hat{\kappa}_i}{\partial \beta}(0,\vec{0}).$

$$\frac{\partial \hat{\kappa}_i}{\partial \beta}(\beta,\vec{0}) = \begin{bmatrix} \frac{\partial \tilde{x}_{i+1}}{\partial \tilde{x}_i} & \frac{\partial \tilde{x}_{i+1}}{\partial \tilde{v}_i} & \frac{\partial \tilde{x}_{i+1}}{\partial \tilde{t}_{i+1}} \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{x}_i}{\partial \beta}(\beta) \\ \frac{\partial \tilde{v}_i}{\partial \beta}(\beta) \\ \frac{\partial \tilde{t}_{i+1}}{\partial \beta}(\beta) \end{bmatrix} - \frac{\partial \tilde{x}_{i+1}}{\partial \beta}(\beta)$$

and so

$$\frac{\partial \hat{\kappa}_i}{\partial \beta}(0,\vec{0}) = \begin{bmatrix} A_{i+1,1} & B_{i+1,1} & v_{i+1}^{-ref} \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{x}_i}{\partial \beta}(0) \\ \frac{\partial \tilde{v}_i}{\partial \beta}(0) \\ \frac{\partial \tilde{t}_{i+1}}{\partial \beta}(0) \end{bmatrix} - \frac{\partial \tilde{x}_{i+1}}{\partial \beta}(0).$$

In order to evaluate this, consider the function

$$\rho_i(\beta) = \begin{bmatrix} A_{i+1,1} & B_{i+1,1} & v_{i+1}^{-ref} \end{bmatrix} \begin{bmatrix} \tilde{x}_i(\beta) \\ \tilde{v}_i(\beta) \\ \tilde{t}_{i+1}(\beta) \end{bmatrix} - \tilde{x}_{i+1}(\beta).$$

Which has derivative:

$$\frac{\partial \rho_i}{\partial \beta}(\beta) = \begin{bmatrix} A_{i+1,1} & B_{i+1,1} & v_{i+1}^{-ref} \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{x}_i}{\partial \beta}(\beta) \\ \frac{\partial \tilde{v}_i}{\partial \beta}(\beta) \\ \frac{\partial \tilde{t}_{i+1}}{\partial \beta}(\beta) \end{bmatrix} - \frac{\partial \tilde{x}_{i+1}}{\partial \beta}(\beta).$$

Note that while in general $\frac{\partial \hat{\kappa}_i}{\partial \beta}(\beta,\vec{0}) \neq \frac{\partial \rho_i}{\partial \beta}(\beta),$ it is the case that $\frac{\partial \hat{\kappa}_i}{\partial \beta}(0,\vec{0}) = \frac{\partial \rho_i}{\partial \beta}(0).$ Returning to $\rho_i(\beta),$ since each $\tilde{X}(\beta)$ is a solution to a convex subproblem with equality constraints

$\begin{bmatrix} A_{i+1,1} & B_{i+1,1} & v_{i+1}^{-ref} \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \\ \tilde{t}_{i+1} \end{bmatrix} - \tilde{x}_{i+1} = 0$, we see that $\rho_i(\beta) = \vec{0} \ \forall \beta$. Since the derivative of a constant function is 0, $\frac{\partial \hat{\kappa}_i}{\partial \beta}(0, \vec{0}) = \frac{\partial \rho_i}{\partial \beta}(0) = \vec{0}$.

As a result, $\frac{\partial \delta v_i}{\partial \beta}(0) = -\left(\frac{\partial \hat{\kappa}_i}{\partial (\delta v_i)}(0, \vec{0}) \right)^{-1} \frac{\partial \hat{\kappa}_i}{\partial \beta}(0, \vec{0}) = \vec{0}$, from which $\|\delta v(\beta)\| = o(\beta)$. As with the original theorem, this implies the increase to the cost due to the velocity correction is itself $o(\beta)$. If it is the case that $J(X^{ref}) - J(X^{ref} + \tilde{X}(\beta)) = O(\beta)$, then following the original proof it will be the case that there is a range of β such that $J(X^{ref}) - J(X^{ref} + \tilde{X}(\beta) + \delta v(\beta)) > 0$. To see this, note that $\tilde{J}(\beta \tilde{X}(1)) \geq \tilde{J}(\tilde{X}(\beta)) \ \forall \beta$, so $\frac{\tilde{J}(\beta \tilde{X}(1)) - J(0 \cdot \tilde{X}(1))}{\beta} \geq \frac{\tilde{J}(\tilde{X}(\beta)) - \tilde{J}(\tilde{X}(0))}{\beta} \ \forall \beta$ since $0 \cdot \tilde{X}(1) = 0 = \tilde{X}(0)$ Then since $\tilde{X}$ is a descent direction:

$$0 > \nabla \tilde{J}(0) \tilde{X} = \lim_{\beta \to 0^+} \frac{\tilde{J}(\beta \tilde{X}(1)) - J(0 \cdot \tilde{X}(1))}{\beta} \geq \lim_{\beta \to 0^+} \frac{\tilde{J}(\tilde{X}(\beta)) - \tilde{J}(\tilde{X}(0))}{\beta} = \frac{d \tilde{X}}{d \beta} \Big|_{\beta = 0}$$

Thus $J(X^{ref}) - J(X^{ref} + \tilde{X}(\beta)) = O(\beta)$, as desired. This along with $\|\delta v(\beta)\| = o(\beta)$ gives the result that there is a range of β where a velocity correction exists to restore continuity and the cost of the corrected, continuous transfer is less than that of the previous iteration. In this case this is done through reducing the size of the trust region, with no line search added.

Regarding constraints other than continuity, the trust region and coasting time constraints are satisfied automatically. This is because these constraints are on the variables themselves, so no approximation is involved. Thus if they are feasible in the CP they are feasible in the NLP. Constraints that involve propagation such as the velocity or nonimpact constraints are more problematic. If they are satisfied with inequality at the reference, then β may simply be reduced (if needed) so that both $J(X^{ref}) - J(X^{ref} + \tilde{X}(\beta) + \delta v(\beta)) > 0$ and these constraints are still satisfied,

intersecting the two open sets where the cost drops after restoring continuity and the constraint functions are negative. If either of these constraints are met with equality only at the reference, then due to differences between the true constraint functions and the approximations, then the convex solution vector (or fixed fraction thereof) cannot be guaranteed to satisfy the true constraints. In this case, line search must be added to find a valid step length.

Thus the conditions necessary to eliminate line search for a particular iteration are as follows: a C^1 curve of solutions parameterized by β exists within some neighborhood of the reference (assumed), and that the propagated constraints are met with inequality at the reference. It is unlikely the constraint condition will hold throughout the optimization process, but it is still may be useful in some specific iterations.

7.5 Summary

The algorithm that optimizes feasible initial guesses defined in Chapter 6 has been shown to be a global descent method that produces feasible iterates with very minimal assumptions. The proof relies heavily on the first-order conditions of convex functions, the nature of the inequality constraints satisfied with equality forcing variable changes to be in descent directions, and the use of the Implicit Function Theorem to bound the magnitude of the velocity correction terms.

Implications of the proof show that the correction term may be found with an unconstrained optimization problem for faster computation, and that the convexity and nonlinearity of the approximate problem inequality constraints on propagated variables is necessary for a correction based method to guarantee feasible iterates.

---

[1] The invertibility has been assumed throughout the literature on differential correction, such as [Wils98, Sukh04, Marc07]. While no invertibility issues were encountered in testing and no conditioning issues arose in the Phobos tests

(Ch.10), a couple of poorly conditioned matrices were encountered during the NEA interceptor testing (Ch.12). Conditioning issues due to high sensitivity can be resolved using a multiple shooting approach, as is common practice. It is important to note that use of a multiple shooting approach fits well within this theoretical framework, and it is shown in Appendix C in the section on Maneuverless Patch Points that there is no theoretical impact to the descent proof of this chapter by making this practical change in implementation. Also, it was observed in the NEA application that the poorly conditioned matrices involved arcs passing very near the Earth, which in the standard CR3BP coordinates places the trajectory quite near the gravitational singularity. Therefore sside from the use of multiple shooting, it is expected that the use of regularization to remove such a singularity such as in [Nakh13] for Hill's model would also help.

Returning to the issue of invertibility itself, it was suggested in [Wils13] that at least for the relevant models, a change of frame could resolve the issue. However, even if this is not the case or is undesirable, the issue may be resolved by adjusting the block matrix used. Due to the invertibility of the state transition matrix, we know that even if $B_{i+1,i}$ were not invertible with $rank(B_{i+1,i})=2$, that the top three rows must have rank=3. Therefore we may select one of the columns of $A_{i+1,i}$ and adjoin it to two of the indepdendent columns of $B_{i+1,i}$ in order to form a full rank $\hat{B}_{i+1,i}$. (This approach is similar to [Sche06] where alternative submatrices may be selected for their generating function definitions in case of singularity issues) Suppose the columns index of these three vectors are m_1, m_2, m_3 so that the columns of $\hat{B}_{i+1,i}$ are $\vec{c}_{m1}, \vec{c}_{m2}, \vec{c}_{m3}$. Let $\hat{e}_{m1}, \hat{e}_{m2}, \hat{e}_{m3}$ be the corresponding unit length basis vectors in $\mathbb{R}^6$ where the first two are taken to be associated with velocity columns and the last with the position column. Let $\vec{w}=(w_1, w_2, w_3)\in\mathbb{R}^3$. Define a modified discontinuity function $\kappa:\mathbb{R}^4\to\mathbb{R}^3$ given by:

$$\kappa_i(\alpha_i, \vec{w})=x_{i+1}^{-}(\alpha_i, \vec{w})-x_{i+1}(\alpha_i)=x(x_i^{ref}+\alpha_i\tilde{x}_i+w_1\hat{e}_{m1}, v_i^{ref}+\alpha_i\tilde{v}_i+w_2\hat{e}_{m2}+w_3\hat{e}_{m2}, t_{i+1}^{ref}+\alpha_i\tilde{t}_{i+1})-(x_{i+1}^{ref}+\alpha_i\tilde{x}_{i+1}).$$

Then at $(\alpha_i, \vec{w})=(0,\vec{0})$ we have that $\kappa_i(0,\vec{0})=x(x_i^{ref}, v_i^{ref}, t_{i+1}^{ref})-(x_{i+1}^{ref})=x_{i+1}^{-ref}-(x_{i+1}^{ref})=\vec{0}$ and that by the chain rule $\frac{\partial\kappa_i}{\partial w_1}(0,\vec{0})=B_{i+1,i}\hat{e}_{m1}=\vec{c}_{m1}, \frac{\partial\kappa_i}{\partial w_2}(0,\vec{0})=B_{i+1,i}\hat{e}_{m2}=\vec{c}_{m2}, \frac{\partial\kappa_i}{\partial w_3}(0,\vec{0})=A_{i+1,i}\hat{e}_{m3}=\vec{c}_{m3}$ so that finally $\frac{\partial\kappa_i}{\partial\vec{w}}(0,\vec{0})=\hat{B}_{i+1,i}$ which is the augmented and invertible matrix defined above. As in the original case the Implicit Function Theorem can be applied so that there are open sets within which $\vec{w}=\vec{w}(\alpha)$ and for which $\kappa_i(\alpha_i, \vec{w}(\alpha))=\vec{0}$ and so continuity holds. The above change does not affect the calculation that

$$\frac{\partial\kappa_i}{\partial\alpha_i}(0,\vec{0})=\begin{bmatrix}A_{i+1,1} & B_{i+1,1} & v_{i+1}^{-ref}\end{bmatrix}\begin{bmatrix}\tilde{x}_i\\ \tilde{v}_i\\ \tilde{t}_{i+1}\end{bmatrix}-\tilde{x}_{i+1}=\vec{0}$$

for CP solutions, and so the result $\vec{w}(\alpha)=o(\alpha)$ holds. From here the proof continues as before, with the addition that the magnitude bounds include constant $\|C_{i+1,i}\|$ terms added to the $\|D_{i+1,i}\|$ terms, which does not change the order. One exception can arise in the case the position at the beginning of the arc is already at the boundary of the forbidden/minimum radius region. In this case a unit vector tangential to the minimum velocity sphere should be chosen rather than $\hat{e}_{m3}$ so that the needed adjustments cannot lead to infeasibility. In terms of calculation sequence, such a change requires that the recorrection be done from the last arc to the first so that the prior arc's recorrection process is aware of the initial position change needed for arcs in this special case.

Chapter 8

Zangwill's Global Convergence Theorem

8.1 Introduction

In the last chapter it was shown that the proposed algorithm is a global descent method on the set of feasible trajectories. That is, if the current iterate is a KKT point the process will stop, otherwise the next iterate will be feasible with strictly lower cost. It remains to be shown that these steps progress sufficiently that their limit converges to a KKT point. Note that the focus here is not on a local convergence rate once close to a solution, but showing that the method will indeed drive any (regular) feasible initial guess towards some KKT point. We will show this by proving that the requirements of Zangwill's Global Convergence Theorem are satisfied. The theorem is stated in the case of our application where KKT points of the NLP are desired and the iterates are feasible.

Zangwill's Global Convergence Theorem: [Zang69, Luen08]

Let Ω_f be the set of points defining trajectories feasible in the NLP. Let Ξ be the set of points satisfying the KKT conditions for the NLP. Let $\{X_{(l)}^{ref}\}_{l=0}^{\infty}$ be the sequence of points generated

by the iterative algorithm $X^{ref}_{(l+1)} \in A(X^{ref}_{(l)})$. in chapter 6. Suppose that:

C1) $\{X^{ref}_{(l)}\}_{l=0}^{\infty}$ is contained within a compact subset of Ω_f.

C2) There is a continuous function *J* (here the cost function) such that:

a. if $X^{ref}_{(l)} \notin \Xi$, then $J(Y) < J(X^{ref}_{(l)}) \forall\, Y \in A(X^{ref}_{(l)})$.

b. if $X^{ref}_{(l)} \in \Xi$, then $J(Y) \leq J(X^{ref}_{(l)}) \forall\, Y \in A(X^{ref}_{(l)})$.

C3) The mapping *A* is closed on $\Omega_f \setminus \Xi$, where closed is defined below in Section 8.4.3.

Then the limit of any convergent subsequence is a member of Ξ, i.e. a KKT point of the original NLP.

8.2 Iterates Are Within A Compact Domain

The position terms of any iterate are bounded in position space. The distance from the body center is bounded from below by the minimum radius $\|x_i^{ref} - x_{cm}\| \geq R_{min}$, and from above either by adding an explicit convex constraint $\|x_i^{ref} - x_{cm}\| \leq R_{max}$, or by relying on the closed boundary associated with any maximum feasible energy level. Call this closed and bounded subset $P_i \subset \mathbb{R}^3$. The set of all variables $X \in \mathbb{R}^{7N}$ is closed due to the standard product topology on $\mathbb{R}^n$ since the set defined by each constraint is $P_i \times \mathbb{R}^{7N-3}$ is the product of two closed sets. Similarly for maximum velocity, where within the bounded positions there is a limit on the velocities due to a maximum feasible energy level, or some explicit convex constraint may be added. Again, the resulting sets are closed, and bound the velocities. The coasting time variables are limited by the min/max coasting time constraints, thus are bounded in a closed set. The intersection of these closed sets is closed, and taken all together bound every variable. Thus the intersection Σ is a closed and bounded set. $\Sigma \cap \Omega_f$ is then a closed (on the subset topology on

$\Omega_f \subset \mathbb{R}^{7N}$) and bounded subset of $\Omega_f \subset \mathbb{R}^{7N}$, thus a compact subset of Ω_f.

8.3 The Cost is a Global Descent Function of the Algorithm

This is the primary result of Chapter 7, as stated in 7.2. Thus here it is assumed that the initial condition satisfies one of the several allowable constraint qualifications.

8.4 The Algorithm is Closed

8.4.1 General Point to Set Maps

As the Second Order Sufficient Conditions are not necessarily satisfied for the minimum fuel problem, the solution set may not be unique. Additionally, depending on starting parameters, inexact line search methods may not return the same single point. Thus point to set maps and generalizations of the concept of continuity are needed.

Point to set mappings are generally stated within the framework of metric spaces. In our case Λ, D are subsets of $\mathbb{R}^q, \mathbb{R}^r$. A point to set map takes point in Λ and associates a subset of *D*, hence an element of the power set of *D*. Thus $\Gamma : \Lambda \to 2^D$. For such maps, there are the following definitions:[Bank83]

Closed: Γ is closed at $\bar{u}$ if for each pair of sequences $\{u_m\}_{m=1}^{\infty} \subset \Lambda, \{x_m\}_{m=1}^{\infty} \subset D$ such that $u_m \to \bar{u}, x_m \in \Gamma(u_m) \forall m, x_m \to x_\infty$, it follows that $x_\infty \in \Gamma(\bar{u})$.

Lower Semicontinuous in the sense of Hausdorff (LSC-H): Γ is LSC-H at $\bar{u}$ if for any $\epsilon > 0$ there exists $\delta > 0$ s.t. $d(x, \Gamma(u)) < \epsilon$ for all $x \in \Gamma(\bar{u})$ and *u* where $d(u, \bar{u}) < \delta$.

Upper Semicontinuous in the sense of Hausdorff (USC-H): Γ is USC-H at $\bar{u}$ if for any $\epsilon > 0$

there exists $\delta>0$ s.t. $d(x,\Gamma(\bar{u}))<\epsilon$ for all $x\in\Gamma(u)$ and u where $d(u,\bar{u})<\delta$.

Lower Semicontinuous in the sense of Berge (LSC-B): Γ is LSC-B at $\bar{u}$ if for each open set U satisfying $U\cap\Gamma(\bar{u})\neq 0,$ there exists a $\delta>0$ such that $U\cap\Gamma(u)\neq 0,$ for all u s.t. $d(u,\bar{u})<\delta$.

Upper Semicontinuous in the sense of Berge (USC-B): Γ is USC-B at $\bar{u}$ if for each open set U satisfying $\Gamma(\bar{u})\subset U,$ there exists a $\delta>0$ such that $\Gamma(u)\subset U$ for all u s.t. $d(u,\bar{u})<\delta$.

Continuous: Γ is continuous at $\bar{u}$ if it is USC-H and LSC-B at $\bar{u}$.

These definitions have the following useful properties:

P1) USC-B implies USC-H

P2) LSC-H implies LSC-B

P3) If Γ is USC-H at $\bar{u}$ and if the set $\Gamma(\bar{u})$ is closed, then Γ is closed at $\bar{u}$.

8.4.2 The Feasible Set Map

Recall that the cost and constraint functions, and hence the feasible and optimal set, are themselves by the feasible reference trajectory/state at the beginning of each major iteration, X^{ref}. The feasible set is determined by the following point to set map:

$$M(X^{ref})=\left\{\tilde{X}\in\Omega:\begin{matrix}\tilde{h}_j(X^{ref},\tilde{X})=0\\ \tilde{g}_k(X^{ref},\tilde{X})\leq 0\end{matrix}\right\}, X^{ref}\ feasible$$

and results in the parameterized programming problem

$$min\{\tilde{J}(X^{ref},\tilde{X}):\tilde{X}\in M(X^{ref})\}$$

We are interested in the case where a particular X_0^{ref} is not isolated, thus there are feasible trajectories (perhaps forming a continuous set) arbitrarily close to X_0^{ref}. Otherwise being closed follows trivially. For compactness we will use the parameter $u=X^{ref}-X_0^{ref}$ for any feasible points near X_0^{ref}. Let $\Lambda=\{u:u+X_0^{ref}\text{ feasible in NLP}\}\subset B(0,\epsilon_1)$ for some small ϵ_1. Thus we

have the point to set feasible set map: $M(u)=\left\{\tilde{X}\in\Omega:\begin{matrix}\tilde{h}_j(u,\tilde{X})=0\\ \tilde{g}_k(u,\tilde{X})\le 0\end{matrix}\right\}, u\in\Lambda,$ and parameterized problem $min\{\tilde{J}(u,\tilde{X}):\tilde{X}\in M(u)\}$. Note that u=0 corresponds to the original CP defined at X_0^{ref}.

Lemma 8.4.2.1 [Zlob09]: If $M(u)=\left\{\tilde{X}\in\Omega:\begin{matrix}\tilde{h}_j(u,\tilde{X})=0\\ \tilde{g}_k(u,\tilde{X})\le 0\end{matrix}\right\}, u\in\Lambda,$ defines a convex programming problem $\forall u\in\Lambda$ and Slater's Condition holds at $\bar{u},$ then *M* is continuous at $\bar{u}$.

Corollary 8.4.2.2: M is LSC-B and USC-H at $\bar{u}=0$.

Proof: $\bar{u}=0$ corresponds to the problem defined in terms of X_0^{ref}, an iterate of the optimization process defined in Chapter 6. By Appendix A - Proposition A.4, X_0^{ref} satisfies the CRMFCQ, which was shown in the SOCP/QCQP subproblems to be equivalent to Slater's condition.

Lemma 8.4.2.3: M(0) is closed and convex.

Proof: The argument showing the set satisfying each constraint is closed is identical to that of Chapter 8.2. The feasible set is then the intersection of closed sets and hence closed. It is obviously convex by the definition of a CP, where convex inequalities and affine equality constraints define a convex set.

8.4.3 The SOCP/QCQP Solution Map is Closed

Define the solution map as $S(u)=arg\,min_{\tilde{X}\in M(u)}\tilde{J}(u,\tilde{X}),$ the set of points attaining the optimum value. Note that the minimum fuel problem does not necessarily (though likely) satisfy the Second Order Sufficiency Conditions for an isolated minimum, so the output may not be a single point.

Corollary 4.3.3.3 from [Bank83]: Since Conditions 1-7 below are satisfied, then S is USC-B at $\bar{u}=0$.

C1) $\tilde{X}\in\mathbb{R}^{7N}$. This is part of the CP definition in Chapter 5.

C2) M is LSC-B at $\bar{u}=0\in\Lambda$. This is implied by Lemma 8.4.2.2.

C3) $S(0)$ is nonempty and bounded. We know a feasible point exists at $\tilde{X}=0$, so a solution must exist. It must lie within the feasible set, including trust regions, which is bounded. Thus the solution set is bounded.

C4) $J(\bar{u}=0,\cdot)$ is continuous. [This is stronger than the needed assumptions on semi-continuity]

C5) $J(\bar{u}=0,\cdot)$ is quasi-convex. $J(\bar{u}=0,\cdot)$ is convex, a stronger condition.

C6) $M(0)$ is closed and convex. Shown in Lemma 8.4.2.3.

C7) M is USC-H at $\bar{u}=0\in\Lambda$. This is implied by Lemma 8.4.2.2.

Corollary 8.4.3.1: If the solution map $S(u)$ is USC-H at $\bar{u}$ then it is closed at $\bar{u}$.

Proof: Let $p^*(\bar{u})$ be the optimal value of the problem defined at $\bar{u}$. We know it is a finite real number since each problem has $\tilde{X}=0$ as a feasible point. As a single value, $p^*(\bar{u})$ is a closed set. $\tilde{J}(u,\cdot)$ is continuous, thus the inverse image of the optimum, $S(\bar{u})=(\tilde{J}(\bar{u},\cdot))^{-1}(p^*(\bar{u}))$ is closed. USC-B implies USC-H by Property P1, and this combined with $S(\bar{u})$ being a closed set makes S closed at $\bar{u}$ by property P3.

Corollary 8.4.3.2: S is closed at X_0^{ref} w.r.t Λ.

Proof: S is closed at $\bar{u}=0$, which by definition is equivalent to the map being closed at X_0^{ref}. T

In conclusion, the solution map associating a feasible reference value X^{ref} with the

solution set of the SOCP/QCQP is closed.

8.4.4 Line Search is Closed

It is a standard result that exact and common inexact line search algorithms (Armijo, etc.) are themselves closed maps. The line search method used in this algorithm is modified in that addition of a continuous function $\delta v(\tilde{X}^*, \alpha), \alpha \le \alpha_{desc}$ is being composed into the line search algorithm. The composition of a continuous point-to-point map into a closed map is itself closed. [Luen08] However, the fact this composite map is closed will be directly verified for the case of an exact line search. It follows, essentially line by line, the proof that a standard exact line search is closed.[Luen08] Let

$$LS(X^{ref}, \tilde{X}^*) = \{Y \in \mathbb{R}^{7N} : Y = X^{ref} + \alpha \tilde{X}^* + \delta v(\alpha, \tilde{X}^*), 0 \le \alpha \le \bar{\alpha}, f(Y) = min_\alpha J(X^{ref} + \alpha \tilde{X}^* + \delta v(\alpha, \tilde{X}^*))\}$$

Suppose $\{X_k^{ref}\}_{k=1}^{\infty}, \{\tilde{X}_k^*\}_{k=1}^{\infty}$ are sequences such that $X_k^{ref} \to X_\infty^{ref}, \tilde{X}_k^* \to \tilde{X}_\infty^* \neq 0$. Suppose that $\{Y_k\}_{k=1}^{\infty}$ is a sequence s.t. $Y_k \in LS(X_k^{ref}, \tilde{X}_k^*) \forall k$ and $Y_k \to Y_\infty$. We need to show that $Y_\infty \in LS(X_\infty^{ref}, \tilde{X}_\infty^*)$.

For each integer *k*, $Y_k = X_k^{ref} + \alpha_k \tilde{X}_k^* + \delta v(\alpha_k, \tilde{X}_k^*)$ for some $\alpha_k > 0$. Since $\lim Y_k = Y_\infty$ is assumed to exist, then since $\delta v(\alpha, \tilde{X}^*)$ is continuously differentiable (see below), and passing to a convergence subsequence of alpha converging to α_∞ if necessary,

$$Y_\infty = \lim Y_k = \lim (X_k^{ref} + \alpha_k \tilde{X}_k^* + \delta v(\alpha_k, \tilde{X}_k^*))$$

$$= \lim X_k^{ref} + \lim \alpha_k \lim \tilde{X}_k^* + \lim \delta v(\alpha_k, \tilde{X}_k^*)$$

$$= X_\infty^{ref} + \lim \alpha_k \lim \tilde{X}_\infty^* + \delta v(\lim \alpha_k, \tilde{X}_\infty^*)$$

Implying that $Y_\infty = X_\infty^{ref} + \alpha_\infty \tilde{X}_\infty^* + \delta v(\alpha_\infty, \tilde{X}_\infty^*)$.

It must be shown that Y_∞ minimizes *J* along the curve $X_\infty^{ref} + \alpha \tilde{X}_\infty^* + \delta v(\alpha, \tilde{X}_\infty^*)$. For each *k* and $\alpha \in [0, \bar{\alpha}]$, by the definition of $LS(X_k^{ref}, \tilde{X}_k^*)$

$$J(Y_k) \le J(X_k^{ref} + \alpha \tilde{X}_k^* + \delta v(\alpha, \tilde{X}_k^*)) \forall k, \alpha \in [0, \bar{\alpha}].$$

By the continuity of *J* in our feasible domain, taking $k \to \infty$ leads to

$J(Y_\infty) \le J(X_\infty^{ref} + \alpha \tilde{X}_\infty^* + \delta v(\alpha, \tilde{X}_\infty^*)), \alpha \in [0, \bar{\alpha}]$. Thus by definition, $Y_\infty \in LS(X_\infty^{ref}, \tilde{X}_\infty^*)$. Therefore the line search algorithm is closed.

Regarding the continuity of $\delta v(\alpha, \tilde{X}^*)$ in both variables, this seems clear, but can be shown by using the discontinuity function from Chapter 7 with a larger set of inputs. Namely, for each patch point, between arcs *i* and *i+1*, let $\kappa : \mathbb{R}^{11} \to \mathbb{R}^3$ be given by:

$$\begin{aligned} \kappa_i((\alpha_i, \tilde{x}_i, \tilde{v}_i, \tilde{t}_{i+1}, \tilde{x}_{i+1}), \delta v_i) &= x_{i+1}^-((\alpha_i, \tilde{x}_i, \tilde{v}_i, \tilde{t}_{i+1}, \tilde{x}_{i+1}), \delta v_i) - x_{i+1}((\alpha_i, \tilde{x}_i, \tilde{v}_i, \tilde{t}_{i+1}, \tilde{x}_{i+1})) \\ &= x(x_i^{ref} + \alpha_i \tilde{x}_i, v_i^{ref} + \alpha_i \tilde{v}_i + \delta v_i, t_{i+1}^{ref} + \alpha_i \tilde{t}_{i+1}) - (x_{i+1}^{ref} + \alpha_i \tilde{x}_{i+1}). \end{aligned}$$

This function defines the gap between arc endpoints, which means there is position continuity when the function value is the 0 vector. This function satisfies

$$\kappa_i((0, \tilde{x}_i, \tilde{v}_i, \tilde{t}_{i+1}, \tilde{x}_{i+1}), \vec{0}) = x(x_i^{ref}, v_i^{ref}, t_{i+1}^{ref}) - (x_{i+1}^{ref}) = x_{i+1}^{-ref} - (x_{i+1}^{ref}) = \vec{0}.$$

Also, at $((0, \tilde{x}_i, \tilde{v}_i, \tilde{t}_{i+1}, \tilde{x}_{i+1}), \vec{0})$ we have $\frac{\partial \kappa_i}{\partial(\delta v_i)}(0, \vec{0}) = B_{i+1,1}$, which by assumption is invertible (or correctable with a change of frame). Finally, as in Chapter 7, $\varkappa$ is continuously differentiable everywhere in the domain.

Therefore the conditions of the Implicit Function Theorem are satisfied and so δv_i is a continuously differentiable function of both α_i and $(\tilde{x}_i, \tilde{v}_i, \tilde{t}_{i+1}, \tilde{x}_{i+1})$. Thus the adjustment to the whole state $\delta v(\alpha, \tilde{X}^*) \in \mathbb{R}^{7N}$ is a continuously differentiable.

8.4.5 Closure Under Composition

Our search direction selection map, which is just the SOCP/QCQP solution map, is closed by Corollary 8.4.3.2. The modified line search is closed whenever a search direction is defined, i.e.

not at a KKT point. Thus, under the same condition as the standard combination of define direction and search [Zang69], namely that both maps are closed and the set of possible X^* is limited to a compact set, the composition is closed when not at a KKT point. Thus the algorithm is closed on the set of feasible points that are not KKT points.

8.5 Conclusion

The results of sections 8.2, 8.3, 8.4 show that the conditions of Zangwill's Global Convergence Theorem are satisfied. Therefore the limit of any convergent subsequence generated by the algorithm is a KKT point of the original NLP.

Chapter 9

Initial Correction of Graph Search Based Initial Guesses

9.1 Introduction

The previous chapters of this section have described a Sequential Convex Programming based optimization method, including the important property that the method creates feasible iterates when starting from a feasible initial guess. Indeed, the assumption of a feasible reference was used throughout the results of Chapter 7. This chapter describes how to bridge the gap between the set of arcs generate by the graph search and arc selection process, and the feasible guess needed by the optimization process. It should be noted that while this chapter comes after those describing the fuel optimization problem, the initial correction takes place before the optimization process in the transfer design problem. This presentation order is primarily due to the fact that an optimization problem and its constraints must be described *before* one can state what qualifies as a feasible initial guess for that problem.

9.2 Feasibility and Infeasibility of the Arc Selection Process Results

Proper construction and pruning of the directed graph (Chapter 3.4.1, 3.4.3) results in an initial guess generated by the search process (Chapter 4) that is nearly feasible. For each of the constraints given in Chapter 5, the extent to which they are feasible after the arc selection process is summarized in this section.

Continuity: As described in Chapter 4.4, the arcs selected will begin and end within the same region. Therefore they are not strictly continuous in position, but the discontinuity is bounded from above by the maximum distance between points within the region.

Minimum and Maximum Coasting Time: During the graph construction process, ballistic arcs whose coasting times are outside of the allowable limits are excluded (Chapter 10.4.2). Therefore these constraints are already satisfied by the initial guess.

Maximum Thrust per Maneuver: During the graph construction process, impulsive links with cost greater than the thrust limit may be eliminated. Since this cost is an approximation of the impulse needed, this does not strictly imply feasibility. It does however, mean that the constraint will be nearly satisfied. The same holds for thrust direction constraints.

Impact Avoidance: Only arcs that do not violate the minimum altitude limit are included in the graph construction process, therefore this constraint is satisfied unless there is a large model mismatch. This may be avoided by either including forces large enough to cause impact when transitioning between models, and verifying that arcs that pass near the body do not impact when additional forces are added.

In cases where the minimum altitude is associated regions of convergence of complex gravity models, it may be the advisable that this constraint be added directly to an infeasibility minimization problem rather than merely penalized.

9.3 Infeasibility Minimization

In this section, a variation on the convex subproblems given in Chapter 5 is given which may be used to create a different sequential convex programming method to penalize infeasibility in an initial guess. This process is shown to have convergence properties like those shown in Chapter 7 and 8, though with fewer restrictions. If the infeasibility cost is minimized with cost equal to zero, a feasible transfer will have been found. If the minimum is not zero, then the search and select process would need to be rerun to provide a higher quality initial guess.

9.3.1 A Standard Second Order Cone Form for Infeasibility Minimization

As a SOCP, the inequality constraints of the optimization problem approximations – whether linear, quadratic, or a full second order cone - are of the form:

$$\|A\tilde{X}-b\|\leq c^T\tilde{X}+d$$

Continuity in position is naturally expressed as an equality constraint. However, note that continuity will hold if $x_{init}=x_1, x_{i+1}=\bar{x}_{i+1}, x_{final}=\bar{x}_{N+1}$, which is alternately expressed as $\|x_i-\bar{x}_i\|=0$ or even $\|x_i-\bar{x}_i\|\leq 0$ due to the nonnegativity of norms. In order to approximate this function as part of an iterative solution process, for some reference trajectory (here either the guess provided by the search and arc selection or the result of a previous iteration), the same small variation variables and STM based approximations in Chapter 5 will be used. This results in a set of second order cone constraints (with $c=\vec{0}, d=0$) of the form:

$$\left\|M_{i+1}\tilde{X}-n_{i+1}\right\|\leq 0$$

where

$$M_1=\begin{bmatrix}-I_{3x3} & 0 & \dots\end{bmatrix} \qquad n_1=x_1^{ref}-x_{init}$$
$$M_{1+1}=\begin{bmatrix}\dots & A_{i+1,i} & B_{i+1,i} & v_{i+1}^{-ref} & -I_{3x3} & \dots\end{bmatrix} \quad , \quad n_{i+1}=x_{i+1}^{ref}-x_{i+1}^{-ref}$$
$$M_{N+1}=\begin{bmatrix}\dots & A_{N+1,N} & B_{N+1,N} & v_{N+1}^{-ref}\end{bmatrix} \qquad n_{N+1}=x_{final}-x_{N+1}^{-ref}$$

where the state transition submatrices and endpoint values are those defined in Chapter 5. This form is not used at all in the minimum fuel or energy problem of previous chapters and indeed will not be used as is here, but will along with the inequalities will serve as the basis for a set of modified constraints in the next section. In any case, for all constraints, the standard second order cone form $\|A\tilde{X}-b\|\le c^T\tilde{X}+d$ may be used to define an optimization problem to minimize the degree to which the desired constraints are violated.

9.3.2 The Infeasibility Minimization Problem SOCP and NLP

With the introduction of an additional variable ς the constraints of the optimization problem defined in Chapter 5 may be used to construct a new problem to drive an infeasible initial guess towards a feasible but non-optimal transfer. Consider now the following SOCP:

For variables $\tilde{X},\varsigma$, *minimize the cost function*

$$\tilde{J}_{feas}(\tilde{X},\varsigma)=\varsigma$$

with constraints

$$\left\{\left\|\begin{bmatrix}A & \vec{0}\end{bmatrix}\begin{bmatrix}\tilde{X}\\ \varsigma\end{bmatrix}-b\right\|\le\begin{bmatrix}c^T & 1\end{bmatrix}\begin{bmatrix}\tilde{X}\\ \varsigma\end{bmatrix}+d : \text{for all constraints } \|A\tilde{X}-b\|\le c^T\tilde{X}+d \text{ of the min fuel problem}\right\}$$

and

$$\varsigma\ge 0$$

and trust region constraints as defined in Chapter 5.

Here the second order cone inequalities are just the standard form of the more intuitively clear $\|A\tilde{X}-b\|\le c^T\tilde{X}+d+\varsigma$. That is, every constraint of the optimization problem is "loosened" by ς. Or alternately, the minimum value of ς gives the maximum amount any of the

constraints are violated.

This SOCP could have alternately have been derived by applying the same process as in Chapter 5 to an infeasibility minimization NLP:

For variables X, ς *minimize the cost function*

$$J_{feas}(X, \varsigma) = \varsigma$$

with constraints

$$\left\{ g_k(X) \le \varsigma, \|h_i(X)\| \le \varsigma : \text{for all constraints } g_k(X) \le 0, h_i(X) = 0 \text{ of the min fuel problem} \right\}$$

and

$$\varsigma \ge 0$$

Therefore all of the constraints will be satisfied if $J_{feas}(X, \varsigma) = 0$. An iterative process like that described in Chapter 6 may be used (omitting the recorrection step) to minimize the infeasibility.

9.3.3 Problem Analysis

As was done in Chapters 7, 8, and Appendix A for the fuel minimization problem, it will be shown that sequential convex programming using the above SOCP results in a convergent sequence of points for which each subproblem satisfying Slater's Condition.

Proposition 9.3.3.1: A point (X, ς) satisfies the KKT conditions of the infeasibility minimization NLP if and only if $(\tilde{X} = 0, \varsigma)$ satisfies the KKT conditions of the infeasibility minimization SOCP defined in terms of X as the reference. Also, if at a point X $J_{feas}(X, \varsigma) = 0$, then $\tilde{J}_{feas}(0, \varsigma) = 0$.

Proof: As shown in Chapter 5, the second order cone constraints are valid to at least first order. Also, the ς cost and constraint functions are identical. Therefore since the function values and gradients at the reference are equal, the terms in their respective KKT conditions are identical. Since the function values are equal at the reference, in particular $J_{feas}(X, \varsigma) = 0$ iff $\tilde{J}_{feas}(0, \varsigma) = 0$.

Proposition 9.3.3.2: Slater's Condition holds for the infeasibility minimization SOCP.

Proof: Let $\varsigma_0 = max_k\{\|b_k\| - d_k + 1, 1\}$ where k indexes all of the second order constraints of the form $\|A_k\tilde{X} - b_k\| \le c_k^T\tilde{X} + d_k + \varsigma$. Consider the point $(\tilde{X}, \varsigma) = (0, \varsigma_0)$. With $\tilde{X} = 0, \varsigma_0 \ge 1$ clearly the trust region and $\varsigma \ge 0$ constraints are satisfied strictly. At $(\tilde{X}, \varsigma) = (0, \varsigma_0)$, the constraints $\|A_k\tilde{X} - b_k\| \le c_k^T\tilde{X} + d_k + \varsigma$ become $\|b_k\| \le d_k + \varsigma_0$ or $\|b_k\| - d_k \le \varsigma_0 = max_k\{\|b_k\| - d_k + 1, 1\}$. This holds strictly for any of the constraints. As there are no equality constraints, $(\tilde{X}, \varsigma) = (0, \varsigma_0)$ is then an interior point. Since the problem is convex, this implies Slater's Condition holds.

Lemma 9.3.3.3: The infeasibility minimization SOCP solution map is closed.

Proof: Since Slater's condition holds for any of the SOCPs without condition and any point X in the domain may be used as a reference, Lemma 8.4.2.1 and Corollary 8.4.2.2 hold and so the feasible set map is LSC-B and USC-H. Since the constraints are second order cones, a closed half plane, and the trust region norm balls – which are topologically closed – the feasible set consisting of their intersection is also closed and convex.

Together these show that conditions C2, C6, and C7 of Bank's Corollary 4.3.3.3 (see Chapter 8.4.3) hold, and C1 holds trivially. In showing an interior point exists, we know the SOCP has a feasible point and hence a solution. Trust regions guarantee the solution set is bounded, this C3 holds. The cost function $\tilde{J}_{feas}(\tilde{X}, \varsigma) = \varsigma$ is trivially continuous and convex, thus C4, C5 hold, and thus the corollary holds. This in turn satisfies the conditions in the proofs of Corollaries 8.4.3.1 and 8.4.3.2, and so the solution set map is closed at the reference.

Lemma 9.3.3.4: The infeasibility cost ς is a global descent function of the iterative algorithm.

Proof: Suppose $(X^{ref}, \varsigma^{ref})$ is a KKT point of the infeasibility minimization NLP. Then it is a KKT

point of the infeasibility minimization SOCP. For a convex problem, KKT conditions imply a minimum and so $(\tilde{X}=0,\varsigma^{ref})$ is a solution to the SOCP, ending the algorithm with no increase to the cost.

If $(X^{ref},\varsigma^{ref})$ is not a KKT point of the infeasibility NLP, then it is not a KKT point of the infeasibility minimization SOCP. Therefore $(\tilde{X}=0,\varsigma^{ref})$ cannot be a solution to the SOCP since if Slater's Condition holds $(\tilde{X}=0,\varsigma^{ref})$ being a solution implies the KKT conditions would hold. As Slater's Condition does hold by Proposition 9.3.3.2, this cannot be the case. Therefore there exists a solution $(\tilde{X^*}=0,\varsigma^*),\varsigma^*<\varsigma^{ref}$. Since there is a strict decrease to the cost, there is a strict decrease in the constraint functions for which $g_k(X^{ref})=\varsigma^{ref}$, $(\tilde{X}^*,\varsigma^*-\varsigma^{ref})$ defines a descent direction for these functions since they are convex. For other functions, $g_k(X^{ref})<\varsigma^{ref}$ strictly. Therefore there is a nonempty range of step sizes such that the constraints g_k satisfying $g_k(X^{ref})=\varsigma^{ref}$ decrease in value, and such that for others we still have $g_k(X^{ref})<\varsigma^{ref}$. Therefore by backstepping line search a step size α exists such that $(X^{ref}+\alpha\tilde{X},\varsigma^{ref}+\alpha(\varsigma^*-\varsigma^{ref}))$ satisfies the constraints of the infeasibility minimization NLP with cost $\varsigma^{ref}+\alpha(\varsigma^*-\varsigma^{ref})<\varsigma^{ref}$. Therefore at points that are not KKT points of the infeasibility minimization NLP, the algorithm produces a point with strictly lower cost. Therefore the cost is a global descent function of the algorithm.

Proposition 9.3.3.5: Zangwill's Convergence Theorem applies for the infeasibility minimization algorithm.

Proof: The domain can be selected to be compact, as limited either by energy or an arbitrary distance from the target body. As shown in the above Lemma 9.3.3.4, the cost is a global descent method. Also, Lemma 9.3.3.3 showed the SOCP solution map is closed, which by the standard result on composition with line searches (see Chapter 8), means that the algorithm is closed. Taken

together, these imply the result of Zangwill's Convergence Theorem as stated in Chapter 8.1.

This shows that for any initial guess provided by the search and arc selection process, the algorithm will converge to a KKT point of the infeasibility minimization NLP. Given that the arcs selected already satisfy many of the constraints such as nonimpact and coasting times, and nearly satisfy the remainder such as continuity and thrust constraints, process should converge to a value of 0 – a feasible point. If not then the initial guess is not in the basin of convergence of a feasible transfer, but rather an infeasible local minimum. In such a case the search process is to be rerun with the graph link corresponding to the infeasible transition removed. This will generate a different itinerary, and the process repeated.

9.4 Summary

A sequential convex programming approach to penalize infeasibility has been stated and shown to satisfy the conditions of Zangwill's Convergence Theorem as well as satisfy Slater's Condition so that an interior point exists for every subproblem. For a high quality result from the search and select process, the local minimum achieved should result in a feasible initial guess for the fuel minimization process. Otherwise the search and select process can be rerun with the problem link removed in order to find a better initial guess.

Chapter 10

Application to an Orbiter at Phobos

10.1 Introduction

This chapter describes the application of the onboard focused redesign tool described throughout the thesis to the problem of onboard replanning for a Phobos orbiter. Questions that must be addressed for any application include what domain is appropriate, how finely to partition that domain given the strength of the differential corrector used, and which forces to include in the program models. Beyond these general adaptations, there are design challenges specific to Phobos. These are caused by factors such as the moon's highly irregular shape and gravity field, large size relative to its libration points, and the rapid speed of impact or ejection. These facets of the Phobos system cause a large reduction in the available periodic orbits, invariant manifolds, and stable arcs that can be used for the transfer construction process. Moreover, at the same time Phobos limits available options, it demands more rapid computation than other targets. Several trade-offs exist when deciding how to balance these priorities, and this chapter will discuss the decisions made in tailoring the method to this challenging system.

10.1.1 Relevance

The moons of Mars – Phobos and Deimos – are being investigated as possible targets for future human exploration and as a stepping stone to Mars itself.[Wall12, Hoff11, Bosa14] In addition to a future manned mission, robotic exploration, such the attempted sample return of Phobos-Grunt,[Maro04] will likely proceed first to gather scouting information and other important science data.

In a paper on orbital operations for Phobos and Deimos,[Wall12] the necessity of using several types of periodic orbits to fulfill various science and safety needs was described. Furthermore, due to the frequency of station keeping maneuvers on unstable orbits, switching between orbits, and recovering from imperfect navigation data, spacecraft autonomy is seen to be an essential component of a Phobos orbiter mission,[Wall12] as it will be in other future missions [Cang12]. Beyond autonomous control to stay along pre-planned reference trajectories, a truly robust autonomous system will need to handle situations where the *original plan is no longer applicable because of missed or poorly executed maneuvers, aborting to a safe orbit from a large perturbation, target changes, or other unexpected circumstances*. In such cases the ability to quickly re-plan the reference trajectory onboard is needed in addition to small corrections. However, most initial guesses are the product of designers on the ground while automated approaches such as evolutionary methods can be very time consuming. There is thus a need for a method that combines a systematic yet automated initial guess generation process when needed, and a complementary local correction process to create a flyable transfer.

10.1.2 Contribution

This chapter addresses several of the design challenges posed by the application to Phobos, relating to available orbits, discretization structure, model selection, and interactions between different design decisions. There are multiple features of Phobos that make the setup phases of

partitioning and discretization distinct. These include the large size of Phobos relative to the libration point region, a domain extending well beyond the libration points, and the high number of impacting trajectories. For onboard phases, increased complexity is added due to more challenging dynamics in the correction process and from the ability to close the loop and use both local only and graph planning methods.

The redesign tool will be demonstrated on several examples related to the operation concepts outlined by JPL engineers. [Wall12] This includes including transfers between periodic orbit families used for different mission stages such as arrival and descent staging. Beyond switching between these orbits exactly – the realm of trajectory design on the ground – for a redesign tool it is more relevant to demonstrate such a transfer with a very perturbed initial condition. This fits more closely the situation where such a system would be called on as well as providing a case where a reference trajectory does not exist. Additionally, in order to "close the loop", random errors will be added within a transfer, providing an opportunity for comparison between the results of a local correction only, and correction after a graph based redesign. Computation times and ΔV's will be provided for each case.

10.2 Models Used

Please note that the models and forces discussed in the following sections are provided in detail in Appendix C. The following focuses on why the model choices were made in this application.

10.2.1 Discretization and Arc Database Model

In order to avoid increasing the size of the discretization by an order of magnitude or more, it is desirable that a time invariant approximation is used, assuming this captures the forces with the highest relative magnitude. Forces like the higher harmonics of Mars, gravity of the Sun, and

solar radiation pressure are time varying (and indeed can have different frequencies), and so not included in the discretization model. The point mass gravities of Phobos and Mars are time invariant in the CR3BP, but particularly for irregularly shaped bodies like Phobos, the higher harmonics of the moon must be analyzed.

An important property of many systems of interest is that the moon is tidally locked with the planet, meaning that its rotation about its axis has the same period as its orbit about the planet. This includes both of the moons of Mars, the Galilean moons with Jupiter, Enceladus and Titan with Saturn, and of course the Earth's Moon. [McFa06] This means that in the CR3BP rotating frame the moons are fixed and do not rotate, and thus the system with higher order harmonics is still time invariant. Therefore the complex gravity field of Phobos may be included without adding an extra dimension to a partition for orbit phase.

Note that in the standard CR3BP equations of motion the rightmost terms describe the forces due to the gravity of Phobos. To include a more complex gravity field, this term is replaced while those due to Mars or frame effects are unchanged. This results in:

$$\ddot{x} = x + 2\dot{y} - \frac{1-\mu}{r_1^3}(x+\mu) - F_x(\vec{r}_2)$$

$$\ddot{y} = y - 2\dot{x} - \frac{1-\mu}{r_1^3}y - F_y(\vec{r}_2)$$

$$\ddot{z} = -\frac{1-\mu}{r_1^3} - F_z(\vec{r}_2)$$

where the new terms are the components of the spherical harmonic gravity model as described in the previous section, converted to the normalized units of the CR3BP.

The question of which order and degree to use in the gravity field remains. While no harm would arise from using all of the available information, there is little benefit to use terms that are dominated by time-varying forces excluded from the discretization model. It will be seen below

that adopting this standard results in the model of the CR3BP with 4th order and degree harmonic gravity for Phobos.

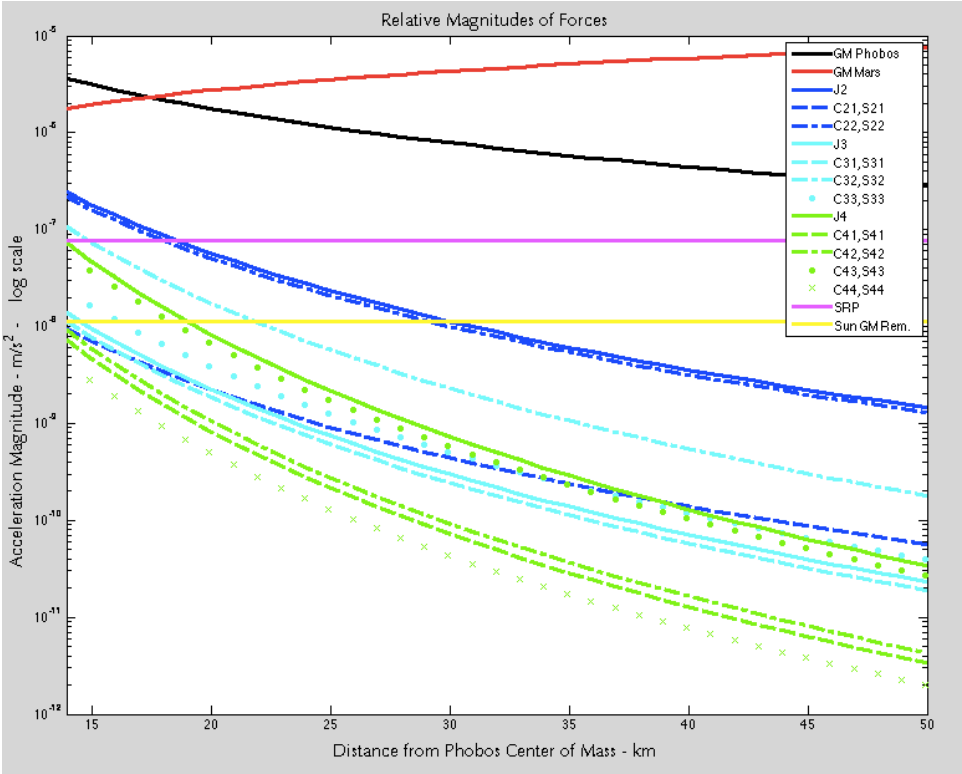


Figure 10.1: Relative magnitudes of forces due to point mass gravity, gravitational harmonics, and SRP in an ideal plate model using the average distance of Mars from the Sun.

10.2.2 Onboard Model

Ultimately, the limit on onboard model accuracy is time. The faster the processor and more time allowed, the more accurate the onboard model may be. It is likely that a full ephemeris model is impractical. As the final model is based on spacecraft dependent factors such as computational ability and navigation accuracy, it is not appropriate to make such a decision here. Rather, the goal is to point out that additional forces – especially time varying forces – may be incorporated into the onboard model that may have been omitted in the discretization model.

Examples that have been tested in Matlab for the Phobos application include up to 12x12 gravity for Phobos and an ideal flat plate model for Solar Radiation Pressure (Appendix C). While

the gravity of Deimos was not tested in any simulations at this moment, the addition of the Moon to a Sun/Earth system in a different application showed that additional massive bodies may be included without problem (Chapter 12). In order to have identical models for comparing calculation times for the test cases between Chapters 10 here and the the RAD750 implementation in Chapter 11, the time invariant model for the discretization is used for the examples given below. Chapter 12 includes a case where the correction/optimization model are time-varying.

10.3 Periodic Orbits and Invariant Manifolds

Despite the lack of standard Keplerian orbits due to Phobos containing its sphere of influence, periodic orbits can still play an important role for a mission to this moon. In particular, both stable and unstable orbit families derived from the Three-Body Problem play an essential part in existing orbital operation plans. [Wall12] Despite the importance of libration points in this plan, their stable and unstable manifolds play less of a role than they do for missions such as Artemis due to the size of Phobos itself blocking these structures.

10.3.1 Orbits in JPL Mission Scenarios

Vertical Lyapunov orbits are libration point orbits which, like the more well known planar Lyapunov orbits, bifurcate directly out of the L1 and L2 libration points. These orbits have the highest out of plane amplitude of the libration point orbits, and as such provide the best polar viewing capabilities. As the energy of the orbit increases, the minimum altitude decreases from that of the associated libration point, creating a limit on the energy and hence z-amplitude due to impact. These orbits are very unstable, and JPL's stationkeeping methods are unable to converge to the orbit when navigation errors as small as 10m were left uncorrected for two hours.

Halo orbits are a family of non-planar orbits that bifurcate out of the planar Lyapunov family of orbits, and are well studied in theory [Farq66, Howe84] and in practice [Lo01, Farq01].

Halo orbits provide a wide viewing area, but also present opportunities for descent staging. This is because the interior branch of their unstable manifolds can provide a relatively low velocity descent to the surface of Phobos including Stickney Crater, and the stable manifold provides a low-energy return path. These orbits, like the Vertical Lyapunov orbits, are very unstable and require frequent station keeping and capability to switch to more stable parking orbits to ensure mission safety [Wall12].

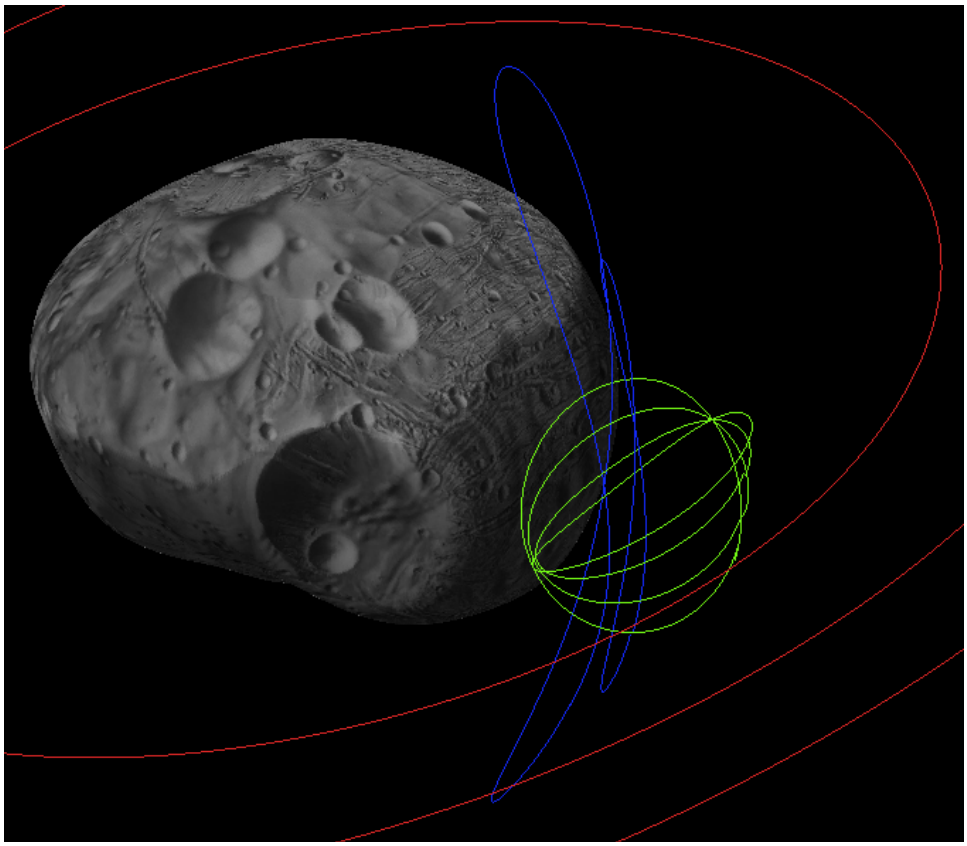


Figure 10.2: Examples of Vertical Lyapunov (blue), Halo (green), and DROs (red) in the Phobos system after the introduction of the time invariant perturbation of 4x4 gravity.

Distant Retrograde Orbits (DROs) are an important class orbits for Phobos operations. In contrast to libration point orbits, DROs and their quasi-periodic equivalents at Phobos are stable when nearly planar.[Wall12] Due to the low cost to transfer between DROs and various libration

point orbit families – on the order of 10-20 meters per second [Wall12] – DROs provide a safe option between higher risk mission segments as well as a target for an abort scenario from an unstable periodic orbit.

All of the above orbits play important roles and should be available as boundary conditions. In reality, only quasi-periodic orbits remain if other forces such as the time-varying influence of the Sun, solar radiation pressure, the gravity of Deimos, etc. are included. Regardless of whether such a complex model is used by the correction process, the underlying periodic orbits in the design model should be accurately calculated. Should the correction model be the same as the design model, then the orbits can be used directly. Should the model be more complex than the design model, then the periodic orbit will provide a good initial guess for the quasi-periodic orbit (or other approximate orbit should the results not fit the formal definition of quasi-periodic yet still remain useful) that the corrector is targeting. Examples are shown in Figure 10.2.

10.3.2 Other Orbit Families

Due to the large size of Phobos relative to its sphere of influence and libration points, many entire orbit families impact the surface. This includes the axial family of libration point orbits, prograde-retrograde switching orbits, symmetric prograde and Prograde-to-Lyapunov families,[Lo04] and indeed prograde orbits in general. This is intuitively clear since anything moving in a prograde motion with a minimum distance greater than or equal to the libration points will escape – and this is very close to the surface of Phobos itself. Also, due to the fact that Phobos contains its sphere of influence (7.3km vs. the 13.4 km radius in its largest dimension), Keplerian-like orbits do not persist in this application. [Wall12]

10.3.3 Invariant Manifolds

It should be noted that while the inner branches of the invariant manifolds of Halo orbits may be useful for descent staging to the surface of Phobos, such manifolds cannot be used as in

other cases [Koon08, Vill04] for low-energy orbital transfers in this system. This is because for those libration point orbits that do not themselves impact Phobos, simulations done as part of this research have shown that the interior branches of their stable and unstable manifolds all impact the surface. Outer branches may be useful for planning an orbital insertion directly into libration point orbits from outside the system, but these manifolds cannot be used to transfer between L1 and L2.

10.4 Algorithm Setup

This section provides details on the partition and resulting graph created for this application by following the procedure described in Chapter 3.

10.4.1 Domain Partition

Poincare sections were created for the following CR3BP energy levels: -1.5000130, -1.5000125, -1.5000120, -1.5000115, -1.5000110, -1.5000100, -1.5000050, -1.5000000, -1.4999990, -1.4999970, -1.4999940. The lower energy levels with the fine gradations correspond to those energies at which the relevant libration point orbits exist and correspondingly those energy levels for which transit into the Phobos region from the exterior region may occur due to the opening of the zero-velocity surface. [Koon08] The fine gradations are due to the large variation in orbit characteristics relative to changes in energy. At energy levels beyond which the libration point orbits are impacting, the gradations increase due to the lower sensitivity of DROs to changes in energy levels. The highest energy level is that at which the periapses of DROs are just within the maximum radius of the domain.

Within each Poincare section, the regions are next defined by the following ranges of radii, given in distance from the center of mass of Phobos: 14-15km, 15-16km, 16-17km, 17-18km, 18-20km, 20-25km, 25-30km. Recall that as a periapsis Poincare section (Chapter 3), and so

trajectories for which these values are closest approaches cover a larger general domain. In particular DROs attaining periapsis within this range of values actually extend out to a maximum distance closer to 50km, a limit suggested by JPL staff advising on this project. The minimum radius is determined by the fact that the radius of convergence of the spherical harmonic gravity model is 13.93km, thus it is necessary that all arcs considered have closest approaches slightly greater than this value. Note that the gradations are more fine closer to the surface due to more sensitive dynamics near the surface and in particular in the libration point regions. The remaining coordinates in the representation described in Chapter 3 – θ, ϕ, α – are divided into intervals of equal size. Since the regions of operational interest are within a limited band of z amplitude and polar orbits seem to be impacting at the energies considered, ϕ has been limited to $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$ in order to save memory on unused space.

This approach results in 6,300 regions per Poincare section for a total of 69,300 total regions, and thus the same number of graph nodes.

10.4.2 Graph Creation

For each energy level, there were 324,764 points in the grid of initial conditions, for a total of 3,572,404 possible initial conditions. For lower energies considered, some of these points correspond to positions with potential energy greater than the energy of the Poincare section, and thus such positions would be omitted for that particular energy level. The ballistic arcs generated in this way were combined with the periodic orbits described above in 10.3.1. These orbits could have been calculated directly via continuation using AUTO [Doed97], however in this case they were differentially corrected into the higher order gravity model directly using a modified version of the software described in Chapter 9. Invariant manifolds were not included for the Phobos application for the reasons described in 10.3.3.

Using these ballistic arcs, the graph was then constructed following the approach described in Chapter 3.4, with the same resulting layered structure as in Chapter 3.5 where ballistic connections exist within each energy level, and impulsive connections exist between regions with matching positions. As is shown in Figure 10.3, of the $(69300)^2$ possible connections that may exist in an arbitrary graph, the actual graph has 3.3 million connections in a banded structure, resulting in a connection density of only 0.07% nonzero entries, or about 48 connections per node. Most of these are impulsive and the search heuristic used efficiently ignores unhelpful maneuvers. Methods to further reduce this memory footprint for onboard use are discussed in Chapter 11.

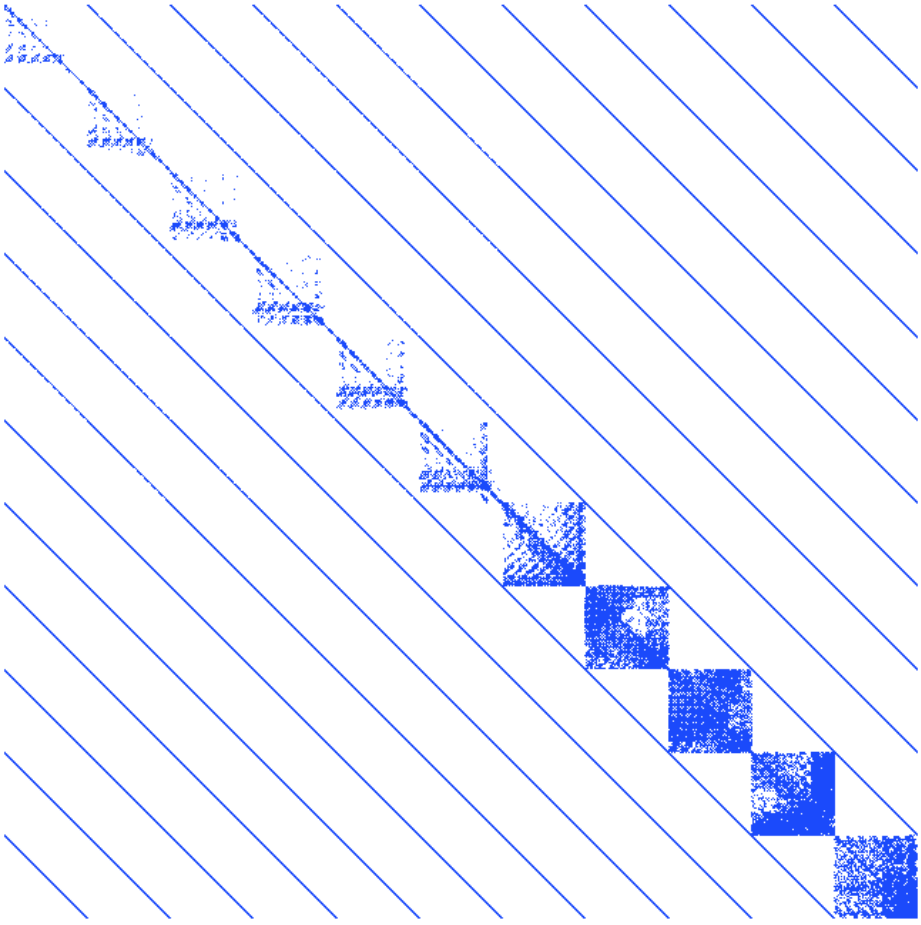


Figure 10.3: Sparsity diagram with a blue pixel representing a nonzero graph entry. Scaling effects make this image appear more dense than in fact.

Furthermore, constraint based pruning has been applied to ensure that initial guesses satisfy those constraints that may be enforced before the corrections process. Thus, only ballistic arcs whose coasting times are within the minimum and maximum are included. Additionally, impulsive connections with cost greater than the maximum fuel cost per maneuver are removed. Lastly, any ballistic arcs that violate the minimum radius / nonimpact constraints would not attain periapsis within the partition, and thus do not contribute to any graph links. Therefore any arc selected for a given ballistic graph link automatically satisfies this constraint, in addition to those governing coasting time. The values of these constraints are discussed in the next section.

10.4.3 Constraint Values

The constraints and associated values included in the test cases that follow are listed in Table 10.1.

Table 10.1: Constraint Values for Phobos Test Cases

Minimum radius from center of mass:	14 km
Minimum coasting time:	0.5 hours
Maximum coasting time:	7.66 hours
Maximum ΔV per maneuver:	12 m/s
Minimum ΔV per maneuver:	0.1 m/s
Position continuity tolerance:	0.94 m
Trust region (see Chapter 5):	Dynamic

The minimum radius / nonimpact constraint value was chosen due to the combination of being 1km above the furthest extent of Phobos and due to the radius of convergence of the spherical harmonic gravity being 13.93km. Minimum and maximum coasting times were set to 30 minutes and 7.66 hours (one orbital period of Phobos) respectively. Minimum coasting time

constraints are in place to allow sufficient time to obtain a navigation fix after a maneuver, as well as the existence of delays between thruster firings. In [Wall12], station-keeping strategies with corrective maneuvers every 30 minutes was considered (with more frequent navigation), therefore it was felt this was a reasonable value for this constraint. The continuity tolerance was selected to be as close to 1 meter as possible when expressed in normalized CR3BP coordinates, in this case 10^{-7} normalized units. This choice follows from JPL estimates of meter to sub-meter optical navigation accuracy. [Wall12] Other constraints in practice are heavily spacecraft dependent, thus at this early stage values were included in order to test functionality, and set conservatively.

10.5 Test Case Descriptions

In order to evaluate whether the design choices made to handle the challenges described in the introduction have proved effective, several scenarios have been considered. These examples were selected in consultation with JPL Mission Design and Navigation (Section 392) staff to be representative both of the types of transfers important to Phobos orbital operations [Wall12] and the abilities of the redesign tool. The objectives for the examples include an abort scenario to a DRO from a trajectory diverging from a Halo orbit, and returning to a Vertical Lyapunov orbit after diverging from the same orbit. The first example plans multiple transfers from scratch using the full graph search and correction method without using any sort of reference trajectory. In the second example, it is assumed an additional significant error to the state has been added part way through the planned transfer. Two transfers are again planned, one using the full replanning process and another correcting and partially optimizing the remainder of the initial transfer from the first example. The third example of returning to a Vertical Lyapunov orbit shows the ability to target an orbit in a very unstable region of the domain.

Several assumptions have been made in these examples. First, the state is exactly known at the time the transfer is needed. Thus, any navigation calculations and uncertainty are not addressed here. Next, the maneuvers are assumed to be impulsive. Additionally, the models described above are assumed to be accurate, so that errors do not arise from things like parameter errors or excluded forces. The large perturbations added to the states to set up the examples are done as needed.

The resulting trajectories are shown in Chapter 10.6 on the transfer results, and the images may be helpful to understand the scenario description.

10.5.1 Test Scenario 1 – Abort to Distant Retrograde Orbit

Halo orbits have been seen to be useful for descent staging purposes, and DROs useful as stable "safe" orbits around Phobos. One transfer of interest then is transitioning between these two orbit families. However, as our interest here lies in onboard replanning rather then initial trajectory design between orbit families, this test case will focus on the case of a large perturbation off of a Halo orbit that uncontrolled would lead to escape from Phobos. It is then from this perturbed initial condition that a transfer to a DRO will be made, simulating an abort scenario. For this case, no initial guesses or reference is provided, thus the process is run using the heuristic search on the directed graph in order to find an itinerary.

Table 10.2: Example 1 Boundary Conditions

Initial State (normalized CR3BP coordinates):

x=0.998157523553723, y=-0.000634429787531433, z=-0.000625648186687050

x'=-0.000520987652511872, y'=0.0013801542149045, z'=-0.000791745441681660

Target Orbit:

DRO at energy -1.49999865 in CR3BP with Phobos 4^{th} order and degree potential.

10.5.2 Test Scenario 2: Closed Loop Redesign in Response to Large Perturbation

This case builds on Transfer 1 of the previous scenario. Here, an additional large perturbation to the state is given part way through the transfer, in order to introduce a substantial error. The redesign system is then called upon to "close the loop" and find new transfers to the original target from the unexpected state.

Table 10.3: Example 2 Boundary Conditions

Initial State (normalized CR3BP coordinates):

x=1.002308393781638, y=-0.000090141995480, z=-0.000448589113068

x'=-0.0010664899953512, y'=-0.005542158428128, z'=0.000156408411279

Target Orbit:

DRO at energy -1.49999865 in CR3BP with Phobos 4th order and degree potential.

10.5.3 Test Scenario 3: Return to Libration Point Orbit

So far, the above examples have dealt with transitions from one orbit family to another, in an abort scenario with and without further error introduced. For this last example, the spacecraft is asked to return to the Vertical Lyapunov orbit at L1 from which it has diverged. Vertical Lyapunov orbits are potentially important due to their good polar viewing properties, given the lack of Keplerian polar viewing orbits. Since the spacecraft is still in the vicinity (it has drifted approximately 3 km, primarily in the -*x* direction) of the orbit, the periodic orbit itself provides a useful reference trajectory. More precisely, pick a state on the orbit close to the initial condition, and choose the orbit period as the initial duration. This arc and time of flight is then used by the differential corrector and optimization process.

Table 10.4: Example 3 Boundary Conditions

Initial State (normalized CR3BP coordinates):

x=0.999230277519748, y=0.000167097229294, z=-0.002765975993450

x'=-0.000534593411918, y'=-0.000564462210736, z'=0.000174481897515

Target State on Orbit:

x=0.9991464837, y=-0.00041269013, z=-0.00196410

x'=-0.00130428661266256, y'=-0.000101645542566986, z'=0.00258735

10.6 Test Case Results

Total fuel cost, computation times, and a description of the resulting transfers are given for each of the test cases. The model used is the Circular Restricted Three Body Problem with 12th order and degree gravity for Phobos. Calculations were performed on an Apple MacBook Air laptop with a 1.3 GHz Intel Core i5 processor and 4 GB of RAM. The executable was built from C and run in Eclipse Juno. Most of the code was converted from Matlab using Matlab Coder, with the remainder being either the C build of the ECOS Second Order Cone Solver or coding directly in C.

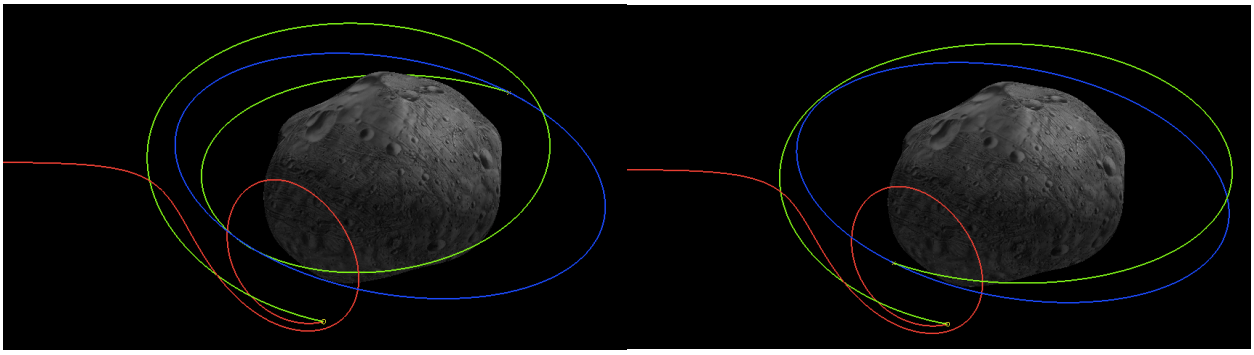


Figure 10.4: Transfers 1 (left) and 2 (right) for Example 1, aborting to a DRO (blue) after diverging from a Halo orbit (red). The transfers are shown in green.

10.6.1 Test Scenario 1

This process generates two transfers in parallel, one to each of the DRO's periapses. These examples are shown in Figure 10.4, with the uncontrolled escaping trajectory in red, the DRO in blue, and the transfers in green.

Table 10.5: Example 1, Transfer 1 Results

Total ΔV:	
After initial correction:	14.1259 m/s
Final:	10.5994 m/s
Computation Time:	
Translation to graph nodes:	0.0025 seconds
Load and search graph:	0.2218 seconds
Selecting arcs:	0.0377 seconds
Correction to feasibility:	0.0298 seconds
Optimization for fuel reduction:	0.4283 seconds
Total:	**0.7201 seconds**

Table 10.6: Example 1, Transfer 2 Results

Total ΔV:	
After initial correction:	11.3977 m/s
Final:	10.2878 m/s
Computation Time:	
Translation to graph nodes:	0.0004 seconds
Load and search graph:	0.1877 seconds
Selecting arcs:	0.0161 seconds

Correction to feasibility:	0.0242 seconds
Optimization for fuel reduction:	0.2366 seconds
Total:	**0.4650 seconds**

Example 1 demonstrates the ability to switch between orbit families with the addition that the initial condition is an off-nominal situation in a highly unstable region. In [Wall12], JPL engineers noted that the ΔV to transfer between DROs and either Halo or Vertical Lyapunov orbits is in the range of 10-20m/s, without providing the specific orbit pair leading to the minimum 10m/s value. *Therefore the fact that for this example a 10.2878 m/s transfer was found with no prior initial guess in less than half of a second demonstrates the method potential.*

Comparing the cost before and after optimization, it is worth noting that the fuel costs of initial guesses provided to the optimizer by the search and correction process were not more than a few meters per second above the final value. This shows that even with a somewhat coarse discretization – particularly for velocity directions – the initial guesses are quite reasonable. Keeping in mind the feasible iterate requirement in case of a system interrupt, it is important that these intermediate iterates have reasonable cost themselves, which is the case in this example. This example also shows the ability to generate multiple transfers to approach periodic orbits, even before the graph feedback step of the algorithm. In fact, in the multithreaded architecture developed for the C implementation of this project, the two transfer problems are handled by different calculation threads and so can be run in parallel.

10.6.2 Test Scenario 2

Since the original transfer is available as a reference, two types of transfers may easily be generated. First, the original remainder of the transfer may be differentially corrected to take the new initial condition into account, which is essentially a partial *local* optimization. In addition, the

whole process including the graph search, arc selection, and correction process may be run with the new boundary conditions. The result has a lower final ΔV than the local approach, and indeed is a very different transfer with an extra revolution. This shows the potential benefits the search process may give even when a reference trajectory is available, gaining a bit more global insight in exchange for a relatively small amount of extra time. The resulting trajectories are shown in Figure 10.5, with the original transfer in green (dash-dot for the first arc it expected to be executed, dot only for the remainder used as a reference trajectory), the local correction in dashed yellow, and the transfer from the full algorithm in solid teal.

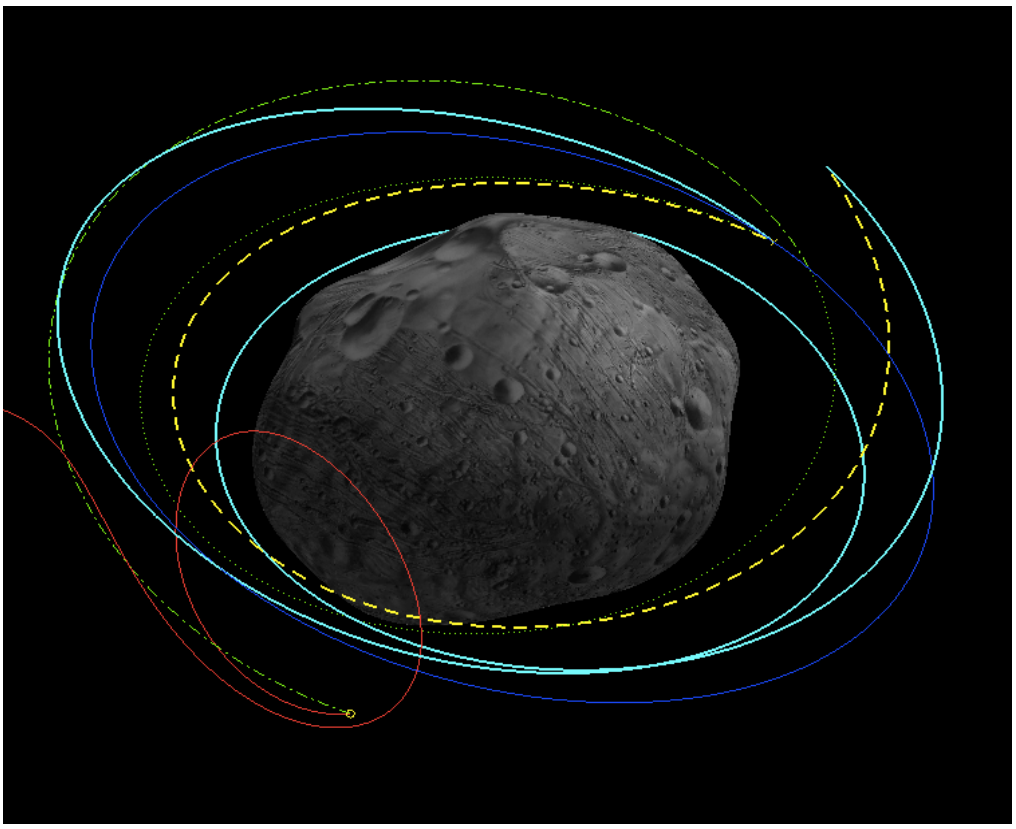


Figure 10.5: Replanning after an additional large perturbation part way through the transfer of Example 1. Graph search approach is shown in solid teal, local correction only in yellow dashes.

Table 10.7: Example 2, Local Correction and Optimization with Reference Results

Total ΔV:	
After initial correction	5.2837 m/s
Final:	3.1278 m/s
Computation Time:	
Correction to feasibility:	0.0197 seconds
Optimization for fuel reduction:	0.2655 seconds
Total:	**0.2852 seconds**

Table 10.8: Example 2, Full Replanning Results

Total ΔV:	
After initial correction:	4.5629 m/s
Final:	2.6957 m/s
Computation Time:	
Translation to graph nodes:	0.0005 seconds
Searching for itinerary:	0.1705 seconds
Selecting arcs:	0.0228 seconds
Correction to feasibility:	0.0333 seconds
Optimization for fuel reduction:	0.4430 seconds
Total:	**0.6701 seconds**

10.6.3 Test Scenario 3

Figure 10.6 shows the results of this scenario. The Vertical Lyapunov orbit is shown in blue dashes. The diverging trajectory is shown in red dots. The transfer from the initial condition on the

diverging trajectory to the orbit is shown in solid green. Note that this is ballistic with the exception of the endpoints, since the initial guess was a single revolution of the orbit. Breaking up the orbit into multiple smaller arcs would likely lead to a lower fuel cost and is possible within this method, but was not done in this example to demonstrate a single arc transfer in a dynamic region where single shooting is considered difficult. This sensitivity is also the reason the computation time is relatively long for this example despite only using a single arc. Since the step sizes are tied to the state transition matrix eigenvalues, the step sizes are very small in this regime, and are reduced even more if the prior step led to an increase in the cost function.

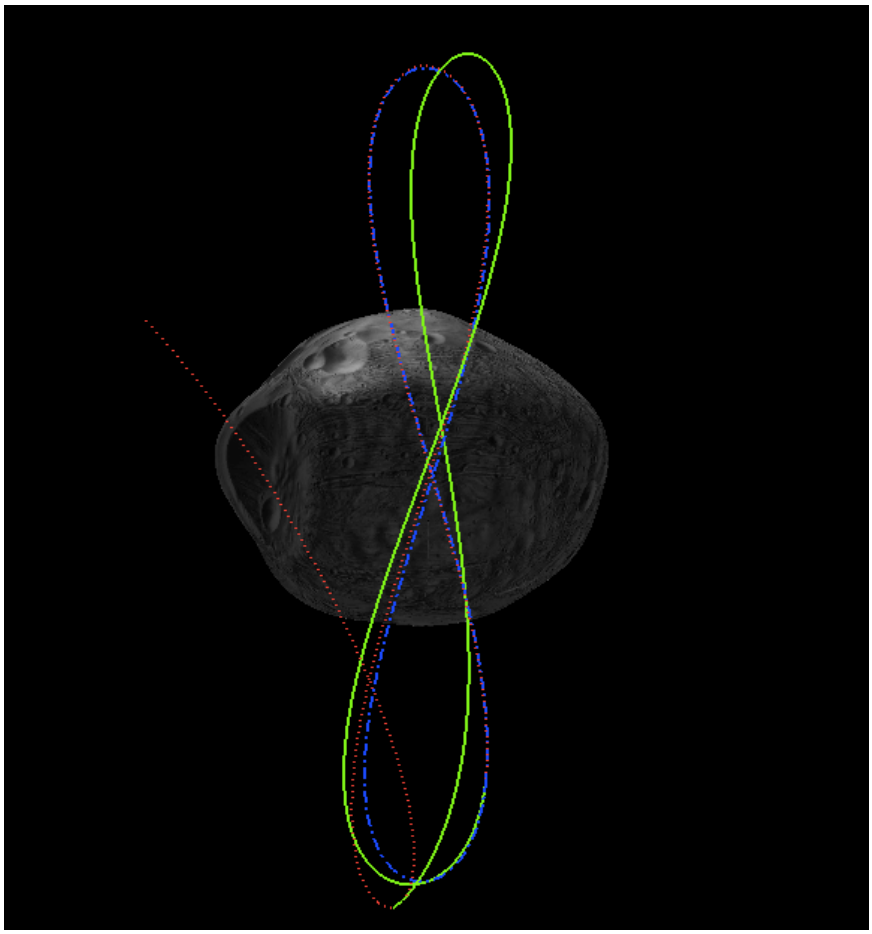


Figure 10.6: Before escaping (red dots) from a Vertical Lyapunov orbit (blue dash-dot), a transfer (solid green) is planned returning the spacecraft to the desired periodic orbit.

Table 10.9: Example 3 Results

Total ΔV:	
After initial correction:	4.4177 m/s
Final:	3.7269 m/s
Computation Time (local only):	
Correction to feasibility:	0.0304 seconds
Optimization for fuel reduction:	0.4462 seconds
Total:	**0.4766 seconds**

10.6.4 Further Computation Time Analysis

Additional information may be drawn by closer inspection of the computation times of different stages of the algorithm. The values that follow are averaged across the examples above.

The first observation is that during the search phase, the vast majority of the time is taken up by reading in the graph file. Indeed, on average only 5.95% of search phase – here defined as the reception of the start and goal nodes up until the graph itinerary is returned to the calling function – is taken up by the A* search itself. This suggests two things. One is that the basic A*algorithm is sufficiently fast without necessarily needing to use more complex but perhaps faster algorithm. It also suggests that better compression techniques would not only reduce the memory footprint but also the search phase run time in total. Compressed Column Storage [Duff89] is a common method that was not implemented in this build. A variation on this approach is used by Matlab, with the result that while the 70MB row/column/value format used in a .txt file is stored by Matlab as a 9MB .mat file.

Next are the breakdowns of the Correction (Chapter 9) and Local Optimization (Chapter 5) stages. For the Correction stage: Setup 89.82%, SOCP Solver 2.86%, Evaluation 7.32%. For the Optimization stage: Setup 44.48%, SOCP Solver 4.98%, Recorrection 44.18%, Evaluation 6.36%.

In both cases the Setup stage consists primarily of the arc endpoint and State Transition Matrix calculations that are determined via integration. Evaluation involves verification of cost reduction and constraints, also involving integration. Recorrection is the unconstrained minimization of position discontinuity (Chapter 6), also requiring STM information. Combining these, it can be seen that the vast majority of the time for both correction and optimization consists in integration needed to get the STM information. Note that such information would be necessary even if simpler QP subproblems were to replace the SOCP subproblems. The fact that the SOCP solver constitutes such a minor portion of the process suggests that trying to trim from this 3-5% of the process by using QP subproblems may be counter productive. The subproblem time savings as a percentage of the whole is minimal, even if the 3-5% were eliminated completely. Furthermore, as the linear constraints would reduce the accuracy of the approximations, more iterations and hence more time consuming integrations would need to be performed. The results do suggest however that different integration methods other than the standard Runge-Kutta should be investigated to determine the best fit for the application.

10.7 Summary

This chapter began by describing the basics of the motivation, dynamic environment, and challenges of an orbital operations plan at Phobos. Next, it provided details on how the trajectory redesign method described in Chapters 3-9 may be applied to a Phobos orbiter. Test cases that cover key off-nominal situations were discussed, with results showing efficient transfers generated in less than a second even with no reference trajectory provided. In addition to the individual test case results, analysis of the computation times by task provided insight into the strengths and improvable aspects of this approach.

Chapter 11

Implementation of Phobos Test Scenarios on Jet Propulsion Laboratory Flight Hardware

11.1 Introduction

Many algorithms in spaceflight research are developed with the claims of potential use onboard a spacecraft. However, it cannot always be assumed that any method developed on a desktop will be suitable to run on flight hardware without prior consideration of the restrictions of the environment. In this case, implementation is not simply a last step after the development of an algorithm, it must inform the theory and design of the algorithm itself.

Flight approved, radiation hardened platforms such as the RAD750 used on the Mars Science Laboratory, and other recent missions were developed over ten years ago. As a result, the computational abilities are quite limited by modern standards. In addition to hardware limitations,

challenges are also posed by additional rules due to fault protection, allowable libraries, etc. Lastly, real time systems face system interrupts, which must be accounted for within complex or long running processes. Therefore when considering all of these factors, it can be see that there is a sharp and growing mismatch between a flight system and the everyday platforms used to develop and benchmark algorithms.

The method developed and tested here is not mature enough yet for mission use, indeed it is important to note that it is not designed to replace the well tested guidance and control algorithms in place on the spacecraft. However, it is designed to supplement such systems where an off-nominal situation renders such methods insufficient and yet rapid action is needed onboard. Therefore, it has always been a goal of this project to successfully run a current incarnation of the method on Jet Propulsion Laboratory's RAD750 based, flight-like test platform. This was done in August 2014. This chapter summarizes the process to achieve this, highlighting those aspects where limitations of an onboard system informed the algorithm design.

11.2 The Test Platform

The RAD750 is a radiation hardened processor made by BAE Systems Inc. Developed in 2001, it was first deployed on the 2005 Deep Impact and Mars Reconnaissance Orbiter [Berg07] missions and continues to be used for current NASA/JPL missions including MSL [Gost13] and Juno.[Dodg07] The CPU clock speed is 200MHz, and test platform has 128MB of local DRAM, with additional (but slower) flash memory. Therefore both processing speed and memory are severely limited by the standards of even today's inexpensive laptop. The test platform and existing missions are run by the VxWorks real-time operating system developed by Wind River Systems Inc.

Flight Software Core (FSW Core) is a new flight software architecture and infrastructure developed by NASA/JPL in order facilitate consistency, reusability, and ease of development of

code. One feature of this system is the ability to compile and run code for the VxWorks operating system on desktop platforms (VxWorks Sim) so that testing and debugging are completed before needing to use the limited resource that is the actual flight-like hardware. FSW Core then allows an easy transition to run the executable on the RAD750/ VxWorks Flight test bed.

11.3 Review of Design Decisions Informed by Onboard Application

Here a short review will be given of the algorithmic choices from Chapters 3 through 9 that were influenced by the intent to develop an onboard oriented approach.

11.3.1 Feasible Major Iterates of the Optimization Process

The most important example of this in this project is the development of the modified SCP optimization method tailored to the application with a feasible major iterate requirement (Chapter 7) in case of a system interrupt. The optimization process is the lengthiest portion of the transfer design and so it is necessary to not use a method where only the final result is guaranteed to be feasible. Therefore once feasibility of *both* nonlinear (and indeed numeric) equality and inequality constraints is established by the faster initial correction process, the best feasible result is always stored after each major iteration and available to the larger system. This led away from standard implementations of SQP or SCP (or even "Feasible SQP" due to nonlinear equality constraints [Lawr96, Lawr01]) to an inclusion of the recorrection steps described in chapters 6 and 7.

11.3.2 Second Order Cone Problems and Interior Point Solvers

As described in Chapter 5.7, the cost and most of the constraints of the impulsive transfer NLP satisfy the structure of a convex problem once State Transition Matrix based approximations of the dynamics are used. As a result, convex subproblems suggest themselves as a natural choice. However, the use of Second Order Cone Problems (5.3) via the epigraph transformation (5.7.3, 5.7.7) is due to implementation requirements. In Chapter 5.3.2, results from SOCP researchers in

[Nest94, Peng09] were provided that certain classes of interior point methods are guaranteed to converge and indeed within a calculate, finite number of iterations for a given tolerance provided interior points exist. Although in some spaceflight applications the existence of interior points is often assumed without note due to physical arguments, Appendix A demonstrates that the iterative process will not eliminate this property.

Aside from the desirable convergence properties, there exist open source SOCP solvers using such interior point methods. ECOS [Chu13, Doma13] is such a package that was developed specifically for embedded applications. As a result, its build in C uses very few nonstandard libraries. In particular, all of the ECOS code was able to be used on the VxWorks platform without modification. It also has builds in Python and Matlab for ease of prototyping. Within Matlab it is also able to be called by CVX. [Gran08, Gran13] Without the CVX interface, the input formats match those in the C build. Therefore code may be written to arrange the inputs in the ECOS format and easily verified that the outputs match the results when using CVX as an intermediary. This formatting code may then be converted into C with minimal changes and again the output may be matched to earlier results in Matlab.

11.3.3 A* Search Algorithm

Concerns regarding guaranteed convergence also played a role in the selection of the heuristic search method. So called Greedy Search methods that use a heuristic function only to determine which node to evaluate may be faster than A* in some applications due to steering directly towards the solution rather than incorporating the running cost. However, Greedy first methods may be incomplete, meaning they might not even find the goal node, let alone on the optimal path. [Russ09] A* search is provably complete and with additional conditions (4.3.3, 4.3.4) can be proven to find the optimal path within the graph.

For the examples tested thus far, the memory usage of the A* algorithm itself did not cause

any problems with the limitations of the RAD750 system, however this is not guaranteed. Thankfully, memory bounded variations of the algorithm exist that would be suitable to guarantee no such issues occur. Given a specified cap on available memory for the search process, the SMA* (Simplified Memory-bounded A*) algorithm has been shown to be complete if any solution exists within the memory cap.[Russ09] Additionally, if the optimal path is reachable within the memory constraint, it will find it, otherwise it finds the best reachable path. [Russ09]

13.3.4 Integration: A Remaining Challenge

At present, standard Runge-Kutta integration methods are being used in this project. 8^{th} order Runge-Kutta integration has been implemented on the AutoNav autonomous navigation software [RiedAN] flown on multiple missions including Deep Space 1.

Although such methods have been used onboard, there is perhaps room for improvement. For a fixed accuracy, an indeterminate number steps needed per integration due to the variable step sizes, which poses a problem in onboard use. Otherwise to mitigate this, as in the case of AutoNav, [RiedAN] step size limits must be implemented which can affect accuracy. Future work would be beneficial to determine whether suitable analytic approximations exist for the trajectories and State Transition Matrices (for short timesteps), or whether new integrators under development can be shown to avoid this problem at least for the relevant application. One approach that looks promising is that of Modified Picard Integrators using the Parker-Sochacki method which uses coordinate transformations and a simple set of operations on resulting polynomial representations of the transformed variable. [Nakh14] While not allowable for arbitrary systems, it has been shown to be applicable to multibody gravity as well as spherical harmonic gravity models. These forces dominate the dynamics of most planetary moon environments, although further work will be needed to see if Solar Radiation Pressure may be incorporated. Another possible benefit that needs to be investigated is whether the simplicity of the method that allows it to run with the calculation

restrictions of Graphical Processing Units (GPUs) [Nakh14] will also allow it to be run using Field Programmable Gate Arrays (FPGAs). While GPUs may eventually be cleared for space qualified platforms, FPGAs are currently approved and are used to quickly perform certain types of parallel computations. [Mour09]

11.4 Graph Memory Reduction via Run Time Impulsive Connections

One situation did arise where a major change was needed due to platform limitations that was not discovered until the porting process. Due to memory limitations, a change was made in which impulsive links in the directed graph (see 3.4.3) were no longer stored in a data file along with the ballistic links. Instead, they can be calculated in real time as part of the search process.

11.4.1 Background

The flight like computing platform has 128MB of DRAM local memory available to the processor without needing to access slower, flash memory. Running the VxWorks flight operating system alone reduces this to approximately 96MB. [Mang14] Additional programs will reduce this amount further, creating a limit on available memory.

Relative to the above limitations, the file containing the graph used in Chapter 10 is 70MB in size. Now, as was stated in 10.6.4, the same data was stored in Matlab as a 9MB file using their version of the Compressed Column Storage (CCS) [Duff89] file format. Further possible steps could include dividing the file into multiple files with one per energy level, accessing on those needed at any given time. These alternatives were considered insufficient for various reasons. Opening and closing sections of the graph is hampered by the fact that at each node, adding adjacent nodes to the frontier (Chapter 4.3.2) involves impulsive connections with multiple energy levels at every step as well as ballistic connections with positions in very different positions. Therefore dividing the graph into multiple files – and accessing each as needed – by either energy or position would

result in significant file I/O. In the examples tested, time spent during the search phase due to file I/O took an order of magnitude more time than running the A* algorithm itself. As a result, such an approach would greatly increase the search time.

A better storage format such as CCS should definitely be considered over the simplistic row, column, value format currently in use. In the case of the 70MB graph, then a reduction to 9MB would likely be sufficient. However, the large size of Phobos relative to its libration point region (creating a large forbidden region) and other features of the system lead to a partition that is limited in extent and resolution but still suitable to the application. A larger graph was created that may be more representative of graphs in other applications, and is 330MB in the row, column, value format and approximately 50MB in the Matlab CCS format. In such a case the format change is an improvement but not sufficient by itself, therefore additional measures are needed.

Comparing the relative memory usage due to impulsive and ballistic connections, it was seen that impulsive connections constituted over 90% of the 70MB graph and 95% of the 330MB graph. Therefore impulsive connections became the target for memory reduction. The approach taken is to calculate the impulsive connections when needed during the search process.

11.4.2 Identifying Connection Candidates

Looping through the entire list of nodes to determine which nodes to add impulsive connections to is inefficient, and storing a list of approximately one hundred such nodes per every node would require a significant amount of memory. Therefore, the list of nodes with matching positions is calculated with a few operations on lists of indices, as is now described.

Determining which nodes to add impulsive connections to begins with the property that impulsive connections only exist between nodes representing the same positions, but with different velocities. The regions are grouped by $J, r, \theta, \phi, \alpha$ in that order. Let R be the number of regions per energy layer, and A be the number of velocity directions. Then for node n (which is simply an

integer starting from 0), $L_1=(n-mod(n,A),n-mod(n,A)+1,\ldots,n-mod(n,A)+(A-1))$ gives the list of nodes which match in all aspected except possibly velocity direction. Let $L_2=mod(L_1,R)$, where the modulus is done element-wise on L_1. This list of values then signifies that the A nodes that match in position within each energy layer occupy the $L_2^{(0)}$th through $L_2^{(A-1)}$th nodes of that layer. Assume there are E number of energy layers. Therefore, with the addition here signifying each element of L_2 is added to by the constant, $L_3=(L_2,R+L_2,2R+L_2,\ldots,(E-1)R+L_2)$ provides the list of $E\times A$ nodes that match in position with node n. The final list of nodes L is created by removing node n itself, and any nodes that have been marked as having a greater potential energy than the energy of their layer and are thus inaccessible. It is this short list of nodes that is then evaluated to determine the impulsive link cost. Assembling such a list directly using index arithmetic instead of iterating through every node greatly reduces the number of iterations that need to be performed.

As a simplified example, suppose the graph in question has 3 layers of 100 nodes, with each layer defined by 20 positions and 5 possible velocity directions. Consider the list of possible impulsive connections for node 213. From the above, we see that $L_1=(210,211,212,213,214)$ is the list of nodes with the same energy level (the fourth) and position region (the third). $L_2=(10,11,12,13,14)$ signifies that the 10th through 14th node (counting from 0) on each layer will have the same position. Therefore

$$L_3=(10,11,12,13,14,110,111,112,113,114,210,211,212,213,214)$$

is the list of all nodes with the same position as node 213. Removing node 213 itself results in the final list of candidates for impulsive connections:

$$L=(10,11,12,13,14,110,111,112,113,114,210,211,212,214)$$

11.4.3 Link Costs

In reality, the $\|\Delta V\|$ in the constructed transfer corresponding to an impulsive connection between nodes n_j and n_k is a function of the ballistic arc which ends in the region associated with n_j and with the ballistic arc beginning in the region associated with node n_k. Regardless of the approach taken to assign a cost to the n_j to n_k transition, it is simply an estimate of a value that is unknowable at the time of the search process. In Chapter 3.4.3, the average $\|\Delta V\|$ between the states of two regions with matching position was calculated via an iterated integral, taken as the impulsive link cost, and stored in the graph file. For less memory limited applications this approach can still used. However, in order to maintain a fast search process, it is desirable to avoid integration if possible during the search process. One method is to determine what the average velocity vector is for each region. The link cost is then taken to be the $\|\Delta V\|$ between these two vectors. For *disjoint* sets such as the regions in question, these two values can be quite similar and can be shown to be identical for disjoint sets in one dimension. Indeed, the same paths were selected for the examples from Chapter 10, resulting in the same arcs being selected and hence the same initial guesses.

In order to facilitate quick calculations at the time of the search process, the average velocities vectors are stored for each node in the Cartesian CR3BP coordinates. Such a table has as many rows as nodes and four columns, storing $(n,(\bar{v}_n)_x,(\bar{v}_n)_y,(\bar{v}_n)_z)$ per row where $\bar{v}_n$ is the average velocity associated with node n. During the search process, when constructing the frontier for some node n_j, the list of nodes $L(n_j)$ that will have impulsive connections is quickly determined using the index calculations above. For each node $n_k \in L(n_j)$, an impulsive connection is added between nodes n_j and n_k with link cost $\|\bar{v}_{nj}-\bar{v}_{nk}\|$.

11.4.4 Memory Reduction and Search Time Impact

Recall that the original graph sizes storing both ballistic and impulsive connections are 70MB for the Chapter 10 example and 330 MB for an alternative larger graph relying on fewer application specific reductions. Using the above approach, these graph files are replaced with a graph file only containing ballistic connections, and a file containing the average velocity for each node which is then used to calculate the impulsive connections during the search process. The 70MB graph was reduced to a 2.4MB ballistic connection graph file and a 2.8MB file containing the average velocities per node, 5.2 MB in total. As impulsive connections were even a higher percentage of the 330MB, the reduction was even more drastic, with a combined 13MB total. Note that this reduction did not involve a difference in how the graph files were compressed. Therefore in terms of memory this process succeeded in its goal.

However, it is also important to examine the impact in search times since more – if simple – calculations are performed during the search process. Indeed, across the test scenarios involving a search phase, the A* search algorithm run times increased 21-27% with the changes just described when tested on the MacBook Air (from 0.011-0.014 seconds to .014-.017 seconds). However, the search algorithm itself is less than 10% of the total run time of the search phase, most of it involving the file I/O of the graph using fscanf. As a result, in all cases the total search phases time was *reduced* due to the much smaller file size even though the A* algorithm itself took slightly longer. The reduction in the search times as a percentage of the total is provided in 11.6.2. Although a more detailed study on the effect of this change to the graph costs overall would need to be performed, with the reduced memory, faster total run time, and matching paths for the few examples considered, this approach was considered to be successful and implemented on FSW Core given the limited access time to the platform.

11.5 Porting Process

11.5.1 Conversion to C

As discussed in 11.3.2, the ECOS SOCP solver has builds in both Matlab and C and so the subproblem solver code was able to be used directly in C. Most of the remaining calculations including converting boundary conditions to nodes (4.2), equations of motion (Appendix C), integration, arc selection (4.4), and cost and constraint matrix construction using arc states and STM submatrices (5.7.6-5.7.7, 9.3.1) were converted from Matlab using Matlab Coder. As a result the code has not been optimized for C, which may lead to further improvements in calculation time. Due to limitations the types of conversions Matlab Coder can handle, iteration control and wrappers around converted code were coded directly in C.

These calculations were distributed within the multiple threads defined by a real time software architecture plan developed with JPL flight software engineer Kim Gostelow. A description of the proposed design with a full calculation breakdown, state machines, and messages is too lengthy for this document. Indeed, with the limited platform access time, only the calculation process was able to be implemented and tested using FSW Core.

11.5.2 Platform Porting Phases

The first step was to move the code from the MacBook Air used for development to a Linux system as a bridge before VxWorks operating systems used in the FSW Core environments. At this stage JPL flight software engineer Lloyd Manglapus was able to identify and correct issues such as missing execution dependencies, code standard issues, and improperly initialized variables.

The next step was to port the code to the FSW Core simulation environment (VxWorks Sim), which uses the same VxWorks operating system and libraries at the target platform. At this stage the graph memory issue discussed in 11.4 above was identified and then resolved. Additional code modifications by Lloyd Manglapus were needed to eliminate the need for libraries that are

required by the converted Matlab code but are not allowed on the VxWorks system. This includes the handling of 'inf' (infinity) and 'NaN' (not a number) values, which were able to be removed in the existing C code. The Matlab Coder produced C code to handle variable matrix sizing also required unallowable libraries and needed to be removed. Changes were made in Matlab to precompute the needed size, and this updated code was then moved through the porting process. The work product was packaged as a FSW Core component and used the FSW Core infrastructure to build the final executable.

After successful testing in the simulation environment – meaning the output matched the results of the C run with the same setup – the executable was then moved to the RAD750 based flight-like platform (VxWorks Flight). The test cases were run on a quiescent system and again verified that the output matched the expected values. These results are presented in the next section.

11.6 Simulation Results

11.6.1 Test Case Performance Data

The test cases and parameters for which calculation time results are given are precisely those of 10.5.1-10.5.3 and 10.6.1-10.6.3. Therefore the scenario descriptions, transfer images, and characteristics will not be repeated here. The data is presented in Tables 11.1-11.5.

Table 11.1: Example 1, Transfer 1 Results

Total ΔV:	
After initial correction	14.1259 m/s
Final	10.5994 m/s
Computation Time:	

Translation to graph nodes:	0.01 seconds
Load and search graph:	4.78 seconds
Selecting arcs:	1.17 seconds
Correction to feasibility:	1.45 seconds
Optimization for fuel reduction:	23.97 seconds
Total:	**31.38 seconds**

Table 11.2: Example 1, Transfer 2 Results

Total ΔV:

After initial correction	11.3977 m/s
Final	10.2878 m/s

Computation Time:

Translation to graph nodes:	0.01 seconds
Load and search graph:	4.56 seconds
Selecting arcs:	.84 seconds
Correction to feasibility:	1.02 seconds
Optimization for fuel reduction:	12.98 seconds
Total:	**19.41 seconds**

Table 11.3: Example 2, Local Correction and Optimization with Reference Results

Total ΔV:

After initial correction	5.2837 m/s
Final:	3.1278 m/s

Computation Time:

Correction to feasibility:	0.89 seconds

Optimization for fuel reduction:	13.48 seconds
Total:	**14.38 seconds**

Table 11.4: Example 2, Full Replanning Results

Total ΔV:

After initial correction:	4.5629 m/s
Final:	2.6957 m/s

Computation Time:

Translation to graph nodes:	0.01 seconds
Searching for itinerary:	4.48 seconds
Selecting arcs:	1.55 seconds
Correction to feasibility:	1.88 seconds
Optimization for fuel reduction:	25.30 seconds
Total:	**33.22 seconds**

Table 11.5: Example 3 Results

Total ΔV:

After initial correction:	4.4177 m/s
Final:	3.7269 m/s

Computation Time (local only):

Correction to feasibility:	1.45 seconds
Optimization for fuel reduction:	24.20 seconds
Total:	**25.66 seconds**

11.6.2 Result Comparison

As expected due to hardware limitations and operating system differences, the run times are longer than those on a new laptop, ranging from approximately 20-35 seconds in these cases. Should the algorithm be considered for a mission, it will need to be determined what computation times will be when run on a non-quiescent system with a potentially more complex model. However, the code would also be optimized rather than using a conversion from Matlab. It is hoped the result will still be less than the round trip communication time from Earth to Mars, which ranges from 6 to 42 minutes, with the MSL landing famously having a 7x2=14 minutes round trip communication delay.

One item to note relates to the calculation at search time of the impulsive graph connections. It was claimed above that although the A* run times themselves increased slightly, that this was more than offset by a decrease in file I/O time. Comparing the percentage of the search phase time relative to the whole, there is a decrease between the results of Chapter 10.6 vs. the RAD750 results of this chapter. There was a decrease from 30.8% of the total to 14.3% of the total for transfer 1 of test scenario 1, 40.3% to 23.5% for transfer 2 of test scenario 1, and 25.5% to 13.5% for the full replanning transfer of test scenario 2.

The breakdown of computation time for the correction and optimization stages 10.6.4 is essentially unchanged, with the only notable decrease being the percentage of the optimization process taken up by the SOCP solver showing a decrease from approximately 5% of the whole to 3-4% of the process.

11.7 Future Opportunities: Parallelization

There are efforts underway by NASA, the AFRL, and Boeing to create the next generation of space-qualified hardware. [Doyl14, Some13, Alex12, Doyl12] Such systems will be significantly

more powerful than the RAD750 and will feature multicore processors; in the case of the Tilera64 based Maestro unit under development by Boeing, there are 64 cores. [ViRe11] In a joint NASA/AFRL evaluation for increasing parallelization in flight processors, a preference was established for multiple general purpose multi-core processors in the future rather than parallelization via FPGAs or GPUs. [Doyl14] These advances should drive investigations on how to speed up the existing capabilities of the transfer redesign tool as well as the possibility of additional functionality.

11.7.1 Naive Transfer Problem Parallelization

Multiple, independent transfer problems may be created when there are multiple goal states to be evaluated. Recall that in Chapter 4.2, (or indeed Chapter 12) that periodic boundary conditions can allow multiple transfer problems to be created, each designed using different departure/insertion points on the orbits. Each of the resulting problems are completely independent of one another and so may be run in parallel. Another example is discussed for designing flyby transfers in 12.3.5. In this case full states are provided as input to the design process with the final relative velocity removed from the cost function to enable flybys. As the optimization process is a local one, the resulting transfer is dependent upon this initial guess. Flyby transfer design would be greatly enhanced by the ability to use multiple approach directions as initial guesses in order to find transfers that are both lower cost and improved viewing properties.

In a more general context, as spacecraft become more autonomous, even higher level decision making will be handled by the spacecraft. Thus not only will a transfer need to be designed for a given goal, but the goals themselves will need to be determined autonomously. In such a case, transfer information will need to be provided for multiple options during the evaluation process, not just once a final boundary condition has been provided. In such a case the transfer problems are independent of one another and so may be run in parallel if the resources

allow.

11.7.2 Search Process Parallelization

There has been recent research on parallel implementations of the A* search algorithm. One promising example is Hash-Distributed A* (HDA*). In [Kish09], speed-ups of 3.6-6.3 times were seen on a single 8-core processor as well as on a computing cluster, with speed-ups an order of magnitude above these in clusters with 64 and 128 cores total. It was observed as well that there was almost no CPU idling due to synchronization overhead, in contrast to other attempts at A* parallelization. [Kish09] Depending on how many cores are available to the search process, such an approach can lead to a faster solution. In the case of the 64 core Maestro being developed, this is a likely scenario. The additional cores also increase the amount of fast memory available, reducing the impact of memory-bounded modifications such as SMA* (11.3.3). [Russ09] Note that if GPUs become used on flight systems, similar research is underway for GPU based A* algorithms with impressive results. [Blei08] In addition to improved performance on existing graphs, such changes could also enable the use of larger graphs by the transfer design tool.

11.7.3 Run Time Ballistic Connections and Uncertainty Constraints

Recall that in Chapter 11.4 that a change was made to calculate impulsive connections at run time rather than store these in a graph file due to memory limitations. Suppose such an approach were to be made for ballistic connections. One benefit of doing so would be the use of a higher-fidelity, time varying dynamical model in the search process, eliminating mismatches between the search and optimization models. Another is reducing the memory used for the graph file and arc database, particularly memory used to for portions of the domain unused for some particular transfer. Doing so, however, would require multiple integrations to be performed in order to approximate the system flow of the region corresponding to the current node in order to add ballistic connections to the frontier. It should be evaluated if performing these integration in

parallel make this approach feasible.

Another use of integration during the search process is incorporating some uncertainty analysis during the design process by estimating the state covariance for the transfer being constructed. As this depends on the orbit determination characteristics at the beginning of the transfer, such information cannot be stored ahead of time. Note that the trajectory and STM information needed for propagation using the Extended Kalman Filter requires a single (n^2+n dimension where here n=6) integration [Cras04] and so is not easily parallelized. Propagation using the integration of multiple sigma points to calculate uncertainty statistics used by the Unscented Kalman Filter involves the integration of 2n+1 trajectories which may be done in parallel. [Nakh13] For a 6 dimensional state space, these compare a single integration of 42 states and 13 parallel integrations of 6 states for 78 total. It should be evaluated if the parallelization of the integrations results in faster UKF vs. EKF performance. Note that GPU based integration techniques [Nakh14] mean that should GPUs be used on flight systems this could be parallelized in this way as well on such systems. Additional UKF benefits such as higher accuracy, the ability to use higher order statistics, and possible synergy with run-time ballistic connections described above (by using the same trajectories for both phases) suggest that if the run time due to paralellization is reduced or equivalent, then the increased use of the UKF onboard should be seriously considered with the introduction of multicore processing.

11.7.4 Naive Parallelization during the Optimization Stages

Integration of both ballistic arcs and their State Transition matrices are required throughout the Correction and Optimization phases of the transfer design process described in Chapters 5-9, and in fact constitute the majority of the calculation time. However, these integrations are done on an arc-by-arc basis and thus are independent. Therefore this may be very simply calculated in parallel with the integration of each arc handled by a different core.

11.8 Summary

This chapter began by describing the current flight hardware based test platform used at Jet Propulsion Laboratory, and the resulting limitations this causes for an onboard system. This information, along with requirements for an onboard trajectory design system in general, creates the background from which various algorithm design decisions and necessary changes were made. After the porting process to the VxWorks Flight platform was described, the computation results of the successful implementation on the flight-like system are given. As an algorithm designed with many challenges of onboard implementation in mind, this was a major milestone and rare opportunity.

Chapter 12

Libration Point Orbit Based

Near Earth Asteroid Interceptor

12.1 Introduction

A mission concept is in the early stages of investigation by mission design staff at Goddard Space Flight Center (GSFC) that would keep a constellation of small satellites in a libration point orbit, and from there deploy them to encounter near earth asteroids. As a first step, example transfers from such orbits to several representative asteroids were needed to determine if the ΔV required is excessive for such satellites. Additionally, there was a desire for comparisons in performance between flyby and rendezvous encounters, orbit types, orbit energies, asteroid energy, and asteroid inclination. Due to a gap in existing tools to quickly generate such transfers, GSFC extended an invitation for the transfer redesign tool described in this dissertation to be demonstrated for this application.

This chapter begins by describing the candidate orbits and target asteroids for which transfers were generated. Next, the necessary setup is described with a focus on those changes to the partition format, models, and optimization algorithm needed for this application. Example transfers are provided, along with an analysis of the resulting data as a whole.

12.2 Boundary Conditions

12.2.1 Candidate Periodic Orbits

Four different periodic libration point orbits at at the Sun/Earth L1 point were selected as initial conditions for these transfers. These consist of Halo and Vertical Lyapunov orbits at two different energies in order to have examples of both high and low *z*-amplitude initial orbits. Using the mass parameter $\mu = 3.035910E-06$ [Gome05] these energies are J=3.000771793, J=3.000487636 using definition of the augmented potential as given in [Koon08]. The low *z*-amplitude Halo orbit is in fact the orbit used in the design of the Genesis mission, with coordinates as provided in [Gome05]. The high *z*-amplitude Halo was taken at an energy level so that the *z*-amplitude would be approximately double the Genesis orbit. The Vertical Lyapunov orbits have the same energy at the corresponding Halo orbits. These orbits are defined in the Sun/Earth CR3BP by the following states and period, provided in normalized CR3BP units.

L1 Halo (northern family) – Low z-amplitude

x=0.9922709412937017, y=0, z=-.002456251256325228;
xdot=0 , ydot=-.01191138815471799, zdot=0
Period=3.0520117049

L1 Halo (northern family) – High z-amplitude

x=0.995474296562891, y=4.43824956048690e-06, z=-0.00491066844907487
xdot=2.73104523229373e-06, ydot=-0.0212098498347536, zdot=-1.15620247379016e-5
Period=2.96888665668952

L1 Vertical Lyapunov– Low z-amplitude

x=0.990405225236362, y=-3.57127182306636e-05, z=-0.000214573873165003
xdot—9.75826821985566e-05, ydot=0.00100244182138559, zdot=0.0113437719064818
Period=3.21162744779924

L1 Vertical Lyapunov– High z-amplitude

x=0.991328040634016, y=-5.77249799238712e-06, z=-2.26524536307027e-05
xdot=-2.39220898303787e-05, ydot=0.00361046734991688, zdot=0.0204302109956262
Period=3.50025021542029

12.2.2 Initial Conditions Used

For each periodic orbit, 12 initial conditions were considered, evenly spaced around each orbit. These conditions are found by propagating the above initial conditions in the CR3BP by increments of the provided orbital periods divided by 12. This approach is used for multiple reasons. First is that the trajectory design system is set up to depart from and insert into specific states along periodic orbits, rather than searching along the orbit for a global minimum. For this particular application this is not a liability because the approach of an asteroid determines when a satellite will need to depart an orbit, and hence its position on that orbit. Sensitivity to initial orbit phase would also indicate that a small constellations of satellites is truly needed in order to quickly intercept an asteroid within fuel limits.

12.2.3 Candidate Asteroids

The following asteroids were selected due to a range of energies, inclination, and influence on the trajectories by the Earth/Moon system. Initial conditions are provided in normalized CR3BP coordinates. In order to determine these coordinates, first the ephemeris coordinates at the specified date in a Heliocentric inertial frame for both the asteroid and the Earth were retrieved from the JPL Horizons website. The coordinates were transformed so that the Sun-Earth line became the *x*-axis and Earth's velocity lay in the *xy*-plane. The relative position vector of the asteroid relative to the Earth in this frame was then scaled into CR3BP normalized distance, and

then added to the position of the Earth in the CR3BP. Similar calculations were done to establish the angle of the Moon relative to the Sun-Earth line, and so to determine its position at encounter time when the Sun/Earth/Moon Bicircular Four Body Problem (BR4BP, Appendix C) is used as described below.

2006 RH120, January 13, 2007 0:00:00. An example of a low energy "mini-moon", an asteroid temporarily captured by the Earth with multiple, high inclination passes. This is similar to the temporary capture of the comet Oterma by Jupiter during its resonance crossing. [Koon08] Similarly to the situation of temporarily captured bodies such as Oterma, the energy of 2006 RH120 is close to that of various libration point orbits, in particular those of the higher *z*-amplitude periodic orbits taken as initial conditions (J=3.000443957 vs. 3.000487636 for the orbits). Due to its relatively low energy, it was evaluated for both rendezvous and flybys. Its state at encounter is given by:

x=995501880026699, y=0.000682447690650, z=-0.000433111895835
xdot=-0.006647867581849, ydot=0.009173412041385, zdot=-0.028436596556750

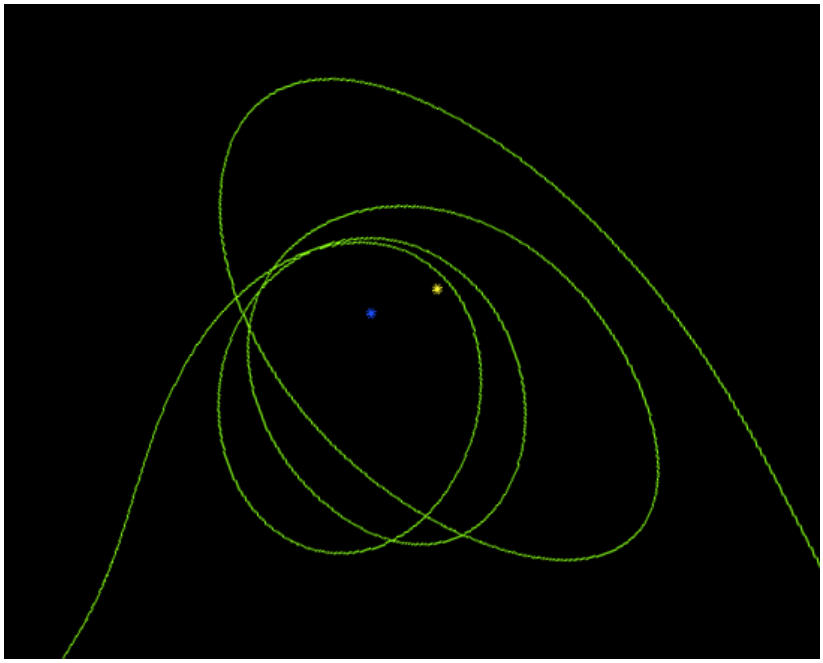


Figure 12.1: Trajectory of 2006 RH120 plotted in an Earth centered, psuedo-inertial frame. Note the temporary capture and high inclination.

2011 UD21, October 13, 0:00:00. Higher energy (*J*=2.998618437) than 2006 RH120, although this asteroid is not temporarily captured, its trajectory is nonetheless highly perturbed by its Earth flyby. Therefore it serves as a transition case between 2006 RH120 and the remaining high energy asteroids. As a result, it was evaluated for both rendezvous and flyby. Its state at encounter is:

x=1.000645292897312, y=0.004444513369061, z=-0.001965555338808
xdot=0.047698817844520;, ydot=-0.003756089804374, zdot=0.017862047954360

2005 YU55, November 9, 2011 0:00:00. A very high energy asteroid (J=2.7906811309) with a moderate inclination. It was evaluated for a flyby only due to its extremely high velocity relative to the Earth. Its state at encounter is:

x=1.000014862362460, y=-0.001912019823162;, z=0.001044249239317
xdot=0.459336314799381, ydot=-0.032365393111594, zdot=0.007631780503715

2004 BL86, January 26, 2015 16:20:00. Another asteroid with a very high velocity relative to the Earth (J=2.7239701943), but with a very high inclination. Evaluated for flyby only. Its state at encounter is:

x=1.006936121626134, y=0.000101706979815, z=-0.004017107759411
xdot=0.263067614613010, ydot=0.019974127647932, zdot=0.455304685964308

12.3 Method Setup and Modifications

12.3.1 Model Selection

As recommended in Chapters 3 and 10, in order to reduce the size of the directed graph used by the search process, a time invariant system is ideal for the discretization model. As such the CR3BP is used (Appendix C), with mass parameter $\mu = 3.035910E-06$. However, as the Moon has a noticeable effect on the dynamics of 2006 RH120 and could potentially be used for a flyby by the optimization process, it could not be omitted from the final model. Therefore the Bicircular Restricted Four Body Problem (BR4BP) is used for the correction and optimization

model, introducing the time varying effect of the Moon's gravity. This model is described in Appendix C with the parameters used for this application.

12.3.2 Domain Partition

Since the initial conditions are known ahead of time to be states along the Halo and Vertical Lyapunov orbits, on-orbit initial conditions and off-orbit intermediate states have been assigned different regions, even if those regions should overlap in position. (The partition definition below clarifies this concept.) This leads to more accurate orbit departure arcs, and is a major change from the coordinate-only discretization of the Phobos application in Chapter 10.4.1. As described in Chapters 3 and 10, the simulations used to generate the ballistic connections and graph layers themselves are associated with a set of specified energies. For each energy level, the partition structure is further split into the following types:

Partition Section 1 – States on Orbits:

1 region per the twelve states – hereafter the "orbit phase" – along the orbits taken as possible initial conditions as described in 12.2.2 above. This results in 12 regions per orbit.

Partition Section 2 – Positions on Orbits, post-Impulse Velocities:

For each orbit phase, the position is fixed to that of the states on the orbit, but velocities are allowed to vary to create departure maneuvers. For each specified energy, the direction sphere is divided into 24x24=576 velocity directions. The velocity magnitude is scaled to match the specified energy of the partition layer. This results in a total of 12x24x24=6912 regions per orbit per energy level.

Partition Section 3 – Standard partition of periapsis Poincaré section:

Position is specified in spherical coordinates as in Chapters 3 and 10. Velocity is then specified by the angle of the velocity relative to the xy-plane.

- r: 14 regions from 10,000-16,000,000 km, with region boundaries given in normalized

units by: 1/1.496e8*[10000:50000:400000,400000:200000:1600000]

- ϕ: From $-\frac{\pi}{2}$ to $\frac{\pi}{2}$, divided into 9 regions.
- θ: From $-\frac{\pi}{12}$ to $2\pi-\frac{\pi}{12}$ divided into 12 regions.
- α: From $-\frac{\pi}{12}$ to $2\pi-\frac{\pi}{12}$ divided into 24 regions.

This results in 14x9x12x24=36288 such regions per energy layer.

12.3.3 Resulting Directed Graph

A directed graph combining ballistic and impulsive connections is created as described in Chapter 3, with a couple of exceptions. Note without a maneuver, states on the orbit remain there. Therefore these form no ballistic connections with the other partition sections, only impulsive connections with partition section 2 – which are states whose positions are on the orbits but with different velocities. Partition section 2 initial conditions are integrated and thus form ballistic connections into partition section 3, which is the standard division of the state space. This approach forces initial conditions – states with a given orbit phase – to deorbit via an impulsive maneuver and then coast for a time before the next maneuver. As regions 1 and 2 are only used to link these non-standard initial conditions on the initial periodic orbits to the standard domain, no links are made *to* them from nodes corresponding to regions of the standard state space domain.

The resulting graph has 358,128 nodes (due to technical reasons needed for fast index manipulation, nodes corresponding to copies of all four orbits and their departure velocities for all three energy levels are repeated per layer, hence a greater number of nodes than the number of regions given in the partition. However, these dummy nodes are unused). An average of 8.7 connections exist per node, resulting in a sparsity of 0.0024%. Of these, over 80% of the connections are impulsive. The resulting .mat file is 11.3 MB.

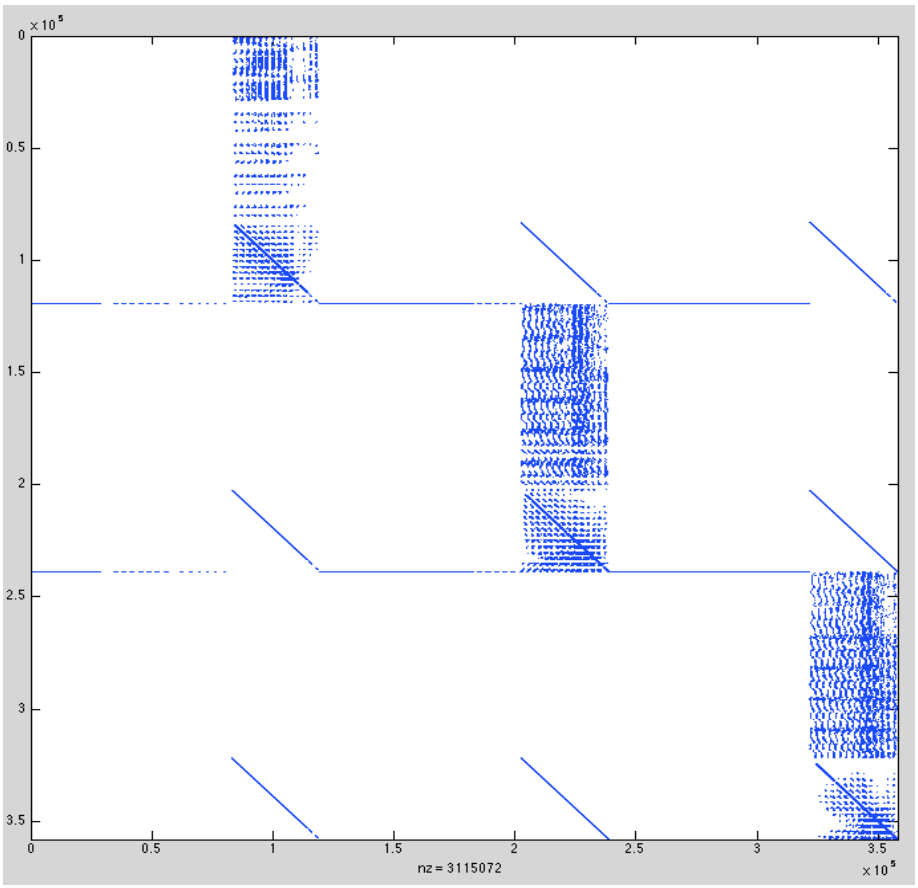


Figure 12.2: Sparsity diagram of the resulting graph described in 12.3.3. Each pixel is a link between two nodes. The 'from' nodes are arranged vertically, the 'to' nodes horizontally.

12.3.4 Optimization Modification for Flybys

For rendezvous trajectories, the correction and optimization process is exactly as described in Chapters 5-9, with the model being the BR4BP. As this method was designed with specific target states in mind as is the case for standard transfer design, an adjustment needs to be made to enable flybys at encounter rather than rendezvous. This is in fact quite simple to do: the final maneuver to match the asteroid velocity is simply dropped from the cost function. The existing constraints guarantee that like a rendezvous the spacecraft must match the asteroid position at the given encounter time, but the relative velocity must no longer be made to match with a final maneuver.

12.3.5 Specifying Boundary Conditions

The transfer initial condition is specified simply as an integer (1-12) corresponding to the orbital phase as described in 12.2.1 and 12.2.2. For rendezvous, the target conditions are precisely the states of the asteroids at encounter given in 12.2.3. As mentioned above, for flybys the final relative velocity is dropped from the cost function. Therefore the state at encounter may still be provided, with only the position and time being true boundary conditions and the velocity of the state providing an initial guess for the process translating states to graph nodes before the search, arc selection, and optimization process. For the high energy 2005 YU55 and 2004 BL86, it was seen to be beneficial to scale these velocities in the flyby case to more closely match the energy of the periodic orbits.

12.4 Results

A total of 255 transfers were generated, varying by initial orbit, target body, flyby vs. rendezvous, and velocity scaling of initial guesses in a few cases (12.3.5). In addition to comparing which orbits had the most efficient transfer to each asteroid, characteristics of the lowest fuel cost transfers found for each pair of initial orbit and target body are provided below. These include the ΔV in km/s, the transfer time given in months, the relative velocity at encounter in km/s, the angle between the satellite velocity at encounter and the direction of the light of the Sun (and hence illumination) given in degrees, and the number of impulses required.

12.4.1 Flybys

Table 12.1 provides a comparison of the ΔV for the lowest cost transfer for each pairing of orbit and asteroid (ranking them in parentheses). Note that for each asteroid, a different orbit provided the most efficient transfer. As a result, there is no clear best single orbit for this set of asteroids in terms of fuel cost.

Table 12.1: Comparison of Lowest ΔV Flyby Transfer Costs for Initial Orbit/Asteroid Pairs

	2006 RH120	2011 UD21	2005 YU55	2004 BL86
L1 Halo (N), J=3.00077179	**0.0798 (1)**	0.1464 (3)	0.5740 (4)	0.3062 (2)
L1 Vertical, J=3.00077179	0.5035 (4)	0.1690 (4)	**0.2645 (1)**	0.3807 (4)
L1 Halo (N), J=3.00048764	0.1552 (3)	**0.1247 (1)**	0.3781 (2)	0.3133 (3)
L1 Vertical, J=3.00048764	0.0931 (2)	0.1275 (2)	0.4959 (3)	**0.2944 (1)**

Tables 12.2-12.5 provide further details about the lowest ΔV transfer per initial orbit, with the fields described in the first paragraph of 12.4. Note that the light angles are in general quite high for the higher energy asteroids, which have the unfortunate property of having a high velocity moving away from the Sun. No constraints have been placed on this angle in these simulations, and in some cases such a constraint would result in very high fuel use. It should be noted however, that any constraints to limit this angle to improve illumination would be convex for angles less than or equal to 90 degrees.

Table 12.2: 2006 RH120 – Characteristics of Lowest ΔV Flyby Transfer per Initial Orbit

	ΔV (km/s)	Transfer Time (months)	Relative Velocity at Encounter (km/s)	Light Angle (degrees)	# of Impulses
L1 Halo (N), J=3.00077179	0.0798	4.8036	1.4714	84.7074	2
L1 Vertical, J=3.00077179	0.5035	5.5921	0.2651	30.7039	2
L1 Halo (N), J=3.00048764	0.1552	2.5742	1.3884	113.1157	2
L1 Vertical, J=3.00048764	0.0931	7.0772	0.2960	45.1799	3

Table 12.3: 2011 UD21 – Characteristics of Lowest ΔV Flyby Transfer per Initial Orbit

	ΔV (km/s)	Transfer Time (months)	Relative Velocity at Encounter (km/s)	Light Angle (degrees)	# of Impulses
L1 Halo (N), J=3.00077179	0.1464	2.3972	2.1075	157.0324	2
L1 Vertical, J=3.00077179	0.1690	3.8716	1.6767	172.6983	3
L1 Halo (N), J=3.00048764	0.1247	5.0231	1.6489	171.5346	2
L1 Vertical, J=3.00048764	0.1275	5.0297	1.6166	168.5095	2

Table 12.4: 2005 YU55 – Characteristics of Lowest ΔV Flyby Transfer per Initial Orbit

	ΔV (km/s)	Transfer Time (months)	Relative Velocity at Encounter (km/s)	Light Angle (degrees)	# of Impulses
L1 Halo (N), J=3.00077179	0.5740	7.4231	12.6185	176.7292	3
L1 Vertical, J=3.00077179	0.2645	3.3126	12.5632	176.0484	2
L1 Halo (N), J=3.00048764	0.3781	1.1948	12.7517	173.2236	1
L1 Vertical, J=3.00048764	0.4959	3.6528	12.5410	174.9136	2

Table 12.5: 2004 BL86 – Characteristics of Lowest ΔV Flyby Transfer per Initial Orbit

	ΔV (km/s)	Transfer Time (months)	Relative Velocity at Encounter (km/s)	Light Angle (degrees)	# of Impulses
L1 Halo (N), J=3.00077179	0.3062	8.3650	15.8090	119.2044	4
L1 Vertical, J=3.00077179	0.3807	14.4091	15.1045	120.1608	6
L1 Halo (N), J=3.00048764	0.3133	8.5328	15.0962	119.6936	3
L1 Vertical, J=3.00048764	0.2944	14.3365	15.0969	120.2754	6

12.4.2 Rendezvous

As with Table 12.1 did in the case of flybys, Table 12.6 provides a comparison of the ΔV for the lowest cost rendezvous for each pairing of orbit and asteroid (ranking them in parentheses). As a rendezvous must match in both position and velocity and since the targets were of moderate to very high inclination, it is not surprising that the Vertical Lyapunov orbits did well for these two cases.

Table 12.6: Comparison of Lowest ΔV Rendezvous Transfer Costs for Initial Orbit/Asteroid Pairs

	2006 RH120	2011 UD21
L1 Halo (N), J=3.00077179	0.8058 (4)	1.2281 (3)
L1 Vertical, J=3.00077179	0.7452 (3)	**1.0184 (1)**
L1 Halo (N), J=3.00048764	0.7072 (2)	1.6277 (4)
L1 Vertical, J=3.00048764	**0.3488 (1)**	1.0485 (2)

Tables 12.7-12.8 provide further details about the lowest ΔV transfer per initial orbit. Note that the light angles that determine illumination are lower than the flyby case. As the transfer in Figure 12.5 shows an example of, with the low relative velocity at encounter, the spacecraft will be roughly between the asteroid and the Sun leading up to encounter.

Table 12.7: 2006 RH120 – Characteristics of Lowest ΔV Rendezvous Transfer per Initial Orbit

	ΔV (km/s)	Transfer Time (months)	Relative Velocity at Encounter (km/s)	Light Angle (degrees)	# of Impulses
L1 Halo (N), J=3.00077179	0.8058	4.2227	0.0197	74.2898	3
L1 Vertical, J=3.00077179	0.7452	5.0287	0.2052	30.4316	3
L1 Halo (N), J=3.00048764	0.7072	12.7771	0.3882	28.7541	7
L1 Vertical, J=3.00048764	0.3488	6.8615	0.1973	27.9372	4

Table 12.8: 2011 UD21 – Characteristics of Lowest ΔV Rendezvous Transfer per Initial Orbit

	ΔV (km/s)	Transfer Time (months)	Relative Velocity at Encounter (km/s)	Light Angle (degrees)	# of Impulses
L1 Halo (N), J=3.00077179	1.2281	6.8595	0.4539	135.8722	4
L1 Vertical, J=3.00077179	1.0184	7.1193	0.6981	132.7321	4
L1 Halo (N), J=3.00048764	1.6277	3.1386	0.7707	142.3417	3
L1 Vertical, J=3.00048764	1.0485	7.3451	0.4533	164.7548	4

12.4.3 Sample Transfer Visualizations

All examples are shown with the computed transfer in green, the asteroid path pre-encounter shown in dashed red, and the asteroid path post encounter in solid red. These are calculated and plotted in BR4BP dynamics. The periodic orbits are plotted using CR3BP dynamics for clarity and are shown in blue for reference, but of course would be perturbed and require active station keeping in the BR4BP. All images below are in the Sun/Earth rotating frame.

The first two examples are flybys to 2006 RH120. As stated above, a rendezvous state is

provided as an initial guess, but with the final velocity difference removed from the cost function to set up a flyby. In these examples note how the transfer from a Halo orbit no longer resembles a rendezvous trajectory at all, with the final velocity coming in from a completely different and more flattened path typical of divergence from and unstable manifolds structure of a Halo orbit. On the other hand, due to the more vertically oriented motion off of a a Vertical Lyapunov orbit, the velocities at encounter are quite similar.

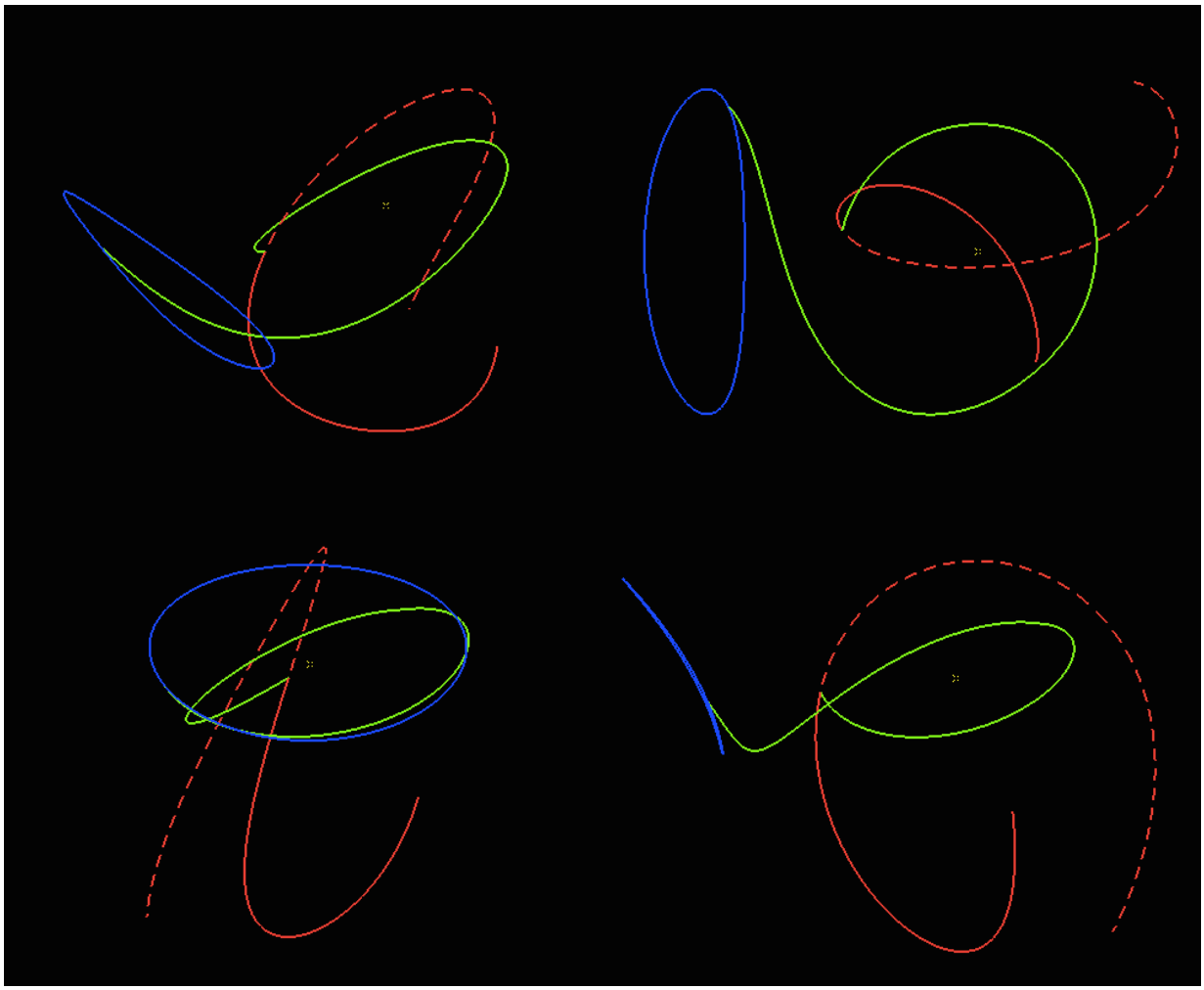


Figure 12.3: Lowest ΔV Flyby of 2006 RH120 from L1 Halo (N), J=3.00077179. Views clockwise from top left are 1) Matlab default, and projections in the 2) *xy*, 3) *yz*, 4) *xz* planes. Transfer (green), asteroid before encounter (dashed red) and after (solid red), Halo(blue).

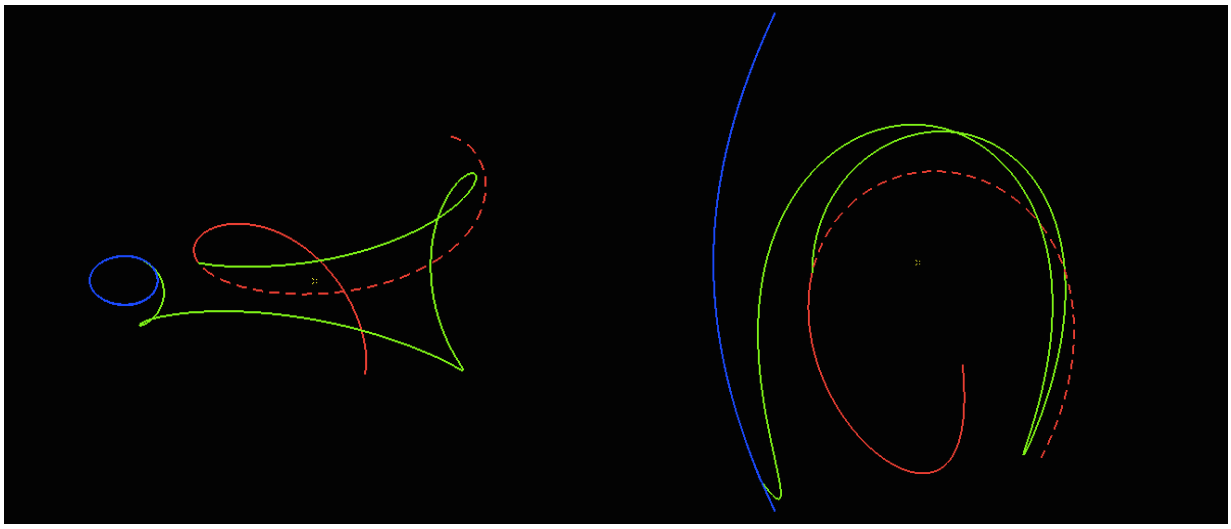


Figure 12.4: Lowest ΔV Flyby of 2006 RH120 from L1 Vertical, J=3.00048764. Transfer (green), asteroid before encounter (dashed red) and after (solid red), and Halo(blue) projected on to the *xy* (left) and *xz* (right) plane.

The rendezvous trajectory used to provide the initial guess for the case shown in Figure 12.4 is provided below for comparison in Figure 12.5. It is also the most efficient rendezvous with 2006 RH120. The asteroid and spacecraft velocities at encounter are noticeably similar than in the flyby case, especially in the projection on to the *xy* plane.

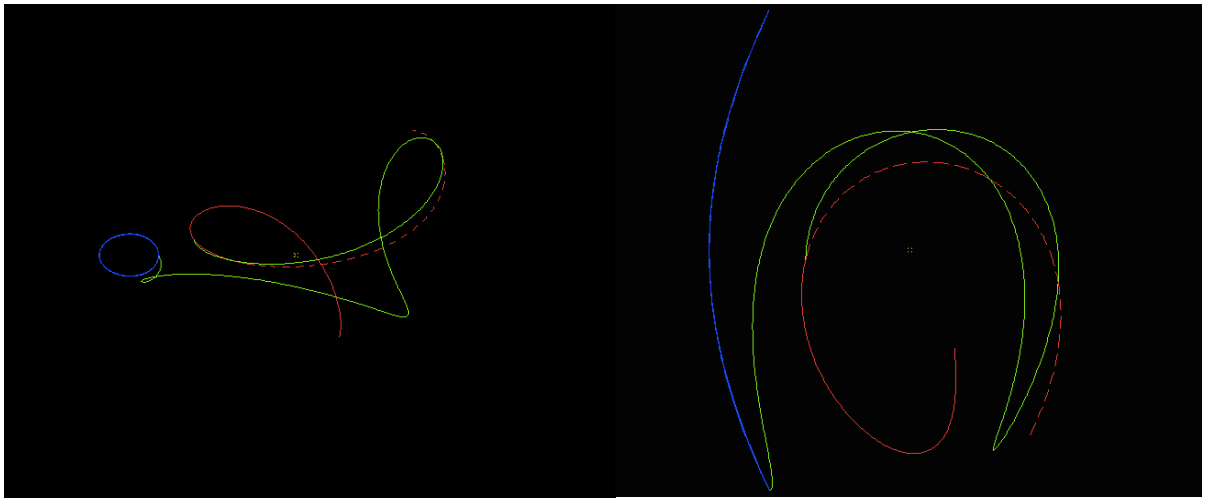


Figure 12.5: Rendezvous used as initial guess for Figure 12.4, and the most efficient rendezvous. Transfer (green), asteroid before encounter (dashed red) and after (solid red), and Halo(blue) projected on to the *xy* (left) and *xz* (right) plane.

Next a flyby of 2011 UD21 is shown in Figure 12.6. In the *xy* plane it is clear that the transfer has a prograde (counterclockwise) direction of motion typical of the manifold structure of libration point orbit. This is in contrast to the asteroid, moving in a clockwise fashion, and is due to the fact that the relative velocity at encounter has been dropped from the cost function.

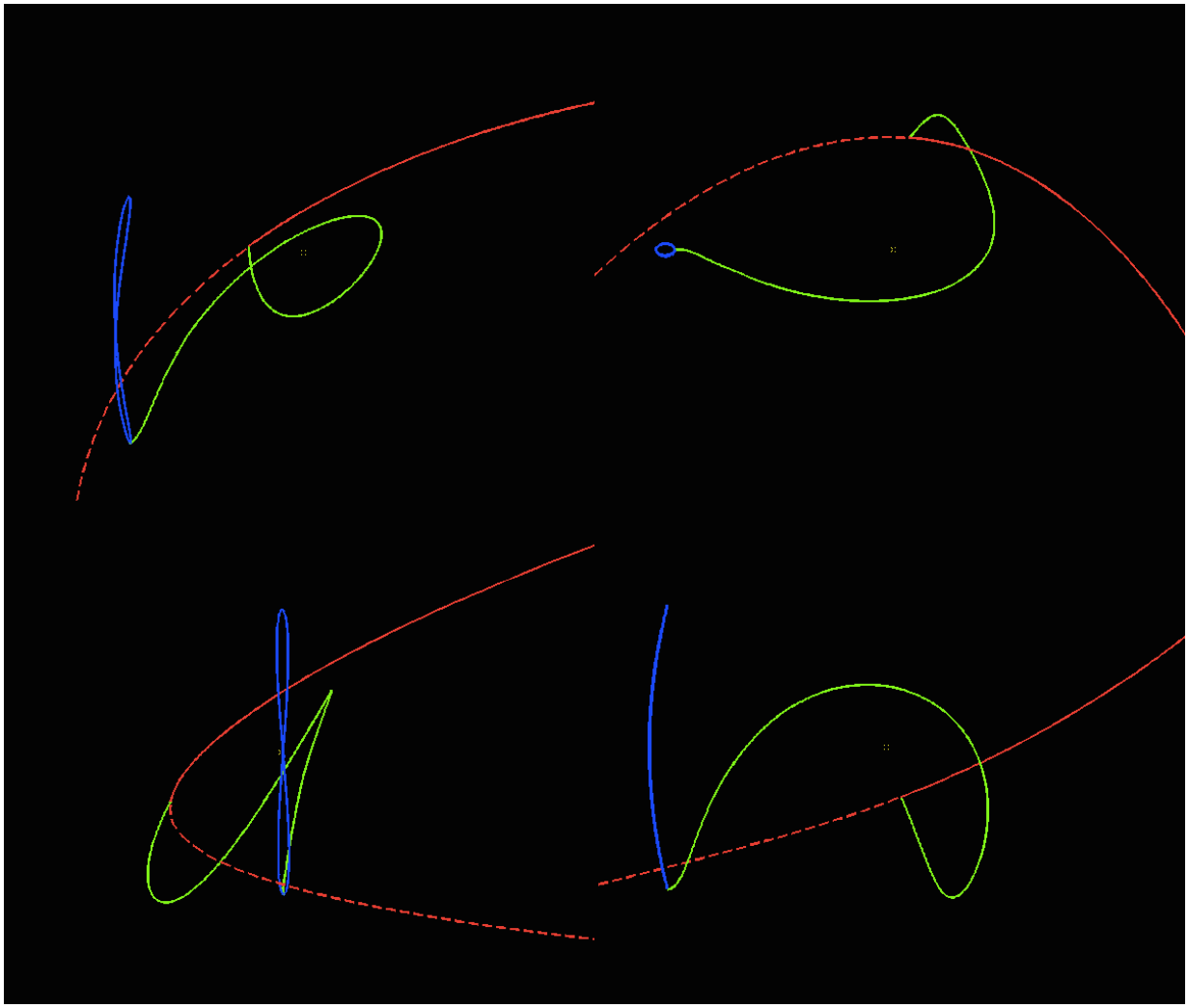


Figure 12.6: Lowest ΔV Flyby of 2011 UD21 from L1 Vertical, J=3.00077179. Views clockwise from top left are 1) Matlab default, and projections in the 2) *xy*, 3) *yz*, 4) *xz* planes. Transfer (green), asteroid before encounter (dashed red) and after (solid red), Halo(blue).

In many cases the best transfers were simple with a low number of impulses, though this was not always the case. Contrast the above flyby with the best flyby transfer to 2004 BL86 found.

It is clear from the near symmetry in the *xy* projection that some dynamical structure is being utilized to efficiently drift towards the rendezvous point over a 14 month period.

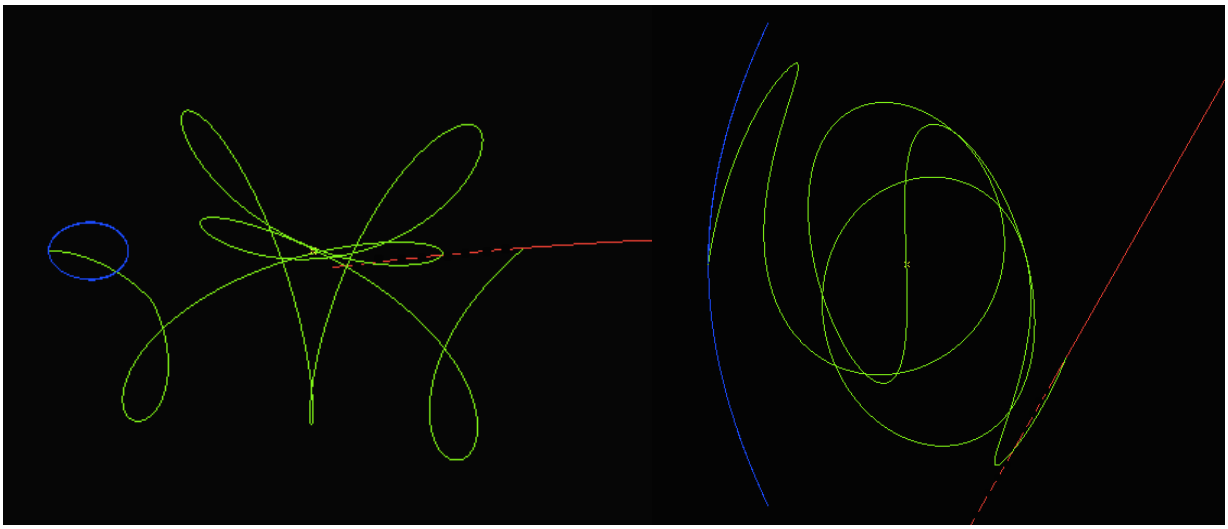


Figure 12.7: Lowest ΔV Flyby of 2004 BL86 from L1 Halo (N), J=3.00048764. Transfer (green), asteroid before encounter (dashed red) and after (solid red), and Halo(blue) projected on to the *xy* (left) and *xz* (right) plane.

12.5 Summary

The transfer redesign tool was adapted to generate hundred of both flyby and rendezvous transfers from libration point orbits to near Earth asteroids as part of an investigation at GSFC. While the mission analysis itself is out of the scope of the transfer construction method itself, sample results and transfers were provided to give a sense of the application. The fact that the method was able to be modified in both the setup and run time phases, have the lengthy partitioning, simulation, and graph construction steps completed, and have hundreds of transfers in the BR4BP generated in a few weeks culminating in a GSFC visit shows the flexibility of the method beyond onboard replanning.

Chapter 13

Conclusion

"All generalizations are false, including this one." - Mark Twain

13.1 Overview

This dissertation has described an impulsive transfer design algorithm for orbiters in multibody gravitational environments. Although useful for on the ground trajectory design, it has been developed with a focus on onboard transfer redesigns to guide the spacecraft to a specified target from off-nominal conditions.

The first section of the dissertation focused on the task of initial guess generation. The use of the apsis condition to define time steps, the coordinates chosen to partition the domain, the tidally locked nature of most moons of interest, the strength of the differential correction method, and precomputing a database of important periodic orbits and other structures combine to allow a reduction in dimension that creates a reasonably sized directed graph representation of the system. Furthermore, the resulting directed graph – although much more complex than simply linking neighboring sets due to long coasting

arcs – possesses enough structure that effective search heuristics may be used to greatly speed up a search process. The A* Search algorithm and related methods are known to converge to the best path within the system discretization, and the graph construction process guarantees that this itinerary may be used to select a set of ballistic arcs. These arcs provide an initial guess for a transfer, an approximation with some constraints feasible by design and others with limited infeasibilities that need to be eliminated in the next phase of the transfer design process.

A rapid differential correction and local optimization process was described in depth, with a focus on proving the properties needed for an onboard implementation. For the fuel minimization process, synthesis of a two level correction structure with steps defined by the solutions of convex programs was shown to possess properties that either approach lacks when taken individually: feasible major iterates with guaranteed lower cost after recorrection. This result, combined with insights on the neglected subjects of Slater's Condition holding throughout the iterative process and the assumed invertibility of certain state transition matrix submatrices, is then used to show that the process will indeed converge to a local minimum should the time be available. Otherwise, even with a system interrupt part way through the process, it will still provide a feasible transfer with a rapid reduction in cost compared to the initial corrected guess.

Applications of the transfer design algorithm were provided in two different cases: onboard redesign for an orbiter at Phobos, and initial design of flyby and rendezvous trajectories from libration point orbits to Near Earth Asteroids. Perhaps the most important demonstration of the method is that the algorithm was successfully modified and run with the stringent requirements and limitations of a RAD750 based flight-like hardware and software system.

In summary, the method developed here is not only quite fast, but is sufficiently lightweight to be run on flight systems and possesses a large number of provable convergence and feasibility properties to provide a major step in the direction of onboard transfer design in complex systems.

13.2 Future Research

Research into integration algorithms is perhaps surprisingly the most important progress that could be made to advance the transfer design method described here. This is for several reasons. As described in the computation time breakdowns in Chapters 10 and 11, those stages requiring integration take up the vast majority of the computation time. Therefore faster integrators than the standard Runge-Kutta methods employed here and by the like of NASA's AutoNav would be of great benefit. Another issue discussed in Chapter 11 is that variable step integrators do not have a deterministic number of steps without minimum step lengths, which can negatively affect the accuracy in sensitive systems.

Fast integration and updated computing platforms would also enable research into building not only the impulsive graph connections at run time, but also the ballistic connections. The primary benefit of this is not in fact the reduced memory footprint. Rather, this would allow uncertainty modeling to be incorporated directly into the initial design process, with a very nice interplay between the trajectories needed for graph construction and the sigma points of the Unscented Kalman Filter. In addition, this eliminates the issue with graph based methods for bodies about which little gravitational information is known before encounter. Rather than relying on pre-encounter parameters, the most recent values would be used. Related to this area of graph construction, as mentioned in Chapter 3 it would be beneficial for the domain partitioning itself to be done automatically rather than through trial and error in ground simulations.

Beyond such integration based extensions, there remain several clear areas for investigation. A* Search was seen to be sufficient for the applications in this dissertation, but as discussed in Chapters 4 and 11 there are variants such as SMA* that explicitly handle the strict memory limitations of a spacecraft. Additional memory gains may be made in the use of Compressed Column Storage over the basic three column storage for sparse matrices used here, however the effect on retrieval times would need to be analyzed. Within the correction and optimization processes, there is always room for the adaptation of additional constraints. Also, the current method is for a fixed number of impulses. Analysis

of the primer vector would indicate whether an additional impulse would be of benefit, though again a good-enough transfer in a short time is generally preferred over the precise optimum in onboard applications.

Lastly, although flight computing platforms will always possess their own distinct requirements and limitations, advancements will occur in the future. In particular, multicore processors will enable parallel processing. Many steps of this algorithm may be done in parallel, either naively or through other more advanced methods under development. A few pages were dedicated to this subject at the end of Chapter 11. Regardless of whether for use on a RAD750 or the processors of the future, it is sometimes said that it is more important how well a component fits into the system as a whole rather than what it can do on its own. Any serious effort to truly make an onboard transfer design program will require serious system integration research.

References

[Acik13]
Açıkmese, Behçet, MiMi Aung, Jordi Casoliva, et al. "Flight Testing of Trajectories Computed by G-FOLD: Fuel Optimal Large Divert Guidance Algorithm for Planetary Landing." AAS/AIAA Spaceflight Mechanics Meeting, Lihue, HI, February 2013.

[Alex12]
Alexander, James, Bradley J. Clement, Kim P. Gostelow, John Y. Lai, “Fault Mitigation Schemes for Future Spaceflight Multicore Processors.” AIAA Infotech Conference, Garden Grove, CA, June 2012.

[Augu12]
Augugliaro, Federico, Angela P. Schoellig, and Raffaello D'Andrea. "Generation of Collision-free Trajectories for a Quadrocopter Fleet: A Sequential Convex Programming Approach." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2012. doi: 10.1109/IROS.2012.6385823.

[Bank83]
Bank, B., et al. *Non-Linear parametric optimization*. Basel: Birkhäuser, 1983. 59-76.

[Berg07]
Berger, R.W. et. al., “RAD750 SpaceWire Enabled Flight Computer for Lunar Reconnaissance Orbiter,” 2007 International SpaceWire Conference, Dundee, Scotland, September 2007.

[Blei08]
Bleiweiss, Avi. "GPU Accelerated Pathfinding." 23rd ACM SIGGRAPH / EUROGRAPHICS Symposium on Graphics Hardware, Aire-la-Ville, Switzerland, 2008.

[Bosa14]
Bosanac, Natasha, et al. "Manned sample return mission to phobos: A technology demonstration for human exploration of Mars." IEEE Aerospace Conference, Big Sky, MT, March 2014. doi: 10.1109/AERO.2014.6836251

[Boyd94]
Boyd, Stephen, et al. *Linear Matrix Inequalities in System and Control Theory*. Philadelphia: Society for Industrial and Applied Mathematics, 1994. 7-8.

[Boyd08]
Boyd, Stephen. "Sequential Convex Programming," Notes, Stanford University, 2008. [Online]. http://www.stanford.edu/class/ee364b/lectures/seqslides.pdf

[Boyd09]
Boyd, Stephen, and Lieven Vandenberghe. *Convex Optimization*. New York: Cambridge University Press, 2009. 21-145, 241-252, 609-620.

[Cang12]
Cangahuala, Al, Shyam Bhaskaran, and Bill Owen. "Science Benefits of Onboard Spacecraft Navigation," *EOS, Transactions American Geophysical Union*, Vol. 93, No. 177, 2012. doi: 10.1029/2012EO180001

[Caso13]
Casoliva, Jordi. *Spacecraft Trajectory Generation by Successive Approximation for Powered Descent and Cyclers*. Ph.D. Dissertation, University of California, Irvine. 2013. 45-76.

[Chu13]
Chu, Eric, et al. "Code Generation for Embedded Second-order Cone Programming." IEEE European Control Conference (ECC), Zurich, 2013.

[Conl68]
Conley, C. C. "Low energy transit orbits in the restricted three-body problems." *SIAM Journal on Applied Mathematics* 16.4 (1968): 732-746.

[Conw07]
Conway, Bruce A., Christian M. Chilan, and Bradley J. Wall. "Evolutionary Principles Applied to Mission Planning Problems." *Celestial Mechanics and Dynamical Astronomy* 97.2 (2007): 73-86.

[Cras04]
Crassidis, John L., John L. Junkins. *Optimal Estimation of Dynamic Systems*. Chapman & Hall / CRC Press, 2004. 285-291.

[Curt13]
Curtis, Howard. *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, 2013.

[Davi12]
Davis, Diane Craig, and Kathleen C. Howell. "Characterization of Trajectories Near the Smaller Primary in the Restricted Problem for Applications." *Journal of Guidance, Control, and Dynamics* 35.1 (2012): 116-128.

[Dell05]
Dellnitz, Michael, et al. "Transport in Dynamical Astronomy and Multibody Problems." *International Journal of Bifurcation and Chaos* 15.03 (2005): 699-727.

[Dell06]
Dellnitz M., and Junge O., “Set Oriented Numerical Methods in Space Mission Design,” *Modern Astrodynamics* (P. Gurfil, ed.), Oxford: Academic Press, 2006.

[Dodg07]
Dodge, Randy, Mark A. Boyles, and Chuck E. Rasbach. "Key and Driving Requirements for the Juno Payload Suite of Instruments." AIAA 2007 SPACE Conference, Long Beach, CA, September 2007.

[Doed97]
Doedel E.J., et al. “Auto97: Continuation and Bifurcation Software for Ordinary Differential Equations”, Technical Report, Concordia University, Montreal, Canada, 1997.

[Doma13]
Domahidi, Alexander, Eric Chu, and Stephen Boyd. "ECOS: An SOCP Solver for Embedded Systems." IEEE European Control Conference (ECC), Zurich, 2013.

[Doyl12]
Doyle, Richard, et al “High Performance Spaceflight Com- puting, An Avionics Formulation Task.” Study Report Executive Summary, NASA Game Changing Development Program, October 2012.

[Doyl14]
Doyle, Richard, et al. “High Performance Spaceflight Computing; Next-Generation Space Processor: A Joint Investment of NASA and AFRL,” International Symposium on Artificial Intelligence, Robotics, and Automation in Space (iSAIRAS 2014), Montreal, Canada, June 2014.

[Duff89]
Duff, Iain S., Roger G. Grimes, and John G. Lewis. "Sparse Matrix Test Problems." *ACM Transactions on Mathematical Software* (TOMS) 15.1 (1989): 1-14.

[Farq66]
Farquhar, R. W. “Station-Keeping in the Vicinity of Collinear Libration Points with an Application to a Lunar Communications Problem.” AAS Spaceflight Mechanics Specialist Conference, Denver, CO, July 1966.

[Farq01]
Farquhar, Robert W. "The Flight of ISEE-3/ICE: Origins, Mission History, and a Legacy." *Journal of the Astronautical Sciences* 49.1 (2001): 23-74.

[Fred97]
Fredman, Michael L., and Robert Endre Tarjan. "Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms." Journal of the ACM (JACM) 34.3 (1987): 596-615.

[Froe97]
Froeschlé, Claude, Elena Lega, and Robert Gonczi. "Fast Lyapunov Indicators. Application to Asteroidal Motion." *Celestial Mechanics and Dynamical Astronomy* 67.1 (1997): 41-62.

[Gask11]
Gaskell, R.W., “Gaskell Phobos Shape Model V1.0. VO1-SA-VISA/VISB-5-PHOBOSSHAPE-V1.0.”

NASA Planetary Data System, 2011.

[Gill02]
Gill, Philip E., Walter Murray, and Michael A. Saunders. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization." *SIAM Journal on Optimization* 12.4 (2002): 979-1006.

[Gome05]
Gómez, Gerard, Manuel Marcote, and Josep J. Masdemont. "Trajectory Correction Manoeuvres in the Transfer to Libration Point Orbits." *Acta Astronautica* 56.7 (2005): 652-669.

[Gost13]
Gostelow, Kim. "Mars Science Laboratory Entry, Descent, and Landing Flight Software." AAS/AIAA Space Flight Mechanics Meeting, Lihue, HI, February 2013.

[Gran08]
Grant, Michael C., and Stephen P. Boyd. "Graph implementations for nonsmooth convex programs." *Recent Advances in Learning and Control*. Springer London, 2008. 95-110.

[Gran13]
Grant, Michael C. and Stephen P. Boyd. "CVX: Matlab Software for Disciplined Convex Programming, Version 2.0 Beta." http://cvxr.com/cvx, September 2013.

[Hira 06]
Hirani, Anil N., and Ryan P. Russell. "Approximations of distant retrograde orbits for mission design." AAS/AIAA Spaceflight Mechanics Meeting, Tampa, FL, January 2006.

[Hoff11]
Hoffman, Stephen J. "A Phobos-Deimos Mission as an Element of the NASA Mars Design Reference Architecture 5.0." (2011). 2nd International Conference on the Exploration of Phobos and Deimos, Moffett Field, CA, March 2011.

[Howe84]
Howell, Kathleen Connor. "Three-Dimensional, Periodic,'Halo' Orbits." *Celestial Mechanics* 32.1 (1984): 53-71.

[Hugh03]
Hughes, Steven P., Laurie M. Mailhe, and Jose J. Guzman. "A Comparison of Trajectory Optimization Methods for the Impulsive Minimum Fuel Rendezvous Problem." 26th AAS Rocky Mountain Guidance and Control Conference, Breckinridge, CO, February 2003.

[Kish09]
Kishimoto, Akihiro, Alex S. Fukunaga, and Adi Botea. "Scalable, Parallel Best-First Search for Optimal Sequential Planning." *Proceedings of the International Conference on Automated Scheduling and Planning: ICAPS-09*, 2009. 201–208.

[Kjel12]
Kjellberg, Henri, and E. Glenn Lightsey. "A Constrained Attitude Control Module for Small Satellites." 26th Annual AIAA Conference on Small Satellites, Logan, UT, 2012.

[Koon08]
Koon W. S., Lo M. W., Marsden J. E., and Ross S.D. *Dynamical Systems, the Three-Body Problem and Space Mission Design*, Marsden Books, 2008. 95-188.

[Krug14]
Kruger, Alexander Y., Leonid Minchenko, and Jiří V. Outrata. "On Relaxing the Mangasarian–Fromovitz Constraint Qualification." *Positivity* 18.1 (2014): 171-189.
[LaVa06]
LaValle, Steven M. *Planning algorithms*. Cambridge university press, 2006. 265-273.

[Lawr96]
Lawrence, Craig T., and André L. Tits. "Nonlinear Equality Constraints in Feasible Sequential Quadratic Programming." *Optimization Methods and Software* 6.4 (1996): 265-282.

[Lawr01]
Lawrence, Craig T., and André L. Tits. "A computationally efficient feasible sequential quadratic programming algorithm." *SIAM Journal on Optimization* 11.4 (2001): 1092-1118.

[Lee12]
Lee, John. *Introduction to Smooth Manifolds*. GTM Vol. 218. Springer, 2003. 155-170.

[Lo01]
Lo, Martin W., et al. "Genesis Mission Design." *Journal of the Astronautical Sciences* 49.1 (2001): 169-184.

[Lo04]
Lo, M.W, and Parker, J.S., "Unstable Resonant Orbits Near Earth and their Applications in Planetary Missions." AIAA/AAS Astrodynamics Specialist Conference, Providence, RI, August, 2004.

[Luen08]
Luenberger, David G., and Yinyu Ye. *Linear and Nonlinear Programming*. Vol. 116. Springer Science & Business Media, 2008.

[Mang14]
Private communication at NASA/CalTech Jet Propulsion Laboratory. August 2014.

[Marc07]
Marchand, Belinda G., Kathleen C. Howell, and Roby S. Wilson. "Improved Corrections Process for Constrained Trajectory Design in the N-Body Problem." *Journal of Spacecraft and Rockets* 44.4 (2007): 884-897.

[Maro04]
Marov, M. Ya, et al. "Phobos-Grunt: Russian Sample Return Mission." *Advances in Space Research* 33.12 (2004): 2276-2280.

[Matt12]
Mattingley, Jacob, and Stephen Boyd. "CVXGEN: a Code Generator for Embedded Convex

Optimization." *Optimization and Engineering* 13.1 (2012): 1-27.

[McFa06]
McFadden, Lucy-Ann, Torrence Johnson, and Paul Weissman, eds. *Encyclopedia of the Solar System*. Academic press, 2006. 365-448.

[McGe69]
McGehee, R. *Some Homoclinic Orbits for the Restricted Three-body Problem.* PhD Thesis, University of Wisconsin, Madison.

[Miel70]
Miele, A., R. Eo Pritchard, and J. N. Damoulakis. "Sequential Gradient-Restoration Algorithm for Optimal Control Problems." Journal of Optimization Theory and Applications 5.4 (1970): 235-282.

[Mond10]
Mondelo, Josep-Maria, Stephen B. Broschart, and Benjamin F. Villac. "Dynamical Analysis of 1:1 Resonances near Asteroids: Application to Vesta." 2010 AAS/AIAA Astrodynamics Specialists Conference, August 2-5, Toronto, Ontario, Canada, 2010.

[Mont11]
Montenbruck, Oliver, and Eberhard Gill. *Satellite Orbits: Models, Methods, and Applications*. Springer, 2011. 53-116.

[Morg13]
Morgan, Daniel, Soon-Jo Chung, and Fred Y. Hadaegh. "Decentralized model predictive control of swarms of spacecraft using sequential convex programming." AAS/AIAA Space Flight Mechanics Meeting, Lihue, HI, 2013.

[Mour09]
Mourikis, Anastasios I., et al. "Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing." *Robotics, IEEE Transactions on* 25.2 (2009): 264-280.

[Nakh13]
Nakhjiri, Navid. *Efficient Methods for Phase Space Analysis in Spaceflight Mechanics: Application to the Optimization of Stable Transfers*. Ph.D. Dissertation, University of California, Irvine. 2013. 89-91, 125-127.

[Nakh14]
Nakhjiri, Navid, and Benjamin Villac. "Modified Picard Integrator for Spaceflight Mechanics." *Journal of Guidance, Control, and Dynamics* 37.5 (2014): 1625-1637.

[Nemi01]
Nemirovski, Arkadi, and Ahron Ben-Tal. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Vol. 2. Society for Industrial and Applied Mathematics, 2001. 166-169.

[Nest94]
Nesterov, Yurii, Arkadii Nemirovskii, and Yinyu Ye. *Interior-Point Polynomial Algorithms in Convex*

Programming. Vol. 13. Philadelphia: Society for Industrial and Applied Mathematics, 1994.

[NRC12]
Steering Committee for NASA Technology Roadmaps, National Research Council. *NASA Space Technology Roadmaps and Priorities*, National Academies Press, 2012. Sections TA05-5.4.3, TA04-4.2.4.

[Osip06]
Osipenko, George. *Dynamical Systems, Graphs, and Algorithms*, Springer, 2006. 15-30.

[Peng09]
Peng, Jiming, Cornelis Roos, and Tamás Terlaky. *Self-Regularity: A New Paradigm for Primal-Dual Interior-Point Algorithms*. Princeton University Press, 2009. 125-152.

[Rich80]
Richardson, David L. "Halo Orbit formulation for the ISEE-3 mission." Journal of Guidance, Control, and Dynamics 3.6 (1980): 543-548.

[Ried09]
Riedel, J. Edmund, Mimi Aung, Christopher A. Grasso, and William M. Owen Jr. "A Survey of Technologies Necessary for the Next Decade of Small Body and Planetary Exploration." Jet Propulsion Laboratory, Pasadena, CA, 2009.

[RiedAN]
Riedel, J. Edmund, et al. “Autonomous Optical Navigation (AutoNav) DS1 Technology Validation Report.” Jet Propulsion Laboratory, Pasadena, CA.

[Russ09]
Russell, Stuart, Peter Norvig. *Artificial Intelligence: A Modern Approach*, Prentice Hall, 3rd Ed., 2009. 81-109.

[Sche01]
Scheeres, Daniel. J., M. D. Guman, and Benjamin. F. Villac. "Stability Analysis of Planetary Satellite Orbiters: Application to the Europa Orbiter." *Journal of Guidance, Control, and Dynamics* 24.4 (2001): 778-787.

[Sche06]
Guibout, V. M., and D. J. Scheeres. "Solving Two Point Boundary Value Problems Using Generating Functions: Theory and Applications to Astrodynamics" *Modern Astrodynamics* (Gurfil, Editor). Butterworth-Heinemann, 2006.

[Some13]
Some, Raphael, et al. “Human and Robotic Mission Use Cases for High-Performance Spaceflight Computing.” AIAA Infotech, Boston, MA, August 2013.

[Sukh04]
Sukhanov, Alexander, and Antonio F. Bertachini A. Prado. "Lambert problem solution in the hill model of motion." Celestial Mechanics and Dynamical Astronomy 90.3-4 (2004): 331-354.

[Suro15]
Surovik, David A., and Daniel J. Scheeres. "Adaptive Reachability Analysis to Achieve Mission Objectives in Strongly Non-Keplerian Systems." *Journal of Guidance, Control, and Dynamics* (2015): 1-10.

[Thom12]
Thomas, Valerie C., et al. "The Dawn Spacecraft." *The Dawn Mission to Minor Planets 4 Vesta and 1 Ceres*. Springer New York, 2012. 175-249.

[Tsir13]
Tsirogiannis, G. A., and V.V. Markellos. "A Greedy Global Search Algorithm for Connecting Unstable Periodic Orbits with Low Energy Cost." *Celestial Mechanics and Dynamical Astronomy* (2013): 1-13.

[Trum11]
Trumbauer, Eric, and Benjamin F. Villac. "An Analysis of Multiple Revolution Third Body Driven Plane Change Maneuvers." AAS/AIAA Astrodynamics Specialist Conference, Girdwood, AK, August 2011.

[Trum12a]
Trumbauer, Eric, and Benjamin F. Villac. "Expanding Transfer Representations in Symbolic Dynamics for Automated Trajectory Design." AAS/AIAA Spaceflight Mechanics Meeting, Charleston,SC, January 2012.

[Trum12b]
Trumbauer, Eric, and Benjamin F. Villac. "Search and Representation Strategies for Automated Trajectory Design." AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, MN, August 2012.

[Trum13]
Trumbauer, Eric, and Benjamin F. Villac. "Sequential Convex Programming for Impulsive Transfer Optimization in Multibody Systems." AAS/AIAA Spaceflight Mechanics Meeting, Kauai, HI, February 2013.

[Trum14a]
Trumbauer, Eric, and Benjamin F. Villac. "Heuristic Search-Based Framework for Onboard Trajectory Redesign." *Journal of Guidance, Control, and Dynamics* 37.1 (2013): 164-175.

[Trum14b]
Trumbauer, Eric, and Benjamin F. Villac. "Autonomous Trajectory Redesign for Phobos Orbital Operations." AAS/AIAA Spaceflight Mechanics Conference, Santa Fe, NM, January 2014.

[Vall01]
Vallado, David A. *Fundamentals of Astrodynamics and Applications*. Vol. 12. Springer, 2001. 546-549.

[Vill03]

Villac, Benjamin F. *Dynamics in the Hill Problem with Applications to Spacecraft Maneuvers*. PhD Thesis, University of Michigan, 2003. 36-46, 71-80.

[Vill04]
Villac, Benjamin F., and Daniel. J. Scheeres. “A Simple Algorithm to Compute Hyperbolic Invariant Manifolds Near L1 and L2.” AAS/AIAA Spaceflight Mechanics Meeting, Maui, Hawaii, February 8-12, 2004.

[Vill08]
Villac, Benjamin F. "Using FLI Maps for Preliminary Spacecraft Trajectory Design in Multi-Body Environments." *Celestial Mechanics and Dynamical Astronomy* 102.1-3 (2008): 29-48.

[Vill09]
Villac, Benjamin F., and Daniel J. Scheeres. "One Impulse vs. Third Body Driven Plane Changes", *Journal of the Astronautical Sciences* 57.3 (2009): 545-559.

[ViRe11]
Villalpando, Carlos, et al. "Reliable Multicore Processors for NASA Space Missions." IEEE Aerospace Conference, Big Sky, MT, 2011.

[Wall12]
Wallace, Mark, Jeffrey Parker, Nathan Strange, and Daniel Grebow. “Orbital Operations for Phobos and Deimos Exploration.” AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, MN, August 13, 2012.

[Wils98]
Wilson, Roby. Trajectory Design in the Sun-Earth-Moon Four Body Problem. PhD Thesis, Purdue University, 1998.

[Wils13]
Wilson, Roby. Private communication at NASA/CalTech Jet Propulsion Laboratory. July 2013.

[Zang69]
Zangwill, Willard I. "Convergence Conditions for Nonlinear Programming Algorithms." *Management Science* 16.1 (1969): 1-13.

[Zill04]
Zillober, Christian, K. Schittkowski, and K. Moritzen. "Very large scale optimization by sequential convex programming." *Optimization Methods and Software* 19.1 (2004): 103-120.

[Zlob09]
Zlobec, Sanjo. "Nondifferentiable optimization: parametric programming Nondifferentiable Optimization: Parametric Programming." *Encyclopedia of Optimization*. Springer US, 2009. 2607-2615.

Appendix A

Constraint Qualifications for Subproblems

A.1 Introduction

It was shown in Chapter 5 that if the Karush-Kuhn-Tucker (KKT) conditions hold for the SOCP/QCQP subproblems at a stationary point, then they will hold for the original NLP. It is therefore important that should one of the subproblems have $\tilde{X}^{*}=0$ as a solution, that the KKT conditions are guaranteed to hold. This will be the case if $\tilde{X}=0$ is a regular point, meaning it satisfies a constraint qualification. These are sufficient conditions that a local optimum at the point in question will satisfy the KKT conditions.

Although regularity is often not addressed in application focused papers, it should not be taken for granted. A constraint qualification must be found such that if it holds for the initial guess, it will be guaranteed to hold true at $\tilde{X}=0$ for any iteration. The weaker the constraint qualification the better, so that a wider variety of initial guesses can provide the needed

characteristics. Aside from this benefit, it seems that while the familiar Linear Independence Constraint Qualification [Luen08] may hold throughout, it does not seem possible to guarantee that.

Lastly, the Interior Point algorithms most suited to onboard application assume that a strictly feasible point exist, i.e. that Slater's Condition holds.[Boyd09] Therefore we seek also to show that Slater's Condition holds for each CP subproblem. We shall see that in fact this is not a separate problem, but for these particular problems is equivalent to a fairly weak constraint qualification.

A.2 Types of Constraint Qualifications

A.2.1 Definitions

Let J be set of all indices for the CP equality constraints $\tilde{h}_j$, and $K_a(\tilde{X}) \subset K$ set of indices of the active inequality constraints $\tilde{g}_k$ at some feasible point $\bar{X} \in \mathbb{R}^{7N}$. Consider the following constraint qualifications: [Luen08, Krug14]

Linear Independence Constraint Qualification (LICQ) holds at $\bar{X}$ if the set of gradients

$$\{\nabla \tilde{h}_j(\bar{X}), \tilde{g}_k(\bar{X}), j \in J, k \in K_a(\bar{X})\}$$

is linearly independent.

Mangasarian-Fromovitz Constraint Qualification (MFCQ) holds at $\bar{X}$ if:

i) the set of gradients $\{\nabla \tilde{h}_j(\bar{X}), j \in J\}$ is linearly independent.

ii) there exists $\tilde{Z} \in \mathbb{R}^{7N}$ such that $\langle \nabla \tilde{h}_j(\bar{X}), \tilde{Z} \rangle = 0 \; \forall \, j \in J, \langle \nabla \tilde{g}_j(\bar{X}), \tilde{Z} \rangle < 0 \; \forall \, k \in K_a(\bar{X})$.

Constant Rank Mangasarian-Fromovitz Constraint Qualification (CRMFCQ) holds at $\bar{X}$ if:

i) the set of gradients $\{\nabla \tilde{h}_j(\tilde{X}), j \in J\}$ has constant rank in some neighborhood of $\bar{X}$.

ii) there exists $\tilde{Z} \in \mathbb{R}^{7N}$ such that $\langle \nabla \tilde{h}_j(\bar{X}), \tilde{Z} \rangle = 0 \; \forall \, j \in J, \langle \nabla \tilde{g}_k(\bar{X}), \tilde{Z} \rangle < 0 \; \forall \, k \in K_a(\bar{X})$.

Slater's Condition holds for the SOCP/QCQP if there exists a point $\tilde{X}_S$ s.t.

$$\nabla \tilde{h}_j(\tilde{X}_S)=0\ j\in J\,,\ \tilde{g}_k(\tilde{X}_S)<0\ \forall\, k\in K$$

A.2.2 Implications and Equivalences

Property 1:

LICQ implies MFCQ, which implies CRMFCQ. [Krug14]

Property 2:

LICQ, MFCQ, CRMFCQ all imply the Abadie Constraint Qualification, which is sufficient to show that a solution satisfies the KKT conditions. [Krug14]

Property 3:

For a convex programming problem, Slater's Condition is sufficient to guarantee that a solution satisfies the KKT conditions and that strong duality holds. [Boyd09]

Proposition A.2.2.1: For a CP with inequality constraints that are differentiable within the feasible set and equality constraints that are linear rather than affine (as would be the case whenever the reference is feasible) Slater's Condition is equivalent to CRMFCQ holding for any feasible point.

Proof: First, note that for a CP the equality constraints are linear, thus the gradients are constant across the whole domain. As a result, the set of gradients of equality constraints has constant rank trivially.

Assume Slater's Condition holds. For any particular feasible point $\bar{X}$, consider the vector $\tilde{Z}=\tilde{X}_S-\bar{X}$. For $\langle \nabla \tilde{h}_j(\bar{X}),\tilde{Z}\rangle = a_j^T\tilde{Z}=a_j^T\tilde{X}_S-a_j^T\bar{X}=0-0=0$ where the constraints are of the form $a_j^T\tilde{X}=0$ and since both $\bar{X},\tilde{X}_S$ are feasible. Since $\tilde{g}_k(\tilde{X}_f)<\tilde{g}_k(\bar{X})\,\forall\, k\in K_a(0)$ and by the FONC of convex functions, $\tilde{Z}=\tilde{X}_f-\bar{X}$ defines a descent direction for all of the active inequality

constraints. Thus $\langle \nabla \tilde{g}_k(\bar{X}), \tilde{Z} \rangle < 0 \forall k \in K_a(\bar{X})$. Therefore CRMFCQ holds for $\bar{X}$, and so for any feasible point.

Assume CRMFCQ holds for some feasible $\bar{X}$. Then $\tilde{Z}$ is a descent direction for every active constraint. Therefore for each $k \in K_a(\bar{X})$ there exists some constant $\bar{\beta}_k$ s.t. $\tilde{g}_k(\bar{X} + \beta \tilde{Z}) < 0 \forall \beta < \bar{\beta}_k$. Additionally, for every inactive constraint there is some neighborhood defined by a ball of radius $\bar{\eta}_k$ where it remains inactive. Therefore for $\epsilon < min\{\bar{\beta}_k, \bar{\eta}_k \forall k \in K\}$, $\tilde{g}_k(\bar{X} + \epsilon \tilde{Z}) < 0 \forall k \in K$. Also, $\tilde{h}_j(\bar{X} + \epsilon \tilde{Z}) = \tilde{h}_j(\bar{X}) + \epsilon \langle \tilde{h}_j(\bar{X}), \tilde{Z} \rangle = 0 + 0 = 0$ from the feasibility of $\bar{X}$ and the CRMFCQ condition on $\tilde{Z}$. Therefore $\bar{X} + \epsilon \tilde{Z}$ satisfies Slater's Condition.

Proposition A.2.2.2: The above equivalence holds for both the SOCP and QCQP formulations in Chapter 5.

Proof: The QCQP formulation satisfies the conditions above since the constraints are differentiable everywhere.

Both the norm ball or quadratic form of the inequality constraints used in the SOCP or QCQP respectively define the same sets. Therefore Slater's Condition holds for one formulation if and only if it holds in the other. Also, the gradient *directions* are identical in the two formulations for all of the shared problem constraints (all SOCP constraints are differentiable within a neighborhood of an active point). A special note is needed for the constraints generated in the epigraph transformation for the SOCP, but if these are active than choosing any higher σ_i value leads to a strictly feasible point or defines a descent direction for the constraint without affecting any of the other variables. Thus CRMFCQ holds for one formulation if and only if it holds for the other. Thus we have:

CRMFCQ in SOCP iff CRMFCQ in QCQP iff Slater's in QCQP iff Slater's in SOCP

Corollary A.2.2.3: If $\tilde{X}=0$ satisfies the CRMFCQ of a SOCP/QCQP subproblem, the minimum $\tilde{X}^*$ does as well.

A.3 Regularity at Each Iteration

Proposition: If $\tilde{X}=0$ satisfies the CRMFCQ for the CP defined at $X_{(l)}^{ref}$, $\tilde{X}=0$ satisfies CRMFCQ for the the next CP defined at $X_{(l+1)}^{ref}$, where this new reference is the result of the iterative process in Chapter 6.

Proof: First, condition i) will be satisfied since the gradients for the new CP will be constant, albeit with different values than the previous iteration.

Now, by the results of Chapter 7.3.2, the open set of step sizes guaranteed to exist - $0<\alpha<\alpha_{desc}=min\{\alpha_{g1},\alpha_{g2},...,\bar{\alpha}\}$ - results in any constraint whose underlying variables underwent a change from the reference to be satisfied with strict inequality at the next iterate. This may in fact be desirable as it will reduce the SOCP/QCQP computation time, since interior point methods need to find a strictly feasible point first. Having $\tilde{X}=0$ be such a point would eliminate these steps. If $X_{(l)}^{ref}$ is strictly feasible then $X_{(l+1)}^{ref}$ can be chosen such that it is strictly feasible and so Slater's Condition will hold at the CP defined in terms of $X_{(l+1)}^{ref}$ with $\tilde{X}_S=0$, and so we are done. If there remain constraints for which all variables involved have undergone no change since the initial guess and are satisfied with equality, a few more steps are needed.

For $X_{(l)}^{ref}$, let $\tilde{Z}_f^{(l)}=\begin{bmatrix}... & \chi_{i-1} & \upsilon_{i-1} & \tau_i & \chi_i & \upsilon_i & \tau_{i+1} & \chi_{i+1} & \upsilon_{i+1} & \tau_{i+2} & \chi_{i+2} & \upsilon_{i+2} & ...\end{bmatrix}^T$

be the vector satisfying condition ii of the CRMFCQ at $\tilde{X}=0$, which we have assumed to hold. Assume there is an active constraint whose underlying variables are fixed. The impulse magnitude constraints would be the most restrictive, so we'll proceed with that, but the logic applies to any.

Then $\tilde{X}^* = \begin{bmatrix} \dots & \tilde{x}^*_{i-1} & \tilde{v}^*_{i-1} & \tilde{t}^*_i & \tilde{x}^*_i = 0 & \tilde{v}^*_i = 0 & \tilde{t}^*_{i+1} = 0 & \tilde{x}^*_{i+1} = 0 & \tilde{v}^*_{i+1} = 0 & \tilde{t}^*_{i+2} & \tilde{x}^*_{i+2} & \tilde{v}^*_{i+2} & \dots \end{bmatrix}$,

where the constraint is at the i+1th patch point. This form, and the fact that that no velocity term to reestablish feasibility at this patch point is added since this portion matches the feasible reference, implies that these variables are identical in both $X^{ref}_{(l)}, X^{ref}_{(l+1)}$. As a result, $\nabla \tilde{h}^{(l+1)}_{i+1}(0) = \nabla \tilde{h}^{(l)}_{i+1}(0), \quad \nabla \tilde{g}^{(l+1)}_{i+1}(0) = \nabla \tilde{g}^{(l)}_{i+1}(0)$ where the change of indices from j,k to $i+1$ signals that these are the constraints that act solely on the variables of the fixed patch point. Now, since the gradients have 0 entries in indices that do not correspond to the underlying variables, we have that:

$$(\nabla \tilde{h}^{(l)}_{i+1}(0))^T \begin{bmatrix} \dots & \chi_{i-1} & \upsilon_{i-1} & \tau_i & \chi_i & \upsilon_i & \tau_{i+1} & \chi_{i+1} & \upsilon_{i+1} & \tau_{i+2} & \chi_{i+2} & \upsilon_{i+2} & \dots \end{bmatrix}^T$$

$$= (\nabla \tilde{h}^{(l)}_{i+1}(0))^T \begin{bmatrix} \dots & x_{i-1} & v_{i-1} & t_i & \chi_i & \upsilon_i & \tau_{i+1} & \chi_{i+1} & \upsilon_{i+1} & t_{i+2} & x_{i+2} & v_{i+2} & \dots \end{bmatrix}^T$$

and also for the inequalities on that patch point:

$$(\nabla \tilde{g}^{(l)}_{i+1}(0))^T \begin{bmatrix} \dots & \chi_{i-1} & \upsilon_{i-1} & \tau_i & \chi_i & \upsilon_i & \tau_{i+1} & \chi_{i+1} & \upsilon_{i+1} & \tau_{i+2} & \chi_{i+2} & \upsilon_{i+2} & \dots \end{bmatrix}^T$$

$$= (\nabla \tilde{g}^{(l)}_{i+1}(0))^T \begin{bmatrix} \dots & x_{i-1} & v_{i-1} & t_i & \chi_i & \upsilon_i & \tau_{i+1} & \chi_{i+1} & \upsilon_{i+1} & t_{i+2} & x_{i+2} & v_{i+2} & \dots \end{bmatrix}^T$$

since only the $\begin{bmatrix} \chi_i & \upsilon_i & \tau_{i+1} & \chi_{i+1} & \upsilon_{i+1} \end{bmatrix}$ terms get multiplied by a nonzero entry.

So, we seek a vector of the form

$$\hat{Z} = \begin{bmatrix} \dots & x_{i-1} & v_{i-1} & t_i & \chi_i & \upsilon_i & \tau_{i+1} & \chi_{i+1} & \upsilon_{i+1} & t_{i+2} & x_{i+2} & v_{i+2} & \dots \end{bmatrix}^T$$

(thus is equal to $\tilde{Z}_f$ at the indices of those unchanged patch points) that also satisfies the continuity constraints for the problem at $X^{ref}_{(l+1)}$. The set of variables at a fixed patch point has elements involved in continuity constraints at three patch points. One is for the fixed patch point, for which

$$(\nabla \tilde{h}^{(l+1)}_{i+1}(0))^T \hat{Z} = (\nabla \tilde{h}^{(l)}_{i+1}(0))^T \hat{Z} = (\nabla \tilde{h}^{(l)}_{i+1}(0))^T \tilde{Z}_f = 0.$$

For the preceding patch point, we have the continuity constraint

$$\begin{bmatrix} A_{i,i-1}^{(l+1)} & B_{i,i-1}^{(l+1)} & v_i^{-ref(l+1)} & -I_{3x3} & 0_{3x3} \end{bmatrix} \begin{bmatrix} x_{i-1} \\ v_{i-1} \\ t_i \\ \chi_i \\ \upsilon_i \end{bmatrix} = 0$$

which since χ_i is fixed reduces to

$$\begin{bmatrix} A_{i,i-1}^{(l+1)} & B_{i,i-1}^{(l+1)} & v_i^{-ref(l+1)} \end{bmatrix} \begin{bmatrix} x_{i-1} \\ v_{i-1} \\ t_i \end{bmatrix} = \chi_i$$

which since $\begin{bmatrix} A_{i,i-1}^{(l+1)} & B_{i,i-1}^{(l+1)} & v_i^{-ref(l+1)} \end{bmatrix}$ is full rank and "fat", there are multiple solutions even with χ_i fixed. For any such initial state and time change, there are multiple solutions for continuity at the previous solution, etc. as the system of linear equations will still be underdetermined.

Similarly, for the next patch point $\begin{bmatrix} A_{i+2,i+1}^{(l+1)} & B_{i+2,i+1}^{(l+1)} & v_{i+2}^{-ref(l+1)} & -I_{3x3} & 0_{3x3} \end{bmatrix} \begin{bmatrix} \chi_{i+1} \\ \upsilon_{i+1} \\ t_{i+2} \\ x_{i+2} \\ v_{i+2} \end{bmatrix} = 0$

$\begin{bmatrix} v_{i+2}^{-ref(l+1)} & -I_{3x3} \end{bmatrix} \begin{bmatrix} t_{i+2} \\ x_{i+2} \end{bmatrix} = -A_{i+2,i+1}^{(l+1)} \chi_{i+1} - B_{i+2,i+1}^{(l+1)} \upsilon_{i+1}$ so again there are multiple solutions. [If the next point were the terminal condition or some other fixed patch point then this would not be the case, but then the arc from the patch point to the fixed patch point would have the same initial conditions, so letting the coasting time match the reference value would yield continuity.] Let $\hat{Z}$ be any member of this set of solutions with the necessary fixed values.

Thus, even though $\hat{Z}$ may not satisfy $(\nabla \tilde{g}_k^{(l+1)}(0))^T \hat{Z} \le 0$ for any constraints with variable changes, it does satisfy $(\nabla \tilde{g}_{i+1}^{(l+1)}(0))^T \hat{Z} = (\nabla \tilde{g}_{i+1}^{(l)}(0))^T \hat{Z} = (\nabla \tilde{g}_{i+1}^{(l)}(0))^T \tilde{Z}_f = -c_k < 0$ for any of the constraints on that patch point (i.e. if those variables are fixed then the impact avoidance

constraint would remain fixed as well, and if active $\tilde{Z}_f$ would still define a descent direction since the variables that multiply nonzero entries in the gradient have not changed) and since it satisfies the linear equality constraints, $(\nabla \tilde{h}_j^{(l+1)}(0))^T \hat{Z}=0$.

If we do allow $\alpha=\alpha_{desc}$ as an option so that some inequality constraints may be satisfied with equality after a change from the reference, an extra step is needed. Otherwise, or in the case $\alpha^*<\alpha_{desc}$ anyways, let d=0 and skip to the nest paragraph. If $\alpha^*=\alpha_{desc}$ consider the function $X(\alpha)=X_{(l)}^{ref}+\alpha\tilde{X}^*+\delta v(\alpha)$, so that $X_{(l+1)}^{ref}=X(\alpha^*)=X_{(l)}^{ref}+\alpha^*\tilde{X}^*+\delta v(\alpha^*)$. Recall that $\delta v(\alpha)$ is continuously differentiable within the set $\alpha<\bar{\alpha}$, which includes α^*. Thus $X(\alpha)=X_{(l)}^{ref}+\alpha\tilde{X}^*+\delta v(\alpha)$ defines a C1 curve from $X_{(l)}^{ref}$ to $X_{(l+1)}^{ref}$. Consider the tangent vector to the curve at α^* in the negative α direction, I.e. $d=\lim_{h\to 0}\frac{X(\alpha^*-h)-X(\alpha^*)}{h}$. Since it is tangent to a curve of continuous/feasible trajectories at $X_{(l+1)}^{ref}$, d will satisfy the linearized equality constraints of the CP, which since they are linear is equivalent to $\langle\nabla\tilde{h}_j^{(l+1)}(0),\tilde{d}\rangle=0\,\forall\, j\in J$. Furthermore, $\alpha<\alpha_{desc}$ will make any active constraints whose underlying variables are nonzero in $\tilde{X}^*$ strictly satisfied. This makes d a descent direction for all such constraints.

Consider now the vector $d+\epsilon\hat{Z}$. Combining the properties of d, $\hat{Z}$:

Equality constraints,

$$\langle\nabla\tilde{h}_j^{(l+1)}(0),d+\epsilon\hat{Z}\rangle=\langle\nabla\tilde{h}_j^{(l+1)}(0),d\rangle+\epsilon\langle\nabla\tilde{h}_j^{(l+1)}(0),\hat{Z}\rangle=0+0=0$$

For inequalities with fixed underlying variables,

$$\langle\nabla\tilde{g}_k^{(l+1)}(0),d+\epsilon\hat{Z}\rangle=\langle\nabla\tilde{g}_k^{(l+1)}(0),d\rangle+\epsilon\langle\nabla\tilde{g}_k^{(l+1)}(0),\hat{Z}\rangle=0+\epsilon\langle\nabla\tilde{g}_k^{(l)}(0),\hat{Z}\rangle=-\epsilon c_k<0$$

For inequalities with a nonzero change to their underlying variables yet are active (if allowed, d=0

only if they are inactive)

$$\langle \nabla \tilde{g}_k^{(l+1)}(0), d+\epsilon \hat{Z} \rangle = \langle \nabla \tilde{g}_k^{(l+1)}(0), d \rangle + \epsilon \langle \nabla \tilde{g}_k^{(l+1)}(0), \hat{Z} \rangle = -b_k + \epsilon \langle \nabla \tilde{g}_k^{(l+1)}(0), \hat{Z} \rangle$$

$$\leq -b_k + \epsilon \| \nabla \tilde{g}_k^{(l+1)}(0) \| \| \hat{Z} \|$$

Then for $0 < \epsilon < \dfrac{min_{k \in K_c} b_k}{max_{k \in K_u} \| \nabla \tilde{g}_k^{(l+1)}(0) \| \| \hat{Z} \|}$ we will have $\langle \nabla \tilde{g}_k^{(l+1)}(0), d+\epsilon \hat{Z} \rangle < 0$. Thus for such a value of $\epsilon, d+\epsilon \hat{Z}$ satisfies condition ii of the CRMFCQ conditions. Since conditions i) holds automatically due to the linear constraints, then for the CP approximation defined at $X_{(l+1)}^{ref}, \tilde{X} = 0$ satisfies the CRMFCQ.

A.4 Summary

The following result combines the result from the previous section into the result on constraint qualifications needed elsewhere.

Proposition A.4: If the initial guess (or any other iteration) satisfies the LICQ, MFCQ, Slater's Condition, or CRMFCQ, then $\tilde{X} = 0, \tilde{X} = \tilde{X}^*$ are regular points and Slater's Condition holds for every SOCP/QCQP subproblem for every iteration thereafter.

Appendix B

Additional Constraints

B.1 Introduction

This appendix discusses additional constraints not included in Chapter 5. These constraints were omitted from the main text because they require lengthy analysis and are not applicable to every scenario. Some are of computational benefit, others are mission or spacecraft dependent.

B.2 Intermediate, Maneuver Free Patch Points

There are multiple reasons why it may make sense to add additional patch points within a coasting arc but not to add maneuvers at those points. Primary of these is the practical benefit that may be achieved from a multiple shooting approach rather than single shooting for long or unstable arcs to improve the convergence properties of the re-correction during the optimization process.[Marc07, Hugh03] Another is to allow constraints along a coasting arc without interference from a maneuver, for example to position a flyover of a region of interest for

photography or other measurements. Even constraints such as the minimum periapsis altitude constraint for impact avoidance may be recast as a constraint on a new, maneuver-free patch point set at the point of minimum altitude rather than a constraint on propagated variables.

B.2.1 The NLP Constraint

Suppose $\bar{k}$ such intermediate, maneuver-free patch points were to be added between two impulses. Let $(x_i, v_i), (x_{i+\bar{k}+1}, v_{i+\bar{k}+1})$ define these two standard, impulsive patch points. Then:

$$x_{i+1} = x_{i+1}^-, k = 1, \ldots, \bar{k} - 1$$
$$v_{i+1} = v_{i+1}^-, k = 1, \ldots, \bar{k} - 1$$
$$x_{i+\bar{k}+1} = x_{i+\bar{k}+1}^-$$

These correspond to continuity in both position and velocity for the intermediate patch points, and position continuity at the end of the ballistic arc, where a maneuver may be present at the standard patch point.

Additionally, if there are coasting time constraints, then for an arc with intermediate, maneuver-free patch points, the constraint would become: $t_{min} \le t_{i+1} + \ldots + t_{i+k+1} \le t_{max}$.

B.2.2 The CP Approximation

In terms of the CP variables and reference values from a feasible prior iteration, these constraints are approximated at a feasible reference by the following linear equality constraints:

$$\begin{bmatrix} A_{i+k,i+k-1} & B_{i+k,i+k-1} & v_{i+k}^{-ref} & -I_{3x3} & 0_{3x3} \\ C_{i+k,i+k-1} & D_{i+k,i+k-1} & a_{i+k}^{-ref} & 0_{3x3} & -I_{3x3} \end{bmatrix} \begin{bmatrix} \tilde{x}_{i+k-1} \\ \tilde{v}_{i+k-1} \\ \tilde{t}_{i+k} \\ \tilde{x}_{i+k} \\ \tilde{v}_{i+k} \end{bmatrix} = 0_{6x1}, k = 1, \ldots, \bar{k} - 1$$

$$\begin{bmatrix} A_{i+\bar{k}+1,i+k} & B_{i+\bar{k}+1,i+k} & v_{i+\bar{k}+1}^{-ref} & -I_{3x3} & 0_{3x3} \end{bmatrix} \begin{bmatrix} \tilde{x}_{i+\bar{k}} \\ \tilde{v}_{i+\bar{k}} \\ \tilde{t}_{i+\bar{k}+1} \\ \tilde{x}_{i+\bar{k}+1} \\ \tilde{v}_{i+\bar{k}+1} \end{bmatrix} = 0_{3x1}$$

B.2.3 Defining the Modified Recorrection Problem

The addition of such maneuver-free patch points introduces a larger set of equality constraints and requires a different type of intermediate correction problem to reestablish position continuity for all patch points and velocity continuity for the intermediate patch points. The first change to note is that for such patch points, the position as well as velocity need to be adjusted.

For the ballistic arc beginning at patch point *i* with $\bar{k}$ intermediate patch points, the variables of the correction process are $\vec{\delta}_i = (\delta v_i, \delta x_{i+1}, \delta x_{i+1}, \ldots, \delta x_{i+\bar{k}}, \delta v_{i+\bar{k}})$: a velocity adjustment to patch point i and position and velocity adjustments to the intermediate, maneuver-free patch points. The goal is to minimize the state discontinuities of the intermediate points and the position discontinuity at the next impulsive patch point to 0. To state this formally, define a generalization of the discontinuity function $\hat{\kappa}_i$ from Chapter 7.3.1 using line search parameter α and the correction adjustments:

$$\hat{\kappa}_i : \mathbb{R}^{6\bar{k}+3+1} \rightarrow \mathbb{R}^{6\bar{k}+3}$$

$$\hat{\kappa}_i(\alpha, \vec{\delta}_i) = \begin{bmatrix} x^-_{i+1}(\alpha, \delta v_i) - x_{i+1}(\alpha, \delta x_{i+1}, \delta v_{i+1}) \\ v^-_{i+1}(\alpha, \delta v_i) - v_{i+1}(\alpha, \delta x_{i+1}, \delta v_{i+1}) \\ \vdots \\ x^-_{i+\bar{k}}(\delta x_{i+\bar{k}-1}, \delta v_{i+\bar{k}-1}, \alpha) - x_{i+\bar{k}}(\delta x_{i+\bar{k}}, \delta v_{i+\bar{k}}, \alpha) \\ v^-_{i+\bar{k}}(\delta x_{i+\bar{k}-1}, \delta v_{i+\bar{k}-1}, \alpha) - v_{i+\bar{k}}(\delta x_{i+\bar{k}}, \delta v_{i+\bar{k}}, \alpha) \\ x^-_{i+\bar{k}+1}(\delta x_{i+\bar{k}}, \delta v_{i+\bar{k}}, \alpha) - x_{i+\bar{k}+1}(\alpha) \end{bmatrix}$$

where

$$x^-_{i+k}(\delta x_{i+k-1}, \delta v_{i+k-1}, \alpha) = x(x^{ref}_{i+k-1} + \alpha \tilde{x}_{i+k-1} + \delta x_{i+k-1}, v^{ref}_{i+k-1} + \alpha \tilde{v}_{i+k-1} + \delta v_{i+k-1}, t^{ref}_{i+k} + \alpha \tilde{t}_{i+k})$$

$$v^-_{i+k}(\delta x_{i+k-1}, \delta v_{i+k-1}, \alpha) = v(x^{ref}_{i+k-1} + \alpha \tilde{x}_{i+k-1} + \delta x_{i+k-1}, v^{ref}_{i+k-1} + \alpha \tilde{v}_{i+k-1} + \delta v_{i+k-1}, t^{ref}_{i+k} + \alpha \tilde{t}_{i+k})$$

define the endpoints of the arcs whose initial conditions are the reference values plus a scaled step in the direction of the CP solution, plus a correction adjustment, as in Chapter 7.3.1. The initial conditions are defined by:

$$x_{i+k}(\delta x_{i+k-1}, \delta v_{i+k-1}, \alpha) = x_{i+k}^{ref} + \alpha \tilde{x}_{i+k-1} + \delta x_{i+k}$$

$$v_{i+k}(\delta x_{i+k-1}, \delta v_{i+k-1}, \alpha) = v_{i+k}^{ref} + \alpha \tilde{v}_{i+k-1} + \delta v_{i+k}$$

The goal then, is to determine if there is a range of step sizes α for which there exist $\vec{\delta}_i(\alpha)$ s.t. $\hat{\kappa}_i(\alpha, \vec{\delta}_i) = \vec{0}$.

B.2.4 Existence of Solutions to the Modified Recorrection Problem

The approach taken follows that of reestablishing continuity in the original case of Chapter 7.3.1., which is to use the Implicit Function Theorem. $\hat{\kappa}_i$ is continuously differentiable, and for a feasible reference $\hat{\kappa}_i(0, \vec{0}) = \vec{0}$. Thus to verify if the Implicit Function Theorem holds it must be shown that

$$\frac{\partial \hat{\kappa}_i}{\partial \vec{\delta}} = \begin{bmatrix} B_{i+1,i} & -I_{3x3} & 0_{3x3} & 0_{3x3} & & & \cdots & 0_{3x3} \\ D_{i+1,i} & 0_{3x3} & -I_{3x3} & 0_{3x3} & & & \cdots & 0_{3x3} \\ 0_{3x3} & A_{i+2,i+1} & B_{i+2,i+1} & -I_{3x3} & 0_{3x3} & & \cdots & 0_{3x3} \\ 0_{3x3} & C_{i+2,i+1} & D_{i+2,i+1} & 0_{3x3} & -I_{3x3} & & \cdots & 0_{3x3} \\ \vdots & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ 0_{3x3} & \cdots & & 0_{3x3} & A_{i+k,i+k-1} & B_{i+k,i+k-1} & -I_{3x3} & 0_{3x3} \\ 0_{3x3} & \cdots & & 0_{3x3} & C_{i+k,i+k-1} & D_{i+k,i+k-1} & 0_{3x3} & -I_{3x3} \\ 0_{3x3} & \cdots & & & \cdots & 0_{3x3} & A_{i+k+1,i+k} & B_{i+k+1,i+k} \end{bmatrix}$$

is invertible. In order to show this, it is beneficial to move the leftmost column to the right column, which has no effect on invertibility. This creates the matrix

$$M = \begin{bmatrix} -I_{3x3} & 0_{3x3} & 0_{3x3} & & & \cdots & 0_{3x3} & B_{i+1,i} \\ 0_{3x3} & -I_{3x3} & 0_{3x3} & & & \cdots & 0_{3x3} & D_{i+1,i} \\ A_{i+2,i+1} & B_{i+2,i+1} & -I_{3x3} & 0_{3x3} & & & \cdots & 0_{3x3} \\ C_{i+2,i+1} & D_{i+2,i+1} & 0_{3x3} & -I_{3x3} & & & \cdots & 0_{3x3} \\ \vdots & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ 0_{3x3} & \cdots & 0_{3x3} & A_{i+k,i+k-1} & B_{i+k,i+k-1} & -I_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & \cdots & 0_{3x3} & C_{i+k,i+k-1} & D_{i+k,i+k-1} & 0_{3x3} & -I_{3x3} & 0_{3x3} \\ 0_{3x3} & \cdots & 0_{3x3} & 0_{3x3} & 0_{3x3} & A_{i+k+1,i+k} & B_{i+k+1,i+k} & 0_{3x3} \end{bmatrix}$$

This may be partitioned as $M=\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ where:

$$M_{11}=\begin{bmatrix} -I_{3x3} & 0_{3x3} & 0_{3x3} & & & \cdots & 0_{3x3} \\ 0_{3x3} & -I_{3x3} & 0_{3x3} & & & \cdots & 0_{3x3} \\ A_{i+2,i+1} & B_{i+2,i+1} & -I_{3x3} & 0_{3x3} & & \cdots & 0_{3x3} \\ C_{i+2,i+1} & D_{i+2,i+1} & 0_{3x3} & -I_{3x3} & & \cdots & 0_{3x3} \\ \vdots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ 0_{3x3} & \cdots & 0_{3x3} & A_{i+k,i+k-1} & B_{i+k,i+k-1} & -I_{3x3} & 0_{3x3} \\ 0_{3x3} & \cdots & 0_{3x3} & C_{i+k,i+k-1} & D_{i+k,i+k-1} & 0_{3x3} & -I_{3x3} \end{bmatrix}$$

$$M_{12}=\begin{bmatrix} B_{i+1,i} \\ D_{i+1,i} \\ 0_{3x3} \\ \vdots \\ 0_{3x3} \end{bmatrix}, M_{21}=\begin{bmatrix} 0_{3x3} & \cdots & 0_{3x3} & A_{i+k+1,i+k} & B_{i+k+1,i+k} \end{bmatrix}, M_{22}=0_{3x3}$$

To reduce clutter, in terms of the full STMs

$$M_{11}=\begin{bmatrix} -I_{6x6} & & & & \\ \Phi_{i+2,i+1} & -I_{6x6} & & & \\ & \Phi_{i+3,i+2} & & & \\ \vdots & & \ddots & & \\ & & & -I_{6x6} & \\ & & & \Phi_{i+k,i+k-1} & -I_{6x6} \end{bmatrix}$$

Note that M_{11} is invertible with inverse

$$M_{11}^{-1}=\begin{bmatrix} -I_{6x6} & & & & \\ -\Phi_{i+2,i+1} & -I_{6x6} & & & \\ -\Phi_{i+3,i+2}\Phi_{i+2,i+1} & -\Phi_{i+3,i+2} & & & \\ \vdots & & \ddots & & \\ & & & -I_{6x6} & \\ -\Phi_{i+k,i+k-1}\cdots\Phi_{i+2,i+1} & \cdots & -\Phi_{i+k,i+k-1}\Phi_{i+k-1,i+k-2} & -\Phi_{i+k,i+k-1} & -I_{6x6} \end{bmatrix}.$$

Using the property of state transition matrices that $\Phi(t_2,t_1)\Phi(t_1,t_0)=\Phi(t_2,t_0),$ this may be more compactly written as

$$M_{11}^{-1}=\begin{bmatrix} -I_{6x6} & & & & \\ -\Phi_{i+2,i+1} & -I_{6x6} & & & \\ -\Phi_{i+3,i+1} & -\Phi_{i+3,i+2} & & & \\ \vdots & & \ddots & & \\ & & & -I_{6x6} & \\ -\Phi_{i+k,i+1} & \cdots & -\Phi_{i+k,i+k-2} & -\Phi_{i+k,i+k-1} & -I_{6x6} \end{bmatrix}$$

Now, since M_{11} is invertible, M will be invertible iff its Schur complement w.r.t. M_{11} Is invertible as well.[Boyd94]. This is defined as:

$$S=M_{22}-M_{21}M_{11}^{-1}M_{12}=\begin{bmatrix} A_{i+k+1,i+k} & B_{i+k+1,i+k} \end{bmatrix}\begin{bmatrix} A_{i+k,i+1} & B_{i+k,i+1} \\ C_{i+k,i+1} & D_{i+k,i+1} \end{bmatrix}\begin{bmatrix} B_{i+1,i} \\ D_{i+1,i} \end{bmatrix}$$

Note that

$$\begin{bmatrix} A_{i+k+1,i} & B_{i+k+1,i} \\ C_{i+k+1,i} & D_{i+k+1,i} \end{bmatrix}=\Phi_{i+k+1,i}=\Phi_{i+k+1,i+k}\Phi_{i+k,i+1}\Phi_{i+1,i}$$
$$=\begin{bmatrix} A_{i+k+1,i+k} & B_{i+k+1,i+k} \\ C_{i+k+1,i+k} & D_{i+k+1,i+k} \end{bmatrix}\begin{bmatrix} A_{i+k,i+1} & B_{i+k,i+1} \\ C_{i+k,i+1} & D_{i+k,i+1} \end{bmatrix}\begin{bmatrix} A_{i+1,i} & B_{i+1,i} \\ C_{i+1,i} & D_{i+1,i} \end{bmatrix}$$

Comparing the bottom expression to the expression for *S*, we see that *S* is equal to the top right entry of $\Phi_{i+k+1,i}$, and so $S=B_{i+k+1,i}$. This matrix is identical to the *B* matrix of Chapter 7.3.1, relating velocity variations in (standard) patch point velocity to the resulting variation in the following patch point position, just as if there were no intermediate patch points added. Thus the condition for the existence of a range of corrections is the same with or without such intermediate maneuver-free points added. As in that case, since this matrix is assumed to be invertible, a set of corrections exist for some range of step size α values in a neighborhood of 0.

This result is not surprising. If there exist adjustments that lead to a ballistic arc between two standard patch points that happen to have some intermediate patch points with no maneuvers, then the resulting arc satisfies position continuity at its endpoint with or without the arc being divided into pieces. Likewise, should an initial velocity correction in a single shooting approach exist creating a continuous transfer, then the difference between points on the arc with the

appropriate coasting times and the corresponding intermediate patch points give a set of adjustments satisfying the above constraints.

B.2.5 Magnitude of Correction

The Implicit Function Theorem also provides an expression for the derivative

$$\frac{\partial \vec{\delta}_i}{\partial \alpha}(0) = -\left(\frac{\partial \hat{\kappa}_i}{\partial \vec{\delta}_i}(0,\vec{0})\right)^{-1} \frac{\partial \hat{\kappa}_i}{\partial \alpha}(0,\vec{0}).$$

For each term of $\hat{\kappa}_i$ the terms of this derivative are 0, due to the constraints the CP solution must satisfy that were derived above.

Position terms -

$$\frac{\partial}{\partial \alpha}|_{(0,\vec{0})}\left[x(x_{i+k-1}^{ref}+\alpha\tilde{x}_{i+k-1}+\delta x_{i+k-1}, v_{i+k-1}^{ref}+\alpha\tilde{v}_{i+k-1}+\delta v_{i+k-1}, t_{i+k}^{ref}+\alpha\tilde{t}_{i+k}) - (x_{i+k}^{ref}+\alpha\tilde{x}_{i+k-1}+\delta x_{i+k})\right]$$

$$= A_{i+k,i+k-1}\tilde{x}_{i+k-1} + B_{i+k,i+k-1}\tilde{v}_{i+k-1} + v_{i+k}^{-ref}\tilde{t}_{i+k} - \tilde{x}_{i+k-1}$$

$$= \begin{bmatrix} A_{i+k,i+k-1} & B_{i+k,i+k-1} & v_{i+k}^{-ref} & -I_{3x3} & 0_{3x3} \end{bmatrix} \begin{bmatrix} \tilde{x}_{i+k-1} \\ \tilde{v}_{i+k-1} \\ \tilde{t}_{i+k} \\ \tilde{x}_{i+k} \\ \tilde{v}_{i+k} \end{bmatrix} = 0$$

Velocity terms -

$$\frac{\partial}{\partial \alpha}|_{(0,\vec{0})}\left[v(x_{i+k-1}^{ref}+\alpha\tilde{x}_{i+k-1}+\delta x_{i+k-1}, v_{i+k-1}^{ref}+\alpha\tilde{v}_{i+k-1}+\delta v_{i+k-1}, t_{i+k}^{ref}+\alpha\tilde{t}_{i+k}) - (v_{i+k}^{ref}+\alpha\tilde{v}_{i+k-1}+\delta v_{i+k})\right]$$

$$= C_{i+k,i+k-1}\tilde{x}_{i+k-1} + D_{i+k,i+k-1}\tilde{v}_{i+k-1} + a_{i+k}^{-ref}\tilde{t}_{i+k} - \tilde{v}_{i+k-1}$$

$$= \begin{bmatrix} C_{i+k,i+k-1} & D_{i+k,i+k-1} & a_{i+k}^{-ref} & 0_{3x3} & -I_{3x3} \end{bmatrix} \begin{bmatrix} \tilde{x}_{i+k-1} \\ \tilde{v}_{i+k-1} \\ \tilde{t}_{i+k} \\ \tilde{x}_{i+k} \\ \tilde{v}_{i+k} \end{bmatrix} = 0$$

Thus $\frac{\partial \vec{\delta}_i}{\partial \alpha}(0) = 0$ as expected and so $\|\vec{\delta}_i(\alpha)\| = o(\alpha)$. By definition of $\vec{\delta}_i$ this implies that for the components as well, $\|\delta v_i(\alpha)\| = o(\alpha), \|\delta x_{i+1}(\alpha)\| = o(\alpha),$ etc. With this order and

existence established, the proof of Chapter 7.3.1 and 7.3.2 may be applied to this case without further alteration. Thus adding intermediate, maneuver-free patch points does not affect the decrease of the cost function or feasibility of the other constraints at each major iteration.

B.3 Thrust Angle Limitations

This constraint handles the case where there is a limitation on the angle between impulse applied and the velocity in order to limit attitude changes. The existence of such a constraint is spacecraft specific, and is not being asserted as necessary for all cases, but is potentially useful. It is assumed the impulse is limited to an angle of θ_{max} from the velocity direction. This forces the new velocity to reside within a cone with vertex centered at, and axis aligned with, the tip of the incoming velocity.

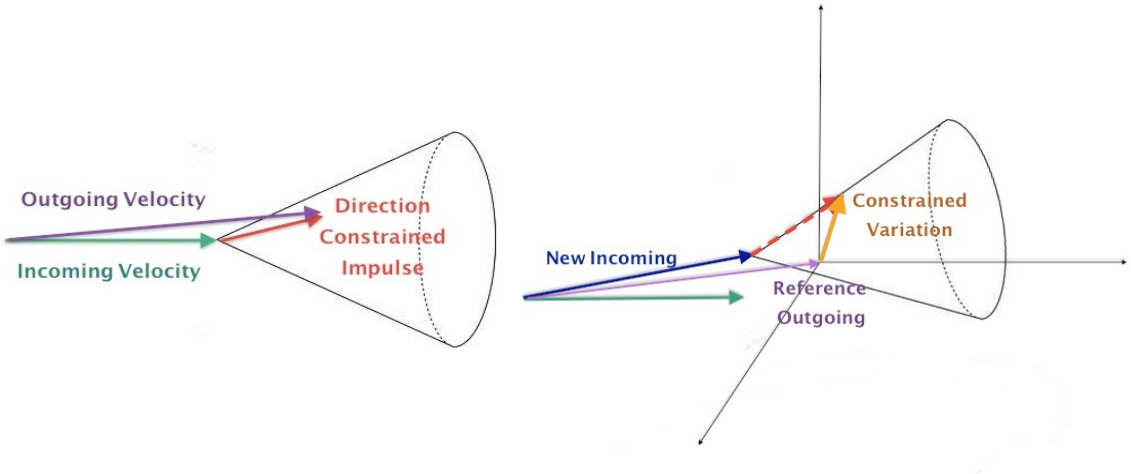


Figure B.1: Thrust direction cone constraint and expression in terms of small variation variables.

B.3.1 The Standard Geometric Cone

For a cone with vertex at the origin, opening in the positive z direction, with half angle θ_{max} the interior is defined by $\frac{\sqrt{x^2+y^2}}{\tan\theta} \le z$ which we can reformat as:

$$\left\| \begin{bmatrix} 1/\tan\theta & 0 & 0 \\ 0 & 1/\tan\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right\| \leq \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Not surprisingly, this geometric cone is a special case of a second order cone.

B.3.2 Defining the Angle Constraint as a Second Order Cone

As opposed to a simply expressed cone in position space such as that described in B.4, this constraint is much more complex because it involves a cone that is positioned and oriented relative to propagated variables.

In order to verify if a candidate velocity after impulse is within θ_{max} of the velocity before impulse, a change of coordinates is needed so that the cone defined above may be used. For any nonzero arc endpoint velocity, let $R_{i+1}(v_{i+1}^-), \rho_{i+1}(v_{i+1}^-)$ define a rotation and translation that maps the ending velocity vector to the positive *z*-axis. Such a transformation then maps the cone aligned with the velocity to the cone defined above. Then $R_{i+1}(v_{i+1}^-)v_{i+1} + \rho_{i+1}(v_{i+1}^-)$ maps potential post-impulse velocities to this frame as well. Thus $R_{i+1}(v_{i+1}^-)v_{i+1} + \rho_{i+1}(v_{i+1}^-)$ resides within the *z*-axis aligned cone above if and only if differed from the incoming velocity by less than θ_{max}.

To save space, let $\Theta_{xy} = \begin{bmatrix} 1/\tan\theta & 0 & 0 \\ 0 & 1/\tan\theta & 0 \\ 0 & 0 & 0 \end{bmatrix}$. The constraint for the original problem is then of the form :

$$\left\| \Theta_{xy} R_{i+1}(v_{i+1}^-)v_{i+1} + \Theta_{xy}\rho_{i+1}(v_{i+1}^-) \right\| < \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} R_{i+1}(v_{i+1}^-)v_{i+1} + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \rho_{i+1}(v_{i+1}^-).$$

Note this itself is not a second order cone due to the nonlinearities/integration involved in expressing the endpoint velocity in terms of the optimization variables. It is however clear that if

the STM based approximations elsewhere are used then the form will be correct.

To find the resulting constraint approximation using the STM approximations, each term will need to be broken down separately. Note that the particular parameterization of the rotation and translation has not been given for generality. Taking derivatives of any such parameterization such as multiplication of two fundamental rotation matrices (first about the *z*-axis to rotate the velocity into the xz-plane, then about the *y*-axis to align the vector with the *z*-axis by angles $-\tan^{-1}\frac{v_y}{v_x}$, $-\left(\frac{\pi}{2}-\sin^{-1}\frac{v_z}{\|v\|}\right)$ respectively), would not simplify the notation.

<u>*Rotation Term:*</u>

$$R_{i+1}(\bar{v}_{i+1})v_{i+1}=R_{i+1}(\bar{v}_{i+1}^{\text{ref}}+\Delta\bar{v}_{i+1})(v_{i+1}^{\text{ref}}+\tilde{v}_{i+1})$$

$$\approx\left(R_{i+1}(\bar{v}_{i+1}^{\text{ref}})+\Delta\bar{v}_{x,i+1}\frac{\partial R}{\partial v_x}(\bar{v}_{i+1}^{\text{ref}})+\Delta\bar{v}_{y,i+1}\frac{\partial R}{\partial v_y}(\bar{v}_{i+1}^{\text{ref}})+\Delta\bar{v}_{z,i+1}\frac{\partial R}{\partial v_z}(\bar{v}_{i+1}^{\text{ref}})\right)(v_{i+1}^{\text{ref}}+\tilde{v}_{i+1})$$

$$\approx R_{i+1}(\bar{v}_{i+1}^{\text{ref}})v_{i+1}^{\text{ref}}+R_{i+1}(\bar{v}_{i+1}^{\text{ref}})\tilde{v}_{i+1}$$
$$+\left(\Delta\bar{v}_{x,i+1}\frac{\partial R}{\partial v_x}(\bar{v}_{i+1}^{\text{ref}})+\Delta\bar{v}_{y,i+1}\frac{\partial R}{\partial v_y}(\bar{v}_{i+1}^{\text{ref}})+\Delta\bar{v}_{z,i+1}\frac{\partial R}{\partial v_z}(\bar{v}_{i+1}^{\text{ref}})\right)v_{i+1}^{\text{ref}}$$

$$\approx R_{i+1}(\bar{v}_{i+1}^{\text{ref}})v_{i+1}^{\text{ref}}+R_{i+1}(\bar{v}_{i+1}^{\text{ref}})\tilde{v}_{i+1}$$
$$+\left(\Delta\bar{v}_{x,i+1}\left(\frac{\partial R}{\partial v_x}(\bar{v}_{i+1}^{\text{ref}})\right)v_{i+1}^{\text{ref}}+\Delta\bar{v}_{y,i+1}\left(\frac{\partial R}{\partial v_y}(\bar{v}_{i+1}^{\text{ref}})\right)v_{i+1}^{\text{ref}}+\Delta\bar{v}_{z,i+1}\left(\frac{\partial R}{\partial v_z}(\bar{v}_{i+1}^{\text{ref}})\right)v_{i+1}^{\text{ref}}\right)$$

$$=R_{i+1}(\bar{v}_{i+1}^{\text{ref}})v_{i+1}^{\text{ref}}+R_{i+1}(\bar{v}_{i+1}^{\text{ref}})\tilde{v}_{i+1}$$
$$+\left[\left(\frac{\partial R}{\partial v_x}(\bar{v}_{i+1}^{\text{ref}})\right)v_{i+1}^{\text{ref}}\quad\left(\frac{\partial R}{\partial v_y}(\bar{v}_{i+1}^{\text{ref}})\right)v_{i+1}^{\text{ref}}\quad\left(\frac{\partial R}{\partial v_z}(\bar{v}_{i+1}^{\text{ref}})\right)v_{i+1}^{\text{ref}}\right]\Delta\bar{v}_{i+1}$$

$$=R_{i+1}(\bar{v}_{i+1}^{\text{ref}})v_{i+1}^{\text{ref}}+R_{i+1}(\bar{v}_{i+1}^{\text{ref}})\tilde{v}_{i+1}$$

$$+\left(\left[\left(\frac{\partial R}{\partial v_x}\left(v_{i+1}^{-\,\mathrm{ref}}\right)\right)v_{i+1}^{\mathrm{ref}} \quad \left(\frac{\partial R}{\partial v_y}\left(v_{i+1}^{-\,\mathrm{ref}}\right)\right)v_{i+1}^{\mathrm{ref}} \quad \left(\frac{\partial R}{\partial v_z}\left(v_{i+1}^{-\,\mathrm{ref}}\right)\right)v_{i+1}^{\mathrm{ref}}\right]\begin{bmatrix} C_{i+1,i} & D_{i+1,i} & a_{i+1}^{\mathrm{ref}}\end{bmatrix}\right)\begin{bmatrix}\tilde{x}_i\\ \tilde{v}_i\\ \tilde{t}_{i+1}\end{bmatrix}$$

Let $\Gamma_{i+1}=\left(\left[\left(\frac{\partial R}{\partial v_x}\left(v_{i+1}^{-\,\mathrm{ref}}\right)\right)v_{i+1}^{\mathrm{ref}} \quad \left(\frac{\partial R}{\partial v_y}\left(v_{i+1}^{-\,\mathrm{ref}}\right)\right)v_{i+1}^{\mathrm{ref}} \quad \left(\frac{\partial R}{\partial v_z}\left(v_{i+1}^{-\,\mathrm{ref}}\right)\right)v_{i+1}^{\mathrm{ref}}\right]\begin{bmatrix} C_{i+1,i} & D_{i+1,i} & a_{i+1}^{\mathrm{ref}}\end{bmatrix}\right)\in\mathbb{R}^{3x7}$.

Then the first-order approximation is finally:

$$R_{i+1}\left(v_{i+1}^{-}\right)v_{i+1}\approx R_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right)v_{i+1}^{\mathrm{ref}}+\begin{bmatrix}\Gamma_{i+1} & 0_{3x3} & R_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right) & 0_{3x1}\end{bmatrix}\begin{bmatrix}\tilde{x}_i\\ \tilde{v}_i\\ \tilde{t}_{i+1}\\ \tilde{x}_{i+1}\\ \tilde{v}_{i+1}\\ \tilde{t}_{i+2}\end{bmatrix}$$

Translation term:

$$\begin{aligned}\rho_{i+1}\left(v_{i+1}^{-}\right)&=\rho_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}+\Delta v_{i+1}^{-}\right)\\ &\approx\rho_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right)+\nabla\rho_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right)\Delta v_{i+1}^{-}\\ &\approx\rho_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right)+\nabla\rho_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right)\begin{bmatrix}C_{i+1,i} & D_{i+1,i} & a_{i+1}^{\mathrm{ref}}\end{bmatrix}\begin{bmatrix}\tilde{x}_i\\ \tilde{v}_i\\ \tilde{t}_{i+1}\end{bmatrix}\end{aligned}$$

Let $P_{i+1}=\nabla\rho_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right)\begin{bmatrix}C_{i+1,i} & D_{i+1,i} & a_{i+1}^{\mathrm{ref}}\end{bmatrix}$. Combining the work above we have the following constraint on the outgoing velocities $\tilde{v}_{i+1}$:

$$\begin{aligned}&\left\|\Theta_{xy}\begin{bmatrix}\ldots & (\Gamma_{i+1}-P_{i+1}) & 0_{3x3} & R_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right) & 0_{3x1} & \ldots\end{bmatrix}\tilde{X}+\Theta_{xy}\left(R_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right)v_{i+1}^{\mathrm{ref}}+\rho_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right)\right)\right\|\\ <&\begin{bmatrix}0 & 0 & 1\end{bmatrix}\begin{bmatrix}(\Gamma_{i+1}-P_{i+1}) & 0_{3x3} & R_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right) & 0_{3x1}\end{bmatrix}\tilde{X}+\begin{bmatrix}0 & 0 & 1\end{bmatrix}\left(R_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right)v_{i+1}^{\mathrm{ref}}+\rho_{i+1}\left(v_{i+1}^{-\,\mathrm{ref}}\right)\right).\end{aligned}$$

This second order cone constraint is the restriction of the variation plus the reference outgoing velocity to lie within the cone with vertex and opening direction based on the first order prediction of the previous arc endpoint position and orientation.

B.3.3 Inclusion into the Proof of Descent with Feasible Iterates

As just shown, the Second Order Cone constraint is given by

$$\begin{aligned}&\left\|\Theta_{xy}\left[\ldots\quad(\Gamma_{i+1}-\mathrm{P}_{i+1})\quad 0_{3\mathrm{x}3}\quad R_{i+1}(v_{i+1}^{\text{- ref}})\quad 0_{3\mathrm{x}1}\quad\ldots\right]\tilde{X}+\Theta_{xy}\left(R_{i+1}(v_{i+1}^{\text{- ref}})v_{i+1}^{\text{ref}}+\rho_{i+1}(v_{i+1}^{\text{- ref}})\right)\right\|\\<&\left[0\quad 0\quad 1\right]\left[(\Gamma_{i+1}-\mathrm{P}_{i+1})\quad 0_{3\mathrm{x}3}\quad R_{i+1}(v_{i+1}^{\text{- ref}})\quad 0_{3\mathrm{x}1}\right]\tilde{X}+\left[0\quad 0\quad 1\right]\left(R_{i+1}(v_{i+1}^{\text{- ref}})v_{i+1}^{\text{ref}}+\rho_{i+1}(v_{i+1}^{\text{- ref}})\right)\end{aligned}$$

The analysis for the Maximum ΔV per Maneuver constraint in Chapter 7.3.2 holds for this constraint in the case where the reference trajectory satisfies this constraint with strict inequality. Additionally, the same arguments hold when the reference satisfies the constraint with equality but the CP solution moves in a descent direction of the constraint function. This is because yet again the increase to the constraint function due to recorrection is $o(\alpha)=O(\|\delta v\|)$. This is because the change to the angle is less than or equal to the worst case scenario where $|\delta\theta|=\sin^{-1}\left(\frac{\|\delta v\|}{\|v^{ref}\|}\right)$.

Taking the Taylor series of this worst case angle change for the outgoing velocity (i.e. the velocity is changing relative to a cone from an unchanged endpoint velocity) in terms of $\|\delta v\|$:

$$\begin{aligned}\delta\theta&=\sin^{-1}\left(\frac{\|\delta v\|}{\|v^{ref}\|}\right)_{\|\delta v\|=0}+\|\delta v\|\cdot\left(\frac{1}{\sqrt{\|v^{ref}\|^2-\|\delta v\|^2}}\right)_{\|\delta v\|=0}+O(\|\delta v\|^2)\\&=\sin^{-1}(0)+\|\delta v\|\cdot\left(\frac{1}{\sqrt{\|v^{ref}\|^2}}\right)+O(\|\delta v\|^2)\\&=\|\delta v\|\cdot\left(\frac{1}{\|v^{ref}\|}\right)+O(\|\delta v\|^2)\\&=O(\|\delta v\|)=o(\alpha)\end{aligned}$$

This result is not surprising, given the standard small angle approximation $\frac{\|\delta v\|}{\|v^{ref}\|}=\sin\delta\theta\approx\delta\theta$. So to first order δθ equals $\|\delta v\|$ multiplied by a constant, thus $\delta\theta=O(\|\delta v\|)=o(\alpha)$. The change to the constraint function due to changing endpoint velocities after reestablishing continuity is the same, other than being in terms of the magnitude of the endpoint velocity. This however, has also

been shown to be $O(\|\delta v\|)=o(\alpha_i)$, albeit with a constant magnified by up to $\|D_{i+1,i}\|$.

Unlike the maximum thrust constraint however, it is possible to move from a reference that satisfies the constraint with equality to a CP solution that satisfies the constraint with equality without moving in a descent direction: the case where the CP solution lies on ray from the vertex through the reference value. Suppose this is the case. Consider any closed ball entirely contained within the velocity direction cone containing the reference value on its boundary (chose the center to be along the inward normal from the reference, a sufficiently small radius may then be calculated). For this iteration only, the cone constraint is replaced with being in the interior of this ball. Note that the reference velocity satisfies this constraint, but no rays along the cone do, preventing the CP result from satisfying the cone constraint without moving in a descent direction of the constraint function. Thus by forcing movement in a descent direction of the constraint function and using the fact that reestablishing continuity increases the constraint function with order $o(\alpha)$, there exists a range of α values such that both:

$$J(X^{\text{ref}})>J(X^{\text{ref}}+\alpha\tilde{X}+\vec{\delta}v)$$
$$g_{angle}(X^{\text{ref}})>g_{angle}(X^{\text{ref}}+\alpha\tilde{X}+\vec{\delta}v)$$

Lastly, the reference trajectory is a fixed point minimum of the CP with the replacement constraint if and only if it is of the CP without the cone constraint. This is because if moving along the ray is a descent direction of the cost then $\nabla\tilde{J}(0)\tilde{X}_{ray}<0$ strictly. Thus there is an open set around $\tilde{X}_{ray}$ of descent directions. Since it is open and the cone and the ball of the replacement constraint meet tangentially, some of these descent directions must be feasible directions for the helper constraint. Thus the replacement constraint would not cause the process to terminate at a fixed point unless it would have for the original cone constraint.

B.4 Conical Observation Regions

Consider the case where specific points on the surface are to be observed by a spacecraft. In order to prevent photographs from being taken at too oblique of an angle, there may be a maximum viewing incidence angle ϑ relative to the surface normal $\hat{n}$. [Suro15] Such a constraint clearly reduces to requiring the spacecraft to pass within a cone opening in the direction of $\hat{n}$ with vertex at the point of interest and half angle ϑ. In this case as opposed to the maximum thrust angle constraint, the rotation and translation matrices needed are independent of the state of the spacecraft. Indeed in the case of a tidally locked moon, the constraint in the rotating frame is time invariant. This case is also simply because the variables involved – the position at the patch point – are the optimization variables themselves with no propagation. Presumably no maneuver is desired at this point, so for a maneuver-free patch point *i*, the constraint in terms of the rotation and translation matrices, and the Θ_{xy} matrix of B.3.1,B.3.2 is:

$$\left\|\Theta_{xy}R(t)x_i+\Theta_{xy}\rho(t)\right\|<\begin{bmatrix}0 & 0 & 1\end{bmatrix}R(t)x_i+\begin{bmatrix}0 & 0 & 1\end{bmatrix}\rho(t).$$

Appendix C

Dynamical Models

C.1 Introduction

This appendix discusses the features of the dynamical model used in the applications discussed in this dissertation. Note that the Restricted Two Body Problem, although historically the basis for most mission analysis, is not used. Not using this model is immediately justified by the fact that for the primary application at Phobos the sphere of influence (SOI) for which this model is valid is in fact contained within Phobos itself. As a result, only three or more body approximations are considered.

C.2 The Circular Restricted Three-Body Problem

For orbiters of bodies such as planetary moons and asteroids, it may be either desirable or necessary to include not only the gravitational effects of the body being orbited by the satellite, but also of the relevant planet or the Sun that the target body is itself orbiting. For example, for an

orbiter at Phobos the gravity of Mars must be included. For bodies with low eccentricity orbits, the Circular Restricted Three Body Problem (CR3BP) is a standard model for this situation. The normal point mass formulation of the CR3BP describes the motion of a spacecraft (assumed to be massless) under the gravitational influence of two massive bodies in circular orbits around their barycenter. The equations of motion are given in the rotating frame with the same frequency as that of the orbits. It is generally given in normalized units so that the distance between the bodies is 1, and that the angular frequency of the orbits is 1. In these units the equations of motion are:

$$\ddot{x} = x + 2\dot{y} - \frac{1-\mu}{r_1^3}(x+\mu) - \frac{\mu}{r_2^3}(x+\mu-1)$$

$$\ddot{y} = y - 2\dot{x} - \frac{1-\mu}{r_1^3}y - \frac{\mu}{r_2^3}y$$

$$\ddot{z} = -\frac{1-\mu}{r_1^3} - \frac{\mu}{r_2^3}z$$

where the parameter μ is defined as the ratio of the mass of the smaller body over the combined mass of the two bodies. Thus for Phobos, it is given by $\mu = \frac{M_{\text{phobos}}}{M_{\text{phobos}} + M_{\text{mars}}} = 1.6559\text{e-}08$. r_1 and r_2 are the distances from the spacecraft to the primary (Mars) and the secondary (Phobos) bodies. In the Phobos/Mars system, one normalized unit of distance corresponds to 9,378.414449 km, and one normalized unit of time is 4,388.556729 seconds. In this frame Mars is located at (-μ,0,0) and Phobos at (1-μ,0,0).

The CR3BP has become the most popular model for analyzing three body systems and the role of libration point orbits, dynamical resonances, and invariant manifolds. [Koon08, Howe84, Lo04] However, for a very irregularly shaped body such as Phobos, even for a simplified discretization model this is an insufficient approximation. Therefore more detail must be included in the model.

C.3 Spherical Harmonic Gravity

Compared to most planetary moons, Phobos is very irregularly shaped, resembling an asteroid perhaps more than it does other major moons. Indeed, within 6 kilometers of the surface of Phobos (a region containing the important libration point orbits used in the orbital operation plan) these perturbations from the irregularity of Phobos are greater than the effects of any non-gravitational forces (see Chapter 10). The parameters up to fourth degree and order are:

$C_{0,0} = 1$	$S_{0,0} = 0$
$C_{1,0} = -0.000160082$	$S_{1,0} = 0$
$C_{1,1} = -0.000053128$	$S_{1,1} = -0.000205951$
$C_{2,0} = -0.029683000$	$S_{2,0} = 0$
$C_{2,1} = 0.000797914$	$S_{2,1} = 0.000083071$
$C_{2,2} = 0.015402100$	$S_{2,2} = 0.000186869$
$C_{3,0} = 0.001488130$	$S_{3,0} = 0$
$C_{3,1} = -0.002231580$	$S_{3,1} = 0.001066390$
$C_{3,2} = -0.004444380$	$S_{3,2} = 0.000364627$
$C_{3,3} = 0.000788783$	$S_{3,3} = -0.006710720$
$C_{4,0} = 0.002655290$	$S_{4,0} = 0$
$C_{4,1} = 0.001244880$	$S_{4,1} = -0.000432743$
$C_{4,2} = -0.000990392$	$S_{4,2} = -0.000621825$
$C_{4,3} = -0.001257780$	$S_{4,3} = 0.001084600$
$C_{4,4} = 0.000347189$	$S_{4,4} = 0.000024846$

These parameters are normalized using the Kaula convention. [Mont11] They were derived at Jet Propulsion Laboratory from an earlier polytopic shape model. [Gask11]

Basic spherical harmonic methods are commonly used to express potential, acceleration, and derivatives of acceleration using spherical coordinates and Legendre polynomials. This form is not ideal for the current application due to the use of Cartesian coordinates in the rotating frame, and because for higher degree and order polynomials either a large number of formulas must be stored with one applied to each term in the expansion, or involve a large number of derivative calculations. The recursive method of Cunningham as provided in [Mont11] gives the needed

accelerations and derivatives in Cartesian coordinates and uses recursion formulas to avoid the higher order derivatives. It also allows for increases or decreases in degree and order as the situation requires.

This method begins by defining terms

$$V_{l,m}=\left(\frac{R}{r}\right)^{l+1}P_{l,m}(\sin\phi)\cos m\theta,\quad W_{l,m}=\left(\frac{R_+}{r}\right)^{l+1}P_{l,m}(\sin\phi)\sin m\theta$$

where $P_{l,m}$ are the Legendre Polynomials of degree l and order m and r,ϕ,θ are the spherical coordinates and R is the radius of the sphere of convergence of the harmonic expansion. For the Phobos spherical harmonic model used, R=13.93km. In order to calculate these terms directly and in Cartesian coordinates (here and in the rest of this section in the Body Centered, Body Fixed frame), the following recurrence relationships have been derived [Mont11]:

$$V_{m,m}=(2m-1)\left[\frac{xR}{r^2}V_{m-1,m-1}-\frac{yR}{r^2}W_{m-1,m-1}\right],\quad W_{m,m}=(2m-1)\left[\frac{xR}{r^2}W_{m-1,m-1}-\frac{yR}{r^2}V_{m-1,m-1}\right]$$

$$V_{l,m}=\frac{2l-1}{l-m}\frac{zR}{r^2}V_{l-1,m}-\frac{l+m-1}{l-m}\frac{R^2}{r^2}V_{l-2,m},\quad W_{l,m}=\frac{2l-1}{l-m}\frac{zR}{r^2}W_{l-1,m}-\frac{l+m-1}{l-m}\frac{R^2}{r^2}W_{l-2,m}$$

where $m\le l$ and $V_{0,0}=\frac{R}{r},\quad W_{0,0}=0$.

Once these terms have been calculated to the necessary degree and order, the potential, acceleration, and partial derivates may then be calculated. The gravitational potential using these terms is then:

$$U=\frac{GM}{R}\sum_{l=0}^{D}\sum_{m=0}^{l}\left(C_{l,m}V_{l,m}+S_{l,m}W_{l,m}\right)$$

and acceleration is given by

$$\ddot{x}=\sum_{l,m}\ddot{x}_{l,m},\ddot{y}=\sum_{l,m}\ddot{y}_{l,m},\ddot{z}=\sum_{l,m}\ddot{z}_{l,m}$$

where

$$\ddot{x}_{l,0}=\frac{GM}{R^2}\left(-C_{l,0}V_{l+1,1}\right),m=0$$

$$\ddot{x}_{l,m}=\frac{GM}{R^2}\frac{1}{2}\left\{\left(-C_{l,m}V_{l+1,m+1}+S_{l,m}W_{l+1,m+1}\right)+\frac{(l-m+2)!}{(l-m)!}\left(-C_{l,m}V_{l+1,m-1}+S_{l,m}W_{l+1,m-1}\right)\right\},m>0$$

$$\ddot{y}_{l,0}=\frac{GM}{R^2}\left(-C_{l,0}W_{l+1,1}\right),m=0$$

$$\ddot{y}_{l,m}=\frac{GM}{R^2}\frac{1}{2}\left\{\left(-C_{l,m}W_{l+1,m+1}+S_{l,m}V_{l+1,m+1}\right)+\frac{(l-m+2)!}{(l-m)!}\left(-C_{l,m}W_{l+1,m-1}+S_{l,m}V_{l+1,m-1}\right)\right\},m>0$$

and

$$\ddot{z}_{l,m}=\frac{GM}{R^2}(l-m+1)\left(-C_{l,m}V_{l+1,m}-S_{l,m}W_{l+1,m}\right)$$

Similar formulas exist for partial derivatives, but are omitted here for brevity as they are quite lengthy. Please see [Mont11]. Note that for a model of degree and order *D,* these terms are calculated up to $l=D$ when only potential is needed, $l=D+1$ when accelerations are needed, and up to $l=D+2$ for partial derivatives.

As mentioned above, these formulas are given in the Body Centered, Body fixed (BCBF) frame. Therefore coordinate transformations are needed between the CR3BP frame and BCBF frame. Note that in the case of tidally locked moons such as Earth's Moon, Phobos, Deimos, Europa, Enceladus, etc., the body is fixed in position and orientation, thus there is only a simple time invariant Euclidean transformation for position, velocity, and acceleration.

C.3 Solar Radiation Pressure

Using the “flat plate model”, the force on a spacecraft due to solar radiation pressure is given by:

$$\vec{F}_{srp}=P_0\left(\frac{R_0}{r_S}\right)^2 A\left[(1-\rho)\vec{e}_0+\rho(\hat{n}\cdot\vec{e}_0)\hat{n}\right]$$

where $P_0 = 4.56 \times 10^{-6} N/m^2$ is the provided reference pressure given at a distance $R_0 = 1\,\text{AU}$, A the assumed surface area of the spacecraft, $\hat{n}$ the outward normal of the plate surface pointing away from the Sun, $\vec{e}_0$ the unit vector from the Sun to the spacecraft, r_S the distance from the center of the Sun, and ρ the reflectivity percentage. [Vall01] Note that changes to the distance from the Sun due to the changes in the spacecraft orbit around Phobos, and indeed the orbit of Phobos around Mars are very small relative to the distance from the Sun to Mars. Thus we take r_S to be the distance from the Sun to Mars. The surface area A is taken to be $A = 40\,m^2$, approximately the combined surface area of the very large panels of the Dawn Spacecraft. [Thom12] Similarly, the mass is taken to be 1,250kg, approximately that of the 1,218 kg Dawn Spacecraft. [Thom12]

An estimate of the resulting acceleration magnitude of the SRP may now be determined. The semi-major axis of the orbit of Mars around the Sun is 1.5237 AU. Thus on average

$$\|F_{s.r.p.}\| \approx P_0 \left(\frac{R_0}{r_{min}} \right)^2 A_{max} = 1.824 \times 10^{-4} \left(\frac{1\,\text{AU}}{1.5237\,\text{AU}} \right)^2 N = 7.856 \times 10^{-5} N$$

Dividing by the assumed mass results in an average acceleration due to SRP of $6.285 \times 10^{-8} m/s^2$.

In practice, even though Mars is in an elliptic rather than circular orbit around the Sun, on the timescale of these transfers position of the Mars/Phobos system to the Sun may be considered constant. Therefore in the rotating frame, the Sun's motion may be modeled by circular motion around the origin of the CR3BP with angular frequency equal to that of the frame itself, as is done in one of the forms of the Bicircular Restricted Four Body Problem. [Koon08] Determining the orientation of the flat plate normal vector is outside of the scope of this work, therefore it is assumed to be provided. For any simulations run as part of this research, it has been assumed to be collinear with the light direction.

C.4 Bicircular Restricted Four Body Problem

This dynamical model is an approximation for systems where a spacecraft (assumed massless) is in a Sun/Planet/Moon system. Thus, when one small body orbits another, and these two bodies orbit an even larger body. As with the CR3BP, these orbits are assumed to be circular. There are two forms of the BR4BP, one case where the rotating frame is the CR3BP frame of the Planet/Moon system, and the Sun is a time varying perturbing force in a circular orbit. The second form is that where the frame is the CR3BP frame for the Sun/Planet system, and the moon is a time varying influence in a circular orbit around the planet. For the Sun/Earth/Moon application, the latter is used since the application is focused on libration point orbits in the Sun/Earth system. In normalized coordinates, the equations of motion are [Koon08]:

$$\dot{x}=u\,,\dot{y}=v\,,\dot{z}=w$$
$$\dot{u}=x+2\mathrm{v}-c_S(x+\mu)-c_E(x-\mu_S)-c_M(x-x_M)$$
$$\dot{v}=y-2u-c_S y-c_E y-c_M(y-y_M)$$
$$\dot{w}=-c_S z-c_E z-c_M z$$

where

$$c_S=\frac{\mu_S}{r_S^3}\,,c_E=\frac{\mu_E}{r_E^3}\,,c_S=\frac{\mu_M}{r_M^3}$$

and

$$r_S=\sqrt{(x+\mu)^2+y^2+z^2}$$
$$r_E=\sqrt{(x-\mu_S)^2+y^2+z^2}$$
$$r_M=\sqrt{(x-x_M)^2+(y-y_M)^2+z^2}$$

and

$$\mu=3.035910E-06$$
$$\mu_S=1-\mu$$
$$\mu_M=3.7339987346E-08$$
$$\mu_E=\mu-\mu_M$$
$$x_M=a_M\cos(\theta_M)$$
$$y_M=a_M\sin(\theta_M)$$
$$\theta_M=\theta_{M0}+\omega_M t$$
$$\omega_M=12.36889493$$
$$a_M=2.573565074$$

Note that the BR4BP provides a very simple way to include Solar Radiation Pressure due to the explicit expression of the location of the Sun.