The Pennsylvania State University

The Graduate School

College of Engineering

USING NUMERICAL OPTIMIZATION TECHNIQUES AND GENERAL PERTURBATION EQUATIONS TO FIND OPTIMAL NEAR-EARTH ORBIT TRANSFERS

A Thesis in

Aerospace Engineering

by

Patrick S. Williams

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2009

The thesis of Patrick S. Williams was reviewed and approved* by the following:

David B. Spencer
Associate Professor of Aerospace Engineering
Thesis Advisor

Robert G. Melton
Professor of Aerospace Engineering

George A. Lesieutre
Professor of Aerospace Engineering
Head of the Department of Aerospace Engineering

*Signatures are on file in the Graduate School.

ABSTRACT

With recent developments in low-thrust optimization techniques, several methods of trajectory optimization can be implemented across various transfers. Much recent work has focused on performing methods of optimal control on low-thrust, near-Earth orbit transfers, achieving maximum efficiency in both energy use and time-of-flight. However, the use of optimal control relies on the exploitation of a satellites' equation of state, which becomes problematic if optimization is to be performed through a "black box" venue, where state equations cannot be manipulated. This situation is particularly evident when attempting to perform trajectory optimization through a commercial off-the-shelf satellite mission modeling software package. Thus, a robust optimization method must be chosen to produce competitive results comparable to optimal control in these types of situations. However, the formulation of an objective function, as well as which type of optimization method to chose is important, since some formulations or algorithms may lead to better or faster convergence when compared to others. To address this issue, several non-linear constrained numerical optimization methods, including classical algorithms and evolutionary strategies, are implemented on simple low-thrust trajectories modeled using Satellite Toolkit's Astrogator®, to determine what methods perform better when applied to these transfers. Once a suitable algorithm is selected, an appropriate objective function and problem formulation based on general perturbation equations is created, which can construct a time-varying thrust vector without manipulating a satellite's equation of state. The objective function is then optimized within the selected algorithm in an attempt to obtain an optimal LEO to Molniya trajectory which can produce results on par with those found using a method of optimal control. Initial testing

has shown that when an evolutionary strategy is applied to the objective function created in these studies, the resulting optimal trajectories are in fact competitive with those found using optimal control.

Table of Contents

List of Tables

List of Figures

AKNOWLEDGEMENTS

I would like to thank several people and organizations who helped me in this study. First, I would like to thank Analytical Graphics Inc. for not only providing me with the software used to complete this thesis, but also with the funding which made obtaining a Masters of Science possible. In particular, I would like to thank Dr. James Woodburn and Frank Linsalata for organizing the funding mechanisms for this research, as well as Matt Berry, Paul Black, and the AGI support team for helping me get the most out of Satellite Toolkit and Astrogator.

I would like to thank my faculty advisor, Dr. David Spencer, and fellow graduate students for their continuous support in this research. I would like to thank the Pennsylvania State University for providing classes and resources which aided in these studies.

I would also like to thank my family, for which their continuous support was greatly appreciated.

Finally, I would like to thank the Brown, Brook and Rainbow Trout which inhabit various central Pennsylvania limestone streams, for without their gracious acceptance of a well presented fly, I could not justify the arctic State College winters.

Chapter 1: Introduction

A spacecraft with a low-thrust propulsion system can achieve numerous types of orbit transfers. Specifically, long duration low-thrust maneuvers to drastically change certain orbital parameters are of primary concern. These transfers could take the form of simply increasing a spacecraft's altitude, or sending a low Earth orbiting spacecraft into a GPS or geosynchronous orbit. While obtaining the proper transfer orbit may consist of a relatively simple finite thrusting maneuver, there are various combinations of engine types, thrust magnitudes and durations which can lead to an almost infinite number of solutions. Thus, it is critical that these transfers be optimized to achieve maximum efficiency in their propellant use.

Since the 1960's much emphasis has been placed on optimizing near-Earth orbit transfers using optimal control [2], and specifically, primer vector theory [3]. These methods involve defining additional state variables (referred to as costates, or adjoint variables) which are integrated along with the state equation, in an effort to converge to a final state which minimizes some cost function. In primer vector theory, these costates are also used to create both a direction of thrust, as well as a "switching function" which dictates times of zero, maximum, or intermediate thrust arcs [3]. The result in both methods is an optimal or near-optimal trajectory between the initial and final states of some orbit transfer.

With the ever growing popularity of commercial satellite mission modeling software, where numerical integration of satellite's equations of motion are performed within the software itself, denying a user the ability to manipulate the state equations, the use of optimal control becomes problematic. This type of numerical integration can be

referred to as a “black box” method of satellite modeling. In this situation, adding costates and performing the necessary numerical integration of those costates may not be possible. Therefore, a method of trajectory optimization must be developed which does not rely on assessing the state equations, but can optimize some cost function for a low-thrust transfer, while staying competitive with other methods of optimal control.

For these studies, low thrust trajectories are modeled in Satellite Toolkit’s Astrogator® to which the optimization algorithms are connected. Astrogator serves as the black box framework to which this new method of trajectory optimization is implemented and tested. To achieve an optimal solution, a non-linear constained numerical optimization algorithm is implemented in accordance with general perturbation equations to optimize the change in certain orbital elements, and thus create a propellant-optimal (and change in velocity-optimal) solution, which also meets final orbital element constraints. Using this commercial satellite mission modeling software package, an optimal solution can be achieved by means of selecting a type of propulsion system, then altering the amount of engine thrust, duration of that thrust, and some initial orbital element values to achieve a desired final orbit while also obtaining a propellant optimal solution.

However, just as there are many possible solutions to each low-thrust transfer problem, there are also varying methods of which to converge to an optimal solution. As is often the case, optimization of these mission parameters is performed in accordance with non-linear constraints imposed on the spacecraft properties and/or its orbital elements, in conjunction with bounds on the control parameters. Therefore, a robust optimization method which can handle non-linear equality constraints must be chosen to

properly address these issues. However, which type of optimization method to chose is important, since some algorithms may lead to better or faster convergence when compared to others. To address this issue, tests which compare a classical Sequential Quadradic Programming method (implemented in Matlab®'s Optimization Toolbox [4]) and an evolutionary strategy (taken from Hansen's Covariance Matrix Adaptation Evolutionary Strategy [CMA-ES] Matlab® source code [7]) are conducted. The transfer for these algorithm test cases will be that of a LEO to 10,000 km semimajor axis with inclination change low-thrust transfer. Selection criteria of the best algorithm will be based upon a combination of the consistency of convergence, the optimal solution found by the algorithms, the effectiveness of constraint handling, and the overall convergence time.

Once the best algorithm is chosen, it is implemented on an objective function based upon using general perturbation equations to create switching functions, which will dictate the engine burn and coast times. Testing of these switching functions will be done by means of optimizing a LEO to Molniya low-thrust transfer, with a fixed total transfer time. These tests will show how the switching functions may be reformulated in order to achieve the most optimal low-thrust trajectory. Additionally, these tests will continue to validate the robustness and effectiveness of the selected numerical optimization algorithm.

Finally, once reformulations are made to the switching functions, comparisons are made to a LEO to Molniya transfer using optimal control and a higher order collocation method studied by Herman and Spencer in 2002 [2]. This comparison will attempt to determine how competitive the methods used in these studies are to the proven

methods in optimal control. All of the results from the testing of the algorithms, the testing of the switching functions, and the comparison with the method of optimal control will then be analyzed, to which future work will be suggested.

Therefore, the contents of this thesis will be presented as follows. First, an explanation of the optimization algorithms (SQP and CMA-ES) will be given as to their implementation, objective function creation, and constraint handling. Additionally, the connection between these algorithms and STK will be described, along with the basic modeling of low-thrust trajectories through STK. Next, a discussion of problem formulations for the testing of the optimization algorithms (circle to circle transfer) and the use of general perturbation equation-based switching functions (LEO to Molniya transfer) will be discussed. This section will include attitude, engine, and force model definitions for conducting these tests, as well as a detailed derivation of how to apply general perturbation equations into various switching functions. Finally, results from the tests will be presented which not only show not only the performance of the optimization algorithms, but also the resulting optimal trajectories, when compared to a method of optimal control. Upon discussion of these results, summations and final conclusions will be drawn, and further work will be presented.

Chapter 2: Optimization Algorithms

The objective of this study is to find optimal trajectories for near-Earth, low-thrust transfers. To achieve this, many have used various methods of optimal control, a process which depends on exploiting a satellites state equation through the addition of costates [3, 13]. However, since satellite trajectory modeling is done though a “black box” numerical integrator in this study, the state vector cannot be manipulated and thus traditional forms of optimal control cannot be used. Due to this fact, optimization must be done in more of a direct method approach in which a function of the satellites trajectory is optimized through the use of a numerical optimization algorithm.

For this study, it is desired to optimize a satellites amount of propellant used in a low-thrust transfer (which also translates to an optimal change in velocity). Depending on the type of optimization algorithm used, the propellant used would be incorporated in some part of an objective function. The objective function represents a function to which an extrema (for these studies, a global minimum) is to be found, and it is evaluated based on some vector of control values, $\vec{u}$.

Additionally, the final trajectory should result in a convergence to certain final orbital elements, which will serve as equality constraints imposed on some optimization algorithm. Depending on the numerical optimization algorithm used, these constraints may be incorporated into the objective function, or used to dictate a search direction within an iteration of the algorithms themselves. These constraints will also be evaluated based on the control vector $\vec{u}$, which may or may not vary linearly with respect to the objective function or other equality constraints.

Due to these needs, a robust nonlinear constrained numerical optimization algorithm must be chosen in order to optimize the propellant used through the transfer, while simultaneously converging to various non-linear equality constraints. This optimization problem could thus be viewed as

Find:

$$\text{Min} \quad F(\vec{u})$$

Subject to:

$$C(\vec{u}) = 0$$

Where $F(\vec{u})$ is the objective function, or function desired to be minimized, and $\vec{u}$ is a vector of independent variables of $F(\vec{u})$, such that $\vec{u} = [u_1, u_2, ..., u_n]$. $C(\vec{u})$ is an [*m* x 1] equality constraint vector where *m* is the number of constraints.

While many types of nonlinear constrained numerical optimization methods exist, two algorithms will be selected for these studies. These algorithms will be a gradient based Sequential Quadratic Programming (SQP) method, implemented in Matlab's Optimization Toolbox [4], and A Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) for which source code was provided by Nikolous Hansen [7]. The SQP method represents a very popular type of efficient classical optimization algorithm, while the CMA-ES algorithm represents a more computationally expensive evolutionary algorithm. Both algorithms have their own strengths and weaknesses, which this study will seek to expose when applied to creating propellant optimal solutions for low-thrust transfers.

Sequential Quadratic Programming Algorithm

In a classical non-linear optimization method, a numerical algorithm will be used to minimize $F(\vec{u})$ while also satisfying the constraints in $C(\vec{u})$. These algorithms vary in their approaches to minimize the objective function, but they all seek the same general convergence criteria. These criteria generally include (but are not limited to)

i) $\left|F(\vec{u}_k) - F(\vec{u}_{k-1})\right|$ is sufficiently small, thus $\vec{u}_k = \vec{u}^*$

ii) $C(\vec{u}^*) = 0$

iii) $g(\vec{u}^*) = 0$

iv) $\lambda_i^* \geq 0$ For $i = 1,2,...,n$

where $\vec{u}^*$ is the optimum value of $\vec{u}$, k denotes an iteration of the algorithm, $g(\vec{u}^*)$ is the gradient of $F(\vec{u})$ evaluated at $\vec{u}^*$, and λ_i^* represent the optimum Lagrange Multipliers, which are a [n x 1] single column matrix, with n being the number of equality constraints [8].

An SQP method in particular seeks to convert a standard non-linear constrained optimization problem into a quadratic programming format, and solved using a standard quadratic programming sub-problem [8]. This method is implemented by first converting $F(\vec{u})$ to a quadratic function $q_k(\delta)$, giving:

$$q_k(\delta) = F(\vec{u}_k) + g(\vec{u}_k)^T \delta + \frac{1}{2}\delta^T W_k \delta \qquad (1)$$

where

$$W_k = \nabla^2 F(\vec{u}_k) - \sum_i \lambda_{k,i} \nabla^2 C_i(\vec{u}_k) \qquad (2)$$

and δ is the difference in control variables between iterations, such that

$$\vec{u}_{k+1} = \vec{u}_k + \delta_k \tag{3}$$

Next, the active constraint functions, $C(\vec{u})$ are adjusted to fit the SQP layout, giving

$$A(\vec{u}_k)^T \delta + C(\vec{u}_k) = 0 \tag{4}$$

where $A(\vec{u}_k)$ represents the constraint vector Jacobian evaluated at iteration *k*. Finally, a standard SQP system can be written as:

$$\begin{bmatrix} W_k & -A(\vec{u}_k) \\ -A(\vec{u}_k)^T & 0 \end{bmatrix} \begin{Bmatrix} \delta_k \\ \lambda_{k+1} \end{Bmatrix} = \begin{Bmatrix} -g(\vec{u}_k) \\ C(\vec{u}_k) \end{Bmatrix} \tag{5}$$

Through iteration of the system of equations in *(5)*, the next iteration of the control vector can be found using equation *(3)*. This system can be iterated until a suitable value for $\vec{u} = \vec{u}^*$ is found. Also, while the above system can give an iterative process for λ, as long as λ_1 is given, it is advised that the value for λ be solved exactly at each iteration, given by the formula:

$$g(\vec{u}_k) = A(\vec{u}_k)\lambda_k \tag{6}$$

which represents the definition of the Lagrange Multiplier. The reason for determining the value of λ at each iteration by equation *(6)*, and not by the SQP system iterative process is because the SQP system value of λ can lead to a final $\vec{u} = \vec{u}^*$ value which is not a local minimizer [8]. A model algorithm for the SQP method can thus be viewed as follows

Set k = 1, and given $\vec{u}_1 \ldots$

i) Compute λ_k, W_k and solve the SQP system
ii) Set $\vec{u}_{k+1} = \vec{u}_k + \delta_k$, and set $k = k + 1$
iii) If convergence criteria has been met, Terminate, Else, go back to i)

Eventually, an algorithm similar to this will converge to an optimal value of $\vec{u}$. Aside from understanding how the SQP method works, it is important to understand the role gradients have on these algorithms. In terms of convergence criteria, simply looking for a location where the gradient is equal to zero may result in finding on a local rather than a global minimum. Thus for multi-modal function spaces, a classical algorithm may not yield a globally optimal solution, and extensive search of the function space would be necessary to obtain such a solution. Secondly, the use of gradients for defining search directions between iterations indicates that function spaces must be continuous, with continuous derivatives. Therefore, classical algorithms such as this SQP method may fail when preformed over complex function spaces.

Covariance Matrix Adaptation Evolutionary Strategy

The Evolutionary Strategy chosen for the proposed optimization problem is a Matlab version of the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) written by Nikolous Hansen. Evolutionary strategies differ from classical algorithms in that they do not rely on the calculation of gradients for their search direction, thus they can operate in function spaces with discontinuities, singularities, or discreteness. Additionally, they search an extensive portion of a function space, and are not directed to find the closest minimum like classical algorithms. Instead, they take many function evaluations, and slowly "evolve" through "generations" of the control variable vector to

an optimal point. This leads evolutionary strategies to be very computationally expensive, but also more robust and more inclined to converge to global extrema than classical algorithms.

The CMA-ES method can be summed up as a real-valued evolutionary strategy, which optimizes some objective function generally viewed to be a black box. Just as a quasi-newton method uses an estimation of a hessian matrix to adapt a convex problem to a quadratic one, the CMA-ES algorithm uses a careful estimation of a covariance matrix to adapt a convex-quadratic function into a spherical one [7].

The algorithm uses a randomized search of the objective function space to obtain a search distribution, which conveniently does not require any direct calculations of gradients or hessians. The covariance matrix of this search distribution is used to calculate the normal distribution, which is defined by the distribution containing the largest entropy in the function space. This normal distribution is then used to calculate each new set of offspring, along with certain selection, recombination and auto adapted Gaussian recombination methods. These are create a self-adapting step sized based on the evolutionary path history, and is why this algorithm can be viewed as an evolutionary strategy [7].

Selection and recombination are implemented primarily by ranking each individual for each generation and placing a specific weight on them, known as recombination weights. These weights can either be distributed superlinearly, linearly, or equally, with superlinear and linear distributions placing higher weights on the fittest individuals. The summation of each individual and their respected weight are then used to calculate the children of the next generation. Thus, if a superlinear or linear

recombination method is selected, the fittest individuals will have more prominence in determining the next generation [7].

Next, by using the number of parents, μ, population size, λ, variables of the previous generation, and their respective weights, a covariance matrix is calculated. This matrix is then manipulated first by a rank-μ-update (ensures the next generation step will move steadily towards the optimum) and rank one update (which ensures the covariance matrix is updated based on a desired evolution path) in order to drive the next search distribution in the general direction of the fittest individuals, while still obtaining a broad enough distribution to avoid premature convergence. Thus, the rank-μ-update and rank one update of the covariance matrix are combined to a new estimate of that matrix, which is used to calculate the normal distribution and ultimately the next generation of individuals. A step size limitation is also placed on the covariance matrix, to ensure that it doesn't result in a step which would converge too rapidly to a local minimum [7].

The CMA-ES algorithm avoids the pitfalls of many genetic and evolutionary algorithms by its ability to handle poorly scaled problems, problems with diverse arrays of population and parent sizes, and avoids premature convergence by limiting the step size taken at each iteration.

However, one obstacle in the CMA-ES algorithm is that it does not handle equality constraints directly. Therefore, a penalty function method is applied to the objective function, in which the discrepancy between a current value of some nonlinear constraint is subtracted from its desired value, and multiplied by a scalar weight [9, 10]. These weights vary in scale for the different orbital element constraints, so as to equalize their contribution to the objective function. If the weights are too high, the algorithm may

pre-converge to satisfying a particular equality constraint, instead of minimizing the propellant used. If the weight is too small, then the equality constraint may not be satisfied to a desired tolerance, or even satisfied at all, upon convergence of a final solution.

A simple example of how a penalty function works is to take the minimization of the function $f(x) = x - 5$ with the equality constraint $x = 1$. This would make the objective function represented by

$$F(x) = (x-5) + \sigma \mid x-1 \mid \tag{7}$$

Where σ represents a penalty weight. The effect of this type of penalty function formulation can be viewed in Figure 1, where it is obvious that the minimum is located where the equality constraint is satisfied. It should be noted that the fact that $F(x)$ has an undefined derivative at the location of the minimum does not matter since evolutionary strategies do not require continuous derivatives to function.

Taking the penalty function formulation into account, the objective function for a propellant optimal trajectory can now be viewed as

$$F(\vec{u}) = Propellant\ Used + \vec{w}\vec{C}^{T} \tag{8}$$

Where the $\vec{w}$ vector represents a penalty weight for each equality constraint, and $\vec{C}$ is a vector of equality constraints. For example, the $\vec{C}$ element for a theoretical final orbit argument of periapsis constraint would be viewed as

$$C_{\omega} = \left| \omega_{current} - \omega_{final} \right| \tag{9}$$

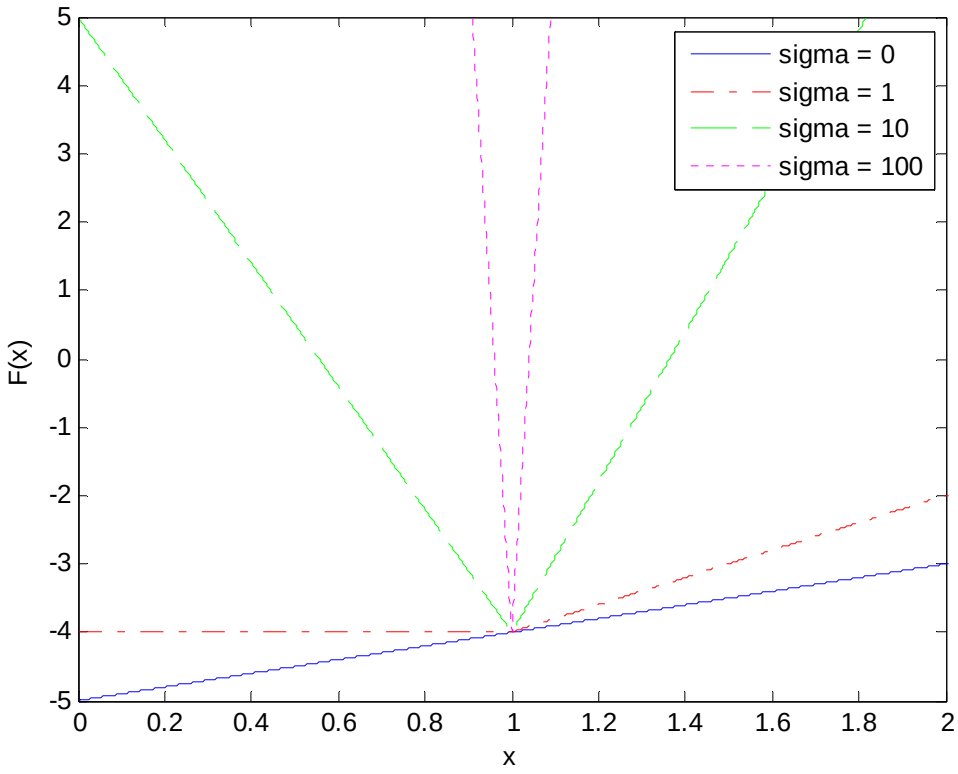


Figure 1: Plot of equation (7) for σ = 0, 1, 10, and 100

Both the SQP and CMA-ES algorithms will be subjected to objective functions and constraints tailored to various low-thrust trajectories. Upon their convergence to an optimal set of control variables, $\vec{u}^*$ for each transfer, these algorithms will be compared in their overall optimal value, time to convergence, and other performance characteristics. However, objective functions, constraint functions, and control variables for these algorithms will be defined within STK. Therefore, before any optimization can be conducted, a method for modeling these trajectories within STK, as well as connecting both the SQP and CMA-ES algorithms to the STK Astrogator interface must be created.

Chapter 3: Commercial Software

For these studies, a commercial software package will be used to model low-thrust transfers and integrate a satellites equation of motion. When choosing the correct software package to accomplish this, there are many considerations which must be made. Primarily, the software should be industry proven so that results can be trusted as being accurate and correct. Secondly, the software should be versatile in both its ability to model multiple low-thrust scenarios, as well as allow for a user to customize force models, attitude definitions, and numerical integration algorithms tailored to various transfers. Finally, the commercial software needs to have some venue in which an external optimization algorithm can be connected in order to perform the necessary optimization of objective function control variables. Due to these reasons, Satellite Toolkit (STK) was chosen as the mission modeling software package for this thesis work. STK is a robust, industry proven off-the-shelf mission modeling software package, which has an extensive software plugin structure (software code written by user), in which the user can customize STK's force models and satellite properties for various transfer scenarios.

Satellite Toolkit and Astrogator

All low-thrust scenarios will be designed using STK's Astrogator ®, a facet of STK that can model various mission segments, a few of which can be described as follows:

Initial State: Represents the initial orbital elements, position and velocity vectors, and spacecraft properties [5].

Propagate: Given initial spacecraft parameters and force model (e.g. two-body), a propagate segment will iterate through a user-specified numerical integration algorithm to determine final position, velocity, orbital elements, etc, of the spacecraft at some user defined stopping condition. A stopping condition could be a specified time duration, date, location in an orbit, or instant some orbital parameter is met a set number of repetitions [5].

Maneuver: Applies user specified inputs for spacecraft attitude, direction and magnitude of thrust vector (can be constant or time-varying) and stopping condition for maneuver (if it is not impulsive). The specified attitude and thrust model (referred to as engine model in STK) will be applied to a spacecraft until the stopping condition is met. The attitude and engine models can either be specified using pre-defined models in STK, or customized by a user using an attitude or engine plugin [5].

Targeting Sequence: Spacecraft or orbit parameters in propagate or maneuver segments can be set as independent variables (controls) and dependant variables (results) which can be placed in some sort of "search" algorithm. This algorithm will perturb the controls and output the results bases on those perturbations. Should the user want to implement their own search algorithm, such as an orbit tuner, parameter search, or optimization algorithm, they can do so through the use of a search plugin [5].

Using these various Astrogator segments, a satellite can be created with some initial state, to which a maneuver is added which transfers it to a specified final state. Control values within the initial state and maneuver segment can be placed inside a targeting sequence which contains a search plugin, connecting STK to an optimization

algorithm. The optimization algorithm can then perturb various control values until an optimal trajectory from the initial to final state has been created.

However, while commercial mission modeling software can be very useful in modeling satellite trajectories, it also has its challenges in that the user may be subject to the limitations of the software itself. STK was not immune to some of these challenges (inability to pass global variables between plugins, for example), primarily version 8.1, the original version used in this study. These challenges made modeling a practical (or real-world applicable) low-thrust transfer difficult within STK 8.1, although conceptual low-thrust transfers could be modeled. Therefore, the testing for determining the effectiveness of the optimization algorithms and perturbation equation-based problem formulations were conducted in this version, since these test cases did not need to reflect practical transfers to give valuable results. Once the best optimization algorithm and problem formulation are chosen based on these results, they can be applied in STK version 9.0, where practical low-thrust trajectories are easier to model. Thus, while the initial testing is conducted in STK 8.1, the final applications are conducted using STK 9.0.

Connection with Optimization Algorithms

Using the Astrogator sequences described perviously, Matlab coded optimization algorithms can be applied to a targeting sequence by means of a custom software plugin. This plugin was taken from an AGI VBScript search plugin template, and modified to fit the needs of these experiments [5]. The VBScript file connects to Astrogators search interface, which represents the targeting sequence mission control segment (MCS) in the

Astrogator Graphical User Interface (GUI), and passes certain STK object classes, as well as values for control parameters, constraints functions and the objective function into the Matlab workspace. Once here, an initial Matlab function will handle the object classes and data, and pass numeric values into separate Matlab based constraint functions and objective functions, which will be used within an optimization algorithm. The initial Matlab function then initiates an optimization algorithm (either the Matlab Optimization Toolbox or CMA-ES algorithm for these studies), and while it iterates, numeric data will be passed back and forth between STK, the VBScript file, the initial Matlab function, the constraint function, the objective function, and the optimization algorithm. This process continues until the algorithm meets some termination criteria, and an optimal trajectory is found. A diagram of this integration process can be seen in Figure 2.

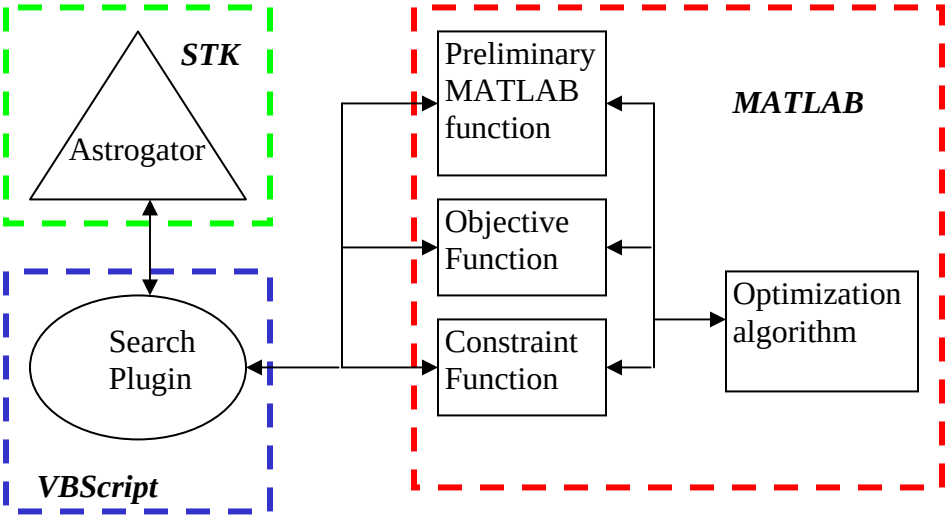


Figure 2: Diagram of STK integration with VBScript and MATLAB

As Figure 2 illustrates, both a method for creating low-thrust trajectories, and connecting an external optimization algorithm to STK have been established. Therefore, the next iteration into the optimization process is to derive various problem formulations

the algorithms will be implemented on. These problem formulations will define spacecraft attitude, force models (including thrusts), initial and final states, which will in turn create a control vector, objective function, and constraint functions to be inputted into an optimization algorithms. These problem formulations will seek two primary objectives, in that they will help determine which optimization algorithm is best suited for solving low-thrust trajectory problems, as well as how to use these algorithms coupled with general perturbation equations to create an optimal LEO to Molniya low-thrust trajectory.

Chapter 4: Low-Thrust Problem Formulations

Low-thrust transfers have been studied extensively in recent decades, primarily for their ability to reduce spacecraft weight, and to employ advanced propulsion techniques. With these advancements in propulsion technology, an engine can achieve a specific impulse on the order of thousands of seconds, leading to very low mass flow rates for a low-thrust engine (thrusts in these studies are on the order of $0 - 5$ N). However, this low thrust will also correlate with a long transfer time, as well as an almost infinite number of possible trajectories. Therefore, for some given transfer time, it becomes necessary to find the most propellant optimal (and correspondingly most change in velocity optimal) transfer from some given initial state to a given final state. Composing this process can be referred to as the problem formulation, and can be viewed two-fold.

First a low-thrust trajectory will be modeled using STK, and a specific objective function will be chosen which seeks to minimize the propellant use while satisfying some final state orbital elements. Control variables in the objective function will change the magnitude and/or direction of the satellites thrust vector, resulting in separate trajectories, representing separate final states and usage of propellant.

The second tier of this approach consists of implementing a non-linear constrained optimization algorithm on this objective function to converge on an optimal trajectory. However, which type of optimization algorithm to choose is unknown, and initial problem formulations must therefore be designed to test the overall robustness of both the SQP and CMA-ES algorithms. Since these preliminary problem formulations are designed simply test the effectiveness of implementing the SQP and CMA-ES algorithms

within a black box framework, they can be relatively rudimentary concerning their applicability to real-world low-thrust trajectories. Thus, a simple problem formulation based on modeling a basic LEO to 10,000 km semimajor axis circle to circle low-thrust transfer consisting of an inclination change will be chosen to complete this task.

Once the best optimization algorithm is chosen, a more complicated problem formulation can be created which seeks to find optimal low-thrust trajectories, producing results comparable to a method of optimal control. This more complicated problem formulation is based on modeling a real-world LEO to Molniya orbit, and will be based on manipulating general perturbation equations.

Creating Dynamics and Attitude Systems

In each problem formulation, a satellite with a mission control sequence is created in Astrogator, to which various mission parameters may be changed for each problem formulation. Each satellite will have initial state properties as defined for each separate problem formulation. Next, the satellite propagates until it reaches periapsis of its initial LEO orbit[*], at which point a thrust maneuver will occur. This maneuver resides within a targeting sequence, to which the VBScript plugin will connect and pass object classes, control variables, and results from the maneuver segment into the Matlab workspace and subsequently into an optimization algorithm. The maneuver will run for a finite amount of time, dictated by a stopping condition, and propagated using a two-body force system with an Runge Kutta 4(5) numerical integrator with a 5 second initial time step.

[*] Note, for a circular orbit, the periapsis can be defined as the point where true anomaly is equal to 0 degrees

Within this maneuver segment the direction of thrust is also specified for each problem formulation, with some being restricted to a particular spacecraft body axis, and others free to rotate in three-dimensional space. The body axis will serve as the reference to the satellite attitude system, which will either be fixed, or oscillating with respect to the satellite. The attitude systems for the various problem formulations are specified within Astrogator, or implemented through a custom attitude plugin, taken and modified from an AGI attitude plugin template [5].

The thrust vector will be created in reference to the particular attitude system selected for the various problem formulations, with its magnitude and direction being controlled by an engine model. Each engine model is created in STK, and applies a constant user-specified thrust, and specific impulse. These engines can either be created from templates in Astrogator, or customized by means of custom engine plugins, taken and modified from and AGI engine plugin template [5].

Problem Formulations (Circle to Circle Transfer)

The first problem formulation in this study is based on a simple LEO to 10,000 km semimajor axis circle to circle low-thrust transfer with an inclination change. The primary objective of this problem formulation is to simulate a very simple low-thrust transfer to which an optimization algorithm can be applied. The problem formulation lacks complexity (and therefore much real-world applicability) because it is desired to isolate the performance of the optimization algorithms from that of the thrusting method. The method of applying thrust, which is presented as a switching function based on general perturbations, will be discussed later in the chapter.

For this simple problem formulation, the spacecraft will undergo a constant applied thrust (i.e. no coast times) from an initial LEO orbit, specified in Table 1, to a final orbit specified in Table 2.

Table 1: Initial state and satellite properties for LEO to 10,000 km semimajor axis with inclination change problem formulation

Spacecraft/orbit property	Value
Semi-major axis (a)	6678.14 km
Eccentricity (e)	0
Inclination (i)	10°
Right Ascension of Ascending Node (Ω)	0°
Argument of Periapsis (ω)	0°
True Anomaly (ν)	0°
Spacecraft Mass (m_f)	100 kg
Propellant Mass (m_{prop})	500 kg
Earth Specific Acceleration (g)	0.00980665 km/sec^2
Engine Specific Impulse (I_{sp})	4000 sec

Table 2: Final state for LEO to 10,000 km semimajor axis with inclination change problem formulation

Spacecraft/orbit property	Value
Semi-major axis (a)	10,000 km
Eccentricity (e)	0
Inclination (i)	0°
Right Ascension of Ascending Node (Ω)	Free
Argument of Periapsis (ω)	Free
True Anomaly (ν)	Free

It should be noted that for this problem, the final inclination and semimajor axis will serve as equality constraints imposed on the objective function. Since this problem formulation is based on how the optimization algorithms perform under simple conditions, the semimajor axis constraint is actually removed as an equality constraint and instead imposed as a stopping condition for the numerical integration (a second stopping condition of an 0.5 year duration will also be imposed). This will effectively

retain the semimajor axis equality constraint without imposing it on the objective function. This is done because imposing equality constraints in a penalty function method greatly complicates the optimization process. The optimization algorithms are thus tested under a simpler problem formulation, to which more complicated problem formulations (and more equality constraints) will be implemented in subsequent tests.

Throughout this transfer, the spacecraft attitude will be set to the velocity, normal, and co-normal ($\hat{V}\hat{N}\hat{C}$) axes system. In this system, the $\hat{V}$ axis is tangential to the spacecraft orbit, in the direction of velocity, the $\hat{N}$ axis is parallel to the angular momentum vector (out of orbital plane), and the $\hat{C}$ axis is normal to the spacecraft orbit. This axes system remains fixed with respect to the instantaneous direction of spacecraft motion, and can allow for thrust vectors which produce changes to both in-plane and out-of-plane orbital elements. A basic diagram of this attitude system is illustrated in Figure 3.

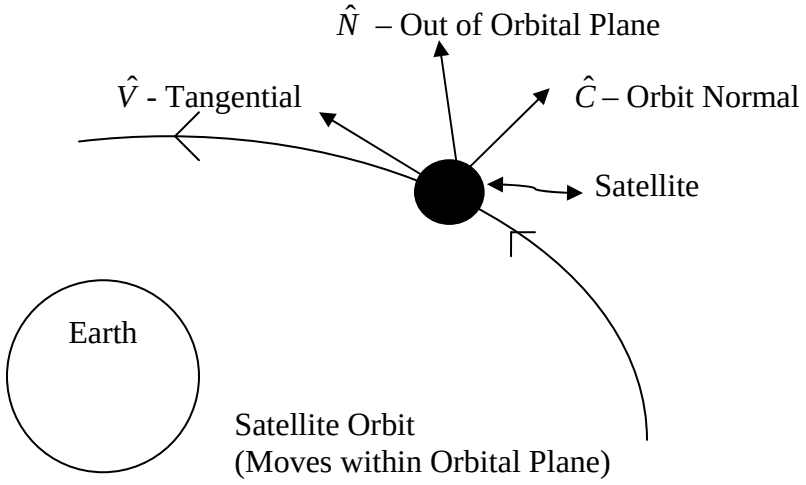


Figure 3: Basic diagram of spacecraft VNC axis.

Thrust within this attitude system will be controlled by a four-engine thruster set, with one thruster acting along the $\hat{V}$ axes, one along the $\hat{C}$ axis, and two acting along both the positive and negative $\hat{N}$ axis'. Each thruster in the set will have a constant specific impulse, as well as a constant thrust applied throughout the maneuver. The value for each thrust in each thruster will serve as four control variables for an optimization algorithm.

The two thrusters along the $\hat{N}$ axes are created to allow for a constant change in inclination throughout the orbit. Should there only thrust in one direction along the $\hat{N}$ axes, the inclination would simply oscillate with respect to time, and no inclination change would be possible. The axis of rotation for the inclination can be viewed as a line connecting the points of transition between ascending and descending portions of the spacecrafts orbit, as illustrated by Figure 4.

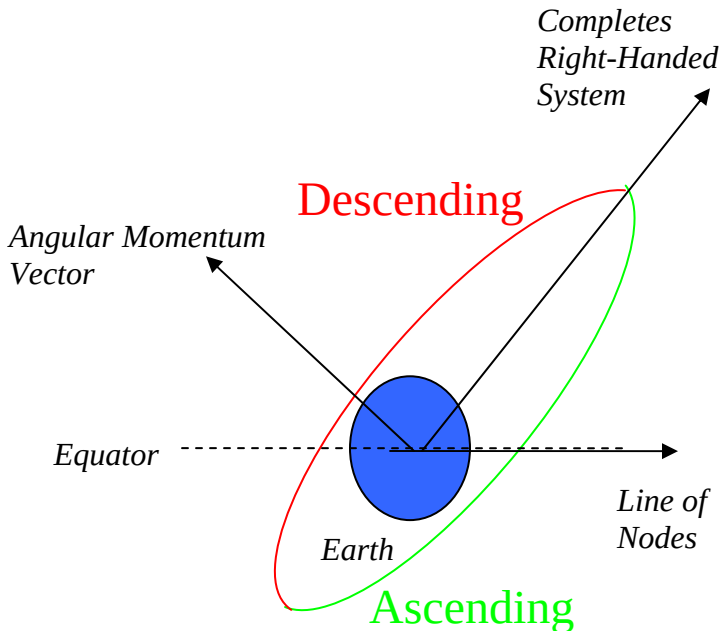


Figure 4: Diagram of ascending and descending portion of spacecraft orbit

Therefore, the out-of-plane thrust direction should be switched from positive to negative, or vice versa at each crossing of these transitional points. In this study, since the desired change in inclination is negative throughout the transfer, there will be one thruster acting in the positive $\hat{N}$ axes, which will be "on" while the spacecraft is ascending, and one in the negative $\hat{N}$ direction which will be "on" when the spacecraft is descending. This will create the desired continuous change in inclination, and not cause the aforementioned wobble effect.

Therefore, constant thrust and I_{sp} engines are created for the $\hat{V}$ and $\hat{C}$ thrusters, and custom engine plugins are created for the positive and negative $\hat{N}$ thrusters to achieve this desired thrusting behavior. The thruster set consisting of each $\hat{V}$, $\hat{N}$ and $\hat{C}$ direction thrusters can is illustrated in Figure 5.

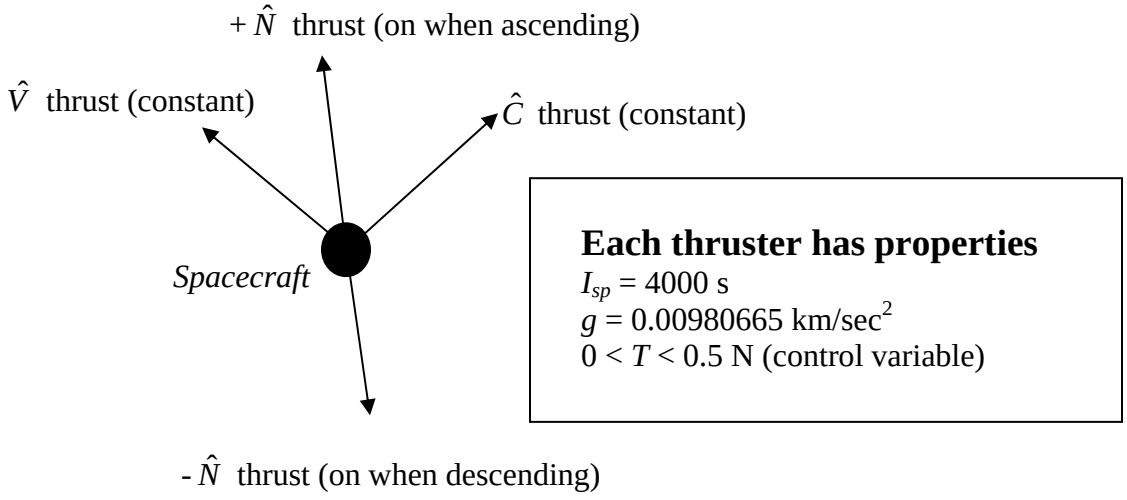


Figure 5: Diagram of spacecraft thruster set for circle to circle transfer problem formulation

Therefore, with the proposed attitude system, and engine thrust force models, the initial LEO to 10,000 km circle to circle low-thrust transfer with inclination change control vector can be viewed as

$$\vec{u} = < T_{\hat{V},+}, T_{\hat{C},+}, T_{\hat{N},+}, T_{\hat{N},-} > \quad (10)$$

where $T_{\hat{V},+}$, $T_{\hat{C},+}$, $T_{\hat{N},+}$, and $T_{\hat{N},-}$ represent the thrust components in the attitude system illustrated in Figure 3, applied to the four-engine thruster set illustrated in Figure 5. The control vector outlined in equation *(10)* is also a preliminary control vector for this particular problem formulation, and will be changed based on testing results in subsequent problem reformulations. Additionally, for this problem formulation, the following bounds bounds are imposed on each control variable

$$0 < T_{\hat{V},+} < 0.5\ N$$

$$0 < T_{\hat{C},+} < 0.5\ N$$

$$0 < T_{\hat{N},+} < 0.5\ N$$

$$0 < T_{\hat{N},-} < 0.5\ N$$

Using the objective function, control vector, attitude definition and engine thruster set, the proposed problem formulation is implemented within both the SQP and CMA-ES algorithm to see which algorithm produces more desired results. Certain problem reformulations will be made in an attempt to expose weaknesses, or display observed strengths of both algorithms to aid in the selection of the best technique. Once an algorithm is chosen, a problem formulation based on using general perturbation equations is created, which when optimized, will lead to results competitive with those found using one method of optimal control.

Use of General Perturbation Equations

The preliminary problem formulation in these studies is created to aid in the selection of an optimization algorithm which performs well on low-thrust trajectories within a black box numerical integration framework such as STK. Once this algorithm is selected, it is then necessary to alter the problem formulation so that the optimal trajectories created by this formulation could produce results competitive with methods of optimal control. In particular, this problem formulation should seek to create one or more switching functions which control burn and coast times of a low-thrust arc. This problem formulation will be used to optimize a low-thrust LEO to Molniya transfer, to which results will be compared to a LEO to Molniya transfer created using methods of optimal control performed by Herman and Spencer [2].

As stated previously, since STK does not allow for the manipulation of the state vector, the use of costates and a traditional primer vector theory type switching function is not possible [3]. However, using general spacecraft perturbation equations, a new type of switching function can be created which does not rely on evaluation of costate variables. This switching function relies on applying maximum spacecraft thrust when the change in certain orbital elements are at a maximum. In the case of a LEO to Molniya orbit transfer, it is the semimajor axis (a), eccentricity (e), inclination (i) and argument of periapsis (ω) which must be altered to go from an initial LEO to a final Molniya orbits within a specified time of flight. Initial conditions for a LEO orbit and final conditions for a Molniya orbit are given in Tables 3 and 4 respectively. The equations for the derivatives with respect to time of these elements are given by Vallado [1] as

$$\frac{da}{dt} = \frac{2e\sin(\nu)F_R}{n\sqrt{1-e^2}} + \frac{2a\sqrt{1-e^2}F_S}{nr} \qquad (11)$$

$$\frac{de}{dt} = \frac{\sqrt{1-e^2}}{na}\left[\sin(\nu)F_R + \left(\cos(\nu) + \frac{e+\cos(\nu)}{1+e\cos(\nu)}\right)F_S\right] \tag{12}$$

$$\frac{di}{dt} = \frac{r\cos(u)F_W}{na^2\sqrt{1-e^2}} \tag{13}$$

$$\frac{d\omega}{dt} = \frac{\sqrt{1-e^2}}{nae}\left[-\cos(\nu)F_R + \frac{2+e\cos(\nu)}{1+e\cos(\nu)}\sin(\nu)F_S\right] - \frac{r\cot(i)\sin(u)}{h}F_W \tag{14}$$

Where n is the mean motion, ν is the true anomaly, u is the mean longitude ($u = \nu + \omega$), and F_S, F_R and F_W represent the thrust acceleration components in the $\hat{R}$ (radial), $\hat{S}$ (transverse), and $\hat{W}$ (orbit normal) axes respectively. For this problem formulation, the $\hat{R}$, $\hat{S}$, $\hat{W}$ system will be used in accordance with the nomenclature of Vallado [1], illustrated in Figure 6. It should be noted that this system is typically referred to as the radial-transverse-normal ($\hat{R}, \hat{T}, \hat{N}$) system.

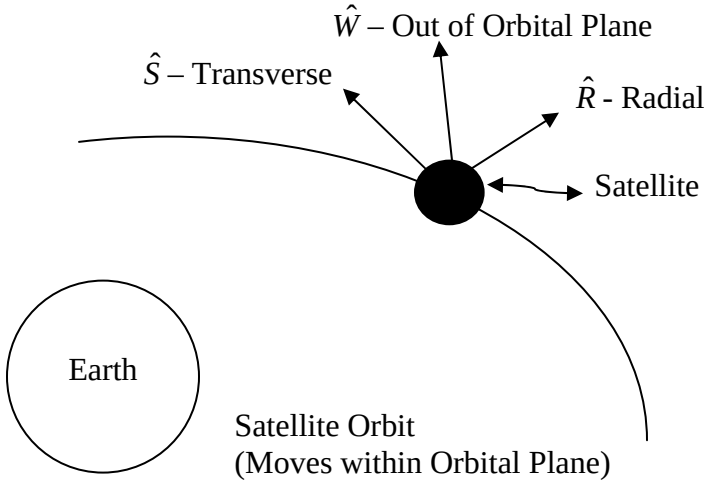


Figure 6: Basic diagram of spacecraft RSW axis.

Table 3: Initial state and satellite properties for LEO orbit

Spacecraft/orbit property	Value
Semimajor axis (a)	6678.14 km
Eccentricity (e)	0
Inclination (i)	28.5°
Right Ascension of Ascending Node (Ω)	Free
Argument of Periapsis (ω)	Free
True Anomaly (ν)	0°
Spacecraft Mass (m)	100 kg
Propellant Mass (m_{prop})	500 kg
Earth Specific Acceleration (g)	0.00980665 km/sec^2
Engine Specific Impulse (I_{sp})	4000 sec

Table 4: Final state properties for Molniya orbit

Spacecraft/orbit property	Value
Semimajor axis (a)	26553.4 km
Eccentricity (e)	0.740969
Inclination (i)	63.4°
Right Ascension of Ascending Node (Ω)	Free
Argument of Periapsis (ω)	270°
True Anomaly (ν)	0°

Based on equations *(11) – (14)*, it can be deduced that in order to perform thrusting maneuvers representing positive changes in semimajor axis, eccentricity, and inclination, there needs to be an engine model constructed of a five-engine thruster set, with thrusters in each of the $\hat{S}$, $\hat{W}$, $\hat{R}$, $-\hat{W}$, and $-\hat{R}$ directions. Note that there is no thrust in the $-\hat{S}$ direction. This is because a propellant optimal solution will also be an energy optimal solution, in which the maximum change in energy should be achieved. Using the vis viva equation *(15)*, it can be shown in equation *(16)* that a maximum change in energy will correspond to a maximum change in semimajor axis, and thus equation *(11)* should remain positive.

$$\varepsilon = -\frac{\mu}{2a} \tag{15}$$

$$\frac{d\varepsilon}{dt} = \frac{\mu}{2a^2}\frac{da}{dt} \tag{16}$$

Since the $\hat{S}$ direction thrust corresponds to a positive value for F_S, and vice versa, thrust should therefore never be in the $-\hat{S}$ direction because it would result in a negative $\hat{S}$ component of equation *(11)*. It will be shown later that due to the sinusoidal behaviors of the $\hat{R}$ and $\hat{W}$ components of equations *(11)*, *(12)* and *(13)*, that thrust in both the positive and negative directions is necessary.

For a minimum propellant and correspondingly minimum change in velocity transfer, these orbital elements should remain on when their derivatives are at their maximum values, thus ensuring the most efficient transfer for a given maximum thrust. In an ideal situation, the most propellant optimal solution would be a series of impulsive thrusts located at the points of maximum change in *a*, *e*, and *i*, and would correspond to

the longest possible transfer time. It should be noted that creating thrusts at these points would result in changes to ω, a problem which will be taken into account later in the problem formulation. Using equations *(11) – (14)*, the true anomaly locations of maximum change in *a*, *e*, and *i* are derived, and given in Table 5.

Table 5: Locations of Optimal Change in a, e and i

Orbital Element	**Thrust Direction**	**Location of Greatest Change ($\frac{d}{dt}|_{\max}$)**
Semimajor Axis (a)	$\hat{S}$	$\nu = 0$
	$\hat{W}$	N/A
	$\hat{R}$	$\nu = \pi/2$
	$-\hat{S}$	N/A
	$-\hat{W}$	N/A
	$-\hat{R}$	$\nu = -\pi/2$
Eccentricity (e)	$\hat{S}$	$\nu = 0$
	$\hat{W}$	N/A
	$\hat{R}$	$\nu = \pi/2$
	$-\hat{S}$	N/A
	$-\hat{W}$	N/A
	$-\hat{R}$	$\nu = -\pi/2$
Inclination (i)	$\hat{S}$	N/A
	$\hat{W}$	$\nu = \sin^{-1}[-e\sin(\omega)] - \omega$
	$\hat{R}$	N/A
	$-\hat{S}$	N/A
	$-\hat{W}$	$\nu = \pi - \sin^{-1}[-e\sin(\omega)] + \omega$
	$-\hat{R}$	N/A

Using results from Table 5, equations *(17) – (21)* can be derived which show the maximum possible values of equations *(11) – (14)* can obtain for a given set of orbital elements along each of the $\hat{S}$, $\hat{W}$, and $\hat{R}$ axes.

$$\frac{da}{dt}|_{\hat{S},\max} = \frac{2a^{3/2}(1+e)F_S}{\sqrt{\mu(1-e^2)}} \qquad (17)$$

$$\frac{da}{dt}\Big|_{\hat{R},\max} = \frac{3ea^{3/2}F_R}{\sqrt{\mu(1-e^2)}} \tag{18}$$

$$\frac{de}{dt}\Big|_{\hat{S},\max} = \frac{2\sqrt{a(1-e^2)}F_S}{\sqrt{\mu}} \tag{19}$$

$$\frac{de}{dt}\Big|_{\hat{R},\max} = \sqrt{\frac{a(1-e^2)}{\mu}}F_R \tag{20}$$

$$\frac{di}{dt}\Big|_{\hat{W},\max} = \frac{\sqrt{\{a(1-e^2)\}[1-e^2\sin^2(\omega)]}F_W}{[1+e\cos(\omega)\sqrt{1-e^2\sin^2(\omega)}-e^2\sin^2(\omega)]\sqrt{\mu}} \tag{21}$$

Using equations *(17) – (21)*, plots can be made which show the contribution each thrusting force makes to changes in *a*, and *e* for $0 \le e \le 0.74969$ and $6678.14 \le a \le 26{,}553.4$ km. It should be noted that for the case of plotting equation *(21)*, since ω is considered a slowly changing variable with respect to *a* and *e*, and it results in an oscillation of equation *(21)*. From equation *(21)*, if a factor of $\sqrt{a}$ is removed, it can be seen that *di/dt* will obviously increase as *a* increases. By setting $a = a_{\max} = 26{,}553.4$ km, plots can be made of *di/dt* at various values of ω and *e*. Plots of equations *(17)* and *(18)* can be seen on Figure 7, plots of equations *(19)* and *(20)* on Figure 8, and finally a plot of the effects of changing ω and *e* on equation *(21)* in Figure 9. It should be noted that in each of these plots the thrust acceleration (either F_S, F_R or F_W) and $\mu^{-1/2}$ are factored out of equations *(17)* - *(21)* since they represent unchanging constants.

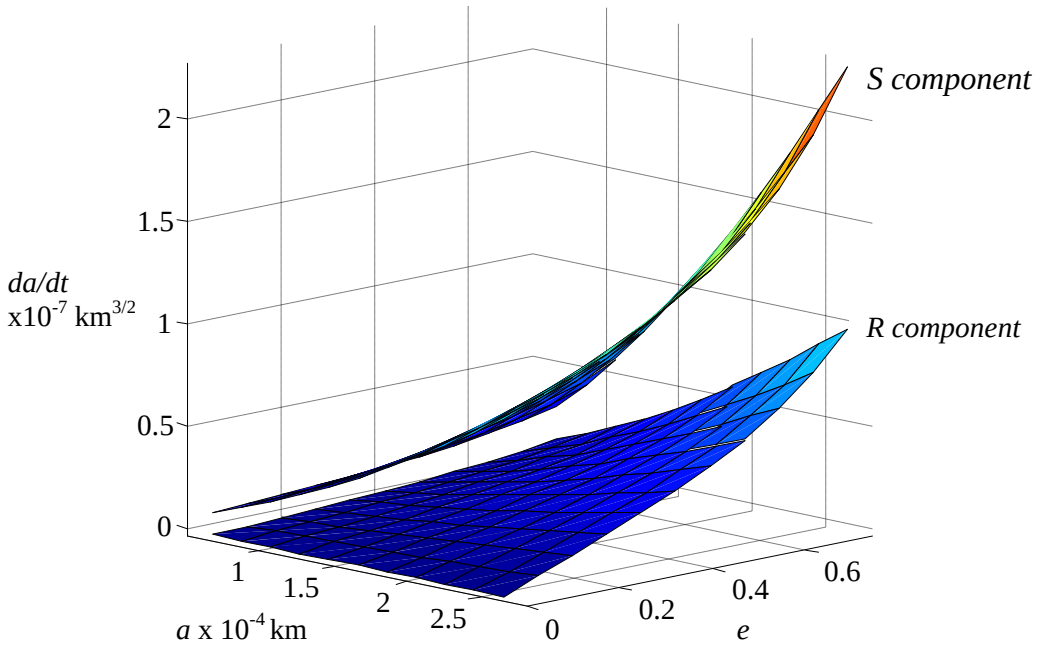


Figure 7: Plots of equations *(7)* and *(8)* for $0 \le e \le 0.74969$ **and** $6678.14 \le a \le 26{,}553.4$

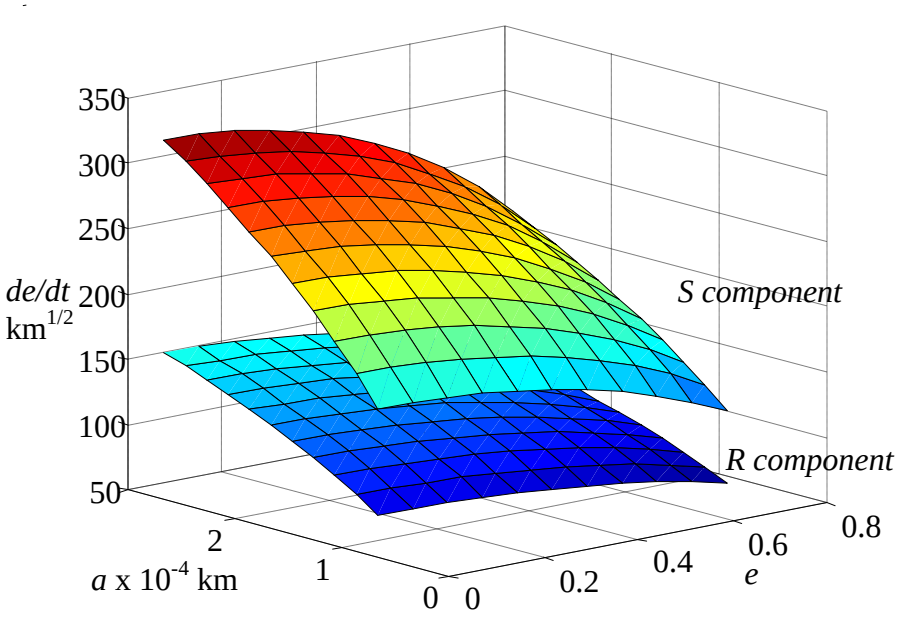


Figure 8: Plots of equations *(9)* and *(10)* for $0 \le e \le 0.74969$ **and** $6678.14 \le a \le 26{,}553.4$

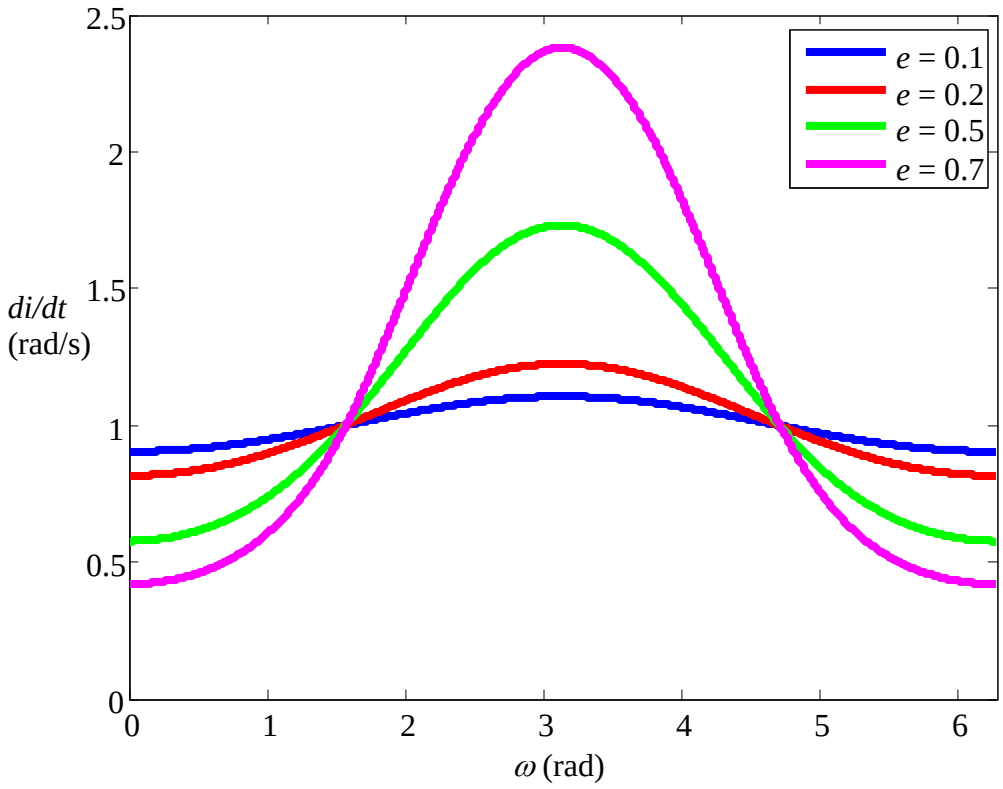


Figure 9: Plot of equation (11) for $0 \le e \le 0.74969$ **,** $0 \le \omega \le 2\pi$ **, and factoring out** $\sqrt{a}$

A trajectory from an initial LEO to final Molniya orbit would represent a line connecting the (*a*, *e*) = (6678.14 km, 0.0) to (*a*, *e*) = (26553.4 km, 0.740969) points on Figures 7 and 8. From Figures 7 and 8, it is clear that within a possible LEO to Molniya transfer there is a definite variation between the best possible places to change *a* or *e*. Specifically, it would make more sense to initiate a semimajor axis change when *a* and *e* are closer to their final values. Likewise, the greatest change in eccentricity would correspond to a large semimajor axis and small eccentricity. However, since equations *(11)* and *(12)* both rely on forces in the $\hat{S}$ and $\hat{R}$ directions, it is impossible to change eccentricity without changing the semimajor axis and vice versa. Additionally, both equations *(11)* and *(12)* are maximized and minimized at the same points along an orbit,

as outlined by Table 5. Due to these relationships, the best option is to simultaneously change the semimajor axis and eccentricity, so that they are both increasing at some rate close to their maximum value. This rate can be accounted for by introducing the scalar ratios, η_a and η_e. These ratios will represent something similar to a switching function, in that if the ratio of the current change of an orbital element value to its maximum change is greater than the value of the ratio scalar, the maximum thrust will be applied. If the change in the element is less than the ratio scalar, no thrust will be applied. Each of the $\hat{S}$ and $\hat{R}$ directions will have their own values of ratio scalars, which will be set as control variables.

An example of this process can best be viewed by first looking at the thrust in the $\hat{S}$ direction. For a thrust in this direction, the ratio of the current change in *a* to its maximum change (for a given value of *a* and *e*) can be views as

$$\frac{da/dt|_{\hat{S},current}}{da/dt|_{\hat{S},\max}} = \frac{\frac{2a\sqrt{1-e^2}F_S}{nr}}{\frac{2a^{3/2}(1+e)F_S}{\sqrt{\mu(1-e^2)}}} = \frac{1+e\cos(\nu)}{1+e} \qquad (22)$$

Evaluating equation *(22)* at each time step of the LEO to Molniya transfer, a statement can be made that for some given value of η_a, if the result of equation *(22)* is greater than η_a, then the criteria for applying a thrust in the $\hat{S}$ direction to change *a* has been met. However, since it is necessary to get the maximum change in both *a* and *e* at each incident of thrust in the $\hat{S}$ direction, an equation for the ratio of current change in *e* to its maximum value (for given *a* and *e*) must be created, and can be written as

$$\frac{de/dt\,|_{\hat{S},current}}{de/dt\,|_{\hat{S},\max}} = \frac{\frac{\sqrt{1-e^2}}{na}\left(\cos(\nu)+\frac{e+\cos(\nu)}{1+e\cos(\nu)}\right)F_S}{\frac{2\sqrt{a(1-e^2)}F_S}{\sqrt{\mu}}} = \frac{\cos(\nu)+\frac{e+\cos(\nu)}{1+e\cos(\nu)}}{2} \qquad (23)$$

Similarly, if equation *(23)* is evaluated at each time step of the transfer, for a given value of η_e, if the result of equation *(23)* is greater than η_e, then the criteria for applying thrust in the $\hat{S}$ direction to change *e* has been met. Therefore, a hypothesized switching function to apply thrust in the $\hat{S}$ direction can be summarized as follows

$$T_{\hat{S}} = \begin{cases} 0 & if \quad \frac{de/dt\,|_{\hat{S},current}}{de/dt\,|_{\hat{S},\max}} \le \eta_{e,\hat{S}} \quad or \quad \frac{da/dt\,|_{\hat{S},current}}{da/dt\,|_{\hat{S},\max}} \le \eta_{a,\hat{S}} \\ T_{\hat{S},\max} & if \quad \frac{de/dt\,|_{\hat{S},current}}{de/dt\,|_{\hat{S},\max}} \ge \eta_{e,\hat{S}} \quad and \quad \frac{da/dt\,|_{\hat{S},current}}{da/dt\,|_{\hat{S},\max}} \ge \eta_{a,\hat{S}} \end{cases} \qquad (24)$$

where $T_{\hat{S}}$ represents the thrust applied in the $\hat{S}$ direction, and $T_{\hat{S},\max}$ is the maximum possible thrust the $\hat{S}$ direction engine can apply.

Next, a switching function must be made for the thrusters in the $\hat{R}$ and -$\hat{R}$ directions. When looking at Figures 7 and 9, it is clear that the contribution to *da/dt* and *de/dt* is much less for $\hat{R}$ direction thrusts than for thrusts in the $\hat{S}$ direction. Mathematically, when comparing the maximum possible *da/dt* and *de/dt* contribution of $\hat{S}$ and $\hat{R}$ direction thrusts per orbit, the ratios can be observed as

$$\frac{da/dt\,|_{\hat{R},\max}}{da/dt\,|_{\hat{S},\max}} = \frac{\frac{2ea^{3/2}F_R}{\sqrt{\mu(1-e^2)}}}{\frac{2a^{3/2}(1+e)F_S}{\sqrt{\mu(1-e^2)}}} = \frac{e}{1+e} \qquad (25)$$

$$\frac{de/dt|_{\hat{R},\max}}{de/dt|_{\hat{S},\max}} = \frac{\sqrt{\frac{a(1-e^2)}{\mu}}F_R}{\frac{2\sqrt{a(1-e^2)}F_S}{\sqrt{\mu}}} = \frac{1}{2} \tag{26}$$

where it should be noted that $F_S = F_R = F_W = \frac{T_{\hat{S},\hat{R},\hat{W},\max}}{m}$. From equations *(25)* and *(26)* it can be seen that an $\hat{R}$ direction thrust will at maximum produce only half the thrust of the maximum $\hat{S}$ direction thrust (assuming there is an upper bound on the eccentricity of e = 1). Due to this fact, it would not be an efficient use of propellant to use the $\hat{R}$ direction thrusters unless the $\hat{S}$ direction thrusters are set to operate with $\eta_{a,\hat{S}}$ and $\eta_{e,\hat{S}}$ less than 0.5 (or 50% maximum *da/dt* and *de/dt* output). Therefore, a switching function for the $\hat{R}$ direction thrust can be described as follows

$$\frac{da/dt|_{\hat{R},current}}{da/dt|_{\hat{S},\max}} = \frac{\frac{2e\sin(\nu)F_R}{n\sqrt{1-e^2}}}{\frac{2a^{3/2}(1+e)F_S}{\sqrt{\mu(1-e^2)}}} = \frac{e\sin(\nu)}{1+e} \tag{27}$$

$$\frac{de/dt|_{\hat{R},current}}{de/dt|_{\hat{S},\max}} = \frac{\frac{\sqrt{1-e^2}}{na}\sin(\nu)F_R}{\frac{2\sqrt{a(1-e^2)}F_S}{\sqrt{\mu}}} = \frac{\sin(\nu)}{2} \tag{28}$$

$$T_{\hat{R}} = \begin{cases} 0 & if \quad -\eta_{a,\hat{R}} \le \frac{da/dt|_{\hat{R},current}}{da/dt|_{\hat{S},\max}} \le \eta_{a,\hat{R}} \quad or \quad -\eta_{e,\hat{R}} \le \frac{de/dt|_{\hat{R},current}}{de/dt|_{\hat{S},\max}} \le \eta_{e,\hat{R}} \\ T_{\hat{R},\max} & if \quad \eta_{a,\hat{R}} \le \frac{da/dt|_{\hat{R},current}}{da/dt|_{\hat{S},\max}} \quad and \quad \eta_{e,\hat{R}} \le \frac{de/dt|_{\hat{R},current}}{de/dt|_{\hat{S},\max}} \\ -T_{\hat{R},\max} & if \quad \frac{da/dt|_{\hat{R},current}}{da/dt|_{\hat{S},\max}} \le -\eta_{a.\hat{R}} \quad and \quad \frac{de/dt|_{\hat{R},current}}{de/dt|_{\hat{S},\max}} \le -\eta_{e,\hat{R}} \end{cases} \tag{29}$$

It should be noted that in equation *(29)*, a constraint is also put on the control values $\eta_{a,\hat{R}}$ and $\eta_{e,\hat{R}}$ in that $\eta_{a,\hat{R}} \geq \eta_{a,\hat{S}}$ and $\eta_{e,\hat{R}} \geq \eta_{e,\hat{S}}$. This is due to the fact that the maximum contributions to equations *(11)* and *(12)* will always be greater in the $\hat{S}$ component than the $\hat{R}$ component, as illustrated by equations *(25)*, *(26)* and Figures 7 and 8. Therefore, contributions to *da/dt* and *de/dt* in the $\hat{R}$ direction should not be allowed to dip below the contributions in the $\hat{S}$ direction, resulting in an inefficient use of propellant. The constraint placed on $\eta_{a,\hat{R}}$ and $\eta_{e,\hat{R}}$ ensures that this inefficiency will not take place.

Finally, a switching function for the $\hat{W}$ direction thrust must be created. By observing equation *(13)* it can be seen that any change in inclination caused by a thrust in a $\hat{W}$ direction will have no effect on the semimajor axis or eccentricity. Since forces in the $\hat{W}$ direction do not have any effects on other in plane elements, the $\hat{W}$ direction thrust therefore does not have to be activated at the same time as the thrusts in the $\hat{S}$ and $\hat{R}$ directions. Additionally, from Figure 9, Table 5, and observations of equation *(21)* it can be seen that the maximum possible value of *di/dt* will occur when $\nu = \sin^{-1}[-e\sin(\omega)] - \omega$, $e = e_{\max} = 0.740969$, $a = a_{\max} = 26553.4$ km, and $\omega = \pi$ for thrust in the $\hat{W}$ direction, ($\nu = \pi - \sin^{-1}[-e\sin(\omega)] + \omega$, $\omega = 0$ for thrust in the -$\hat{W}$ direction). Applying these conditions to equation *(13)*, an analytical representation can be derived for the maximum possible change inclination over all possible orbital elements

$$\frac{di}{dt}\Big|_{\max,total} = \frac{\sqrt{a_{\max}(1-e_{\max}{}^2)}F_W}{(1-e_{\max})\sqrt{\mu}} \qquad (30)$$

Since *di/dt* will increase in maximum magnitude as a spacecraft goes from a LEO to Molniya orbit, and because changing the inclination has no effect on changing either the semimajor axis or eccentricity, the inclination will have two ratio scalars associated with it. The first scalar is the ratio of the current *di/dt* with respect to the maximum possible value, denoted as $\eta_{i,tot}$. A switching function based on this value will ensure that the thrust in the $\hat{W}$ direction will not be activated until the most efficient stages of the total transfer. A second ratio scalar, denoted as η_i, will represent a scalar ratio much like η_a and η_e, which will ensure that thrust in the $\hat{W}$ direction, once satisfying the $\eta_{i,tot}$ ratio criteria, will only be activated as the most efficient times per each orbit. Therefore, the switching function for thrust in the $\hat{W}$ direction can be defined as follows

$$\frac{di/dt|_{current}}{di/dt|_{\max,total}} = \frac{\dfrac{r\cos(u)F_W}{na^2\sqrt{1-e^2}}}{\dfrac{\sqrt{a_{\max}(1-e_{\max}{}^2)}F_W}{(1-e_{\max})\sqrt{\mu}}} = \frac{\sqrt{a(1-e^2)}\cos(u)(1-e_{\max})}{\sqrt{a_{\max}(1-e_{\max}{}^2)}\{1+e\cos(u-\omega)\}} \qquad (31)$$

$$\frac{di/dt|_{current}}{di/dt|_{\hat{W},\max}} = \frac{\dfrac{r\cos(u)F_W}{na^2\sqrt{1-e^2}}}{\dfrac{\sqrt{\{a(1-e^2)\}[1-e^2\sin^2(\omega)]}F_W}{[1+e\cos(\omega)\sqrt{1-e^2\sin^2(\omega)}-e^2\sin^2(\omega)]\sqrt{\mu}}}$$

$$= \frac{\cos(u)[1+e\cos(\omega)\sqrt{1-e^2\sin^2(\omega)}-e^2\sin^2(\omega)]}{\sqrt{1-e^2\sin^2(\omega)}\{1+e\cos(u-\omega)\}} \qquad (32)$$

$$T_{\hat{W}} = \begin{cases} 0 & if \quad -\eta_{i,tot} \le \dfrac{di/dt|_{current}}{di/dt|_{\max,total}} \le \eta_{i,tot} \quad or \quad -\eta_i \le \dfrac{di/dt|_{current}}{di/dt|_{\hat{W},\max}} \le \eta_i \\ T_{\hat{W},\max} & if \quad \eta_{i,tot} \le \dfrac{di/dt|_{current}}{di/dt|_{\max,total}} \quad and \quad \eta_i \le \dfrac{di/dt|_{current}}{di/dt|_{\max,total}} \\ -T_{\hat{W},\max} & if \quad \dfrac{di/dt|_{current}}{di/dt|_{\max,total}} \le -\eta_{i,tot} \quad and \quad \dfrac{di/dt|_{current}}{di/dt|_{\max,total}} \le -\eta_i \end{cases} \tag{33}$$

The switching functions outlined in equations *(24)*, *(29)* and *(33)* represent an initial hypothesis as to how to perform low-thrust trajectory optimization, specifically for a LEO to Molniya orbit, without the use of altering the equations of state. This switching function thus serves as a basis, to which alterations will be made due to testing results. Before each test is explained, the switching functions may be altered to meet the needs of a particular test. Once testing has been completed, a final optimal switching function will be decided upon and compared against a higher-order collocation method for solving the low-thrust LEO to Molniya optimal control problem, originally studied in the work of Herman and Spencer [2].

Problem Formulations (LEO to Molniya Transfer)

For the case of testing a LEO to Molniya low-thrust transfer, the switching functions outlined in equations *(24)*, *(29)*, and *(33)* (or slight variations) will be implemented in STK. These switching functions will alter the value of a propellant based objective function, subject to certain final orbit equality constraints. Preliminary tests will be conducted which will show what alterations can be made to equations *(24)*, *(29)*, and *(33)* to aid in the optimality of the final solution, or speed of algorithm convergence. Once testing has concluded what alterations to make, these final switching functions will

be implemented in final test cases. One test case will show the most optimal low-thrust trajectory from a LEO to Molniya orbit, subject to a transfer time of 30 days, while the other will compare results from a time-constrained LEO to Molniya transfer which was found using a higher order collocation method of optimal control [2].

For each problem formulation (both preliminary testing and final results), some or all of the ratio scalar values; $\eta_{a,\hat{S}}$, $\eta_{e,\hat{S}}$, $\eta_{a,\hat{R}}$, $\eta_{e,\hat{R}}$, $\eta_{i,tot}$ and η_i will represent parameters in the switching functions in equations *(24)*, *(29)*, and *(33)*. These values can be selected as control values in Astrogator, and would therefore also represent elements in an objective function control vector. Unless otherwise specified, the initial argument of periapsis, ω_i, will also be selected as a control variable. This is done because of all the effects perturbations in the $\hat{S}$, $\hat{W}$, and $\hat{R}$ directions have on the argument of periapsis, illustrated in equation *(14)*. Any maneuver that changes *a*, *e*, or *i* will also change ω, so instead of attempting to keep ω fixed at some value, it is instead allowed to rotate freely, and its initial position varied so that it will rotate into the correct final position at the end of the maneuver. Therefore, a control vector for the LEO to Molniya problem formulation can be viewed as

$$\vec{u} = < \eta_{a,\hat{S}}, \eta_{e,\hat{S}}, \eta_{a,\hat{R}}, \eta_{e,\hat{R}}, \eta_i, \eta_{i,tot}, \omega_i > \qquad (34)$$

The control vector outlined in equation *(34)* is also a preliminary control vector for this particular problem formulation, and will be changed based on testing results and problem reformulations. Additionally, the following initial bounds will be imposed on these control variables

$$0 \le \eta_{a,\hat{S}} \le 1$$

$$\eta_{a,\hat{S}} \le \eta_{a,\hat{R}} \le 1$$

$$-1 \le \eta_{e,\hat{S}} \le 1$$

$$\eta_{e,\hat{S}} \le \eta_{e,\hat{R}} \le 1$$

$$0 \le \eta_{i,tot} \le 1$$

$$0 \le \eta_{i} \le 1$$

$$0^\circ \le \omega_i \le 360^\circ$$

Also for this particular problem formulation, the spacecraft attitude is set to the $\hat{R}\ \hat{S}\ \hat{W}$ system illustrated in Figure 6, while the thrust is governed by the switching functions. These switching functions will be implemented in three custom engine plugins in STK, representing engines providing thrust in the $\hat{S}$, $\hat{W}$, and $\hat{R}$ directions, which will be referred to as $\hat{S}_{engine}$, $\hat{W}_{engine}$, and $\hat{R}_{engine}$. Five separate engines will be created using these three plugins; one engine representing the $\hat{S}$ direction switching function, using the $\hat{S}_{engine}$ plugin, two engines representing the $\hat{R}$ and -$\hat{R}$ switching functions using the $\hat{R}_{engine}$ plugin, and two engines representing the $\hat{W}$ and -$\hat{W}$ switching functions using the $\hat{W}_{engine}$ plugin. Therefore, a thruster set for this particular problem formulation can is illustrated in Figure 10. The thruster set in Figure 10 represents the thruster set which will be used in the final test cases, performed in STK 9.0.

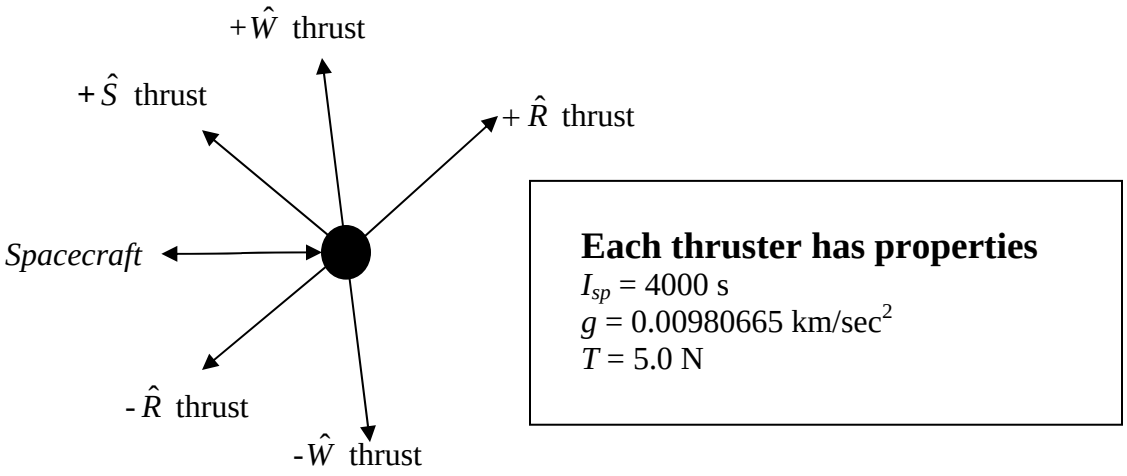


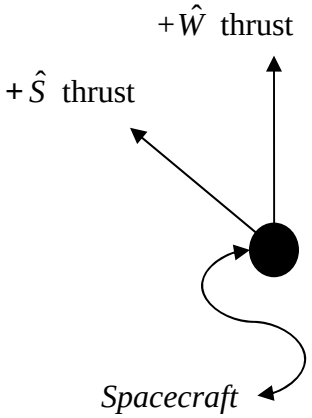
Figure 10: Diagram of final test case spacecraft thruster set for LEO to Molniya Problem Formulation.

Of particular importance are the limitations of STK 8.1 on implementing the $\hat{R}$ and $\hat{W}$ direction switching functions. Recall that while the final test cases will be done in STK 9.0 (and use the thruster set illustrated in Figure 10), all other testing is performed in STK 8.1. Based on the constraint imposed on $\eta_{a,\hat{R}}$ and $\eta_{e,\hat{R}}$ in equation *(29)* the $\hat{R}_{engine}$ plugin will need to reference $\eta_{a,\hat{S}}$ and $\eta_{e,\hat{S}}$ values in the $\hat{S}_{engine}$ plugin. Likewise, in equations *(29)* and *(33)*, it can be seen that both switching functions rely on using one value for $\eta_{a,\hat{R}}$, $\eta_{e,\hat{R}}$, η_i and $\eta_{i,tot}$, but will be implemented across four custom engines (for thrusters in the $\hat{R}$, $-\hat{R}$, $\hat{W}$ and $-\hat{W}$ directions) . In STK 8.1, there is no way to transfer one value between multiple engines. That is, values can be changed for each individual engine, but they cannot represent global variables, used by multiple engines. Therefore, for initial testing of the aforementioned switching functions, the $\hat{R}$ direction switching function will be removed (due to the inability to handle its control variable

constraints) and the $\hat{W}$ direction switching function will be represented as follows (with the $\hat{S}$ direction switching function remaining unchanged)

$$T_{\hat{W}} = \begin{cases} 0 & if \quad \dfrac{di/dt\,|_{current}}{di/dt\,|_{\max,total}} \le \eta_{i,tot} \quad or \quad \dfrac{di/dt\,|_{current}}{di/dt\,|_{\hat{W},\max}} \le \eta_i \\ T_{\hat{W},\max} & if \quad \eta_{i,tot} \le \dfrac{di/dt\,|_{current}}{di/dt\,|_{\max,total}} \quad and \quad \eta_i \le \dfrac{di/dt\,|_{current}}{di/dt\,|_{\hat{W},\max}} \end{cases} \qquad (35)$$

Therefore, for the preliminary test cases of this problem formulation conducted in STK 8.1, the thruster set used can be illustrated in Figure 11.



Each thruster has properties
I_{sp} = 4000 s
g = 0.00980665 km/sec^2
T = 5.0 N

Figure 11: Diagram of spacecraft initial test case thruster set for LEO to Molniya Problem Formulation.

This thruster set, while not as thorough as the one illustrated in Figure 10, will help show how effective the two switching functions outlined in equations (25) and (36) are in creating burn and coast times. However, one major problem with the thruster set illustrated in Figure 11 completely eradicates radial thrusts (since the $\hat{R}$ direction switching function cannot be used effectively in STK 8.1). As a result of this inconvenience, a long transfer time of 30 days is chosen for the preliminary test cases.

This long duration was chosen because optimal values of $\eta_{a,\hat{S}}$ and $\eta_{e,\hat{S}}$ were in the range of ~0.7, much higher than the ~0.5 upper bound on $\eta_{a,\hat{R}}$ and $\eta_{e,\hat{R}}$, indicating radial thrust would not even be activated under the definition of equation *(30)*. Future alterations of the switching functions, as well as the switching functions chosen for the final test cases will be discussed and presented in subsequent tests and problem formulations.

Therefore, with the circle to circle and LEO to Molniya transfer problem formulations defined, their control vectors, objective functions and constraint functions can be implemented in both the SQP and CMA-ES algorithms. As a preliminary test, the circle to circle transfer will be used to help determine the strengths and weaknesses of both algorithms, so an educated choice can be made as to which one is more desirable for solving low-thrust trajectory optimization problems. Once an algorithm is selected, it will be implemented on the LEO to Molniya problem formulation, in an attempt to refine, if necessary, the various switching functions. The overall best configuration of optimization algorithm and set of switching functions will then be selected to replicate a LEO to Molniya transfer studied using optimal control, to determine the overall competitiveness of the methods described in this thesis.

Chapter 5: Initial Testing and Final Results

In this chapter, several initial tests are conducted, both on the SQP and CMA-ES optimization algorithms, as well as the general perturbation switching function problem formulations. These tests will show the strengths and weaknesses in both the optimization algorithms, objective functions and switching functions, so that reformulations can be made, and thus the overall best optimization scheme can be selected.

The optimization algorithms will be tested first, by means of subjecting them to the LEO to 10,000 km semimajor axis with inclination change problem formulation. This simple problem formulation will give insight as to how each algorithm performs within STK's black box framework. Specifically, it will test weather the algorithms converge to global or local extrema, how well they handle equality constraints, and their overall time to convergence. A final comparison will be made based on these observations, and either the SQP or CMA-ES algorithm will be selected to continue with future testing, as well as the gathering of final results.

With the selection of the proper optimization algorithm, the problem formulation consisting of definitions for switching functions and an objective function, can now be revised and tested. The problem formulation is what dictates how thrust will be applied within the transfer, as well as how a cost variable, control variables, and constraints will be applied to a numerical optimization algorithm. A problem formulation for a LEO to Molniya low-thrust transfer is tested in this stage, to see what modifications should be made to the switching functions and objective function to aid in the speed to convergence, constraint handling, and overall optimal solution. As with the initial testing of the optimization algorithms, the testing of the problem formulation will seek to expose

weaknesses and/or strengths in the formulation, rather than creating real-world applicable optimal low-thrust transfers.

Once the proper reformulations have been made to the switching functions and objective function, the final test cases can be conducted. These final test cases will seek to find real-world applicable optimal low-thrust transfers, rather than testing the algorithms or problem formulations. In these final tests, the selected optimization algorithm, and best set of switching functions are used to determine two optimal low-thrust LEO to Molniya transfers. The first transfer will consist of a 30-day transfer time, while the second will model a transfer created using a higher order collocation method of optimal control presented by Herman and Spencer [2]. In the latter case, the results from the methods imposed in this study will be compared to the results using optimal control to determine if the methods are competitive for solving a LEO to Molniya low-thrust transfer.

Optimization Algorithm Testing: Circle to Circle Transfer

In order to test the performance of both the SQP and CMA-ES algorithms on finding optimal low-thrust trajectories within a black box framework, a simple LEO to 10,000 km semimajor axis with an inclination change problem is created. This problem formulation is based off of finding a propellant optimal solution from a given initial state to a desired final state, subject to equality constraints on the final semimajor axis[*] and inclination. Therefore, this particular optimization problem could be described as

[*] Recall, this constraint is handled by a stopping condition within the numerical integration algorithm, as described in Chapter 4

Given

Initial conditions in Table 1, attitude illustrated in Figure 3, thruster set illustrated in Figure 5, and a_f = 10,000 km

Find

$$\vec{u} = < T_{\hat{V},+}, T_{\hat{C},+}, T_{\hat{N},+}, T_{\hat{N},-} >$$

That Minimizes

$$F(\vec{u}) = Propellant\ Used \quad (SQP)$$

$$F(\vec{u}) = Propellant\ Used + w_i \left| i - i_f \right| \quad (CMA\text{-}ES)$$

Subject to:

$$0 < T_{x,+} < 0.5\ N$$
$$0 < T_{z,+} < 0.5\ N$$
$$0 < T_{y,+} < 0.5\ N$$
$$0 < T_{y,-} < 0.5\ N$$
$$i_f = 0\ Degrees$$

where a_f and i_f is the final orbit semimajor axis and inclination respectively, and w_i is a scalar weight penalty associated with the inclination constraint (used for testing CMA-ES algorithm only). It should be noted that there are two possible objective functions, one for the SQP algorithm, and one for the CMA-ES algorithm. This is due to the fact that the CMA-ES algorithm must handle equality constraints by imposing them as penalty functions within the objective functions, while the SQP algorithm handles then directly.

Since there is an equality constraint associated with this problem formulation, it becomes necessary to implement the proper weight for the objective function used in the CMA-ES algorithm. For this case, initial weights (w_i) of 1, 10, and 100 are applied to the CMA-ES objective function, with results displayed in Table 6. It should be noted that there are four runs of w_i = 10 and w_i = 100 only one for w_i = 1 because it was obvious that after one run of w_i = 1, the objective function did not accurately match the desired function space to be minimized. Specifically, for $w = 1$, the weight was not big enough,

and the inclination constraint was not even met because its penalty was not high enough to properly alter the function space. Results from execution of the SQP algorithm can also be seen in Table 7. Note, in Table 7, and future result tables, the number of function evaluations (*nfe*) the given optimization algorithm performed before convergence is given.

Table 6: CMA-ES executions of circle to circle transfer with inclination change for weights of w_i = *1, 10, and 100*.

w_i	nfe	$T_{\hat{V},+}$ (N)	$T_{\hat{C},+}$ (N)	$T_{\hat{N},+}$ (N)	$T_{\hat{N},-}$ (N)	Propellant (kg)	i_f (deg)	a_f (km)
1	6290	0.4687	0.0000	0.0000	0.0000	21.2406	9.9924	10,000.0
10	17978	0.2193	0.0000	0.3499	0.2444	48.8721	0.0000	10,000.0
10	9138	0.3178	0.0000	0.4002	0.4615	48.8606	0.0000	10,000.0
10	13586	0.3462	0.0000	0.4449	0.4939	48.8598	0.0000	10,000.0
10	10050	0.3365	0.0000	0.4288	0.4838	48.8599	0.0000	10,000.0
100	10010	0.2803	0.0000	0.4834	0.2815	49.0146	0.0000	10,000.0
100	16818	0.2117	0.0000	0.1135	0.4606	48.8870	0.0000	10,000.0
100	12986	0.3018	0.0002	0.3434	0.4766	48.9317	0.0000	10,000.0
100	22498	0.3268	0.0000	0.4069	0.4817	48.9318	0.0000	10,000.0

Table 7: SQP execution of circle to circle low-thrust transfer with inclination change.

nfe	15
$T_{\hat{V},+}$ ***(N)***	0.3698
$T_{\hat{C},+}$ ***(N)***	0.2026
$T_{\hat{N},+}$ ***(N)***	0.4999
$T_{\hat{N},-}$ ***(N)***	0.4999
Propellant (kg)	59.562
i_f ***(deg)***	0.0286
a_f ***(km)***	10,000.0

As Table 6 shows, while the difference in optimal propellant consumption differs slightly from $w_i = 10$ and $w = 100$, it can still be seen that a weight of $w_i = 10$ performs faintly better than $w_i = 100$. However, between the four runs at $w_i = 10$, different optimal

control values for $T_{\hat{V},+}$, $T_{\hat{N},+}$, and $T_{\hat{N},-}$ are calculated. While the CMA-ES algorithm can handle multi-modal function spaces, when presented with noise in the function space, it can still be plagued by inconsistent convergence. However, while the CMA-ES algorithm converged to different control variable values for a weight of *w = 10*, the overall propellant use converged to the same general value. This could be an indication that while this problem may have a global minimum; it is it is hidden within an area of small amplitude noise, making convergence to a single point difficult. If this is the case, then an optimal solution could be viewed as any set of values within that domain, and not necessarily one point (that is, if the noise amplitude is small). Taking into account the fact that STK uses numerical integration to solve a spacecraft's equations of motion (which includes truncation and round-off error), coupled with a termination criteria of the final semimajor axis (which is met to a tolerance, and not exactly), stating that there may be some small amplitude noise within this black box function space is a valid assumption.

As for the classical algorithm runs seen in Table 7, the SQP algorithm converged in far fewer function evaluations than the CMA-ES algorithm, but obviously converged to a local minimum. The SQP algorithm converged to a final propellant use of about 10 kg higher than the CMA-ES algorithm, no doubt because it converged to the first stationary point ($\nabla F(\vec{x}) = 0$) within the vicinity of the initial control vector, rather than performing a more thorough search of the function space. Also of note is that the SQP algorithm did not solve the final inclination constraint as strictly as the CMA-ES algorithm.

Due to these observations seen in Table 6, the problem formulation is altered in several ways to attempt to simplify the problem, and create a more narrow focus on

where the optimal value may lie. This may help differentiate whether there is a distinguishable global minimum, or whether noise within STK still makes pinpointing an exact global minimum difficult. Testing of this reformulation will be done exclusively within the CMA-ES algorithm. Should this noise still be present, the optimal control vector will occur within a domain, rather than one specific point.

In this problem reformulation, a weight of $w_i = 10$ is used for all remaining CMA-ES trial runs, because it was shown to perform the best between the weights in Table 6. Second, the $T_{\hat{C},+}$ component of thrust is removed from the control vector, and set to zero. This was done because of the tendency of this value to converge to 0 for each weight case in Table 6. While this value did not converge to zero for the SQP algorithm (Table 7), it produced a less optimal result than in Table 6, and therefore removal of this variable is still justified. Thus, another set of runs is made with there revisions to the problem formulation, to yield the results seen in Table 8.

Table 8: CMA-ES executions of low-thrust circle to circle transfer with inclination change for weights of $w_i = 10$, and no co-normal thrust

(*w*)	nfe	$T_{\hat{V},+}$ (N)	$T_{\hat{N},+}$ (N)	$T_{\hat{N},-}$ (N)	Propellant (kg)	i_f (deg)	a_f (km)
10	5252	0.303021	0.485979	0.335582	48.86628	0.000001	10,000.0
10	3740	0.187206	0.024786	0.482654	48.86973	0.000008	10,000.0
10	3446	0.213766	0.286897	0.292444	48.87665	0.000000	10,000.0
10	4531	0.000016	0.018510	0.139496	31.60728	0.002579	6605.25
10	4762	0.000001	0.157959	0.000047	31.59386	0.038367	6604.50

From Table 8, it becomes obvious (at least for the last two executions) that the CMA-ES algorithm has found a minimum around the vicinity of the lower bounds (specifically, thrust = 0) for the control values. From Table 8, it can be seen that for these runs, the semi-major axis constraint was not met; only the inclination constraint was. This is due to the fact that the semi-major axis constraint was removed as an equality

constraint and replaced as a stopping condition on the thrust segment propagation. By recollection, another stopping condition, that of thrust duration of 0.5 years was also placed on the thrust segment propagation. Therefore, if the velocity-direction thrust is driven to zero, it turns out that the inclination constraint can be met using less propellant than meeting both the inclination and semi-major axis constraints. In this case, the algorithm converges to values which result in an activation of the thrust duration, not final semi-major axis stopping condition.

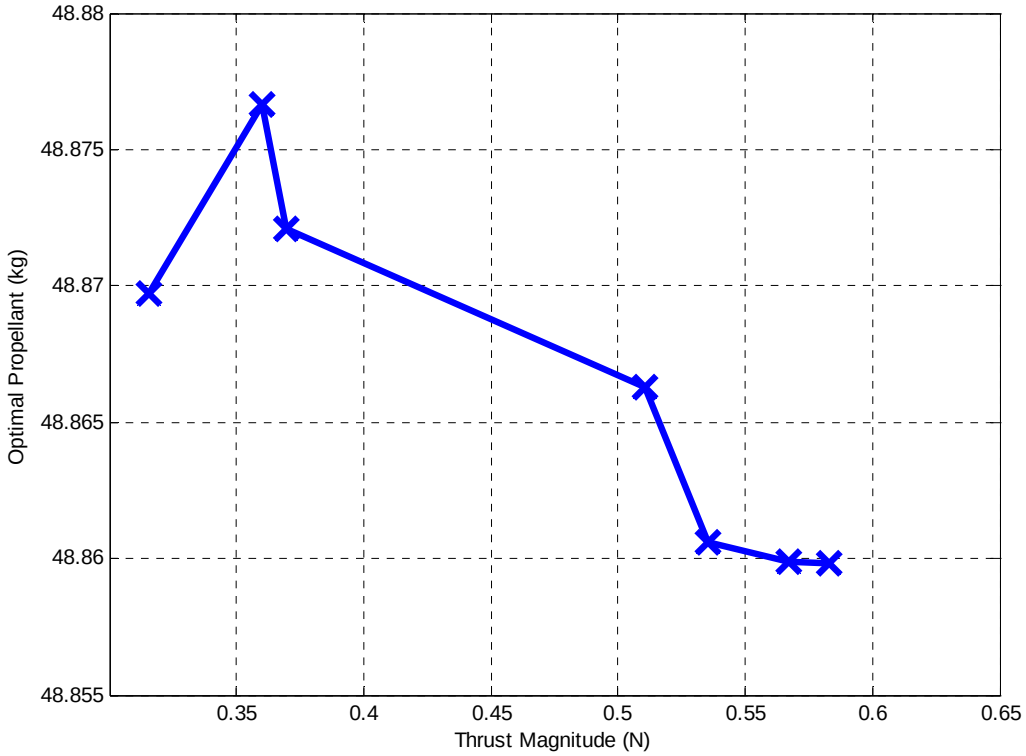


Figure 12: Plot of optimal propellant vs. thrust magnitude for the circle to circle low-thrust transfer (contains data from all *w* = 10 runs in Table 6 and first 3 runs in Table 8)

It can also be observed from Table 6 and Table 8 that for w_i = 10, increasing the values for $T_{\hat{V},+}$, $T_{\hat{N},+}$, and $T_{\hat{N},-}$ seem to cause the algorithm to gravitate to a slightly (on

the order of 0.02 kg) more optimal values for propellant used. This observation can be better illustrated in Figure 12 which represents the magnitude of the thrust vectors (excluding the last two runs in Table 8) plotted against the over all propellant optimal solution.

As a result of this observation, three more tests are created, one in which the value for $T_{\hat{V},+}$ is removed as a control value, and set to its upper bound of 0.5 N, another in which the values for $T_{\hat{N},+}$, and $T_{\hat{N},-}$ are removed as control values and set to their upper bound of 0.5 N, and a third where velocity and orbit-normal direction thrusts are kept as control values but their domains are changed from:

$$0 < T_{\hat{V},+} < 0.5\ N$$

$$0 < T_{\hat{N},+} < 0.5\ N$$

$$0 < T_{\hat{N},-} < 0.5\ N$$

to

$$0.3 < T_{\hat{V},+} < 0.4\ N$$

$$0.4 < T_{\hat{N},+} < 0.5\ N$$

$$0.4 < T_{\hat{N},-} < 0.5\ N$$

Which were found to be the areas consisting of the most propellant optimal solutions based on Table 6. These tests are executed in an attempt to see whether a consistent globally optimal solution can be achieved when one thrust direction is set to its maximum, and the other merely adjusts to meet the equality constraints, or whether a maximum still occurs in an area between those extremes. Results for the maximum velocity direction thrust, maximum orbit-normal direction thrusts, and reduced domain

thrusts formulations tested using the CMA-ES algorithm can be seen in Table 9, Table 10, and Table 11 respectively. A table containing data from the SQP algorithm for the velocity maximum and orbit-normal maximum thrust problem formulations can also be seen in Table 12.

Table 9: CMA-ES executions of circle to circle low-thrust transfer with inclination change for maximum thrust in velocity direction

Run	nfe	$T_{\hat{N},+}$ (N)	$T_{\hat{N},-}$ (N)	Propellant (kg)	i_f (deg)	a_f (km)
1	188	0.500000	0.500000	41.72685	2.627770	10,000.0
2	140	0.500000	0.500000	41.72685	2.627770	10,000.0
3	194	0.500000	0.500000	41.72685	2.627770	10,000.0

Table 10: CMA-ES executions of circle to circle low-thrust transfer with inclination change for maximum thrust in orbit-normal directions

Run	nfe	$T_{\hat{V},+}$ (N)	Propellant (kg)	i_f (deg)	a_f (km)
1	606	0.368481	48.87583	0.000016	10,000.0
2	790	0.368476	48.87616	0.000015	10,000.0
3	854	0.368460	48.87739	0.000012	10,000.0

Table 11: CMA-ES executions of circle to circle low-thrust transfer with inclination change for bounded thrust in velocity and orbit-normal directions

Run	nfe	$T_{\hat{V},+}$ (N)	$T_{\hat{N},+}$ (N)	$T_{\hat{N},-}$ (N)	Propellant (kg)	i_f (deg)	a_f (km)
1	4671	0.329753	0.446036	0.448111	48.85946	0.000007	10,000.0
2	6043	0.300265	0.410944	0.403217	48.86426	0.000004	10,000.0
3	4664	0.322764	0.475257	0.400068	48.86168	0.000000	10,000.0

Table 12: SQP execution of circle to circle low-thrust transfer with inclination change maximum velocity direction and maximum orbit-normal direction thrust.

	Problem Formulation	
	Velocity Thrust Max	**Orbit-Normal Thrust Max**
nfe	Did Not Converge	101
$T_{\hat{V},+}$ ***(N)***	N/A	0.3702
$T_{\hat{N},+}$ ***(N)***	N/A	0.5000
$T_{\hat{N},-}$ ***(N)***	N/A	0.5000
Propellant (kg)	N/A	48.748
i_f ***(deg)***	N/A	0.045
a_f ***(km)***	N/A	10,000.0

As seen in Table 9, if the velocity direction thrust is maintained at a maximum, even with the maximum amount of orbit-normal thrust, the spacecraft cannot change its inclination enough to satisfy the inclination constraint given the short time it takes for the thrusting maneuver to reach its stopping condition of a 10,000 km semi-major axis orbit. As a result, the CMA-ES algorithm converges to maximize the orbit-normal direction thrust, which places the spacecraft in a final orbit which most closely satisfies the inclination equality constraint, but ultimately does not achieve a final inclination of 0.0 degrees.

However, the SQP algorithm did not converge at all for this test case. This could be due to the fact that the algorithm approached an area where the gradient was ill-defined, where there may have been discontinuities within the black box function space, or the algorithm could have simply failed. In any case, it showed that this particular SQP algorithm might not be as consistently reliable (in terms of simply converging) as the CMA-ES algorithm when performed within a black box framework.

For the case of maximum orbit-normal thrust, the CMA-ES algorithm consistently converged to the same optimal velocity direction thrust, as can be seen from Table 10. However, the optimal propellant used in this problem formulation is slightly higher than some results obtained in Table 6 and Table 8, which leads to a speculation that the optimal thrust magnitudes will not actually occur when one of the spacecraft thrust is set to a maximum. However, the SQP method converged to a propellant use less than any previous convergence of the CMA-ES algorithm. This decrease was extremely small (~0.1 kg) and can could be attributed to the fact that the final inclination constraint was not met to a tolerance as exact as the CMA-ES algorithm. This test case was more

encouraging for the SQP algorithm, because it shows that it has the potential to find results on par with the CMA-ES algorithm, given the proper starting values.

When observing the results for the more tightly bounded control variables, seen in Table 11, the optimal propellant used converged to values on par with the lowest currently seen in any problem formulation. However, these values are still smaller by rather insignificant amounts when compared to other optimal values, which can lead to the assumption that any set of control variables within a specific domain can be considered "optimal". For this particular problem, it could be safe to say that any set of control variables which converged to a propellant use of approximately 48.87 kg or lower would be considered optimal. Based on observations, values of propellant concurrent with this optimal value will typically occur within the following approximate set of control variables

$$0.3 < T_{\hat{V},+} < 0.4\ N$$

$$0.4 < T_{\hat{N},+} < 0.5\ N$$

$$0.4 < T_{\hat{N},-} < 0.5\ N$$

$$T_{\hat{C},+} = 0\ N$$

As a final test, the SQP algorithm is executed with initial control values corresponding to the best results observed thus far. From the results of the maximum orbit-normal thrusts, it is hypothesized that given the correct initial control vector, the SQP algorithm may be able to effectively find a minimum competitive with the CMA-ES algorithm in a fraction of the time. Thus, the SQP algorithm is executed with a initial

control vector of $<T_{\hat{V},+}, T_{\hat{C},+}, T_{\hat{N},+}, T_{\hat{N},-}>$ = <0.32, 0.00, 0.44, 0.44> *N* and subject to the optimal bounds of those variables, with results found in Table 13.

Table 13: SQP execution of circle to circle low-thrust transfer with inclination change, and initial values of $<T_{\hat{V},+}, T_{\hat{C},+}, T_{\hat{N},+}, T_{\hat{N},-}>$ = <0.32, 0.00, 0.44, 0.44> *N*

nfe	158
$T_{\hat{V},+}$ **(N)**	0.3247
$T_{\hat{C},+}$ **(N)**	0.00000
$T_{\hat{N},+}$ **(N)**	0.4000
$T_{\hat{N},-}$ **(N)**	0.4841
Propellant (kg)	48.972
i_f **(deg)**	0.003
a_f ***(km)***	10,000.0

As can be seen from Table 13, even given a more ideal initial guess on the control vector, the SQP algorithm produced results similar to the CMA-ES algorithm, but failed to get a better optimal value of propellant used.

Selection of Best Optimization Algorithm

Based on observations from several tests of both the SQP and CMA-ES algorithm, various conclusions can be made concerning the overall performance of these two algorithms, as well as which should be used in further tests.

With regard to the classical-based SQP algorithm, while convergence was usually met (except in one case), and computational costs were small, the algorithm was plagued by problems common with classical algorithms. The SQP algorithm appeared to converge to local minimums often, and would require a good initial guess of the control vector in

order to converge to an optimal solution competitive with that of the CMA-ES algorithm. This is almost certainly due to the fact that search directions in classical algorithms such as this are dictated by gradients, and are therefore susceptible to converging to the closest point where the gradient is equal to zero, thus hindering a more global search, and locating of multiple extrema. A more global search method, based on several executions of the SQP algorithms for randomized initial control variables would probably be necessary to alleviate converging to local rather than global minimums.

As for the CMA-ES algorithm, performance was more consistent than in the SQP algorithm. The evolutionary-computation-based CMA-ES algorithm always converged to optimal values either equal or substantially better than the SQP algorithm, and never failed to converge. The CMA-ES algorithm, with its more global, and non-gradient based search was no doubt more robust in finding global minimums within the black box function space as compared to the SQP algorithm. However, one significant drawback to this robust global search was that the CMA-ES algorithm was very computationally expensive, generating anywhere from approximately 20 ~ 200 times longer to converge than the SQP algorithm. Additionally, constraints were met to suitable tolerances using the weighted penalty function method, and were often met to more accurate tolerances than the SQP method.

Due to these facts, it is decided that the CMA-ES algorithm, with its more robust global search method yielding better optimal results than the SQP algorithm, will be used for the remaining tests in these studies.

Switching Function Testing: LEO to Molniya Transfer

As a preliminary test, using the CMA-ES algorithm, the switching functions outlined in equations *(24)* and *(35)* are implemented in STK 8.1, so as to gain insight converning the effectiveness of these functions. Additionally, the objective function, outlined in equation *(8)* is tested as to its effectiveness in both minimizing the problem and satisfying the equality constraints. For this particular problem formulation, the constraint vector and corresponding weights can be viewed as

$$\vec{w}\vec{C}^{T} = w_a \mid a - a_f \mid + w_e \mid e - e_f \mid + w_i \mid i - i_f \mid + w_\omega \mid \omega - \omega_f \mid \qquad (36)$$

Where w_a, w_e, w_i, and w_ω are the scalar weights for the semimajor axis, eccentricity, inclination, and argument of periapsis constraints, respectively. A total transfer time stopping condition of 30 days is used to eliminate negative effects from the exclusion of equation *(29)*. This particular problem formulation can be viewed as

Given:

Initial conditions in table 1, using switching functions outlined in equations (24) and (36) with $T_{\hat{S},\max} = T_{\hat{W},\max} = 5.0N$ *and a total transfer time = 30 days*

Find:

$$\vec{u} = \{ \eta_{a,\hat{S}}, \eta_{e,\hat{S}}, \eta_{i,tot}, \eta_i, \omega_i \}$$

That minimizes:

$$F(\vec{u}) = \text{Propellant Use} + w_a \mid a - a_f \mid + w_e \mid e - e_f \mid + w_i \mid i - i_f \mid + w_\omega \mid \omega - \omega_f \mid$$

Subject to:

$$0 \le \eta_{a,\hat{S}} \le 1$$
$$-1 \le \eta_{e,\hat{S}} \le 1$$
$$0 \le \eta_{i,tot} \le 1$$
$$0 \le \eta_i \le 1$$
$$0^\circ \le \omega_i \le 360^\circ$$

a_f =26553.4 km
e_f = 0.740969
i_f = 63.4 deg
ω_f = 270 deg

Preliminary tests have shown that $\{ w_a , w_e , w_i , w_\omega \}$ = {0.001, 500, 500, 500} for penalty constraint weights work well in satisfying the various end condition equality constraints without pre-convergence. Implementing the CMA-ES algorithm on this problem formulation in STK 8.1 results in an optimal trajectory as seen in Table 14.

Table 14: Results from CMA-ES implementation of preliminary LEO to Molniya test case

nfe	2401
$\eta_{a,\hat{S}}$	0.0979
$\eta_{e,\hat{S}}$	0.6887
$\eta_{i,tot}$	0.1351
η_i	0.8212
ω_i *(deg)*	290.48
a_f *(km)*	26553.6
e_f	0.7268
i_f *(deg)*	63.392
ω_f *(deg)*	270.03
Propellant Used (kg)	90.508
Δv *(km/s)*	6.3498

From Table 14, several observations are made. Primarily, it can be seen that $\eta_{a,\hat{S}}$ converges to a very low value. Given the definition of equation *(24)*, it can be inferred that since equations *(11)* and *(12)* are maximized at the same points and equation *(11)* will always be positive in the $\hat{S}$ direction, that optimization of a single ratio scalar can produce a desired result. When deciding which ratio scalar to terminate as a control variable, it is easy to see that $\eta_{a,\hat{S}}$ should be eliminated. This is because the $\hat{S}$ component of equation *(11)* is constant for an initial LEO orbit and would always satisfy

equation *(24)*, negating the purpose of the switching function. Therefore, in subsequent tests, the ratio scalars $\eta_{a,\hat{S}}$ and $\eta_{a,\hat{R}}$ are removed from the switching function equations *(24)* and *(29)*, resulting in

$$T_{\hat{S}} = \begin{cases} 0 & if \quad \dfrac{de/dt|_{\hat{S},current}}{de/dt|_{\hat{S},\max}} \le \eta_{e,\hat{S}} \\ T_{\hat{S},\max} & if \quad \eta_{e,\hat{S}} \le \dfrac{de/dt|_{\hat{S},current}}{de/dt|_{\hat{S},\max}} \end{cases} \tag{37}$$

and

$$T_{\hat{R}} = \begin{cases} 0 & if \quad -\eta_{e,\hat{R}} \le \dfrac{de/dt|_{\hat{R},current}}{de/dt|_{\hat{S},\max}} \le \eta_{e,\hat{R}} \\ T_{\hat{R},\max} & if \quad \eta_{e,\hat{R}} \le \dfrac{de/dt|_{\hat{R},current}}{de/dt|_{\hat{S},\max}} \\ -T_{\hat{R},\max} & if \quad \dfrac{de/dt|_{\hat{R},current}}{de/dt|_{\hat{S},\max}} \le -\eta_{e,\hat{R}} \end{cases} \tag{38}$$

Taking into account the elimination of the control variables $\eta_{a,\hat{S}}$ and $\eta_{a,\hat{R}}$, the optimization procedure is repeated, and results are shown in Table 15, with plots of the change in semimajor axis, eccentricity, inclination and argument of periapsis in Figures 13, 14, 15 and 16 respectively.

From Table 15, it can be seen that removing the $\eta_{a,\hat{S}}$ control variable does little to transform the optimal solution and even leads to a faster evaluation time due to the decrease in problem complexity. However, when viewing Figure 15, it appears that the inclination is not changing as hypothesized. Theoretically, equation *(35)* should cause the inclination begin to change closer to the end of the transfer, not throughout its duration. However, $\eta_{i,tot}$ converged to such a low value that the inclination changed throughout the

orbit, instead of when it was most efficient. Through inspection, it was discovered that the satisfaction of the argument of periapsis equality constraint, coupled with the positive-only $\hat{W}$ direction thrust outlined in equation *(35)* caused this phenomenon.

From Figures 13, 14, and 15, it can also be seen that the switching functions are executing properly, initiating burn and coast segments, indicated by the "stepping" behavior in these plots.

Table 15: Results from CMA-ES implementation of the LEO to Molniya low-thrust transfer, removing semimajor axis ratio scalars

nfe	1617
$\eta_{e,\hat{S}}$	0.7116
$\eta_{i,tot}$	0.0989
η_i	0.8405
ω_i *(deg)*	290.40
a_f *(km)*	26553.7
e_f	0.7290
i_f *(deg)*	63.401
ω_f *(deg)*	269.95
Propellant Used (kg)	89.898
Δv *(km/s)*	6.3345

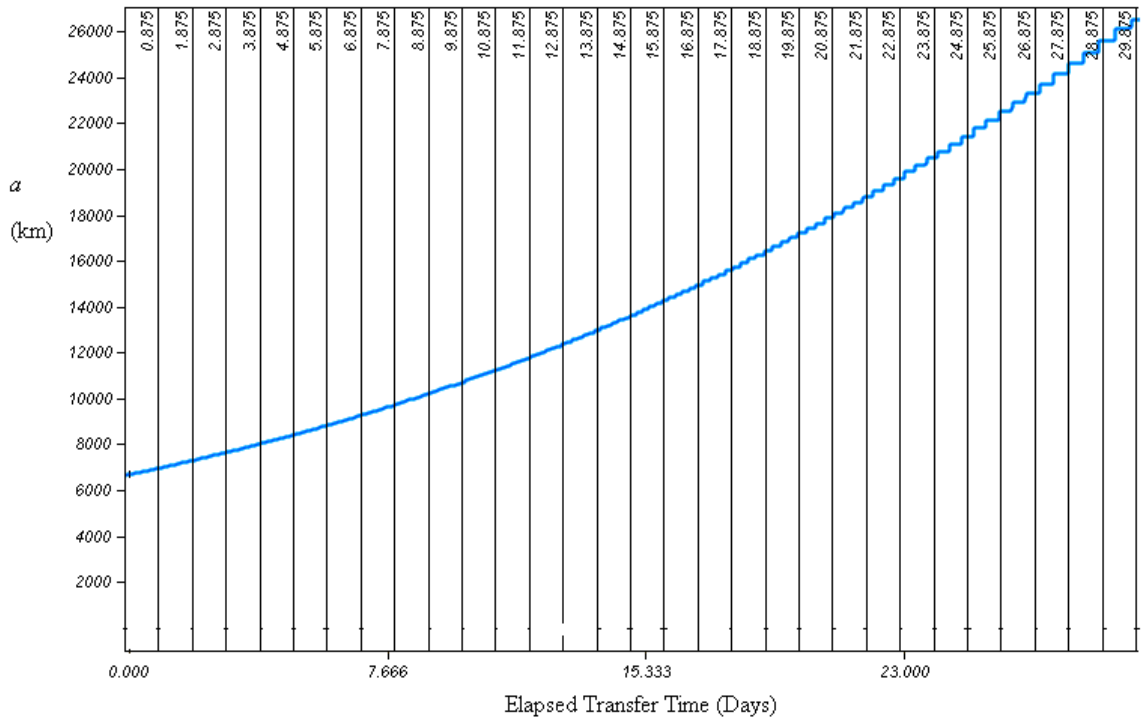


Figure 13: Semimajor axis change during maneuver for switching function alteration in equation *(37)* for the LEO to Molniya low-thrust transfer

Figure 14: Eccentricity change during maneuver for switching function alteration in equation *(37)* for the LEO to Molniya low-thrust transfer

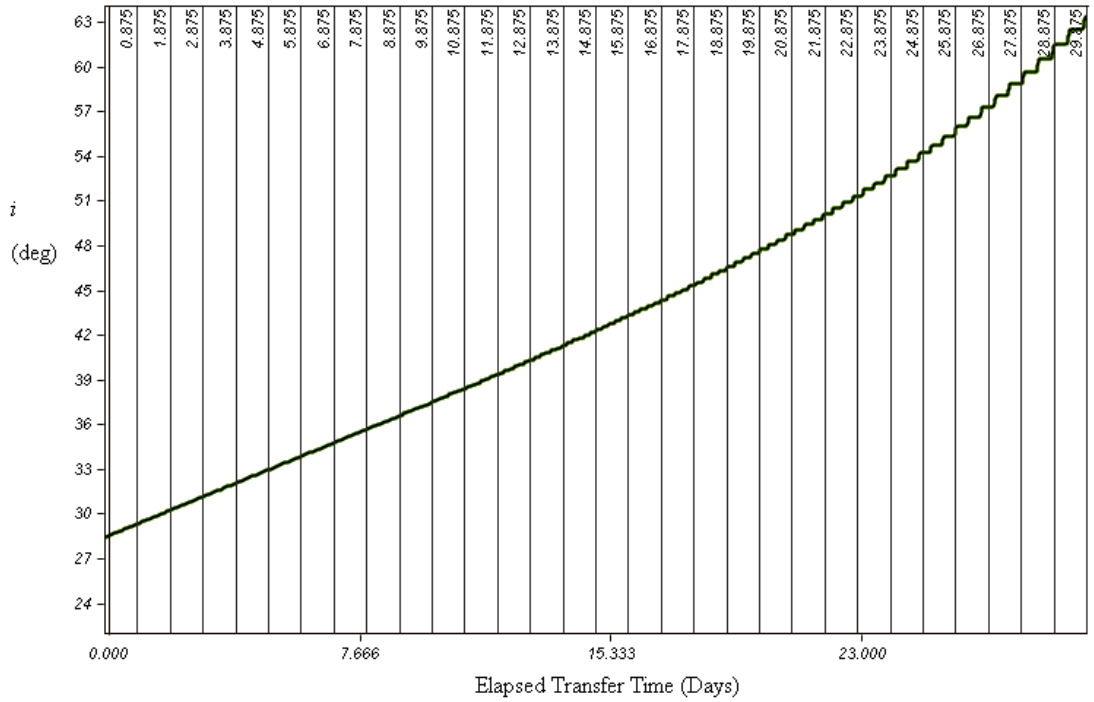


Figure 15: Inclination change during maneuver for switching function alteration in equation *(37)* for the LEO to Molniya low-thrust transfer

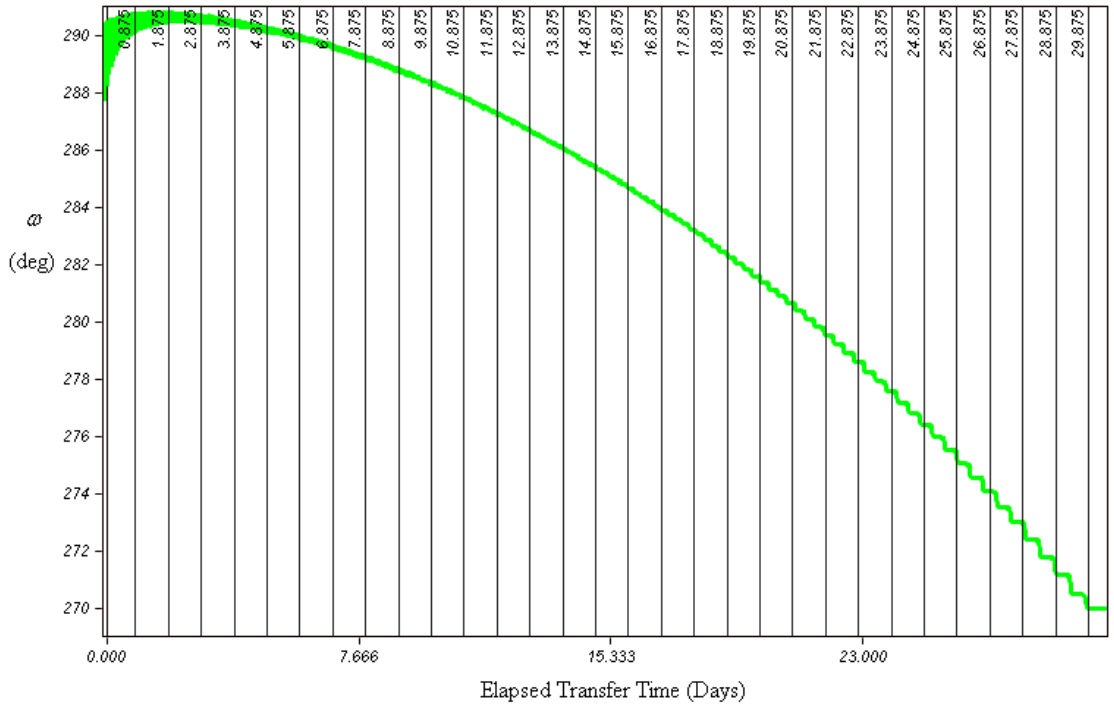


Figure 16: Argument of periapsis change during maneuver for switching function alteration in equation *(37)* for the LEO to Molniya low-thrust transfer

To illustrate how sensitive the argument of periapsis constraint is to the preliminary test formulation, another execution of the CMA-ES algorithm was run on a similar problem formulation, except that ω_i is set to an initial value of 180 degrees (its optimal value for inclination changes in the $\hat{W}$ direction) and the argument of periapsis constraint is removed from the objective function. Results from this simulation can be seen in Table 16, with a plot of the inclination change over the trajectory illustrated in Figure 17. From Table 16, it is clear that eliminating the argument of periapsis constraint and placing its initial value at the optimal value can have drastic changes on the optimal solution. Additionally, Figure 17 shows that when the final argument of periapsis constraint is removed, the inclination can be changed at its most efficient time, thus causing a drastic improvement to the optimal solution. This particular test illustrates how sensitive the optimal solution can be to the various final orbital element equality constraints.

Table 16: Results from CMA-ES implementation of the LEO to Molniya transfer, removing semimajor axis ratio scalars, and argument of periapsis constraint

nfe	6809
$\eta_{e,\hat{S}}$	0.6903
$\eta_{i,tot}$	0.7789
η_i	0.5992
ω_i *(deg)*	172.37
a_f *(km)*	26553.5
e_f	0.7270
i_f *(deg)*	63.415
ω_f *(deg)*	176.17
Propellant Used (kg)	54.188
Δv *(km/s)*	3.7130

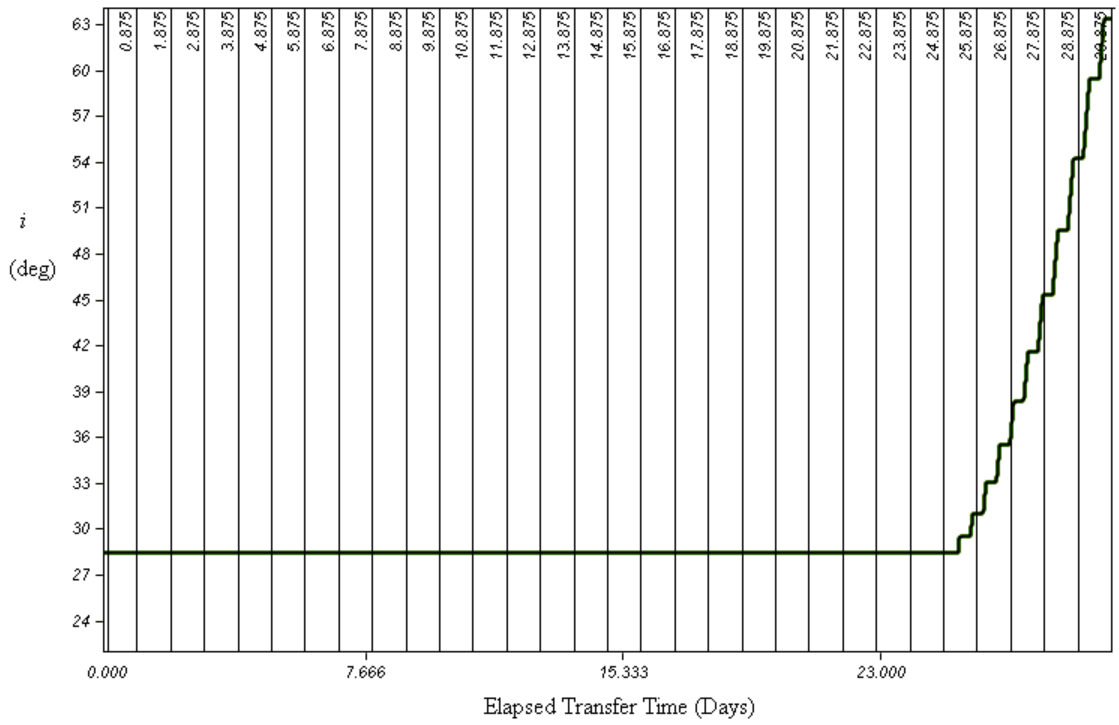


Figure 17: Inclination change during maneuver for switching function alteration in equation *(37)* and removal of argument of periapsis constraint for LEO to Molniya transfer

Selection of Best Problem Formulation

Due to the results of these three preliminary test cases, a final set of switching functions and objective function can be created for testing in STK 9.0, which allows for thrusts in the $-\hat{W}$, $\hat{R}$ and $-\hat{R}$ directions. Keeping this in mind, the radial switching function, given by equation *(38)* can be used for the first time in these studies. It should be refreshed that equation *(38)* takes into account the removal of the $\eta_{a,\hat{R}}$ scalar control variable, since the semimajor axis and eccentricity can both be maximized simultaneously solely using $\eta_{e,\hat{R}}$. Secondly, the preliminary tests were given a long 30-day transfer time in order to eliminate the need for radial thrust (since $\eta_{e,\hat{S}}$ converged to

values greater than ~0.5). The first final test case will seek to reinforce or debunk this hypothesis, by implementing equation *(38)* using a 30-day transfer time.

Additionally, due to the ability of STK 9.0 to implement $-\hat{W}$ direction thrusts, the out-of-plane switching function in equation *(35)* can be replaced with equation *(33)*. This replacement should aid in changing the inclination at a faster rate, allowing for a longer time to initiate the out-of-plane thrust, decreasing the amount of propellant used in the transfer. Also, the addition of a $-\hat{W}$ direction thrust should help create a wobble effect on the argument of periapsis, making its equality constraint less of a burden than in the preliminary LEO to Molniya test cases.

As testing showed, the $\hat{S}$ direction switching function given by equation *(37)* produced the best results, and will be implemented in the final test cases. furthermore, the constraint handling, given by equation *(36)*, with weights $\{ w_a , w_e , w_i , w_\omega \} = \{0.001, 500, 500, 500\}$ was shown to allow for convergence to an optimal value of propellant used while satisfying the various final state equality constraints, and will be used in the final test cases.

Therefore, the final problem formulation will use switching functions given by equation *(37)* for the $\hat{S}$ direction, equation *(33)* for the $\hat{W}$ direction, and equation *(38)* for the $\hat{R}$ direction. The final test case objective function will be to minimize the propellant use during the transfer, subject to the weighted penalty parameters given by equation *(36)*.

Final Application: 30 day LEO to Molniya Low-Thrust Transfer

From the preliminary test cases, adjustments have been made to the switching functions to ensure both a reliable, and competitive optimal solution for the LEO to Molniya transfer. These adjustments include the elimination of the $\eta_{a,\hat{S}}$ and $\eta_{a,\hat{R}}$ control variables, and use of STK 9.0 to include thrusts in the $-\hat{W}$, $\hat{R}$ and $-\hat{R}$ directions. The inclusion of these thrust directions will lead to a real-world applicable optimal solution, as well as a problem formulation which is ready to be tested against other methods of optimal control.

Before a comparison to the work of Herman and Spencer is made, the optimization of the preliminary LEO to Molniya test case is conducted, to ensure that the inclusion of the $-\hat{W}$, $\hat{R}$ and $-\hat{R}$ thrusting directions does indeed improve the overall solution as hypothesized. Secondly, the results of this optimization simulation will be concluded as the most optimal 30-day LEO to Molniya trajectory for the given initial and final orbital elements, initial spacecraft and engine configuration, and problem formulation (based on general perturbation equations). Therefore, this final optimization problem, can be viewed as

Given:

Initial conditions in table 1 and using switching functions outlined in equations (33), (37) and (38), with $T_{\hat{S},\max} = T_{\hat{R},\max} = T_{\hat{W},\max} = 5.0N$ *and a total transfer time = 30 days*

Find:

$$\vec{u} = \{\ \eta_{e,\hat{S}},\ \eta_{e,\hat{R}},\ \eta_{i,tot},\ \eta_i,\ \omega_i\ \}$$

That minimizes:

$$F(\vec{u}) = \text{Propellant Use} + w_a\,|\,a - a_f\,| + w_e\,|\,e - e_f\,| + w_i\,|\,i - i_f\,| + w_\omega\,|\,\omega - \omega_f\,|$$

Subject to:

$$-1 \le \eta_{e,\hat{S}} \le 1$$

$$\eta_{e,\hat{S}} \le \eta_{e,\hat{R}} \le 1$$

$$0 \le \eta_{i,tot} \le 1$$

$$0 \le \eta_i \le 1$$

$$0^\circ \le \omega_i \le 360^\circ$$

$$a_f = 26553.4 \text{ km}$$

$$e_f = 0.740969$$

$$i_f = 63.4 \text{ deg}$$

$$\omega_f = 270 \text{ deg}$$

Implementing the CMA-ES algorithm on this problem formulation in STK 9.0, the following results of an optimal trajectory can be seen on Table 17. Additionally, plots of inclination and argument of periapsis are seen in Figures 18 and 19 respectively.

Table 17: Results from CMA-ES implementation of optimal LEO to Molniya 30 day transfer problem formulation

nfe	2074
$\eta_{e,\hat{S}}$	0.6977
$\eta_{e,\hat{R}}$	0.6881
$\eta_{i,tot}$	0.2518
η_i	0.4624
ω_i *(deg)*	269.18
a_f *(km)*	26555.9
e_f	0.7278
i_f *(deg)*	63.400
ω_f *(deg)*	269.98
Propellant Used (kg)	82.875
Δv *(km/s)*	5.8308

As predicted, the addition of thrust in the -$\hat{W}$ direction has allowed for a much more optimal solution, with approximately an 8% decrease in propellant used and Δv in comparison to the preliminary test cases. This is in part because more thrust along the $\hat{W}$ axes means that the inclination can be changed at a later and concordantly more efficient

time in the total trajectory, represented by the increase in both $\eta_{i,tot}$ and η_i. This effect can also be seen in Figure 18, by observing that the inclination remains fixed at 28.5 degrees until approximately 10 days into the maneuver.

Aside from the inclination change, the addition of thrust in the $-\hat{W}$ direction also leads to less drift in the argument of periapsis, aiding in a more optimal solution than in preliminary test cases. As can be seen from Figure 19, thrust in both the $\hat{W}$ and $-\hat{W}$ directions leads to a stabilization effect in the argument of periapsis, which may allow for easier satisfaction of its final state equality constraint.

Additionally, it can be seen that the ratio scalar $\eta_{e,\hat{R}}$ converged to a value higher than 0.5. Any convergence above this upper bound indicates that the $\hat{R}$ direction thrust would not be utilized, due to its propellant inefficiency with respect to the current problem formulation. The particular value of its convergence is arbitrary, should it converge to a value above the upper bound in a case such as this. It is hypothesized that should the transfer time be drastically decreased, leading to the value for $\eta_{e,\hat{S}}$ to converge below 0.5, then $\eta_{e,\hat{R}}$ would also converge below 0.5, and cause activation in the $\hat{R}$ direction thrust.

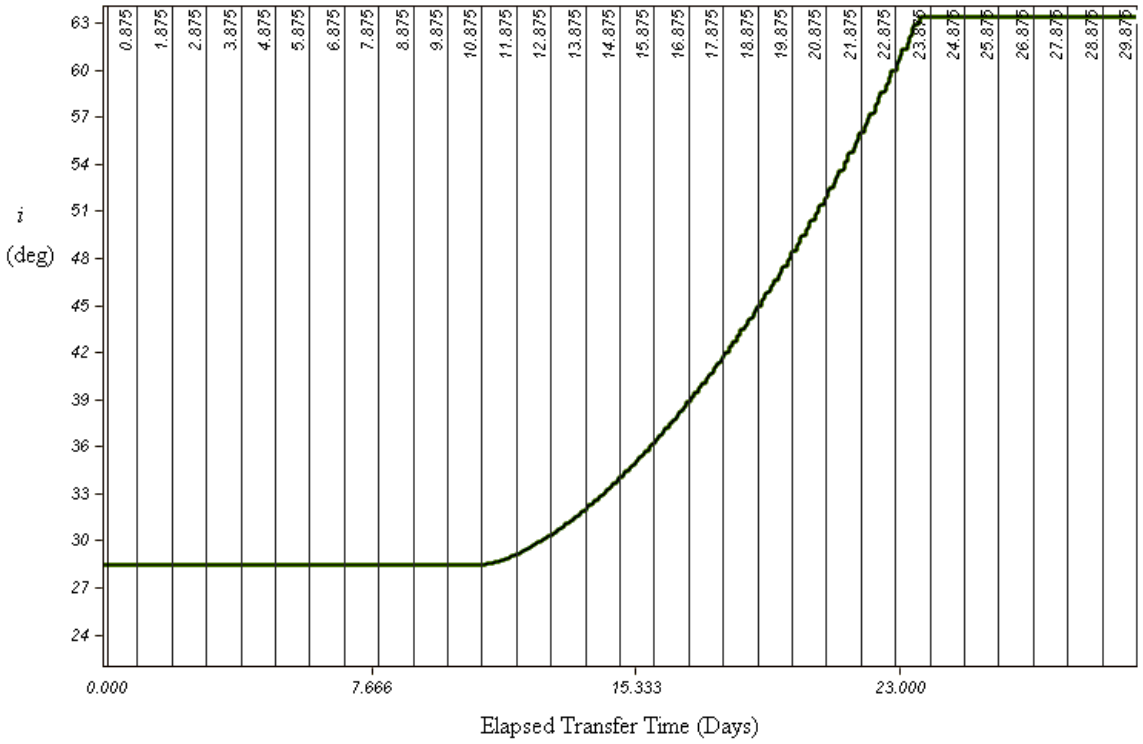


Figure 18: Inclination change during maneuver for optimal LEO to Molniya 30 day transfer problem formulation

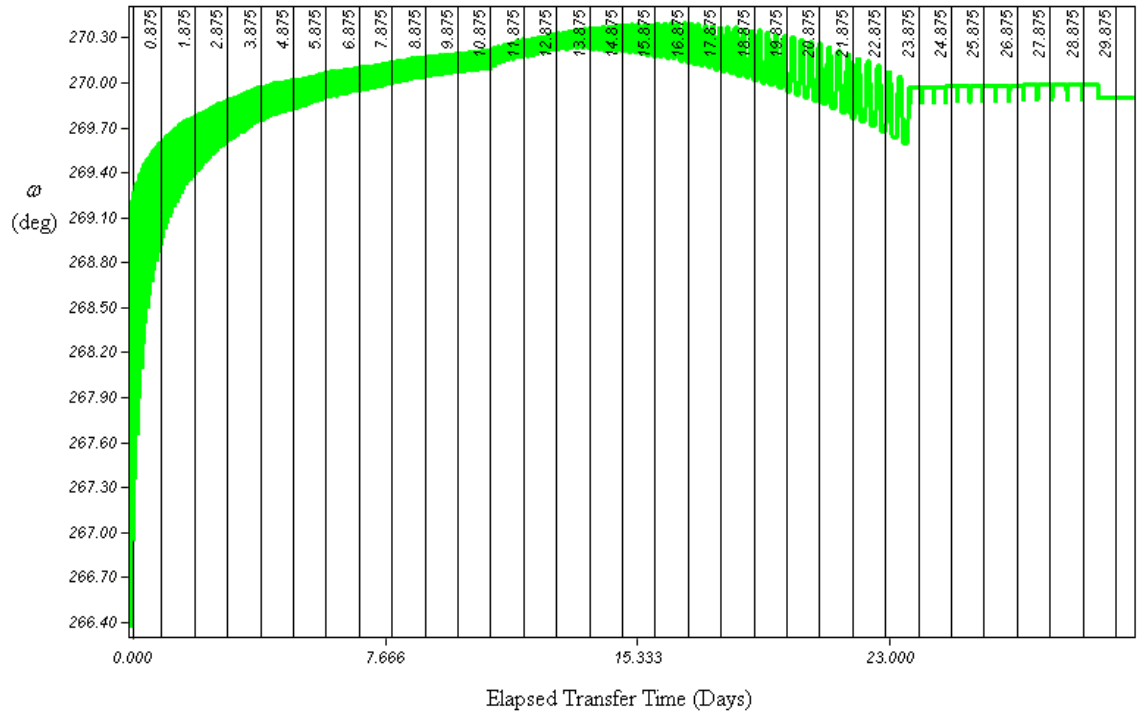


Figure 19: Argument of periapsis change during maneuver for optimal LEO to Molniya 30 day transfer problem formulation

Final Application: Comparison to Optimal Control Method

Due to the positive results from the pervious simulation, the current problem formulation is tested against an optimal LEO to Molniya low-thrust trajectory found in the work of Herman and Spencer. In this work, Herman and Spencer used a standard optimal control problem formulation, and were able to directly solve a cost function minimizing the Δv of several LEO based, continuous thrust-acceleration low-thrust transfers through the use of a Higher Order Collocation 7^{th} degree system. A more detailed explanation of these methods can be found in reference [2].

Table 18: Initial state and satellite properties for LEO orbit, outlined in [2]

Spacecraft/orbit property	Value
Semi-major axis (a)	7003 km
Eccentricity (e)	0
Inclination (i)	28.5°
Right Ascension of Ascending Node (Ω)	0º
Argument of Periapsis (ω)	0°
True Anomaly (ν)	0°
Thrust Acceleration (TA)	0.01 m/s^2

Table 19: Final state properties for Molniya orbit outlined in [2]

Spacecraft/orbit property	Value
Semi-major axis (a)	26578 km
Eccentricity (e)	0.73646
Inclination (i)	63.435°
Right Ascension of Ascending Node (Ω)	Free
Argument of Periapsis (ω)	0°
Total Transfer Time	159.75 hr

In [2], initial and final orbital elements identical to those in Tables 18 and 19 were used. It is the objective of the comparison between the methods described in this work to those in [2] to create competitive optimal results for Δv using identical thrust

accelerations and transfer time, for the given initial and final states. Therefore, a problem formulation for this comparison can be summarized as follows:

Given:

Initial conditions in table 18 and using switching functions outlined in equations (33), (37) and (38) with $F_S = F_R = F_W = 10^{-2}$ *m/s*^{2} *and a total transfer time = 159.75 hr*

Find:

$$\vec{u} = \{\ \eta_{e,\hat{S}},\ \eta_{e,\hat{R}},\ \eta_{i,tot},\ \eta_i,\ \omega_i\ \}$$

To Minimize:

$$F(\vec{u}) = \text{Propellant Use} + w_a \mid a - a_f \mid + w_e \mid e - e_f \mid + w_i \mid i - i_f \mid + w_\omega \mid \omega - \omega_f \mid$$

Subject to:

$$-1 \le \eta_{e,\hat{S}} \le 1$$

$$\eta_{e,\hat{S}} \le \eta_{e,\hat{R}} \le 1$$

$$0 \le \eta_{i,tot} \le 1$$

$$0 \le \eta_i \le 1$$

$$0^\circ \le \omega_i \le 360^\circ$$

$$a_f = 26578 km$$

$$e_f = 0.73646$$

$$i_f = 63.435 deg$$

$$\omega_f = 0 deg$$

One matter of concern when comparing the aforementioned problem formulation to that of Herman and Spencer is the thrust acceleration. In their work, the thrust acceleration was kept constant at 10^{-2} m/s^2, which was done to create a problem in which spacecraft mass and thrust were not necessary to define. However, their work did not use a switching function, so the thrust acceleration was kept at the value of 10^{-2} m/s^2 for the duration of the trajectory. Using the switching function formulation described in these studies, this thrust acceleration will not be constantly applied. Furthermore, since each engine in the $\hat{S}$, $\hat{W}$, and $\hat{R}$ directions will have a thrust acceleration of 10^{-2} m/s^2, the

total thrust acceleration could be anywhere in the range of $0 \le F_{total} \le \sqrt{3}10^{-2}$ m/s^2. Due to these facts, the comparison to reference [2] should be taken as an approximate, and not an exact comparison. Therefore, should the optimal Δv found using the switching function method be slightly better or worse than the value found by Herman and Spencer, it would not be appropriate to conclude which method is indeed superior (unless the discrepancies in optimal Δv are drastic). Instead, should the two methods give similar results, it can be said that the switching function method outlined in this work is competitive to the method of optimal control used by Herman and Spencer.

Using the problem formulation described previously, results from the comparison of the two low-thrust trajectory optimization methods can be seen in Table 20, with plots of semimajor axis, eccentricity and inclination throughout the trajectory in Figures 20, 21, and 22 respectively.

Table 20: Results from CMA-ES implementation of Herman and Spencer problem formulation

nfe	3082
$\eta_{e,\hat{S}}$	-0.164
$\eta_{e,\hat{R}}$	0.1597
$\eta_{i,tot}$	0.4276
η_i	0.5324
ω_i *(deg)*	3.0586
a_f *(km)*	26578.7
e_f	0.7329
i_f *(deg)*	63.449
ω_f *(deg)*	0.0129
Propellant Used (kg)	102.14
Δv *(km/s)*	6.0638
Δv *Herman and Spencer (km/s)*	6.1090
% Change	-0.73 %

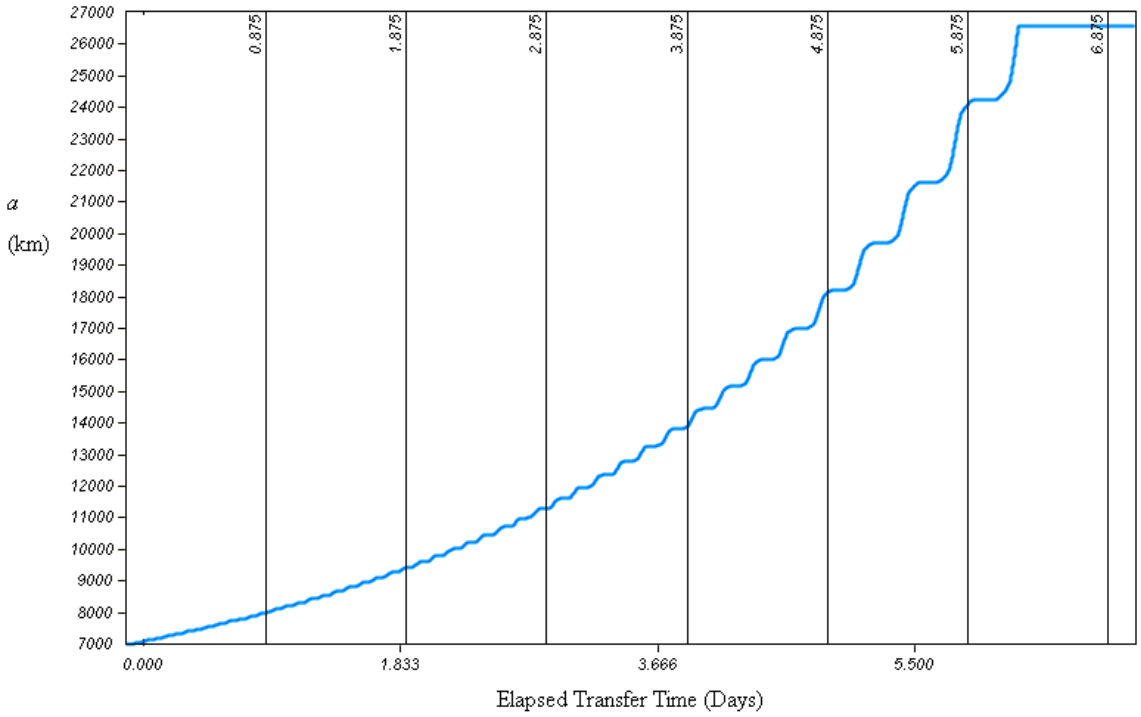


Figure 20: Semimajor axes change during maneuver for Herman and Spencer problem formulation

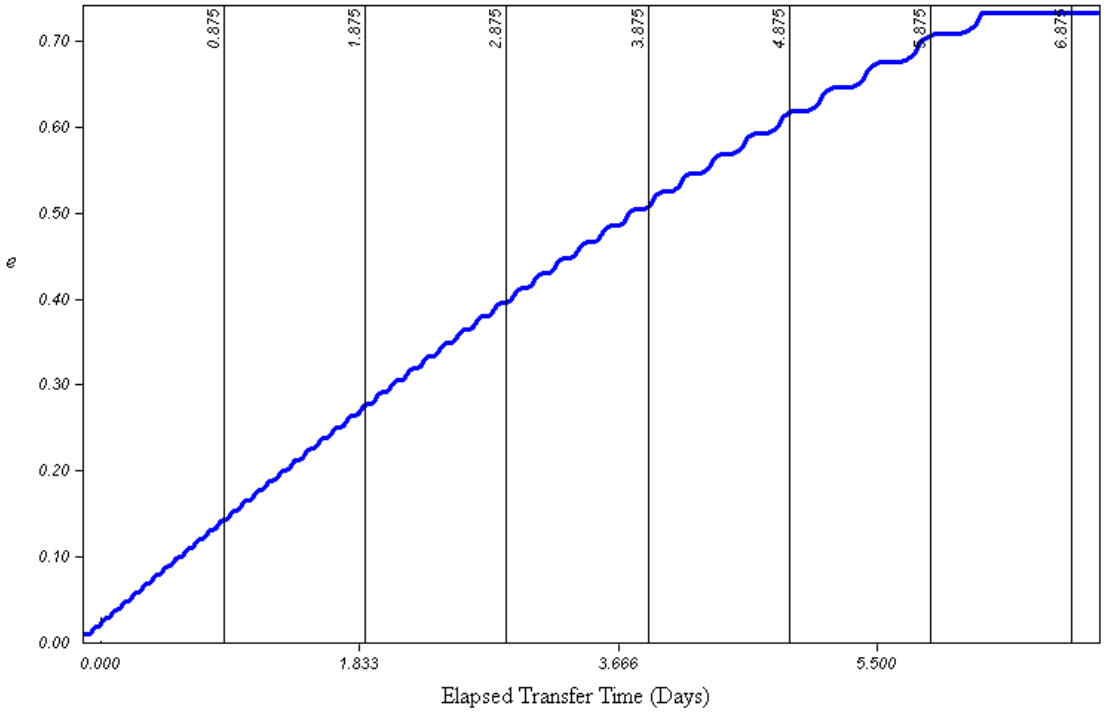


Figure 21: Eccentricity change during maneuver for Herman and Spencer problem formulation

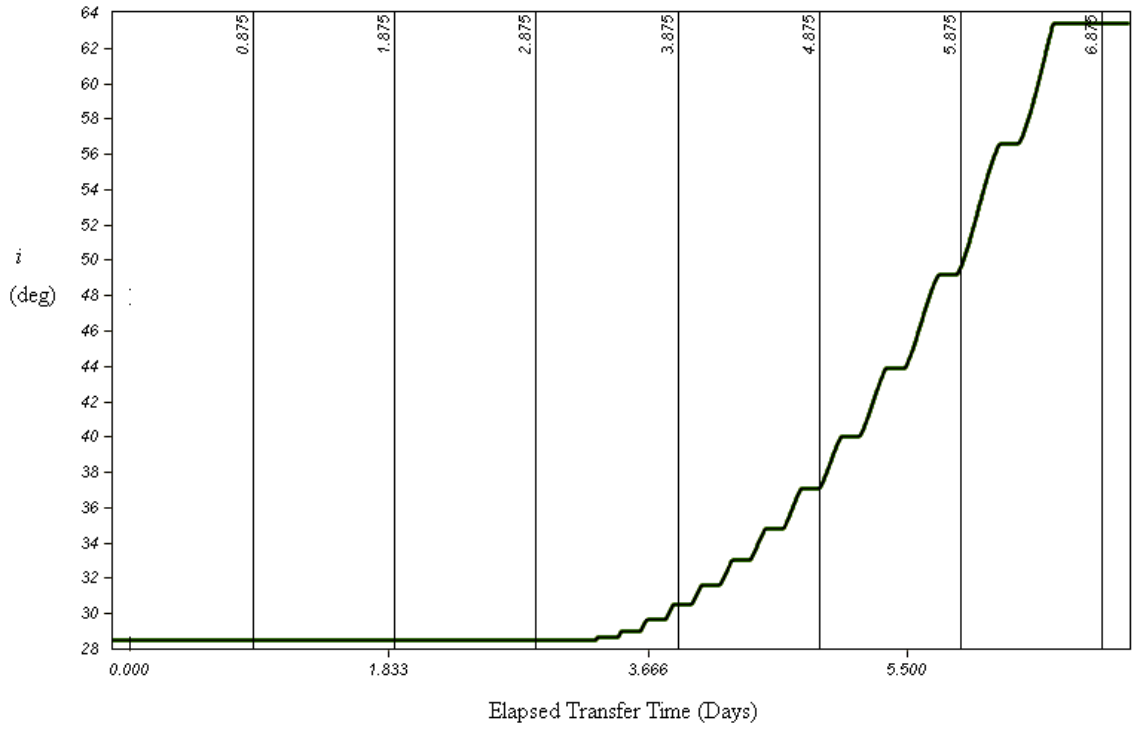


Figure 22: Inclination change during maneuver for Herman and Spencer problem formulation

Based on these results, it can be seen that there is a 0.73% decrease in Δv from the problem formulation in this study as opposed reference [2]. Again, this does not give any definitive results as to which method is better, but it does indicate that the optimization approach through use of general perturbation equation switching functions in conjunction with an evolutionary strategy may be competitive to certain methods of optimal control.

Additionally, Table 20 shows that for the first time in these studies, the criteria for using $\hat{R}$ direction thrust has been met ($\eta_{e,\hat{S}}$ < ~ 0.5), and radial thrusts were initiated. Thus the decrease in transfer time and convergence of $\eta_{e,\hat{S}}$ < 0.5 resulted in the activation of the $\hat{R}$ direction thrust, just as hypothesized.

In terms of the computational costs associated with finding this optimal trajectory using the CMA-ES algorithm, the number of function evaluations (3082) was reflective of roughly a 6 hour time to convergence. While no data in reference [2] was given to show the time to convergence for that study, it should be noted that in practical applications, the convergence time would play a large role in the overall comparison of two optimization methods.

Chapter 6: Summary, Conclusions, and Further Study

Based on the results for both the testing of the CMA-ES and SQP algorithms, several conclusions can be drawn concerning the effectiveness of evolutionary and classical algorithms with regard to solving low-thrust optimization problems within a black box environment. Additionally, the overall usefulness of creating switching functions using general perturbation equations can be determined.

Testing of the CMA-ES and SQP Algorithms

In the preliminary stages of these studies, a proper non-linear constrained optimization algorithm needed to be selected. This algorithm would serve a primary objective of finding optimal solutions to various low-thrust transfer problems, while simultaneously converging to an array of end condition equality constraints. Selection of an algorithm which met these criteria would be based on the consistency of convergence, the optimal value converged upon, the handling of equality constraints, and the overall time to convergence. The algorithms tested in these studies reflected two methods of numerical optimization. The first algorithm, representing a classical, gradient-based method of optimization was an SQP algorithm performed within Matlab's Optimization Toolbox. The second algorithm, based on evolutionary computation was the CMA-ES algorithm created by Hansen. These algorithms were tested on a LEO to 10,000 km semimajor axis low-thrust transfer, which included an inclination change. This problem was reformulated several times in order to get a thorough test of both algorithms, so that a proper assessment of their performance could be made.

In terms of the SQP algorithms performance, convergence was met in all but one test case. Since the SQP algorithm, like any other classical algorithm, is based upon creating search directions from gradients, it can be pre-conditioned to avoid ill convergence within function spaces that are discrete, discontinuous, or contain areas where the gradient cannot be easily calculated. Since optimization in these studies was conducted within a black box, it could be a reasonable assumption that noise within this framework could lead to problems calculating a gradient, resulting in the lone case of non-convergence. In the cases where the SQP algorithm did converge, the optimal values were either on par with those of the CMA-ES algorithm, or they were significantly worse. This phenomena can easily be explained by a classical algorithms pre-disposition to converge to local rather than global extrema. Therefore, if this low-thrust optimization problem had many local minimums, the SQP algorithm would converge to the one it was closest too, and would only converge to a global minimum if it were given a starting point within a proximity of that minimum. The SQP algorithm also handled the equality constraints it was presented with, to only a slightly less accurate tolerance than the CMA-ES algorithm. The one substantial advantage the SQP algorithm had when compared to the CMA-ES algorithm was that its convergence time was significantly lower, and was thus much less computationally expensive.

As for the CMA-ES algorithm, convergence was always met, and performance was rather consistent. The CMA-ES algorithm was shown to consistently converge to an equally, or more optimal solution than the SQP algorithm. This is due to the more thorough search process within the evolutionary strategy, allowing it to observe a larger portion of the function space than the SQP algorithm. However, the CMA-ES algorithm

would converge to inconsistent optimal locations, which had near-consistent optimal values. This was most likely due to low-amplitude noise within the black box framework. Should noise like this be present in the function space, an algorithm would have trouble converging when the noise is greater than or equal to the average objective function change within iterations of the search direction. Therefore, it was decided that the CMA-ES algorithm was converging to a domain for optimal results, and a location of the global minimum was almost "hidden" within some low level of noise. Regardless of this hypothesis, the CMA-ES algorithm still outperformed the SQP algorithm in its optimal solutions in almost every test case.

One drawback to the CMA-ES algorithm was that it did not directly handle equality constraints, which therefore had to be implemented by infusing weighted penalty functions within the objective function. This method of handling equality constraints proved to work, so long as the weight associated with the equality constraint was chosen correctly. In the first test case of the CMA-ES algorithm on the LEO to 10,000 km semimajor axis transfer, the weight chosen for the inclination constraint was too small, and the algorithm did not converge to meet the constraint. Once weights were increased, the equality constraint was consistently satisfied. It was also hypothesized that should the weights get too high, the algorithm may pre-converge, but this behavior was not seen in these preliminary tests. Another drawback to the CMA-ES algorithm was that it was very computationally expensive, and required much time to converge.

Taking into account the results of testing these two algorithms, it was eventually decided that the ability of the CMA-ES algorithm to consistently find global extrema

(within a domain) outweighed all aspects of performance, and thus the CMA-ES algorithm was selected as the best algorithm.

Creation of Switching Functions and Comparison to Optimal Control

Once the appropriate numerical optimization algorithm was chosen, it could be applied to a more complex low-thrust trajectory optimization problem. This problem would seek to create switching functions for transfer burn and coast times which did not rely on manipulating a satellite's state vector, as methods of optimal control do. The creation of these switching functions was based on the use of general perturbation equations, tailored specifically to a LEO to Molniya low-thrust transfer. These switching functions would be used in accordance with a propellant-optimal objective function which imposed weighted penalty functions for equality constraints of the final orbit semimajor axis, eccentricity, inclination and argument of periapsis. Once reformulations were made to these switching functions based on initial tests, they were compared to results for a LEO to Molniya transfer using a method of optimal control to test the overall competitiveness of the methods employed in these studies.

Primarily, these studies showed that when there can be no mathematical manipulation of a satellites' state vector, switching functions to maximize the change in certain orbital elements can be created using general perturbation equations. For the LEO to Molniya transfer studied, using the $\hat{R}\ \hat{S}\ \hat{W}$ coordinate system presented by Vallado, the best set of switching functions used were given in equations *(33)*, *(37)* and *(38)*. In these equations, the ratio scalars $\eta_{e,\hat{S}}$, $\eta_{e,\hat{R}}$, η_i, $\eta_{i,tot}$ and the initial argument of periapsis were control variables which could be inputted into the CMA-ES algorithm. These

switching functions were ideal for the specific LEO to Molniya transfer, due primarily to the fact that the semimajor axes and eccentricity are maximally perturbed at the same locations in an orbit, allowing for the elimination of the $\eta_{a,\hat{S}}$ and $\eta_{a,\hat{R}}$ control variables. In other types of low thrust transfers, the optimal switching functions may change formulations, but the general approach of maximizing the rates of change of certain orbital elements would still apply.

To converge to the final orbital elements of a Molniya orbit, appropriate weights were chosen to be imposed on the penalty functions within the objective function. The weights for these penalty parameters were chosen based on weights which were found to satisfy all equality constraints, without causing pre-convergence or divergence of any constraint variables. However, results showed that equality constraints were met to accurate tolerances, and constraints were rarely met exactly. This was due to both the chosen weights being slightly too small, and to the increased complexity to the function space from the addition of four equality constraints. However, while the constraint convergences were not exact, these were still close enough to the desired final values to be considered acceptable.

When comparing these methods an optimal control method, the technique in this study were found to be effective in providing optimal results for LEO to Molniya low-thrust transfers, competitive with one method of optimal control implemented by Herman and Spencer. However, due to some aspects of the transfer simulated in STK being unable to replicate the transfer Herman and Spencer used, along with possible uncertainties in STK (such as convergence tolerances, truncation error, round-off error, etc) a conclusion as to which method provided a better final solution could not be

determined. However, this initial comparison did show that the methods in this study were at least viable to those of optimal control, barring further study.

Further Study

The findings in these studies will be used to improve this approach to low-thrust trajectory optimization, as well as conduct further testing into its overall competitiveness to optimal control. In particular, problems such as a LEO to GEO, LEO to Moon, and interplanetary problem formulations will be examined to improve the robustness of the switching functions. Additionally, these methods will be tested against different schemes of optimal control, including primer vector theory, to gain more insight as to where these methods would be at an advantage or disadvantage when compared to optimal control.

Furthermore, the CMA-ES algorithm was used more or less to many of its default settings. Further research should be done to find the best possible configuration of this algorithm to these types of optimization problems, as well as find other methods of handling equality constraints, or even optimizing the penalty function weights so that those constraints are met exactly, and not approximately. Also, other evolutionary strategies could be applied to the LEO to Molniya problem formulations, in an attempt to find the most robust algorithm possible.

Bibliography

[1] Vallado, D.A., *Fundamentals of Astrodynamics and Applications.* Microcosm Press, 2007. pp 625-633.

[2] Herman, A.L. and Spencer, D.B. "Optimal, Low-Thrust Earth-Orbit Transfers Using Higher-Order Collocation Methods", *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, Jan. 2002, pp 40-47.

[3] Lawden, D.F., *Optimal Trajectories for Space Navigation*, Butterworths, London, 1963. p 16.

[4] The Mathworks, *Optimization Toolbox Documentation*. Version 3.1.1 (R2007a), 2007.

[5] Analytical Graphics, *Satellite Toolkit Documentation*. http://www.stk.com. Release 8.1.1, 2007. Cited 2009

[6] Curtis, H.D., *Orbital Mechanics for Engineering Students*. Elsevier, Oxford, 2005. pp 182-183.

[7] Hansen, N., *The CMA Evolution Strategy: A Tutorial.* http://www.lri.fr/~hansen/cmatutorial.pdf. 2007. p 3. Cited 2009

[8] Fletcher, R. *Practical Methods of Optimization,* John Wiley and Sons, 1987. pp 277-280

[9] Nocedal, J., and Wright, S.J. *Numerical Optimization*, Springer, 2006. p 498.

[10] Gill, P.E, Murray, W., and Wright, M.H., *Practical Optimization*, Academic Press, 1981. pp 207-209.

[11] Williams, Patrick S, and Spencer, David B. "Preliminary Findings Concerning Applications of Non-Linear Constrained Optimization Methods on Low-Thrust Orbit Transfers" *AAIA/AAS Astrodynamics Specialist Conference*, Honolulu, HI, August 2008. pp 1-24.

[12] Williams, P.S, and Spencer, D.B. "Applications of an Evolutionary Strategy on General Perturbation Equations for Optimization of Low-Thrust Near-Earth Orbit Transfers" *AAS Spaceflight Mechanics Meeting*, Savannah, GA, February 2009, pp 1-20.

[13] Bryson, A.E, and Ho, Y. *Applied Optimal Control*, Blaisdell Publishing Company, 1969, pp 65-69.