The Pennsylvania State University

The Graduate School

College of Engineering

OPTIMAL ORBIT RAISING VIA PARTICLE SWARM OPTIMIZATION

A Thesis in

Aerospace Engineering

by

Ghanghoon Paik

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

May 2015

The thesis of Ghanghoon Paik was reviewed and approved* by the following:

Robert G. Melton

Professor of Aerospace Engineering

Thesis Advisor, Chair of Committee

David B. Spencer

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

A spacecraft in one orbit may need to move to another orbit. Apoapsis orbit raising, in particular, takes a spacecraft from a circular orbit to an elliptical orbit by thrusting at a periapsis. This technique was applied as the initial stage for the lunar (LADEE), GEO (ARTEMIS), and interplanetary (Mangalyaan) missions to save propellant usage and raise the apoapsis distance of the orbits. These projects show that the apoapsis orbit raising can be applied to various types of missions.

In this thesis, by applying the Particle Swarm Optimization (PSO) algorithm to the five finite thrust maneuvers, evaluation of an optimal solution that derives optimized propellant usage is presented. Each transfer orbit pushes out the apoapsis of the trajectory, depending on the thrust duration and the thrust-on location. The final orbit of the optimal solution of the problem should meet two criteria: the line of apside (LOA) alignment and the apoapsis distance.

The PSO is a computational method that is inspired by a swarm movement. By sharing information obtained by each member, the entire swarm (set of possible solutions) can find the best location efficiently and rapidly. This algorithm highly depends on size of a swarm and number of iterations as well as an initial solution set. In this thesis, a modification is applied to the PSO to handle equality constraints.

The PSO application to apoapsis orbit raising shows the feasibility of determining an optimized trajectory to reach a target orbit and gives required propellant for each maneuver in terms of a thrust duration. The optimal results are acquired by the PSO algorithm and all the requirements are satisfied.

Table of Contents

List of Figures vi

List of Tables vii

List of Symbols viii

Acknowledgments ix

Chapter 1
Introduction 1

Chapter 2
Particle Swarm Optimization and Apoapsis Orbit Raising 4
2.1 Particle Swarm Optimization 4

Chapter 3
Apoapsis Orbit Raising Application 9
3.1 Orbit Raising with various final radii 9
3.2 Problem definition 10

Chapter 4
Result and Discussion 14
4.1 Results 14
4.2 Discussion 20

Chapter 5
Conclusion and Future Work 22
5.1 Conclusion 22
5.2 Future Work 23

Appendix A
Plots of objective function vs. iteration 24
A.1 Objective function over iterations at $\beta = 1.5$ 24
A.2 Objective function over iterations for $\beta = 2$ 25
A.3 Objective function over iterations for $\beta = 5$ 26
A.4 Objective function over iteration for $\beta = 8$ 27
A.5 Objective function over iterations for $\beta = 10$ 28

Appendix B
Trajectory plots and properties 29
B.1 Properties of the trajectory for $\beta = 1.5$ 29
B.2 Properties of the trajectory for $\beta = 2$ 31
B.3 Properties of the trajectory for $\beta = 5$ 35
B.4 Properties of the trajectory for $\beta = 8$ 37
B.5 Properties of the trajectory for $\beta = 10$ 40

References **43**

List of Figures

2.1 Flow chart of basic PSO algorithm 7

3.1 Line of apsides for two types of transfer orbits 10
3.2 Definition of thrust angle δ 12

4.1 Value of objective function over number of iteration for $\beta = 1.5$ with 5 independent orbits 16
4.2 Trajectory of orbit for $\beta = 1.5$ with thrust-off locations 18
4.3 Angular displacement and apoapsis distance for thrusts 19
4.4 Paths of burn arcs for $\beta = 1.5$ 19
4.5 Thrust pointing angle of burn arcs for $\beta = 1.5$ 20
4.6 Percent error of LOA alignment and distance of final orbit over number of iteration 21

A.1 Value of objective function over number of iterations for $\beta = 1.5$ with 5 independent orbits 24
A.2 Value of objective function over number of iterations for $\beta = 2$ with 5 independent orbits 25
A.3 Value of objective function over number of iterations for $\beta = 5$ with 5 independent orbits 26
A.4 Value of objective function over number of iterations for $\beta = 8$ with 5 independent orbits 27
A.5 Value of objective function over number of iterations for $\beta = 10$ with 5 independent orbits 28

B.1 Trajectory of orbit for $\beta = 1.5$ with thrust-off locations 29
B.2 Paths of burn arcs for $\beta = 1.5$ 30
B.3 Thrust pointing angle of burn arcs for $\beta = 1.5$ 30
B.4 Angular displacement and apoapsis distance for thrusts 31
B.5 Trajectory of orbit for $\beta = 2$ with thrust-off locations 32
B.6 Paths of burn arcs for $\beta = 2$ 32
B.7 Thrust pointing angle of burn arcs for $\beta = 2$ 33
B.8 Angular displacement and apoapsis distance for thrusts for $\beta = 2$ 34
B.9 Trajectory of orbit for $\beta = 5$ with thrust-off locations 35
B.10 Paths of burn arcs for $\beta = 5$ 36
B.11 Thrust pointing angle of burn arcs for $\beta = 5$ 36
B.12 Angular displacement and apoapsis distance for thrusts for $\beta = 5$ 37
B.13 Trajectory of orbit for $\beta = 8$ with thrust-off locations 38
B.14 Paths of burn arcs for $\beta = 8$ 38
B.15 Thrust pointing angle of burn arcs for $\beta = 8$ 39
B.16 Angular displacement and apoapsis distance for thrusts for $\beta = 8$ 39
B.17 Trajectory of orbit for $\beta = 10$ with thrust-off locations 40
B.18 Paths of burn arcs for $\beta = 10$ 41
B.19 Thrust pointing angle of burn arcs for $\beta = 10$ 41
B.20 Angular displacement and apoapsis distance for thrusts for $\beta = 10$ 42

List of Tables

3.1 Initial and target orbit relationship with β 10

4.1 c and n_0 with respect to β 15
4.2 Average percent error of LOA alignment and distance and objective function value 17
4.3 Error rate of LOA alignment and distance and objective function value for $\beta = 1.5$ 17
4.4 Thrust duration and LOA displacement for $\beta = 1.5$ 18

A.1 Error rate of LOA alignment and distance and objective function value for $\beta = 1.5$ 25
A.2 Error rate of LOA alignment and distance and objective function value for $\beta = 2$ 25
A.3 Error rate of LOA alignment and distance and objective function value for $\beta = 5$ 26
A.4 Error rate of LOA alignment and distance and objective function value for $\beta = 8$ 27
A.5 Error rate of LOA alignment and distance and objective function value for $\beta = 10$ 28

B.1 Thrust duration and LOA displacement for $\beta = 1.5$ 31
B.2 Thrust duration and LOA displacement for $\beta = 2$ 33
B.3 Thrust duration and LOA displacement for $\beta = 5$ 35
B.4 Thrust duration and LOA displacement for $\beta = 8$ 37
B.5 Thrust duration and LOA displacement for $\beta = 10$ 40

List of Symbols

c_C	cognitive weighting coefficient
c_I	inertial weighting coefficient
c_S	social weighting coefficient
J	objective (or cost) function
$\tilde{J}$	objective function with penalty terms
N	number of particles
N_{IT}	number of iterations
n	number of parameters
α_r	weighting coefficient for penalty term r
d	equality constraint
β	target-initial orbit ratio
μ	standard gravitational constant [DU^3/TU^2]
T/m	thrust-to-mass ratio
δ	thrust pointing angle [rad or degree]
ϑ	coefficient of thrust pointing angle function
c	thrust level
n_0	thrust-mass ratio at t_0
a	semi-major axis [DU]
e	eccentricity
f	true anomaly [rad or degree]
E	eccentric anomaly [rad or degree]
M	mean anomaly [1/TU]
T	period of the orbit [TU]
r	radius of the orbit [DU]
v_r	radial velocity of the spacecraft [DU/TU]
ξ	angular displacement from x-axis [rad or degree]
v_θ	transverse velocity of the spacecraft [DU/TU]

Acknowledgments

I would like to acknowledge people who supported me during my writing of this thesis. First of all, I would like to thank my advisor, Dr. Robert G. Melton, for his continued help and support throughout the process. It would not have been able to finish this work without his patience, encouragement, and advice. I would also like to thank Dr. Cengiz Camci, Dr. David B. Spencer, and Dr. Michael M. Micci for their support of me as a graduate student and teaching me. Dr. George A. Lesieutre, the head of Aerospace Engineering department, deserves a special thanks for his strong leadership.

Additionally, I would like to thank my friends, including Kaushik Basu, Vidullan Surendran, Bradley J. Sottile, who made a lot of contributions and have been nice friends. As a great friend and teaching assistant, Kaushik Basu gave me advice and helped me. Vidullan Surrendran helped me with programming related problems and was a nice friend. Bradley J. Sottile was a helpful teaching assistant and a friend since my undergraduate years.

Finally, I would like to thank my parents and family who have done more than enough. This work could never have been completed without their support. They provided me such a great opportunity to study in the U.S. Their devotion and love brought me here and made everything I have done possible.

I would also like to thank others not mentioned here, including non-departmental friends and friends back home, who gave me great help throughout the process.

Chapter 1

Introduction

For orbit transfer, minimizing propellant usage is one of the major concerns. To achieve the best possible result, impulsive maneuvers would be optional since they do not lose any velocity (gravity loss). Unlike the impulsive maneuver, the finite thrust has two different velocity components, the radial and the transverse, which cause loss. However, a true impulsive maneuver is a mathematical model and finite burns are performed in a real world situation, which implies a shorter thrust has better efficiency. Therefore, employing short several thrusts model could be a solution to save propellant [1–3].

Particle Swarm Optimization (PSO) is a heuristic and swarm intelligence method that was first introduced by Eberhart and Kennedy in 1995. To optimize a solution, this technique mimics behavior of birds flocking. Each bird in a flock or a swarm is called a particle and all the particles have individual position and velocity vectors. The position vector is associated with unknown parameters that determine a possible solution and the velocity vector updates the position vector over iterations. The optimal solution is defined by the social behavior which is represented by a local best position ($pBest$) and a global best position ($gBest$). All the swarm members share the information about the best positions ever visited and the global best position is selected among the positions collected. The random generation process and the iteration dependency are a stochastic characteristic of an evolutionary programming. The movement of the particle depends on the inertial and the cognitive effects which attract the particle toward either the local or the global best position respectively based on the weight of the effects. Though it is possible to land on a local solution or a random bad solution, the advantages of the PSO algorithm are simplicity of coding and relatively low computational cost [4–10].

Transferring a spacecraft from one orbit to another is an essential part of most space missions. When an initial orbit is a circular or an elliptical and a spacecraft transfers to an elliptical orbit, either an apoapsis or a periapsis burn technique can be applied. Both techniques use a very similar concept which is turn the thruster on only at the apoapsis or the periapsis. In this thesis, only the apoapsis orbit raising, or periapsis burn, method is covered. However, work related to apoapsis burn technique applied to circular orbit transfers are also reported [11, 12]. The apoapsis orbit raising concept was used in couple of space missions such as the Lunar Atmosphere and Dust Environment Explorer (LADEE) of the National Aeronautics and Space Administration (NASA), the Acceleration, Reconnection, Turbulence and Electrodynamics of the Moon's Interaction with Sun (ARTEMIS) of European Space Agency (ESA), and the Mangalyaan of Indian Space Research Organization (ISRO) [13–19]. The method was applied in the first stage of the maneuvers to save propellant usage before the spacecraft begins to move further. The application in the LADEE mission raised the orbit to reach the moon then the spacecraft inserted into the lunar orbit [13,14]. For ARTEMIS, apoapsis orbit raising pushed the satellite out to the Geosynchronous Equatorial Orbit (GEO) and then it made another propulsive maneuver to circularize the orbit [15–17]. The Mangalyaan mission used a periapsis burn technique before the spacecraft departed to Mars using a Hohmann transfer [18, 19].

The work done by Pontani and Conway on the use of a PSO algorithm in trajectory design forms the basis for the method in this thesis [6]. The paper was published in 2010 and it gives idea of optimization of several different type of orbits. Especially, this thesis is influenced by the Chapter 5 which is 'Optimal Finite Thrust Orbital Transfer Between Two Circular Orbits' since the chapter is introducing finite thrust orbit maneuver between orbits. The initial conditions and elliptical thrust arc are applied into the thesis method. Some equations are modified from the original paper and employed.

The objective of this work is to optimize propellant usage while trasferring a spacecraft from a circular orbit to an elliptical orbit. To get an optimized solution, a few assumptions are made:

1) no perturbation effects, 2) only a central gravitational force exists (2-body problem), 3) the engines on the spacecraft have no throttling capability, and 4) the transfer and the final orbits have semi-major axes larger than that of the initial orbit. The minimum line of apsides (LOA) rotation is considered since that is critical for combined LEO-GEO transfers with a plane-change maneuver [1]. The PSO gives some large LOA rotations on the intermediate ellipses, but the final elliptical path has very small rotation.

Chapter 2

Particle Swarm Optimization and Apoapsis Orbit Raising

2.1 Particle Swarm Optimization

The basic Particle Swarm Optimization (PSO) algorithm is based upon the behavior of swarms of birds searching for food or shelter. Each member of a swarm is called a particle and is represented by a vector of n number of parameters that essentially define the solution. Each particle chases two different targets, one that the particle set by itself and another by the entire swarm. Each particle moves toward one target that weighs more than the other. The objective function (or cost function) gives a reference to tell whether the outcome is considered to be optimal or not. Achieving the minimized objective function value without violating constraints is the goal of the PSO. Details about the PSO algorithm are described in the following paragraphs.

Particle Swarm Optimization requires the initial position of each particle to begin with. The initial position, P_k, of each particle with n unknown parameters is randomly generated within the range of the lower and the upper bound, B_{L,P_k} and B_{U,P_k}:

$$B_{L,P\,k} \leq P_k \leq B_{U,P\,k} \qquad (k = 1, \ldots, n) \tag{2.1}$$

The number of particles in a swarm (N) gives the population size of a position vector and a velocity vectors, and i-th particle of each vector is represented by $\boldsymbol{P}(i)$ and $\boldsymbol{V}(i)$, respectively.

$$\boldsymbol{P}(i) \triangleq [P_1(i) \quad \ldots \quad P_n(i)]^T \qquad (i = 1, \ldots, N) \tag{2.2}$$

Each velocity vector component $V_k(i)$ has minimum and maximum values that come from limits of position vector components in Eq. (2.1). The range of the velocity vector components is defined

as:

$$-(B_{U,P_k} - B_{L,P_k}) \leq V_k \leq (B_{U,P_k} - B_{L,P_k}) \tag{2.3}$$

$$B_{L,V_k} \leq V_k \leq B_{U,V_k} \tag{2.4}$$

Since PSO is mimicking dynamic movement of a swarm, both position and velocity of every particle change over iteration j, where $j = 1, \ldots, N_{IT}$. The new position of a particle $P_k^{(j+1)}(i)$ is defined by adding its velocity to the current position:

$$P_k^{(j+1)} = P_k^{(j)} + V_k^{(j)} \qquad (j = 1, \ldots, N_{IT}) \tag{2.5}$$

In this context, velocity is the rate of change of a particle per iteration. If an updated position violates Eq. (2.1), the new position is set to the lower bound if a value is smaller than the lower limit, or the upper bound if a value is greater than the upper limit . The same process is applied to velocity components to keep values within the proper range [5–7].

Each particle of the swarm has both global and particle optimal positions which are related to global and particle optimal solution of a problem. The particle optimal or particle best position (*pBest*) relies on the best position that a particle has ever visited before. The global best position (*gBest*) is, then, one of the *pBest* values of the swarm that minimizes the objective function [4–7]. The global best position of the swarm is shared with all the particles in the swarm [4, 6]. Finding optimal positions, *pBest* and *gBest*, are based upon the objective function J, where the objective function is minimized such that the position that gives minimized value is picked as the best position [4, 6, 7]. The objective function $J^{(j)}(i)$ for *pBest* of particle i at j-th iteration is:

$$J^{(j)}(i) = min\ J^{(1,\ldots,j)}(i) \tag{2.6}$$

where (*i=1, . . . , N*) then, *pBest* is determined as:

$$\boldsymbol{pBest}^{(j)}(i) = \boldsymbol{P}^{(l)}(i) \qquad \left(l = arg\ min_{p=1,\ldots,j}\ J^{(p)}(i)\right) \tag{2.7}$$

Similarly, the best J among all the swarm is:

$$J_{Best}^{(j)}(i) = min\ J^{(1,\ldots,j)}(i) \tag{2.8}$$

and the *gBest* is:

$$\boldsymbol{gBest}^{(j)} = \boldsymbol{pBest}^{(j)}(q) \qquad \left(q = arg\ min_{i=1,\ldots,N}\ J_{Best}^{(j)}(i)\right) \tag{2.9}$$

The velocity vector also needs to be updated after each iteration by combining three effects: inertial, cognitive, and social. The inertial component tends to keep a particle's velocity unchanged, while the cognitive effect tries to move a particle toward its *pBest* and the social effect leads a particle toward the *gBest* of a swarm [6, 7, 10]. Each of them has its own weight expressed in order:

$$c_I = \frac{1 + r_1(0,1)}{2} \qquad c_C = 1.49445\, r_2(0,1) \qquad c_s = 1.49445\, r_3(0,1) \tag{2.10}$$

where $r_1(0,1)$, $r_2(0,1)$, and $r_3(0,1)$ are random numbers with uniform probability distribution between 0 and 1. Then, the velocity component of a particle is updated by:

$$V_k^{(j+1)}(i) = c_I V_k^{(j)}(i) + c_C \left[pBest_k^{(j)}(i) - P_k^{(j)}(i)\right] + c_S \left[gBest_k^{(j)} - P_k^{(j)}(i)\right] \tag{2.11}$$

The next step is updating the position vector of N particles for all $P_k(i)$ as Eq. (2.5). However, if a new position violates the limits in Eq. (2.1), the velocity of the iteration is set to 0 as [6]:

$$P_k^{(j+1)} = \left\{ \begin{array}{ll} B_{L,P_k} & \text{if} \quad P_k^{(j+1)} < B_{L,P_k} \\ B_{U,P_k} & \text{if} \quad P_k^{(j+1)} > B_{U,P_k} \end{array} \right\} \quad \text{and} \quad V_k^{(j+1)}(i) = 0.$$

Evaluating *gBest* and *pBest* and updating position and velocity vectors are repeated until the number of iteration, N_{IT}, is reached [4–7]. The process of the basic PSO algorithm mentioned above is illustrated in Figure 2.1.

The basic PSO algorithm is able to optimize an unconstrained problem but cannot be used to solve constrained problems. Since engineering problems are mostly constrained optimization

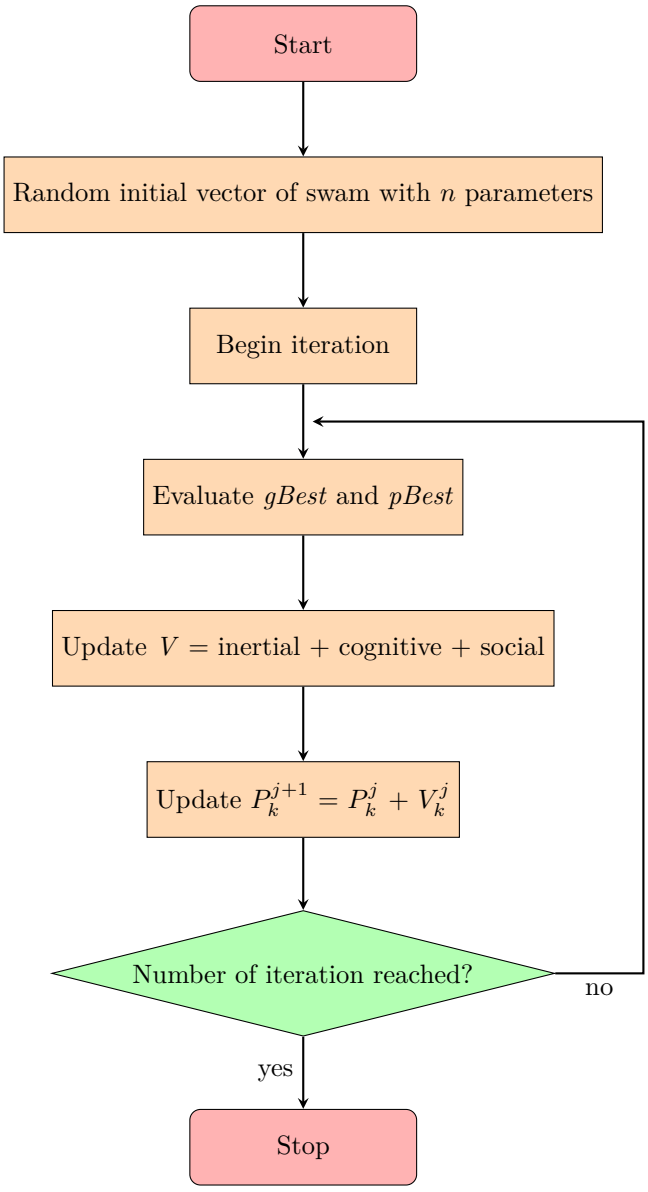

Figure 2.1. Flow chart of basic PSO algorithm

problems, ways to solve constrained problems should be introduced [5, 6]. The basic algorithm requires modification to solve such problems.

An evolutionary computation method, such as PSO, does not easily handle equality constraints. Since the degree of freedom of a problem is reduced by the number of equality constraints m, the number of parameters n is restricted to $m = n$ even if $(m \leq n)$ [6]. Equality constraints would

have the form

$$d_r(\boldsymbol{P}) = 0 \qquad (r = 1, \ldots, m) \tag{2.12}$$

and would require reformulating the problem to make each particle have only m elements. Penalizing constraint violations by adding extra terms to the objective function is a widely used method to deal with constraints, and is also used in this work. Weighting coefficients multiply the constraints and are added to the original objective function:

$$\tilde{J} = J + \sum_{r=1}^{m} \alpha_r |d_r(\boldsymbol{P})| \tag{2.13}$$

Coefficients α_r are problem-dependent and one may find suitable values empirically. Bigger α_r value indicates more weighting on a constraint, and smaller α_r means it is less weighted [6]. A similar method can be used to handle inequality constraints.

Chapter 3

Apoapsis Orbit Raising Application

Apoapsis orbit raising is an orbit maneuver that is possible using periapsis burns. Instead of a single burn technique, the method in this thesis employs multiple finite thrusts. Optimal thrust duration and location to turn on the propulsion system to achieve the target orbit is obtained by using a PSO algorithm.

3.1 Orbit Raising with various final radii

Periapsis-burn orbit raising in this work involves pushing out the apoapsis location while a spacecraft is orbiting around one focal point. The initial orbit of the problem is circular and has radius of 1 (all distance are scaled by the initial circular radius). After passing through several transfer orbits, the spacecraft reaches the elliptical target orbit. Assume the focal point is located at the center of the x-y plane, R_a refers to the distance between the center and the apoapsis of the elliptical orbit. Then, let β equal the ratio of R_a of the target orbit to R_a of the initial orbit.

$$\beta = \frac{R_{a,target}}{R_{p,initial}} = \frac{a_{target}\,(1 + e_{target})}{a_{initial}\,(1 - e_{initial})} \tag{3.1}$$

when e is the eccentricity $0 \leq e \leq 1$ and a is the semi-major axis. The line of apside (LOA) of transfer orbits may or may not rotate in order to get some optimal result and also R_a of the transfer orbit may not increase constantly. However, the amount of LOA rotation and rate of R_a increment are possibly specified based on the problem definition. Figure 3.1 gives the basic idea of two different types of trajectories with different limitations. The right side of the Fig 3.1 describes the apoapsis orbit raising technique studied in this thesis, when intermediate transfer ellipses have significant roration of the LOA. Table 3.1 is the range of β values studied in this

work.

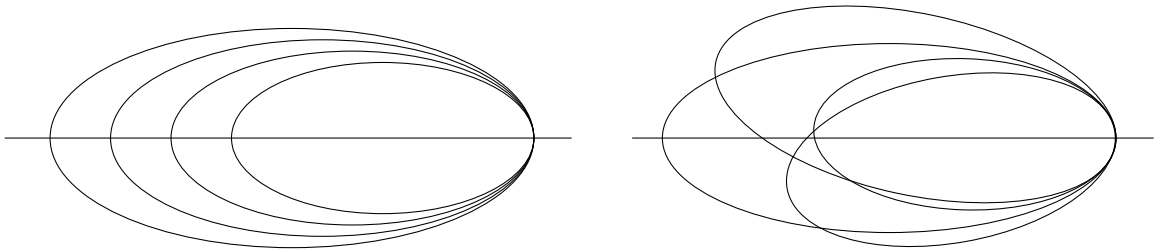

Figure 3.1. Line of apsides for two types of transfer orbits

Table 3.1. Initial and target orbit relationship with β

β	Initial orbit	Target orbit
1.5	1	1.5
2	1	2
5	1	5
8	1	8
10	1	10

3.2 Problem definition

This section deals with optimization of multi-path finite thrust transfer between initial and target orbits using a PSO algorithm. Several equations and conditions come from Pontani and Conway [6].

Beginning with a circular orbit with radius R_1, the spacecraft moves along four transfer orbits and reaches the final orbit with $R_{a,target}$ represented as R_2, where $R_2 > R_1$ and $\beta \triangleq R_2/R_1 > 1$. When r, v_r, ξ, and v_θ denote radius, radial velocity component, angular displacement from x-axis, and transverse velocity component, respectively, initial and final conditions at t_0 and t_f are given by:

$$r(t_0) = R_1 \qquad v_r(t_0) = 0 \qquad \xi(t_0) = 0 \qquad v_\theta(t_0) = \sqrt{\frac{\mu}{R_1}} \tag{3.2}$$

$$r(t_f) = R_2 \qquad v_r(t_f) = 0 \qquad v_\theta(t_f) = \sqrt{\frac{\mu}{R_2}} \tag{3.3}$$

μ in Equation (3.2) and (3.3) are a normalized gravitational parameter of body at the center ($\mu = 1$ DU^3/TU^2). The system of units employed here is as follows: 1) distance unit (DU) = the radius of the initial circular orbit and 2) time unit (TU) = $\frac{\text{period of the initial circular orbit}}{2\pi}$.

A spacecraft flying on this trajectory experiences a total of five propulsive and five non-propulsive maneuvers. A spacecraft employing engines without throttling capability is assumed. So, they are only turned on at maximum thrust or completely off. The thrust-to-mass ratio (T/m) of the engine with thrust level T is:

$$\frac{T}{m} = \begin{cases} \dfrac{T}{m_0 - (T/c)(t_{total} + t)} = \dfrac{cn_0}{c - n_0(t_{total} + t)} & \text{if} \quad 0 \le t \le \Delta t_i \quad (i = 1, \ldots, 5) \\ 0 & \text{if spacecraft is coasting} \end{cases} \tag{3.4}$$

In Equation (3.4), t_{total} is the sum of all previous thrust durations Δt_i (if $i = 1$, $t_{total} = 0$), c represents the effective thrust exhaust velocity, and n_0 and m_0 denote the thrust-to-mass ratio and the mass of the spacecraft at t_0. The state-space equations for r, v_r, ξ, and v_θ for the motion of the spacecraft are:

$$\dot{r} = v_r \tag{3.5}$$

$$\dot{v_r} = -\frac{\mu - r v_\theta^2}{r^2} + \frac{T}{m}\sin\delta \tag{3.6}$$

$$\dot{\xi} = \frac{v_\theta}{r} \tag{3.7}$$

$$\dot{v_\theta} = -\frac{v_r v_\theta}{r} + \frac{T}{m}\cos\delta \tag{3.8}$$

and the state vector is $\boldsymbol{x} = [r \quad v_r \quad \xi \quad v_\theta]^T$. In order to apply finite thrust along the path, a thrust vectoring technique is used, and the angle δ in Equation (3.6) and (3.8) is the thrust angle defined in Figure 3.2 represented as a third degree polynomial as a function of time. The equation for δ has four coefficients ϑ_0, ϑ_1, ϑ_2, and ϑ_3 and the PSO algorithm gives optimal values of $\vartheta_{k,i}$ (k = 0, . . . , 3).

$$\delta_i = \vartheta_{0,i} + \vartheta_{1,i}\, t + \vartheta_{2,i}\, t^2 + \vartheta_{3,i}\, t^3 \qquad \text{if} \quad 0 \le t \le \Delta t_i \tag{3.9}$$

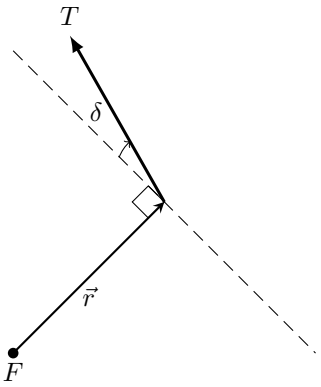

Figure 3.2. Definition of thrust angle δ

Thus, during each thrust arc, δ_i gives the thrust angle.

The flight time on a coasting arc t_{CO} is assumed to be equal to the time duration of the previous thrust arc T_i unless the thrust is turned off below or above the x-axis by a certain amount. The period of i-th thrust arc, T_i, is initially guessed as:

$$T_i = 2\,\pi\sqrt{\frac{a^3}{\mu}} \tag{3.10}$$

where the semi-major axis a is:

$$a = \frac{\mu r_i}{2\mu - r_i(v_{r,i}^2 + v_{\theta,i}^2)} \tag{3.11}$$

when r_i, $v_{r,i}$, ξ_i, and $v_{\theta,i}$ refer to values of each variable at time Δt_i. If the thrust-off location ξ_i is within $\pm\,5$ degree range, t_{co} is set to T_i but if it is either 5 - 60 degrees or 300 - 355 degrees range, adjustment is applied as:

$$t_{CO} = \begin{cases} T_i - \dfrac{M}{2\,\pi\,T_i} & \text{if} \quad 5^\circ \le \xi_i \le 60^\circ \quad \text{or} \quad \pi/\,36 \le \xi_i \le \pi/\,3 \\ T_i + \dfrac{M}{2\,\pi\,T_i} & \text{if} \quad 300^\circ \le \xi_i \le 355^\circ \quad \text{or} \quad -\,\pi/\,3 \le \xi_i \le -\pi/\,36 \\ T_i & \text{if} \quad 355^\circ \le \xi_i \le 5^\circ \quad \text{or} \quad -\,\pi/\,36 \le \xi_i \le \pi/\,6 \end{cases} \tag{3.12}$$

Mean anomaly M in Eq (3.12) is evaluated by eccentricity e, true anomaly f, and eccentric

anomaly E as follows:

$$e = \sqrt{1 - \frac{r_i^2 v_{\theta,i}^2}{\mu\, a}} \tag{3.13}$$

$$f = \tan^{-1}\left(\frac{\sin f}{\cos f}\right) = \frac{v_{r,i}\sqrt{\dfrac{a(1-e^2)}{\mu}}}{v_{\theta,i}\sqrt{\dfrac{a(1-e^2)}{\mu}} - 1} \tag{3.14}$$

$$E = \tan^{-1}\left(\frac{\sin E}{\cos E}\right) = 2\tan^{-1}\left(\tan\left(\frac{f}{2}\right)\sqrt{\frac{1-e}{1+e}}\right) \tag{3.15}$$

$$M = E - e\sin E \tag{3.16}$$

The simplified equations (3.14) and (3.15) are derived from

$$\sin f = \frac{v_{r,1}}{e}\sqrt{\frac{a(1-e^2)}{\mu}} \qquad \text{and} \qquad \cos f = \frac{v_{\theta,1}}{e}\sqrt{\frac{a(1-e^2)}{\mu}} - \frac{1}{e} \tag{3.17}$$

and

$$\sin E = \frac{\sin f\sqrt{1-e^2}}{1+e\cos f} \qquad \text{and} \qquad \cos E = \frac{\cos f + e}{1+e\cos f} \tag{3.18}$$

The objective function J is directly related to the thrust duration Δt_i and defined as:

$$J = \Delta t_1 + \Delta t_2 + \Delta t_3 + \Delta t_4 + \Delta t_5 \tag{3.19}$$

Since the value of the objective function depends only on the thrust duration, minimization of J means minimization of total propellant consumption. In addition, the value of the objective function must be assigned to the infinity if condition $a \leq 0$ is violated in any case.

Chapter 4

Result and Discussion

4.1 Results

The PSO algorithm in this study is written in MATLAB. C++ or Compute Unified Device Architecture (CUDA) may outperform MATLAB in terms of computing time but for this study, the performance of MATLAB is sufficient [9, 20, 21]. MATLAB is well suited for the problem and it shows acceptable performance. The number of iterations (N_{IT}) and particles (N) are set to 150 and 50 respectively in this work. Computing time for a loop of five complete runs takes approximately one hour (3500 - 4200 seconds). Each particle in the swarm is formed with 30 different parameters:

$$\begin{aligned} \boldsymbol{P} = [&\vartheta_{0,1} \quad \vartheta_{1,1} \quad \vartheta_{2,1} \quad \vartheta_{3,1} \quad \vartheta_{0,2} \quad \vartheta_{1,2} \quad \vartheta_{2,2} \quad \vartheta_{3,2} \quad \vartheta_{0,3} \quad \vartheta_{1,3} \quad \vartheta_{2,3} \quad \vartheta_{3,3} \\ &\vartheta_{0,4} \quad \vartheta_{1,4} \quad \vartheta_{2,4} \quad \vartheta_{3,4} \quad \vartheta_{0,5} \quad \vartheta_{1,5} \quad \vartheta_{2,5} \quad \vartheta_{3,5} \quad \Delta t_1 \quad \Delta t_1 \quad \Delta t_2 \quad \Delta t_3 \quad \Delta t_4 \quad \Delta t_5]^T \end{aligned} \tag{4.1}$$

Only thrust pointing angle and duration are considered as parameters because other factors can be calculated from components of thrust arcs. Each unknown parameter has minimum and maximum bounds as Eq. (4.2). A narrow range of thrust durations is empirically determined to manage all five thrusts. A non-zero lower bound is set for Δt to avoid a non-propulsive maneuver while the spacecraft is transferring to the target orbit. All values used in this thesis are presented as canonical units: distance unit (DU) and time unit (TU).

$$-1 \leq \vartheta_{j,i} \leq 1 \qquad 0.00001 \text{ TU} \leq \Delta t_i \leq 0.1 \text{ TU} \qquad (j = 0, \ldots, 3 \quad \text{and} \quad i = 1, \ldots, 5) \tag{4.2}$$

The value of standard gravitational constant μ is 1 DU^3/TU^2. The effective exhaust velocity c and thrust-to-mass ratio at initial time n_0 vary with the semi-major axis of the target orbits and

their values are shown in Table 4.1. Penalty terms are introduced to the original cost function

Table 4.1. c and n_0 with respect to β

β	c	n_0
1.5	0.5	0.3
2	1.5	0.6
5	2.5	1
8	3	1.5
10	5.5	2.2

due to the four equality constraints of the problem which are:

$$d_1 = |r(t_{apo}) - R_2| \qquad d_2 = v_r(t_{apo}) \qquad d_3 = \left| v_\theta(t_{apo}) - \frac{h_f}{r(t_{apo})} \right| \qquad d_4 = \xi(t_{apo}) \tag{4.3}$$

where t_{apo} is time at apoapsis of final orbit and h_f is a angular momentum of final orbit defined as following:

$$h = \sqrt{2\mu}\sqrt{\frac{r_a r_p}{r_a + r_p}} \tag{4.4}$$

By adding penalty terms, the objective function is updated from Equation (3.19) and becomes:

$$\tilde{J} = J + \sum_{k=1}^{4} \alpha_k |d_k| = \Delta t_1 + \Delta t_2 + \Delta t_3 + \Delta t_4 + \Delta t_5 + \sum_{k=1}^{4} \alpha_k |d_k| \tag{4.5}$$

The tolerance of each equality constraint is set to 10^{-3}, so any absolute value of constraint greater than the tolerance is considered as a violation. When they fail to meet the tolerance, the objective function is penalized by adding extra terms. Weighting coefficients α_k are defined as:

$$\alpha_k = \begin{cases} 150 & \text{if} \quad |d_k| > 10^{-3} \quad (k = 1, \ldots, 3) \\ 500 & \text{if} \quad |d_k| > 10^{-3} \quad (k = 4) \\ 0 & \text{if} \quad |d_k| < 10^{-3} \quad (k = 1, \ldots, 4) \end{cases} \tag{4.6}$$

The reason why constraints are not weighted equally is because they represent two different criteria. Weightings on d_1, d_2, and d_3 affect apoapsis distance of final orbit and d_4 adjusts LOA rotation. In case $a \leq 0$, which also violates the condition of an elliptical orbit, the objective function becomes infinity. Figure 4.1 illustrates behaviors of objective function values for $\beta = 1.5$.

Plots for other β values are attached in Appendix A. Plot shows that as the number of iteration increases, the value of the objective function tends to go down and plateau out. Sharp drops shown in the plot occur when the number of equality constraints are satisfied. Average percent

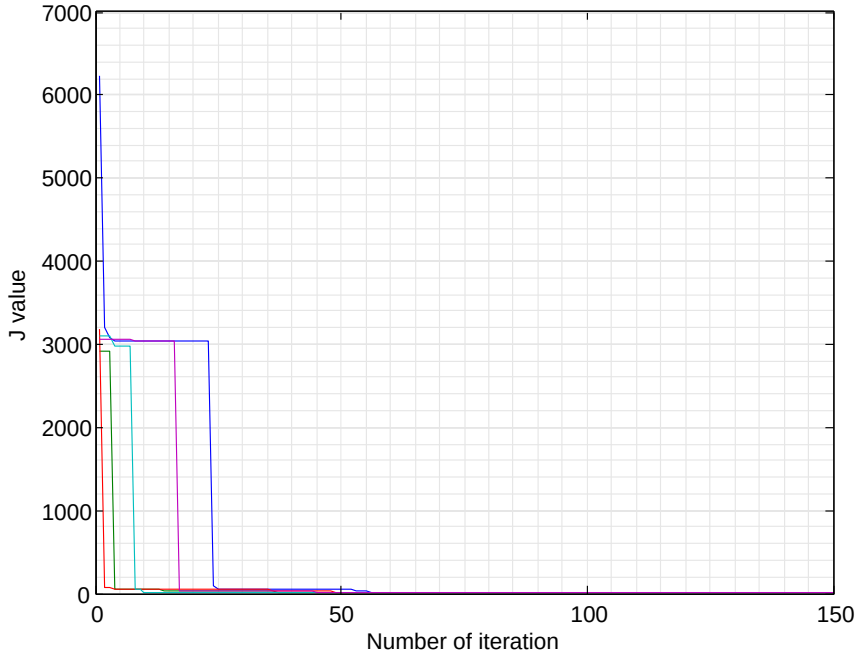

Figure 4.1. Value of objective function over number of iteration for $\beta = 1.5$ with 5 independent orbits

errors of LOA alignment and apopasis distance are calculated to analyze the performance of optimization as seen in Table 4.2. The percent LOA alignment error is calculated as:

$$\text{Percent LOA alignment error} = \frac{|180 - \text{LOA of final orbit}|}{180} \times 100 \tag{4.7}$$

and the equation for the percent distance error is:

$$\text{Percent distance error} = \frac{|R_a \text{ of target orbit} - R_a \text{ of final orbit}|}{R_a \text{ of target orbit}} \times 100 \tag{4.8}$$

Average errors obtained by the PSO are less than 3 percent and mostly less than 1 percent, and the objective functions give reasonably small values. Tables of individual β are attached in Appendix A and Table 4.3 only represents case for $\beta = 1.5$. The objective function values and error rates are mostly linearly related to each other and this tendency indicates that the optimal

Table 4.2. Average percent error of LOA alignment and distance and objective function value

β	Percent LOA alignment error (%)	Percent distance error (%)	Objective function
1.5	0.2670	2.3444	6.2021
2	0.2225	2.6669	8.2543
5	0.2489	0.2688	4.2237
8	0.4515	0.0135	10.6294
10	1.0641	0.0183	18.0720

solution is derived when the objective function value is minimized. Therefore, the result with the minimum objective function is picked as the best result of five runs. However, if either maximum or minimum burn time is recorded, the result is ignored since that implies the thrust duration is insufficient or surpasses the required amount of time.

Table 4.3. Error rate of LOA alignment and distance and objective function value for $\beta = 1.5$

No. run	LOA alignment error (%)	Distance error (%)	Objective function
1	0.4044	0.0666	0.5975
2	0.1699	2.97189	6.9936
3	0.3976	0.0693	0.5944
4	0.0356	0.0425	3.5972
5	0.3272	8.5718	19.5279

The trajectory of each orbit is evaluated from the final position $\boldsymbol{P}$ updated throughout the iterations. The best particle location is selected among all the particles in the swarm based on the objective values. Following data are results of the best particle of the swarm. The path of the full trajectory for $\beta = 1.5$ is shown in Figure 4.2 and other plots with different β can be found in Appendix B. Thrust-on locations are also presented in Figure 4.2 with dots and thrust paths are illustrated in Figure 4.4. The trajectory in Figure 4.2 may not give good intuition of the LOA rotation due to the small eccentricity though it is more obvious for higher β. The LOA rotation, which is not clearly observed in Fig. 4.2, is found in Table 4.4. Figure 4.3 represents change in both the LOA rotation angles and the apoapsis distance for each thrust. The zero degree mark in Fig 4.4 represents either 0 or 360 degrees since coordinates of the plot shown are transformed from polar to Cartesian. The location of burn arcs are near 1 DU because transfer orbits tend to keep

the same periapsis location. Individual plots of thrust pointing angles are presented in Figure 4.5. Due to the correlation between thrust pointing angle and burn arc trajectory, they have similar patterns. The paths of burn arcs and the thrust pointing angles present the movement of the spacecraft while in propulsive maneuvers. The paths shown in Fig. 4.4 show the exact location and behavior of the spacecraft in the burn arcs. Table 4.4 represents thrust duration and LOA displacement from the x-axis of each transfer orbit and more tables are included in Appendix B. The LOA displacement column shows how much each transfer orbit is tilted. The final orbit reaches at apoapsis distance of 1.5010 DU which is reasonably close to the target distance of 1.5 DU.

Table 4.4. Thrust duration and LOA displacement for $\beta = 1.5$

No. thrust	Thrust duration [TU]	LOA displacement [deg.]
1	0.0316	15.2930
2	0.0506	-3.9145
3	0.0771	-1.7912
4	0.0946	4.4879
5	0.0643	-0.7157

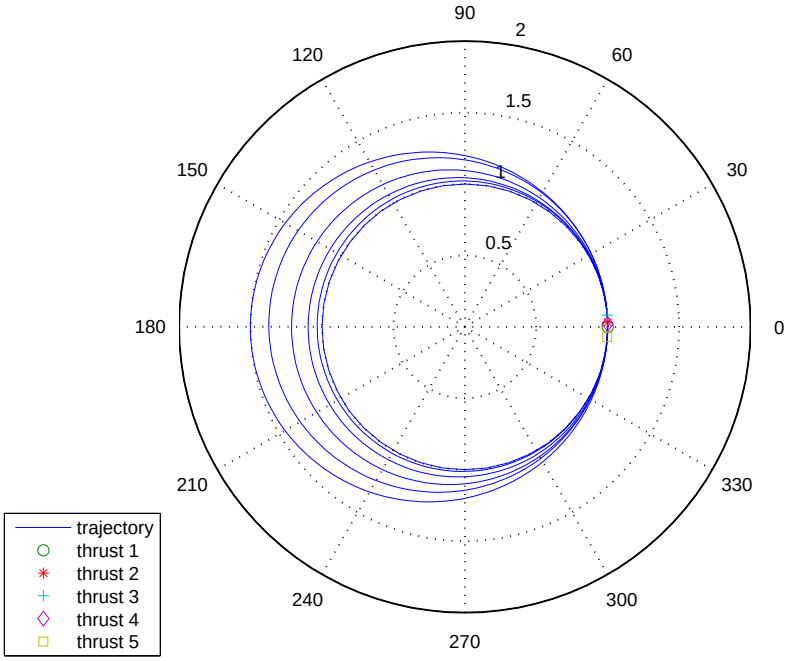

Figure 4.2. Trajectory of orbit for $\beta = 1.5$ with thrust-off locations

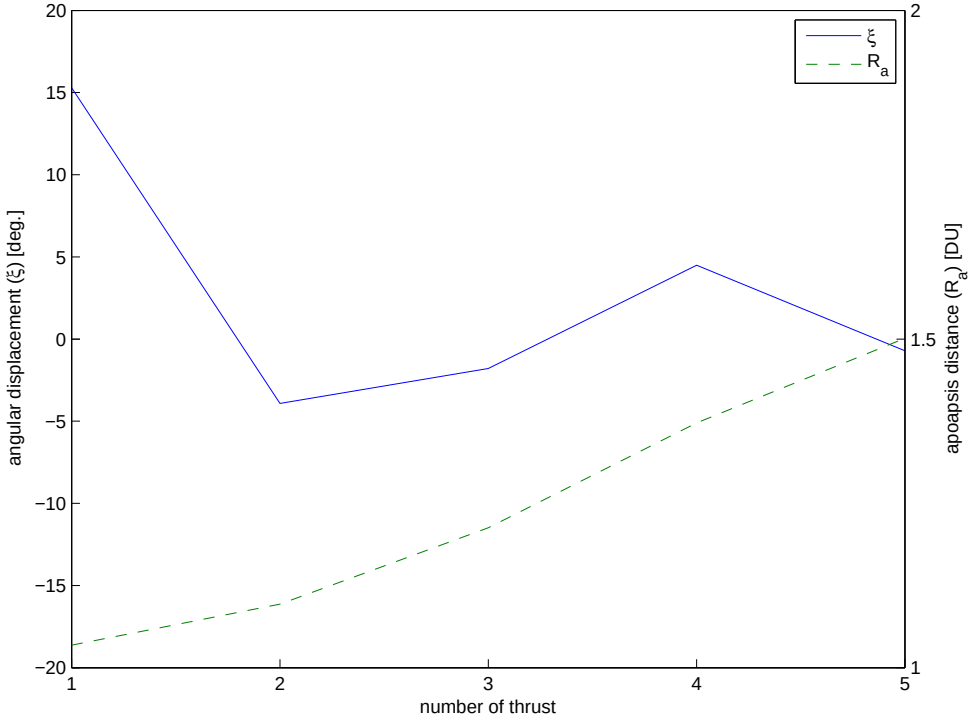

Figure 4.3. Angular displacement and apoapsis distance for thrusts

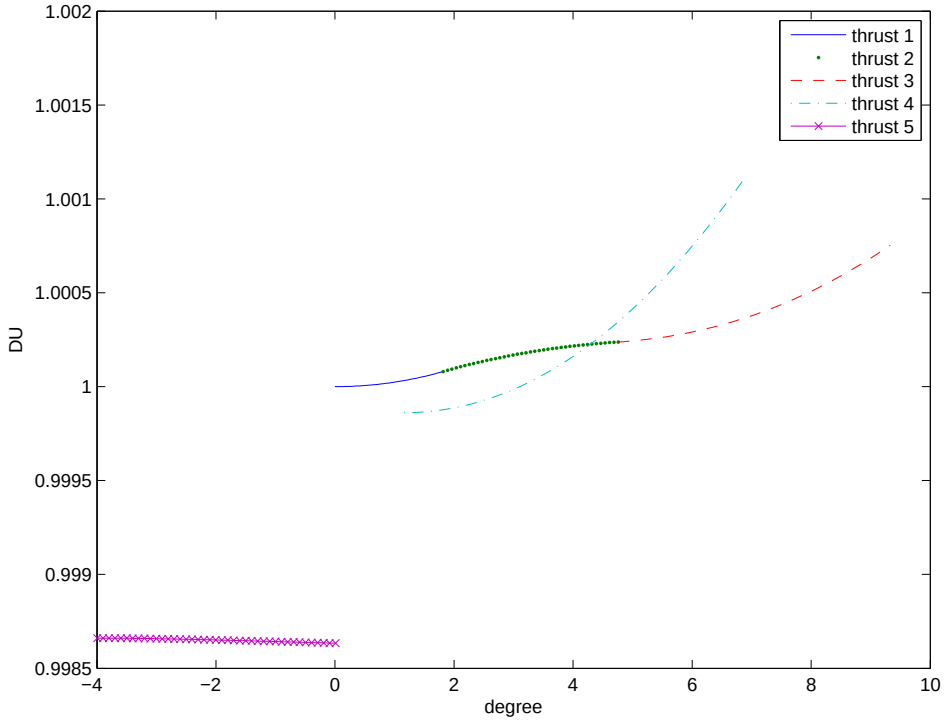

Figure 4.4. Paths of burn arcs for $\beta = 1.5$

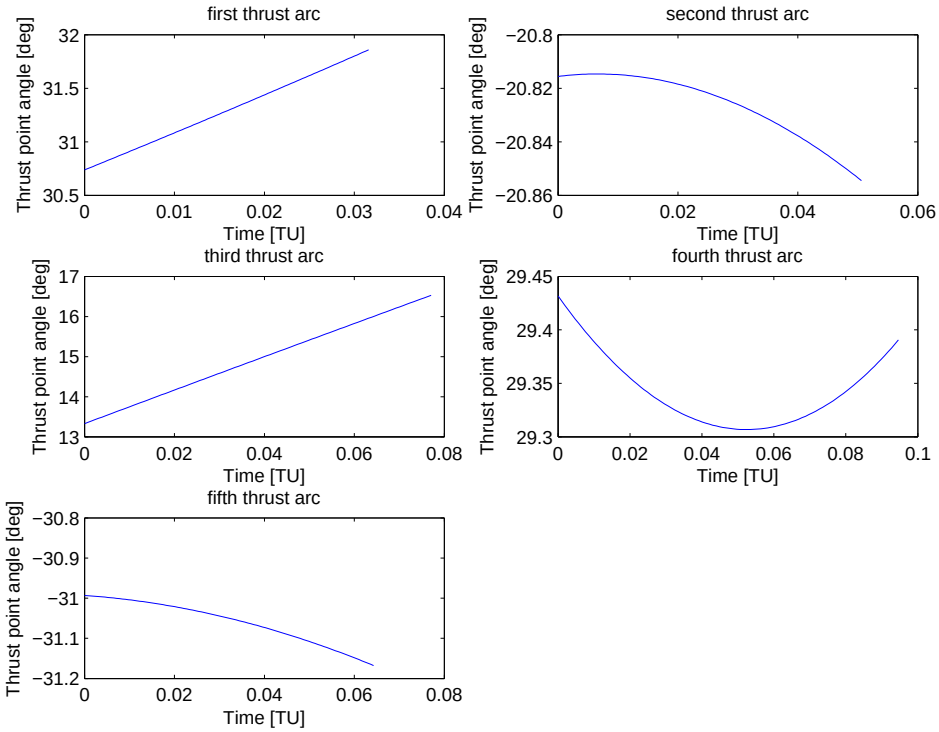

Figure 4.5. Thrust pointing angle of burn arcs for $\beta = 1.5$

4.2 Discussion

According to the results in the previous section, the PSO algorithm gives reasonably good results. Due to the nature of heuristic methods, the PSO gives only rough estimates and requires other methods, i.e. pseudospectral or direct collocation, to achieve precise solutions. Also, the amount of error varies over iteration and results are easily influenced by initial conditions. Figure 4.6 shows how error rates change with number of iteration. Excluding few outliers, errors in both plots decrease drastically after approximately 100 iterations. Yet, percent errors seem to fluctuate even at the higher iterations. This tendency indicates that higher numbers of iterations are definitely recommended to avoid error but cannot always guarantee an optimized solution. Additionally, the PSO easily falls into local or random solution which may not be close to the target solution and the PSO sometimes gets stuck (stagnation effect). However, the PSO is

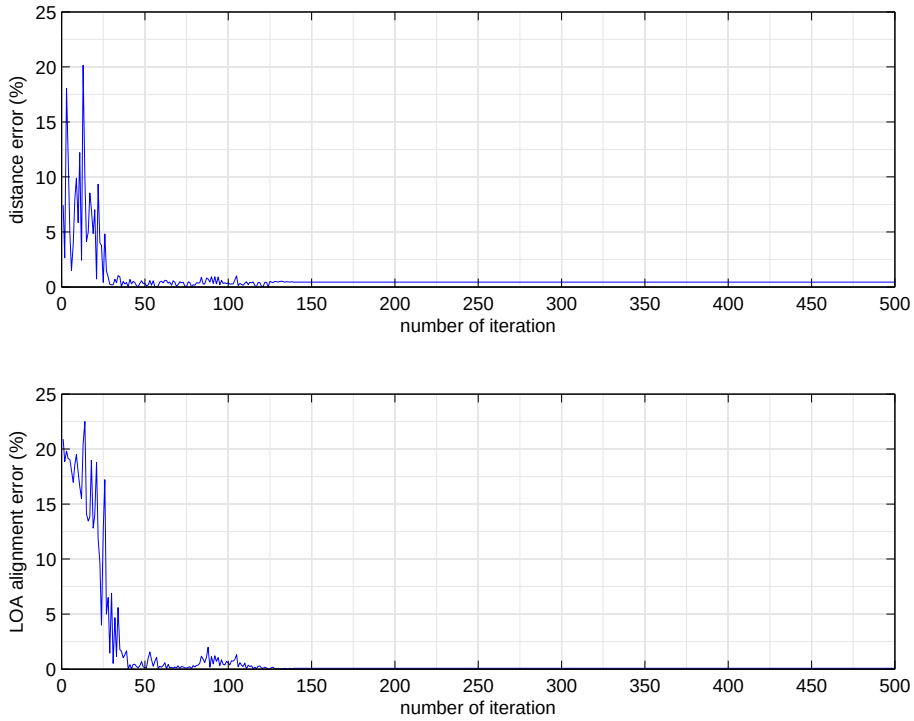

Figure 4.6. Percent error of LOA alignment and distance of final orbit over number of iteration

computationally highly cost effective and easy to code. In addition, PSO does not require any initial estimate of the solution since the program can generate a random solution set to begin with. While there are some limitations of PSO algorithm, results are reasonably satisfactory.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis deals with the application of particle swarm optimization (PSO) to apoapsis orbit raising. The PSO algorithm is a fairly new and popular method for trajectory optimization with a broad range of applications. Apoapsis orbit raising also has been applied to several space missions including, LADEE by NASA, ARTEMIS by ESA, and Mangalyaan by ISRO [13–19].

The PSO is well suited for the apoapsis orbit raising problem and an effective tool to find optimized solution for each problem. Constraints are addressed via the use of penalty functions added to the objective function. At the end of 150 iterations, in most cases the four equality constraints are satisfied to within specified tolerances. Therefore, the best run and the particle are selected according to the value of the objective function. However, if the spacecraft employs the maximum or the minimum thrust duration, the case is rejected due to the possibility that the spacecraft needs more or less thrust than is allocated.

The result of apoapsis orbit raising with 5 finite propulsive maneuvers meets all the requirements for optimizing trajectory: target distance, line of apside (LOA) alignment of the final orbit, and less propellant consumption. The percent errors of LOA alignment and distance gap found in the optimal results are less than 1 percent which indicates the PSO applied to apoapsis orbit raising provides acceptable solutions for this problem.

In conclusion, application of the PSO algorithm to apoapsis orbit raising is an effective approach. The trajectory obtained shows gradual increase in apoapsis distance and small final LOA displacement being the optimized solution to the problem.

5.2 Future Work

In this study, only PSO is used to optimize trajectory. However, there are multiple ways to optimize orbital trajectories. Comparing results and optimization capability on apoapsis orbit raising could be useful. According to Hassan et al., a genetic algorithm (GA) shows better performance than the PSO when the problem is constrained and nonlinear [22]. Since the orbit raising in this thesis has constraints and nonlinear equations of motion, GA may show a better performance than the PSO. Additionally, pseudospectral or direct collocation methods could be interesting since they are considered to be much more accurate than heuristic methods. Solutions of the PSO can be fed into the pseudospectral or direct collocation method if the latter requires an initial solution to begin. Efficiency of the method can be compared by increasing or decreasing the number of iterations. A no LOA rotation or a symmetric rotation could be another technique to optimize propellant consumption. To simulate real situations, adding perturbation effects or three-body problem would be required.

Appendix A

Plots of objective function vs. iteration

This Appendix presents the change in the objective function value over iterations. The objective function values of 5 runs are illustrated in each plot based upon final apoapsis distances. Most of the cases, the objective functions drop down to a single value as shown in the Table 4.3. Detailed J values are also provided in the Appendix A.

A.1 Objective function over iterations at $\beta = 1.5$

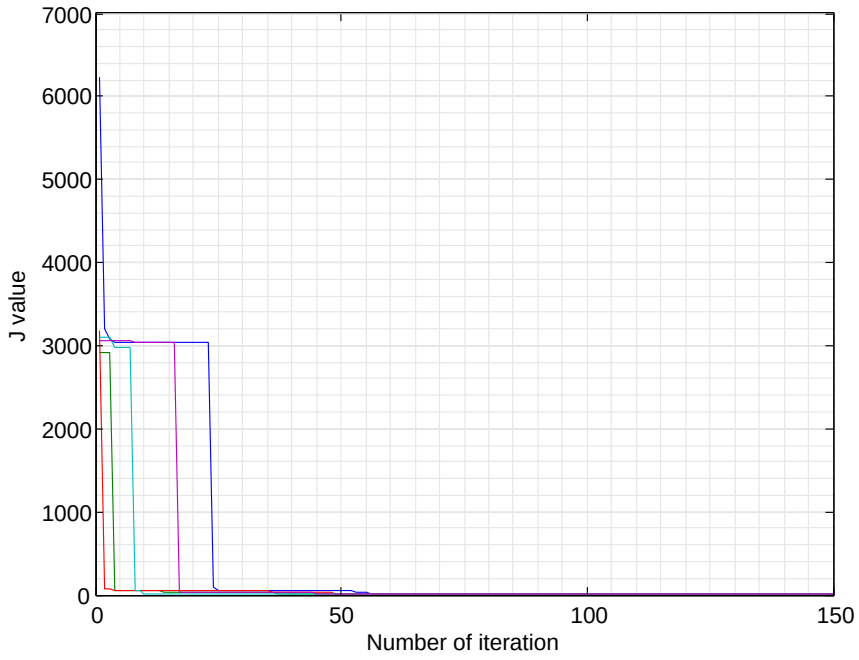

Figure A.1. Value of objective function over number of iterations for $\beta = 1.5$ with 5 independent orbits

Table A.1. Error rate of LOA alignment and distance and objective function value for $\beta = 1.5$

No. run	LOA alignment error (%)	Distance error (%)	Objective function
1	0.4044	0.0666	0.5975
2	0.1699	2.9718	6.9936
3	0.3976	0.0693	0.5944
4	0.0356	0.0425	3.5972
5	0.3272	8.5718	19.5279

A.2 Objective function over iterations for $\beta = 2$

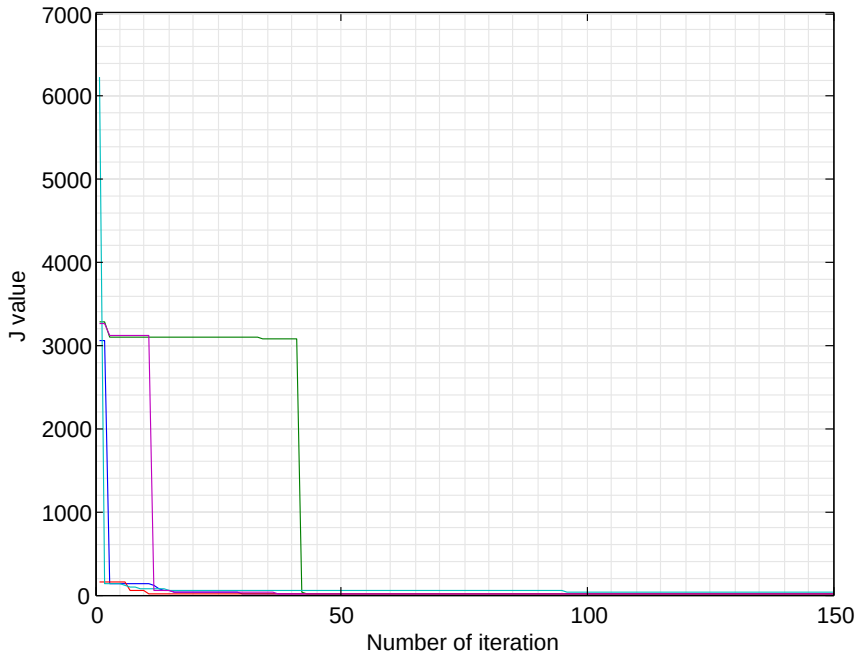

Figure A.2. Value of objective function over number of iterations for $\beta = 2$ with 5 independent orbits

Table A.2. Error rate of LOA alignment and distance and objective function value for $\beta = 2$

No. run	LOA alignment error (%)	Distance error (%)	Objective function
1	0.2091	0.0489	0.2884
2	0.0125	0.0227	0.2660
3	0.0229	0.0248	0.2686
4	0.5982	13.2370	40.1784
5	0.2698	0.0009	0.2699

A.3 Objective function over iterations for $\beta = 5$

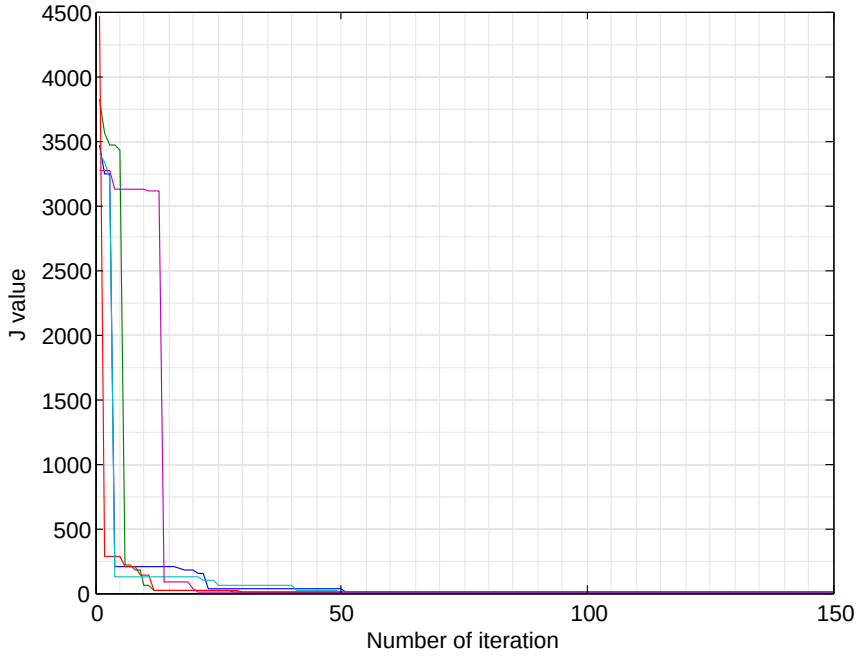

Figure A.3. Value of objective function over number of iterations for $\beta = 5$ with 5 independent orbits

Table A.3. Error rate of LOA alignment and distance and objective function value for $\beta = 5$

No. run	LOA alignment error (%)	Distance error (%)	Objective function
1	0.0137	1.1157	4.0168
2	0.0955	0.0197	3.0871
3	0.0883	0.1696	2.8054
4	0.6084	0.0198	7.8444
5	0.4385	0.0194	3.3647

A.4 Objective function over iteration for $\beta = 8$

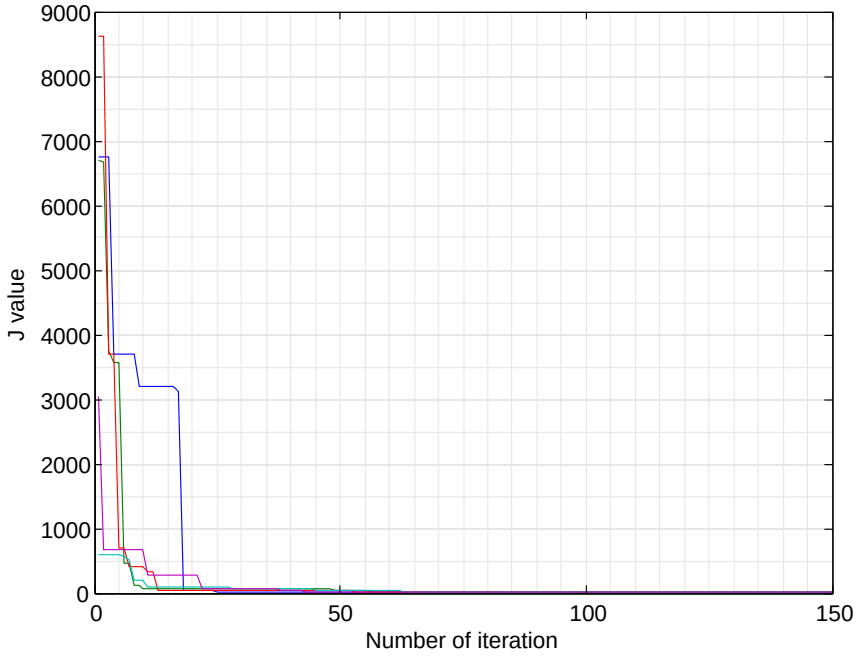

Figure A.4. Value of objective function over number of iterations for $\beta = 8$ with 5 independent orbits

Table A.4. Error rate of LOA alignment and distance and objective function value for $\beta = 8$

No. run	LOA alignment error (%)	Distance error (%)	Objective function
1	0.2112	0.0006	2.7650
2	0.3695	0.0182	20.0057
3	0.3850	0.0051	5.2603
4	0.2204	0.0120	5.6907
5	1.0717	0.0318	19.4254

A.5 Objective function over iterations for $\beta = 10$

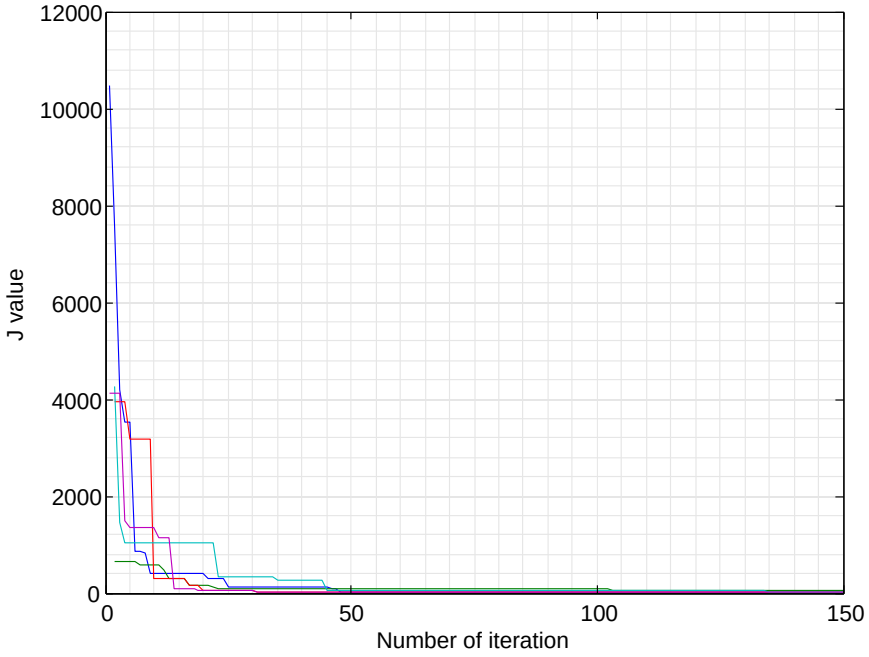

Figure A.5. Value of objective function over number of iterations for $\beta = 10$ with 5 independent orbits

Table A.5. Error rate of LOA alignment and distance and objective function value for $\beta = 10$

No. run	LOA alignment error (%)	Distance error (%)	Objective function
1	1.58974	0.0099	19.2521
2	3.0236	0.0654	51.2160
3	0.0837	0.0011	2.6727
4	0.0982	0.0140	9.5088
5	0.5253	0.0011	7.7104

Appendix B

Trajectory plots and properties

The orbital trajectories, thrust pointing angles, and tables of percent errors of each case are presented in this Appendix. Error tables give data for all 5 runs, though, only the best solution among them is picked. The optimal solutions are selected by the minimum objective function value, except for the case when the $\beta = 2$.

B.1 Properties of the trajectory for $\beta = 1.5$

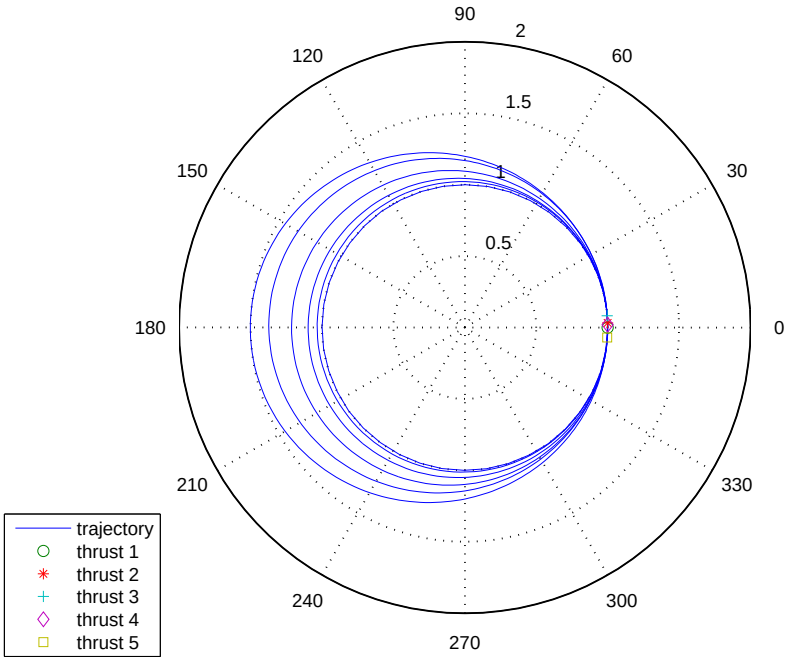

Figure B.1. Trajectory of orbit for $\beta = 1.5$ with thrust-off locations

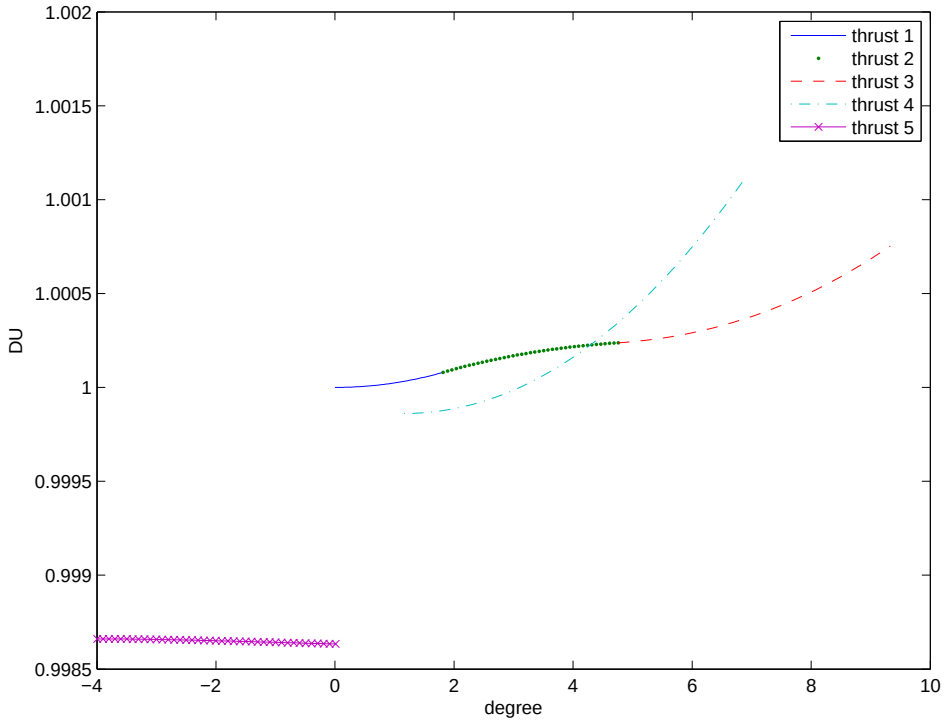

Figure B.2. Paths of burn arcs for $\beta = 1.5$

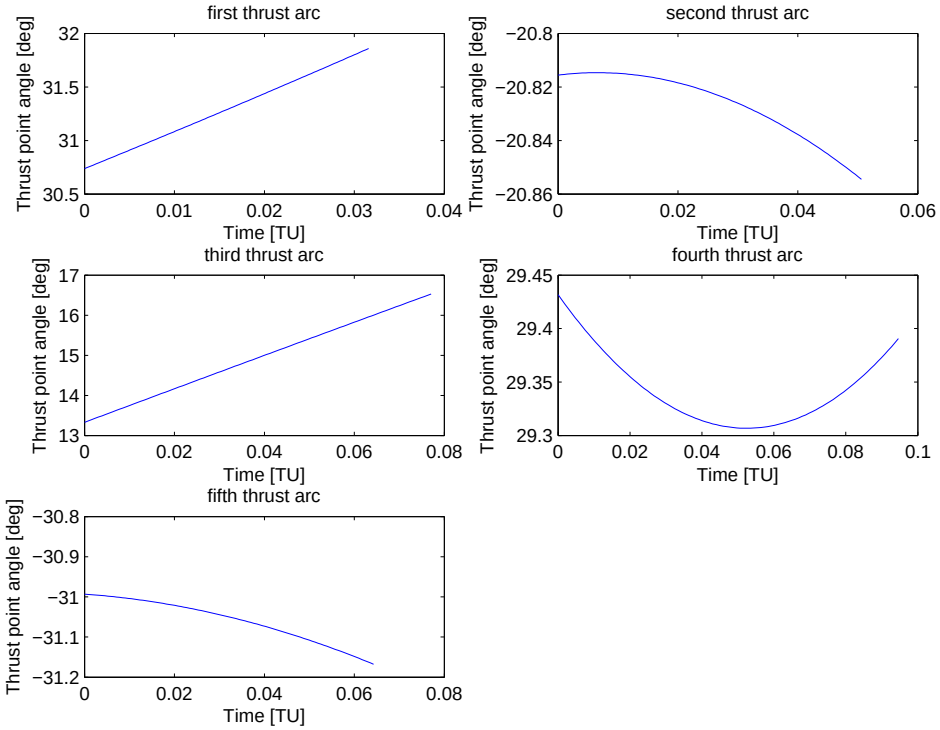

Figure B.3. Thrust pointing angle of burn arcs for $\beta = 1.5$

Table B.1. Thrust duration and LOA displacement for $\beta = 1.5$

No. thrust	Thrust duration [TU]	LOA displacement [deg.]
1	0.0316	15.2930
2	0.0506	-3.9145
3	0.0771	-1.7912
4	0.0946	4.4879
5	0.0643	-0.7157

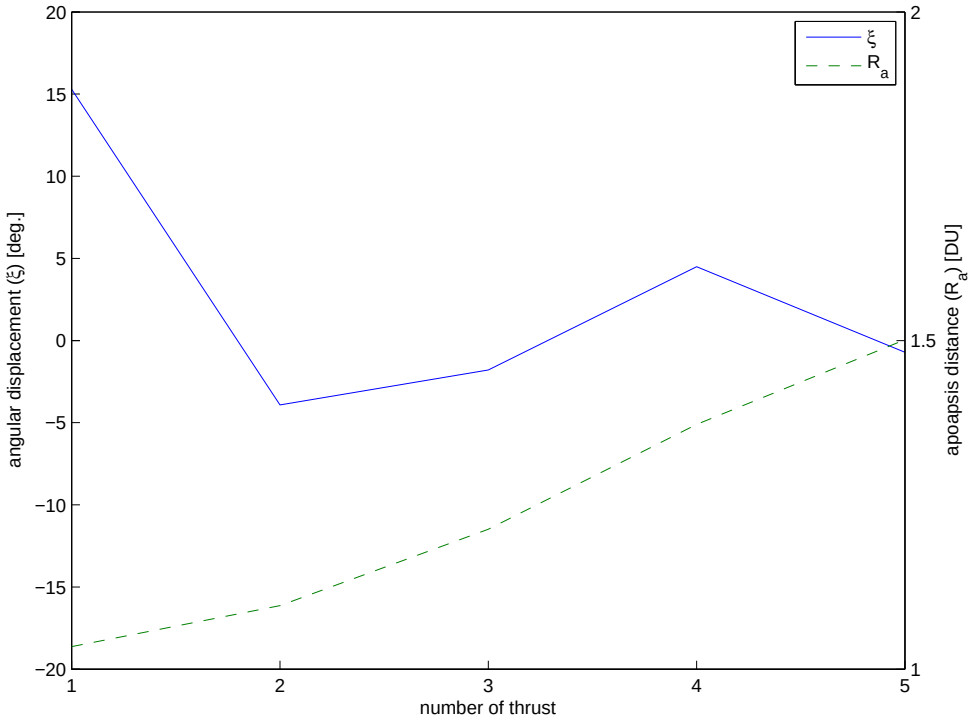

Figure B.4. Angular displacement and apoapsis distance for thrusts

B.2 Properties of the trajectory for $\beta = 2$

For $\beta = 2$, the third run is picked instead of the second run which seems to have the least errors and the smallest objective function value referring to Table A.2 in Appendix A. The reason why the second run result is ignored is because the thrust durations reach both maximum and minimum bounds and that is not recommended. Once the maximum or the minimum bound reached, that means the duration of the thrust at that maneuver is either not enough or exceeding.

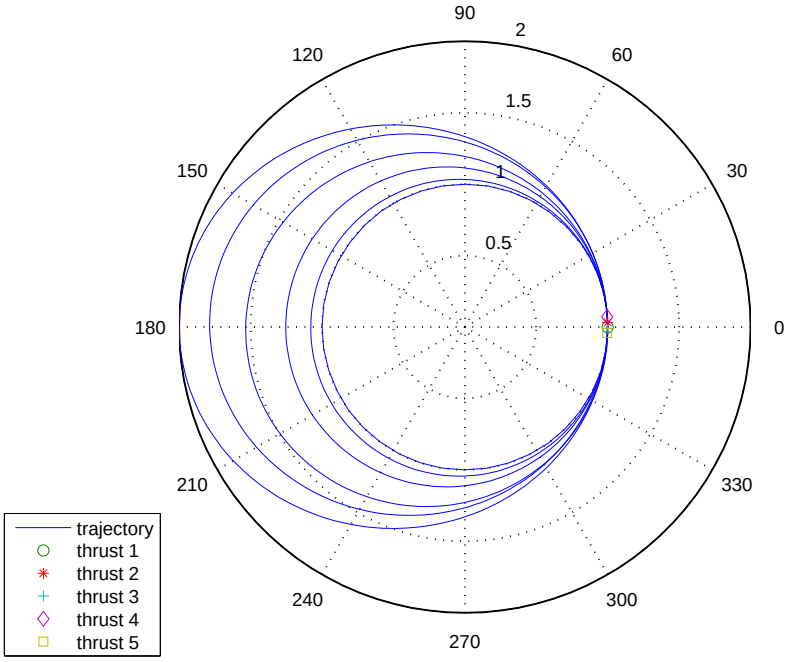

Figure B.5. Trajectory of orbit for $\beta = 2$ with thrust-off locations

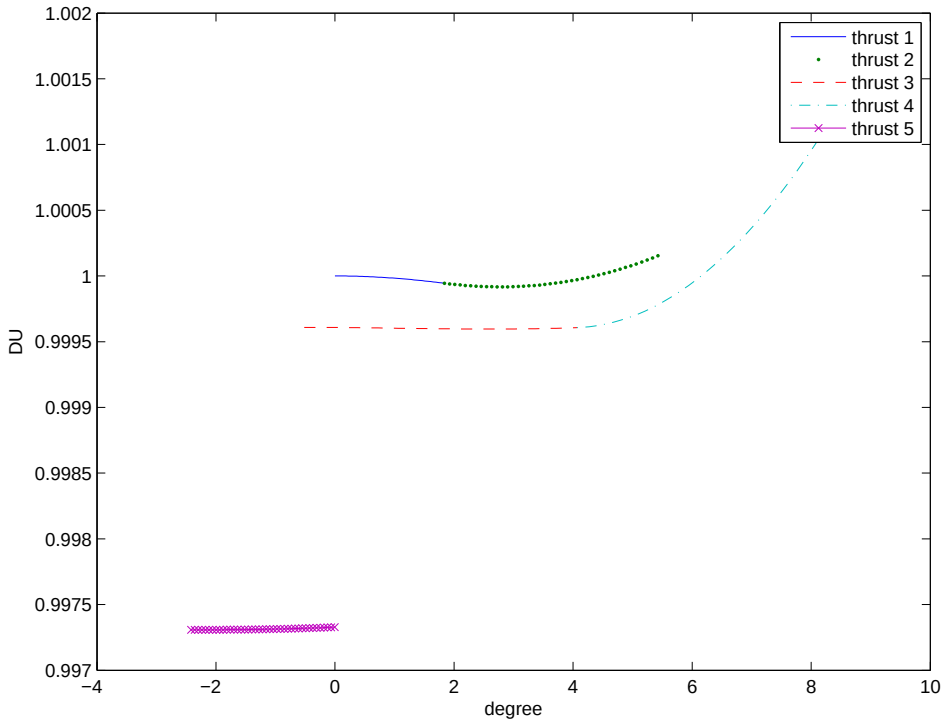

Figure B.6. Paths of burn arcs for $\beta = 2$

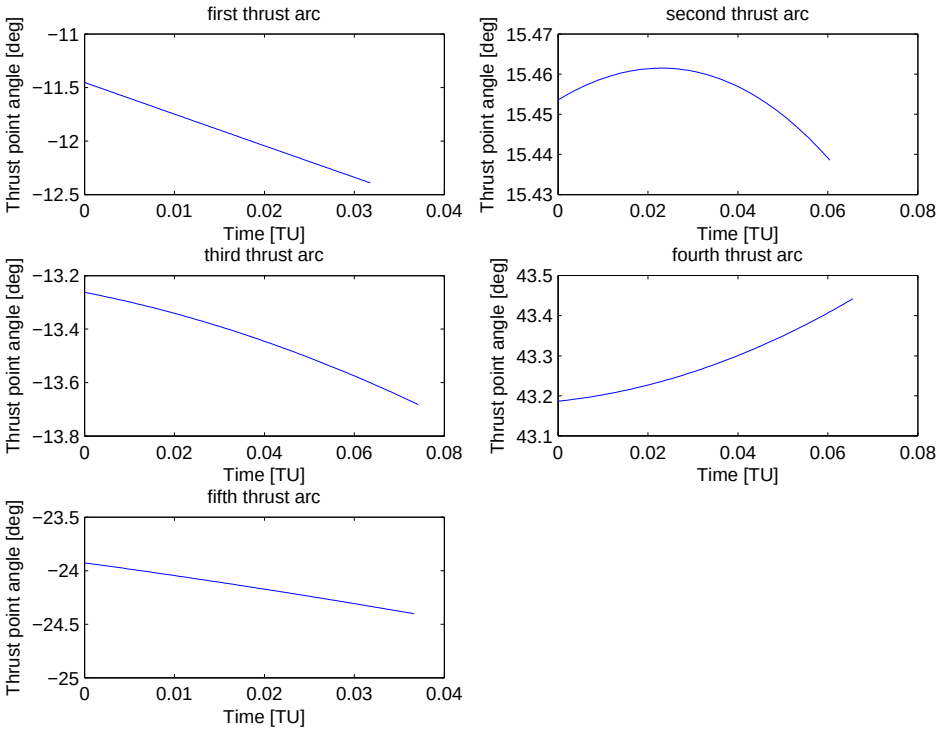

Figure B.7. Thrust pointing angle of burn arcs for $\beta = 2$

Table B.2. Thrust duration and LOA displacement for $\beta = 2$

No. thrust	Thrust duration [TU]	LOA displacement [deg.]
1	0.0318	-6.7276
2	0.0604	1.4131
3	0.0742	-4.1888
4	0.0656	2.1840
5	0.0367	-0.0412

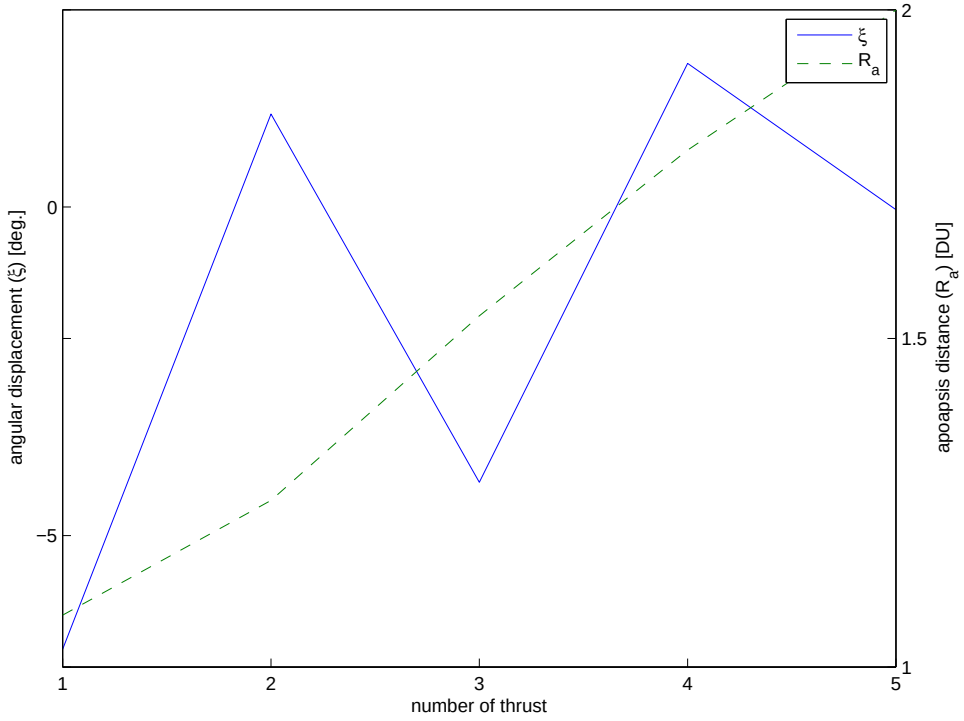

Figure B.8. Angular displacement and apoapsis distance for thrusts for $\beta = 2$

B.3 Properties of the trajectory for $\beta = 5$

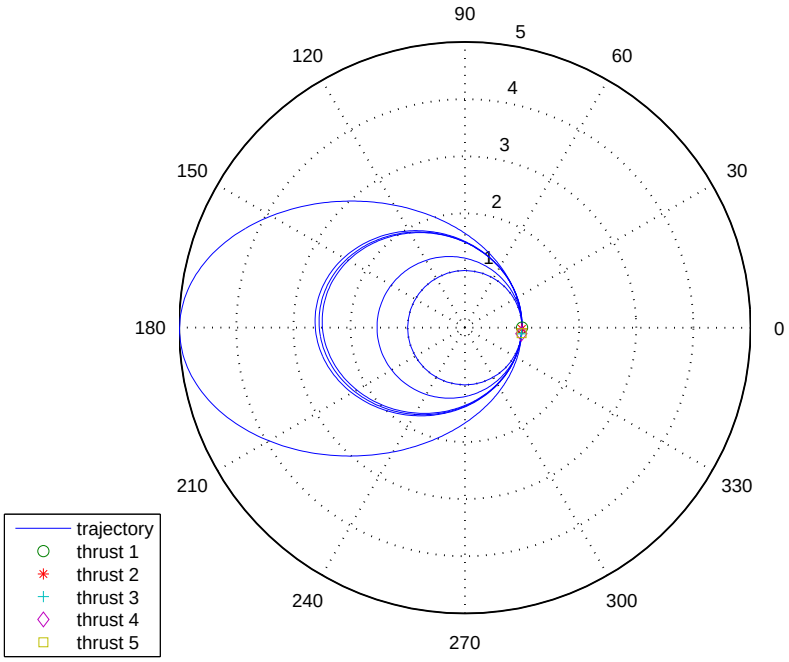

Figure B.9. Trajectory of orbit for $\beta = 5$ with thrust-off locations

Table B.3. Thrust duration and LOA displacement for $\beta = 5$

No. thrust	Thrust duration [TU]	LOA displacement [deg.]
1	0.0994	1.2664
2	0.0960	6.8726
3	0.0050	5.3480
4	0.0043	4.7592
5	0.0887	-0.1590

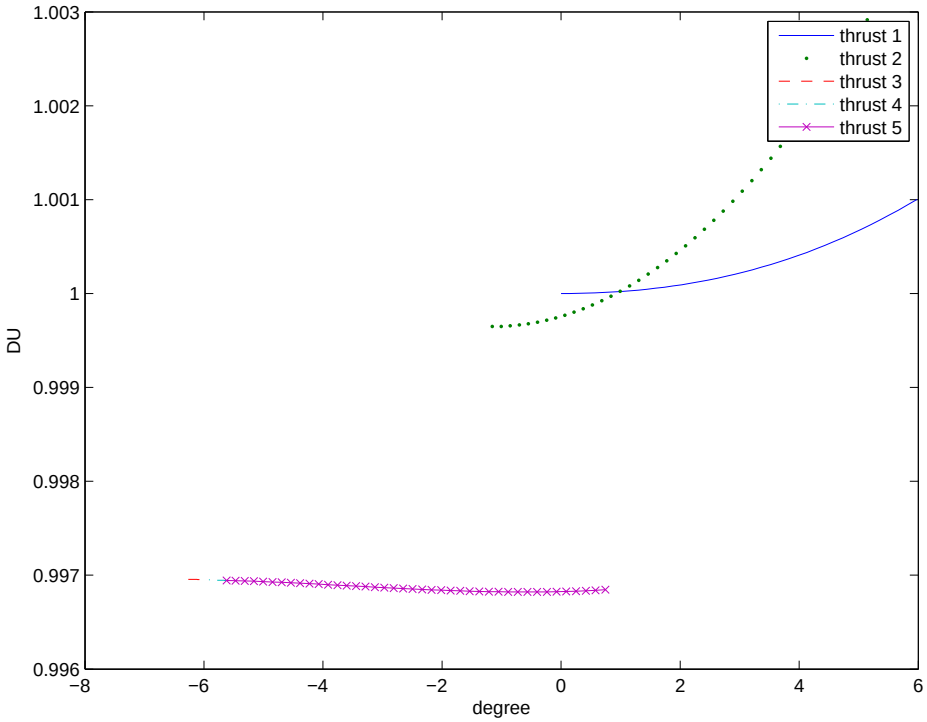

Figure B.10. Paths of burn arcs for $\beta = 5$

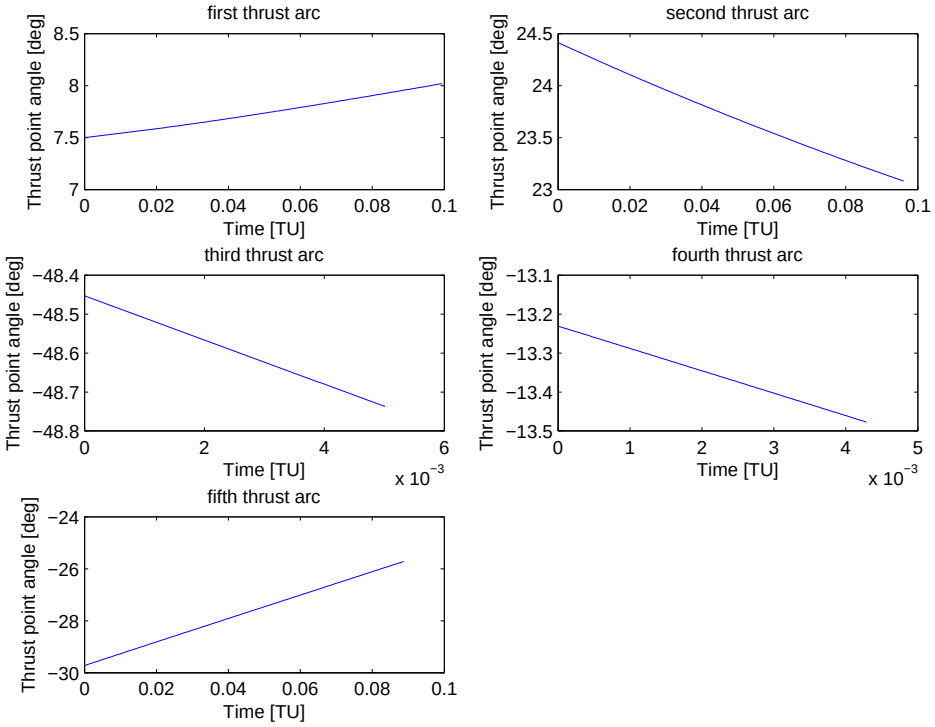

Figure B.11. Thrust pointing angle of burn arcs for $\beta = 5$

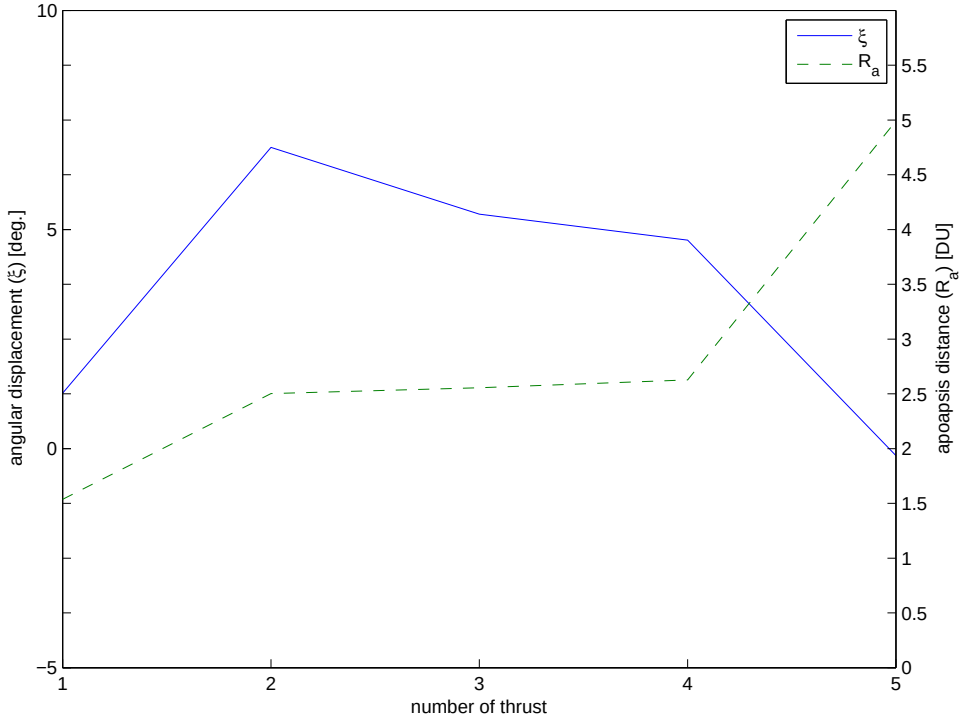

Figure B.12. Angular displacement and apoapsis distance for thrusts for $\beta = 5$

B.4 Properties of the trajectory for $\beta = 8$

Table B.4. Thrust duration and LOA displacement for $\beta = 8$

No. thrust	Thrust duration [TU]	LOA displacement [deg.]
1	0.0223	7.1332
2	0.0840	-4.5236
3	0.0774	5.2871
4	0.0188	6.6776
5	0.0506	-0.3802

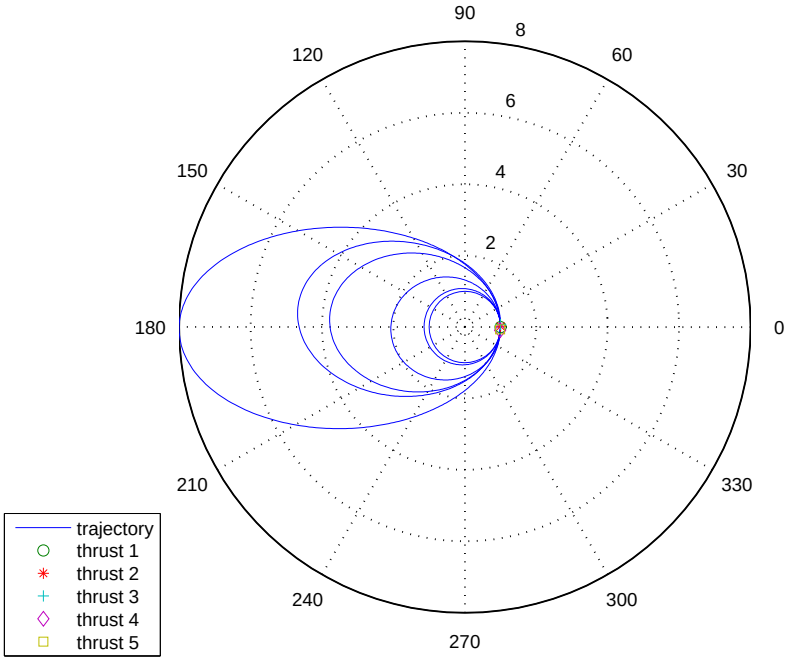

Figure B.13. Trajectory of orbit for $\beta = 8$ with thrust-off locations

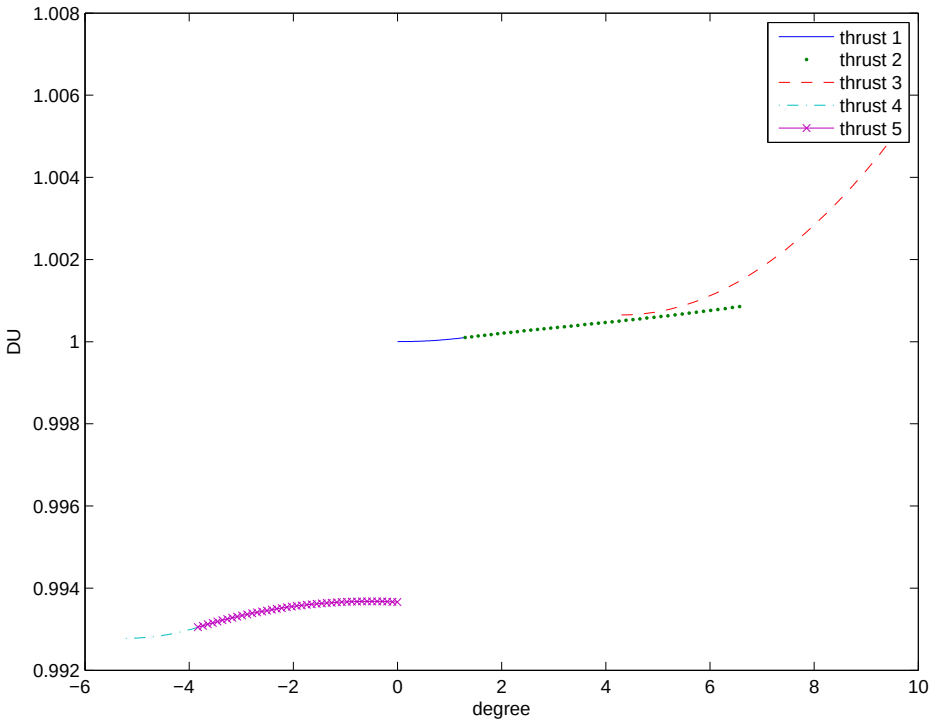

Figure B.14. Paths of burn arcs for $\beta = 8$

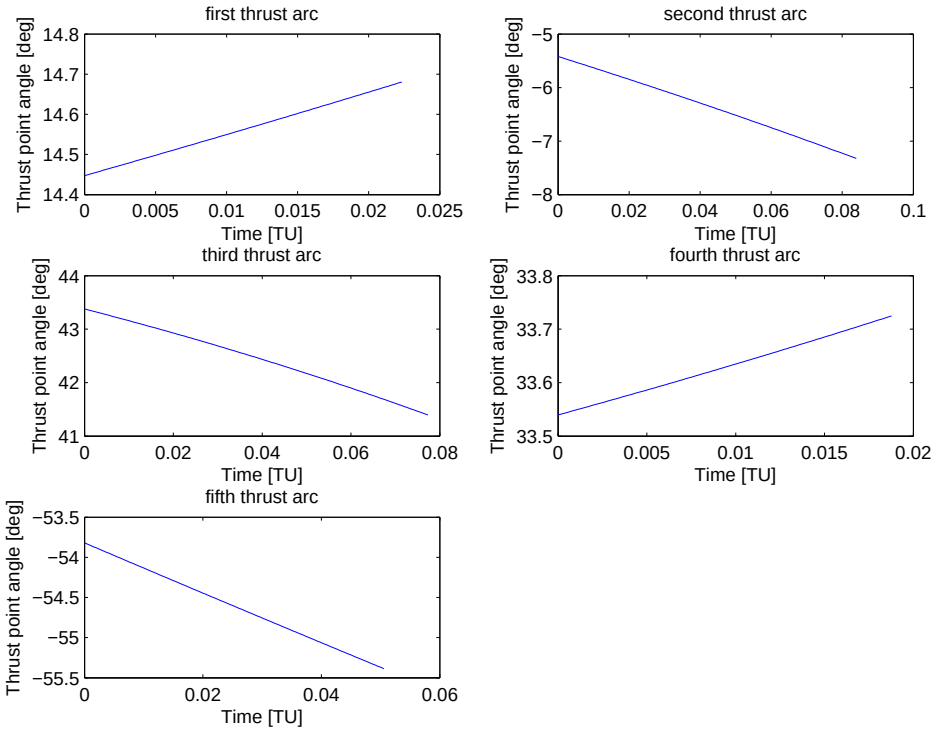

Figure B.15. Thrust pointing angle of burn arcs for $\beta = 8$

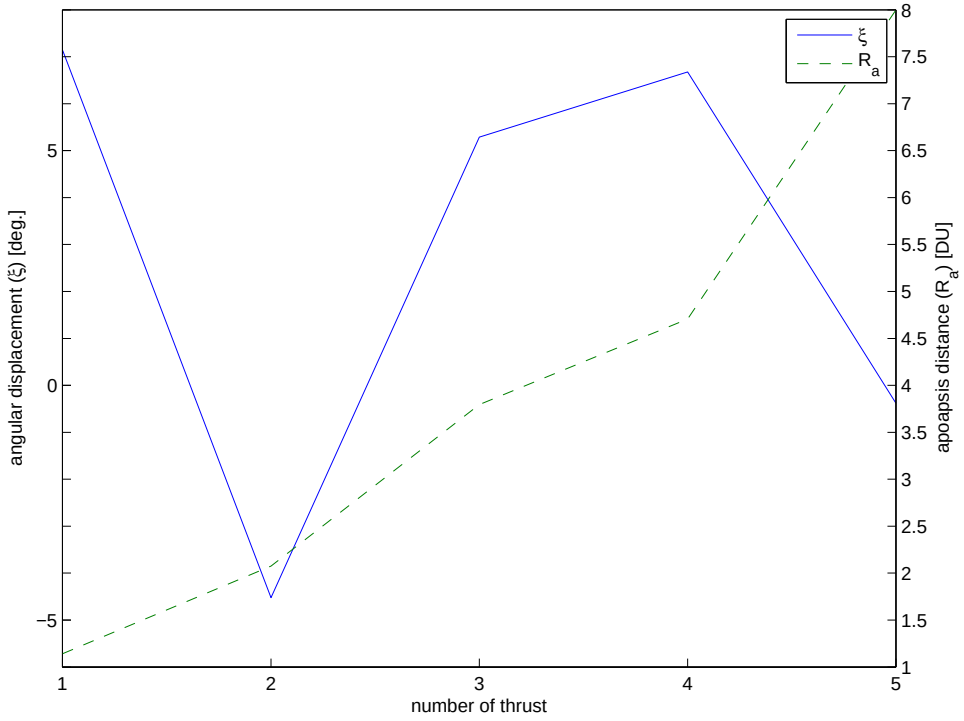

Figure B.16. Angular displacement and apoapsis distance for thrusts for $\beta = 8$

B.5 Properties of the trajectory for $\beta = 10$

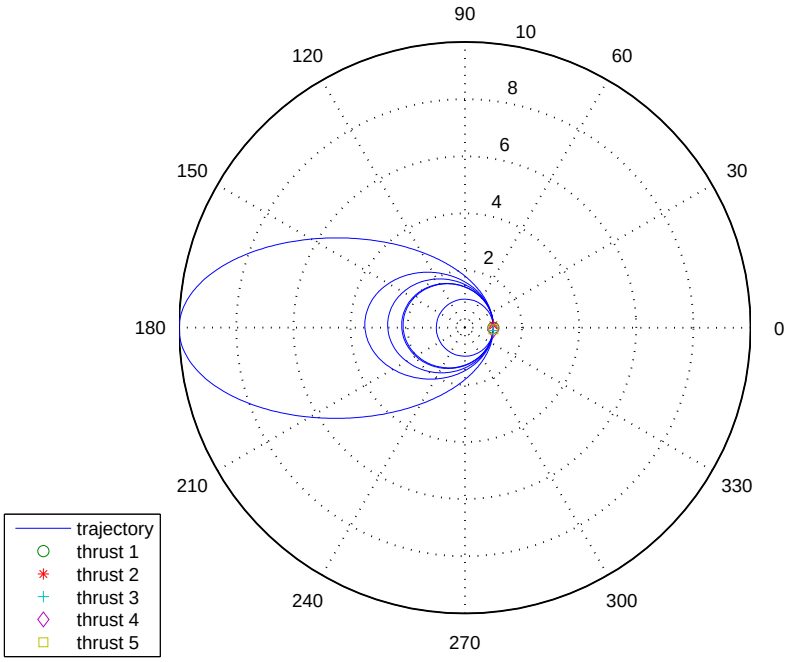

Figure B.17. Trajectory of orbit for $\beta = 10$ with thrust-off locations

Table B.5. Thrust duration and LOA displacement for $\beta = 10$

No. thrust	Thrust duration [TU]	LOA displacement [deg.]
1	0.0785	6.4098
2	0.0022	5.4772
3	0.0156	3.6122
4	0.0173	3.1180
5	0.0456	-0.1506

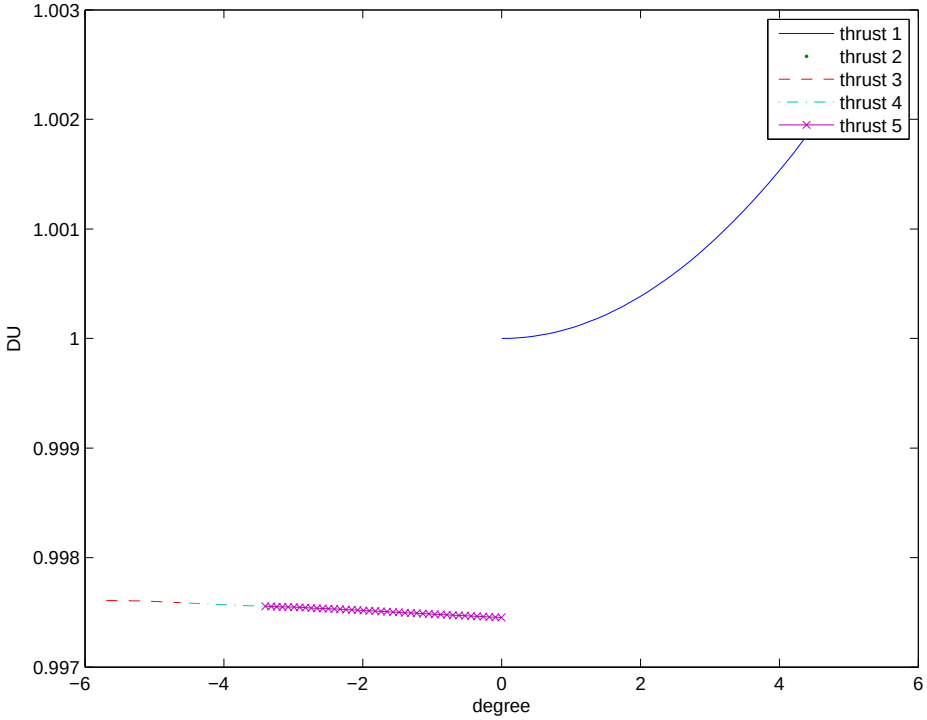

Figure B.18. Paths of burn arcs for $\beta = 10$

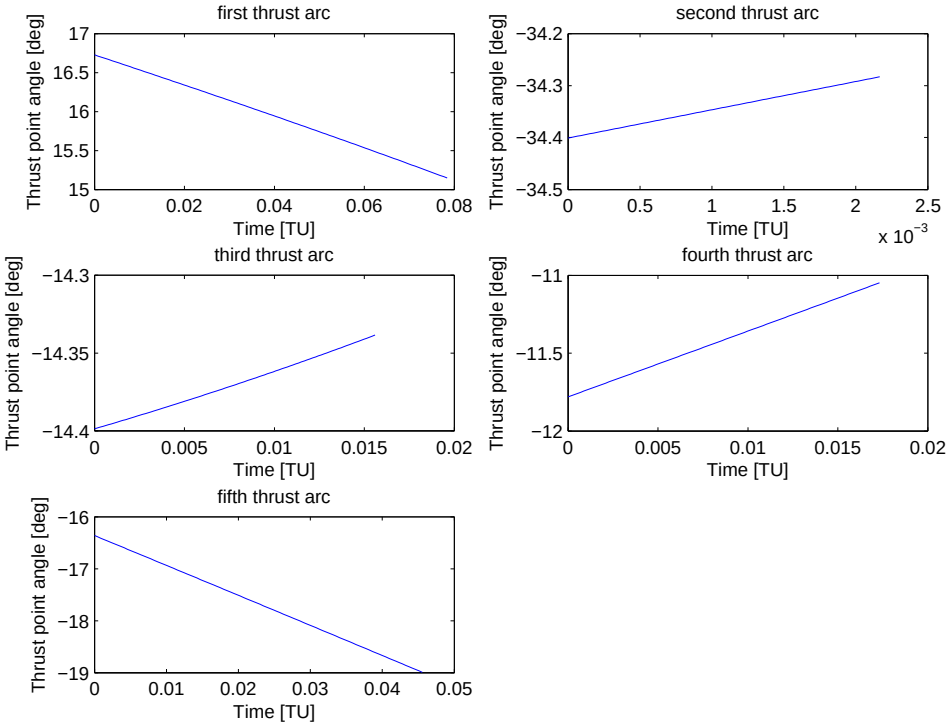

Figure B.19. Thrust pointing angle of burn arcs for $\beta = 10$

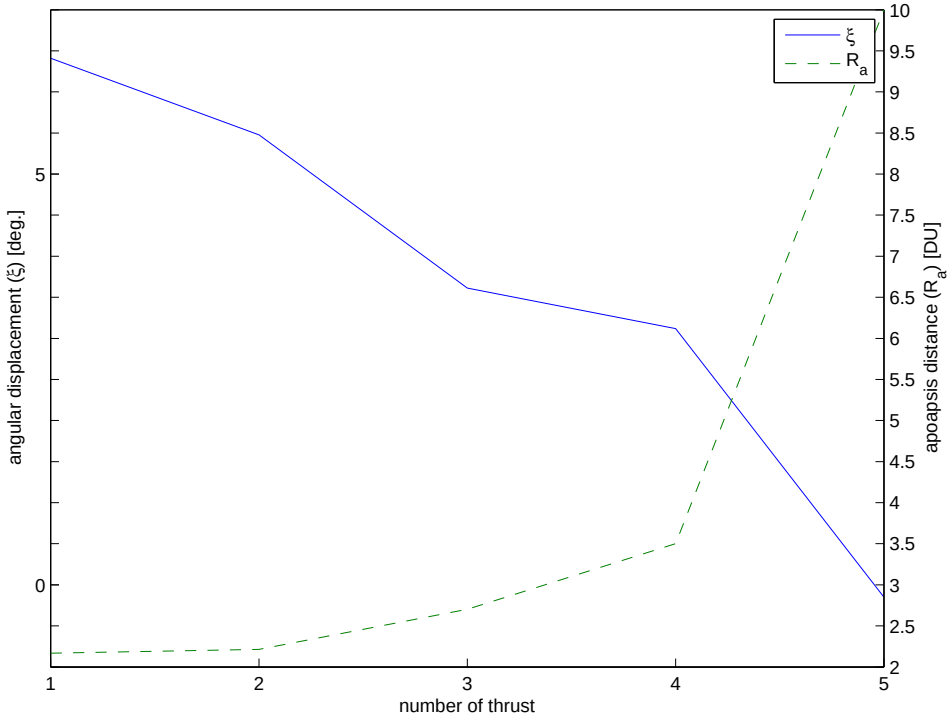

Figure B.20. Angular displacement and apoapsis distance for thrusts for $\beta = 10$

References

[1] Melton, R. G., K. M. Lajoie, and J. W. Woodburn (1989) "Optimum burn scheduling for low-thrust orbital transfers," *Journal of Guidance, Control, and Dynamics*, **12**(1), pp. 13–18.

[2] Robbins, H. M. (1966) "An analytical study of the impulsive approximation," *AIAA Journal*, **4**(8), pp. 1417–1423.

[3] Şenses, B. and A. V. Rao (2013) "Optimal Finite-Thrust Small Spacecraft Aeroassisted Orbital Transfer," *Journal of Guidance, Control, and Dynamics*, **36**(6), pp. 1802–1810.

[4] Kennedy, J. and R. Eberhart (1995) "Particle Swarm Optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948.

[5] Venter, G. and J. Sobieszczanski-Sobieski (2003) "Particle Swarm Optimization," *AIAA Journal*, **41**, pp. 1583–1589.

[6] Pontani, M. and B. A. Conway (2010) "Particle Swarm Optimization Applied to Space Trajectories," *Journal of Guidance, Control, and Dynamics*, **33**, pp. 1429–1441.

[7] Melton, R. G. (2014) "Hybrid methods for determining time-optimal, constrained spacecraft reorientation maneuvers," *Acta Astronautica*, **94**, pp. 294–301.

[8] Kalivarapu, V., J.-L. Foo, and E. Winer (2009) "Improving solution characteristics of particle swarm optimization using digital pheromones," *Structural and Multidisciplinary Optimization*, **37**(4), pp. 415–427.

[9] Kalivarapu, V. and E. Winer (2012) "Graphics Hardware Acceleration of Particle Swarm Optimization with Digital Pheromones using the CUDA Architecture," in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM*.

[10] Shi, Y. and R. Eberhart (1998) "A Modified Particle Swarm Optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pp. 69–73.

[11] Noushabadi, M. F. and N. Assadian (2012) "Optimal apogee burn time for low thrust spinning satellite in low altitude," *European Journal of Operational Research*, **222**(2), p. 386–391.

[12] Oh, D. Y., T. Randolph, S. Kimbrel, and M. Martinez-Sanchez (2004) "End-to-End Optimization of Chemical-Electric Orbit-Raising Missions," *Journal of Spacecraft and Rockets*, **41**(5), pp. 831–839.

[13] Elphic, R. C., B. Hine, G. T. Delory, J. S. Salute, S. Noble, A. Colaprete, M. Horanyi, P. Mahaffy, and the LADEE Science Team (2014) "The Lunar Atmosphere and Dust Environment Explorer (LADEE): Initial Science Results," in *45th Lunar and Planetary Science Conference*.

[14] Halekas, J. S., A. Poppe, G. Delory, R. Elphic, V. Angelopoulos, M. Horányi, and J. Szalay (2014) "ARTEMIS Observations And Data-based Modeling In Support Of LADEE," in *45th Lunar and Planetary Science Conference*.

[15] Whiffen, G. J. and T. H. Sweetser (2012) "Earth orbit raise design for the ARTEMIS mission," in *AIAA/AAS Astrodynamics Specialist Conference*.

[16] Sweetser, T. H., S. B. Broschart, V. Angelopoulos, G. J. Whiffen, D. C. Folta, M.-K. Chung, S. J. Hatch, and M. A. Woodard (2012) "ARTEMIS Mission Design," *Space Science Reviews*, **165**(1-4), pp. 27–57.

[17] ESA (2002), "Artemis starts its journey to final orbit," `http://www.esa.int/Our_Activities/Telecommunications_Integrated_Applications/Artemis_starts_its_journey_to_final_orbit`, Last accessed: March 2015.

[18] Spaceflight101, "Mars Orbiter Mission - Spacecraft & Mission Overview," `http://www.spaceflight101.com/mars-orbiter-mission.html`, Last accessed: March 2015.

[19] ISRO, "Mars Orbiter Mission Profile," `http://www.isro.gov.in/pslv-c25-mars-orbiter-mission/mars-orbiter-mission-profile`, Last accessed: March 2015.

[20] Schutte, J., J. Reinbolt, B. Fregly, R. Haftka, and A. George (2004) "Parallel global optimization with the particle swarm algorithm," *International Journal for Numerical Methods in Engineering*, **61**(13), pp. 2296–2315.

[21] Zhou, Y. and Y. Tan (2009) "GPU-based Parallel Particle Swarm Optimization," in *2009 IEEE Congress on Evolutionary Computation*, IEEE, pp. 1493–1500.

[22] Hassan, R., B. Cohanim, O. de Weck, and G. Venter (2005) "A Comparison of Particle Swarm Optimization and the Genetic Algorithm," in *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*.