Michigan Technological University

Numerical Optimization 5630 Project

Application of Finite Fourier series for Shape-Based Low Thrust Trajectory Optimization

Project Group:

Ehsan Taheri

Elahe Gorgin

Hanif Sadat Hoseini

4/29/2011

1. Abstract

Space mission trajectory design using the low-thrust capabilities is becoming increasingly popular. However, the optimization of the resulting trajectories is a very challenging and time-consuming task. In this paper, we build upon previous and existing shape-based techniques to present an alternative Fourier series approximation for rapid low-thrust rendezvous trajectory construction with thrust acceleration constraint handling capability. The new flexible representation along with the constraint handling capability makes this method a competing candidate for feasibility assessment of a whole bunch of trajectories within the given system propulsive budget. In addition, the provided solutions make good initial guesses for direct optimization techniques. Application of this method on a simple Earth-Mars rendezvous problem is addressed.

2. Introduction

The general problem we are looking at is to transfer a spacecraft from an initial condition, using polar coordinate system, (r_1 and θ_1) to a final orbit (r_2 and θ_2) within a given time t_f and spacecraft propulsive budget $\|T_a\| \le T_{\max}$. Two general techniques are assumed for solving the problem i.e. ***direct*** and ***indirect methods***, both of which require to be initialized with a first guess solution that are time-consuming and inefficient in preliminary rapid approximation of many trajectories. One of the recent methods used for initial guess generation is the shape-based method. The basic idea in shape-based methods is too consider a representative shape for state variables i.e. exponential, polynomial etc and try to find a feasible trajectory using these simple versions of the problem. For example the shape of the trajectory is assumed to have certain shape:

$$r = \frac{1}{a + b\theta + c\theta^2 + d\theta^3 + e\theta^4 + f\theta^5 + g\theta^6}$$

The ultimate goal of this project is to consider application of a new proposed method on typical problems using the MATLAB optimization toolbox and it algorithm and compare their efficiency in terms of time. We will solve both constrained and unconstrained version of the problem with fmincon and fsolve functions respectively, with detailed information about them.

3. Problem description

In this section we will give a brief description of the equations of motion and objective functions used for optimization in both constrained and unconstrained cases.

3.1 Equations of motion

The governing equations of a spacecraft in a two-body gravitational field can be written in the following polar forms using the Newton's gravitational law:

$$2\dot{r}\dot{\theta} + r\ddot{\theta} = T_a \cos\left(\alpha\right)$$

$$\ddot{r} - r\dot{\theta}^2 + \frac{\mu}{r^2} = T_a \sin\left(\alpha\right)$$

Looking at the Fig. 1 the following parameters are defined:

$\vec{r}$ is the radius vector, θ is the polar angle, γ is the flight path,

$\vec{v}$ is the velocity vector, α is the steering angle,

$\vec{T}_a$ is the thrust acceleration, μ the gravitational parameter

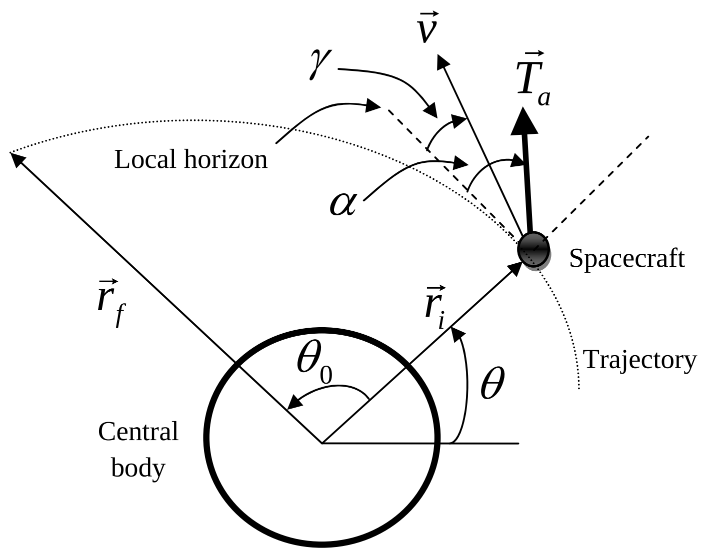


Figure1. Motion states in two-body gravitational field

In the next section this equation of motion along with some assumptions will be used to make the objective function.

3.2 Fourier series approximation

In this method, we approximate radius and polar angle with finite terms of Fourier series. For r approximation one can write:

$$r(t) = \frac{a_0}{2} + \sum_{n=1}^{n_r} \left\{ a_n \cos\left(\frac{n\pi}{T} t\right) + b_n \sin\left(\frac{n\pi}{T} t\right) \right\}$$

For θ approximation we can have the same Fourier-based approximation in general.

$$\theta(t) = \frac{c_0}{2} + \sum_{n=1}^{n_\theta} \left\{ c_n \cos\left(\frac{n\pi}{T} t\right) + d_n \sin\left(\frac{n\pi}{T} t\right) \right\}$$

$$\begin{cases} r(t) = \dfrac{a_0}{2} + \displaystyle\sum_{n=1}^{n_r} \left\{ a_n \cos\left(\dfrac{n\pi}{T} t\right) + b_n \sin\left(\dfrac{n\pi}{T} t\right) \right\} \\ \theta(t) = \dfrac{c_0}{2} + \displaystyle\sum_{n=1}^{n_\theta} \left\{ c_n \cos\left(\dfrac{n\pi}{T} t\right) + d_n \sin\left(\dfrac{n\pi}{T} t\right) \right\} \end{cases}$$

Knowing the values for $r_1, r_2, \dot{r}_1, \dot{r}_2, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2$ at terminal points some of the coefficients can be determined using the boundary conditions resulting eight equations. In general after some manipulation we can write:

$$a_1 = \frac{r_i - r_f}{2} - \sum_{\substack{n=3 \\ \frac{n}{2} \neq 0}}^{n_r} a_n ; n_r \geq 3 \qquad a_2 = \frac{r_i + r_f - a_0}{2} - \sum_{\substack{n=4 \\ \frac{n}{2} = 0}}^{n_r} a_n ; n_r \geq 4$$

$$b_1 = \frac{T}{2\pi}\left(\dot{r}_i - \dot{r}_f\right) - \sum_{\substack{n=3 \\ \frac{n}{2} \neq 0}}^{n_r} \left(n \times b_n\right); n_r \geq 3 \qquad b_2 = \frac{T}{4\pi}\left(\dot{r}_i + \dot{r}_f\right) - \frac{1}{2}\sum_{\substack{n=4 \\ \frac{n}{2} = 0}}^{n_r} \left(n \times b_n\right); n_r \geq 4$$

$$c_1 = \frac{\theta_i - \theta_f}{2} - \sum_{\substack{n=3 \\ \frac{n}{2} \neq 0}}^{n_\theta} c_n ; n_\theta \geq 3 \qquad c_2 = \frac{\theta_i + \theta_f - c_0}{2} - \sum_{\substack{n=4 \\ \frac{n}{2} = 0}}^{n_\theta} c_n ; n_\theta \geq 4$$

$$d_1 = \frac{T}{2\pi}\left(\dot{\theta}_i - \dot{\theta}_f\right) - \sum_{\substack{n=3 \\ \frac{n}{2} \neq 0}}^{n_\theta} \left(n \times d_n\right); n_\theta \geq 3 \qquad d_2 = \frac{T}{4\pi}\left(\dot{\theta}_i + \dot{\theta}_f\right) - \frac{1}{2}\sum_{\substack{n=4 \\ \frac{n}{2} = 0}}^{n_\theta} \left(n \times d_n\right); n_\theta \geq 4$$

From the second relation of motion i.e. in the tangential direction we can combine the two equations and arrive at a final relation for the equation of motion that needs to be satisfied at all points:

$$2\dot{r}\dot{\theta} + r\ddot{\theta} = T_a \cos\left(\alpha\right) \Rightarrow T_a = \frac{2\dot{r}\dot{\theta} + r\ddot{\theta}}{\cos\left(\alpha\right)}$$

Substituting (14) into (13-a) will result:

$$\ddot{r} - r\dot{\theta}^2 + \frac{\mu}{r^2} = T_a \sin\left(\alpha\right)$$

$$\ddot{r} - r\dot{\theta}^2 + \frac{\mu}{r^2} = \frac{2\dot{r}\dot{\theta} + r\ddot{\theta}}{\cos\left(\alpha\right)} \sin\left(\alpha\right) = \left(2\dot{r}\dot{\theta} + r\ddot{\theta}\right)\tan\left(\alpha\right) = \left(2\dot{r}\dot{\theta} + r\ddot{\theta}\right)\tan\left(\gamma\right) \xrightarrow{\tan(\gamma) = \frac{\dot{r}}{r\dot{\theta}}}$$

$$\frac{r^2\ddot{r} - r^3\dot{\theta}^2 + \mu}{r^2} = \left(2\dot{r}\dot{\theta} + r\ddot{\theta}\right)\frac{\dot{r}}{r\dot{\theta}} \Rightarrow \frac{r^2\ddot{r} - r^3\dot{\theta}^2 + \mu}{r} = \frac{2\dot{r}^2\dot{\theta} + r\dot{r}\ddot{\theta}}{\dot{\theta}}$$

$$\left(r^2\ddot{r} - r^3\dot{\theta}^2 + \mu\right)\dot{\theta} = \left(2\dot{r}^2\dot{\theta} + r\dot{r}\ddot{\theta}\right)r$$

$$r^2\ddot{r}\dot{\theta} - \left(r\dot{\theta}\right)^3 + \mu\dot{\theta} = 2r\dot{r}^2\dot{\theta} + r^2\dot{r}\ddot{\theta}$$

$$\Rightarrow \underline{r^2\ddot{r}\dot{\theta}} - \left(r\dot{\theta}\right)^3 + \underline{\underline{\mu\dot{\theta}}} - \underline{\underline{2r\dot{r}^2\dot{\theta}}} - \underline{r^2\dot{r}\ddot{\theta}} = 0 \rightarrow f(r, \dot{r}, \ddot{r}, \dot{\theta}, \ddot{\theta}, \mu, t) = 0$$

$$f : r^2\left(\dot{\theta}\ddot{r} - \dot{r}\ddot{\theta}\right) + \dot{\theta}\left(\mu - 2r\dot{r}^2\right) - \left(r\dot{\theta}\right)^3 = 0$$

Selecting some values for n_r and n_θ the total number of unknowns are $2(n_r + n_\theta) + 2$ eight of which can be calculated in terms of the others using boundary conditions. So the equation converts to a function f which is only a function of time as well as coefficients.

$$f\left(a_0, a_1 \cdots a_{n_r}, b_1 \cdots b_{n_r}, c_0, c_1 \cdots a_{n_\theta}, d_1 \cdots d_{n_\theta}; t\right) = 0$$

3.3 Unconstrained problem and Algorithms

In order to solve for the unknowns this nonlinear equation should be discretized at some points so that we have some number of equations. Depending on the number of equations we would get underdetermined/square/over determined systems. We can use Matlab fsolve function to solve for the unconstrained version of the problem. The fsolve function has four algorithms for solving systems of nonlinear multidimensional equations:

- Trust-region-dogleg
- Trust-region-reflective
- Levenberg-marquardt
- Gauss-Newton

All but the Gauss-Newton method are large-scale;

3.3.1 Trust-Region

Another approach is to solve a linear system of equations to find the search direction, namely, Newton's method says to solve for the search direction d_k such that

$$J(x_k)d_k = -F(x_k)$$
$$x_{k+1} = x_k + d_k$$

where $J(x_k)$ is the n-by-n Jacobian

$$J(x_k) = \begin{bmatrix} \nabla F_1(x_k)^T \\ \nabla F_2(x_k)^T \\ \vdots \\ \nabla F_n(x_k)^T \end{bmatrix}$$

Newton's method can run into difficulties. $J(x_k)$ may be singular, and so the Newton step d_k is not even defined. Also, the exact Newton step d_k may be expensive to compute. In addition, Newton's method may not converge if the starting point is far from the solution.

Using trust-region techniques improves robustness when starting far from the solution and handles the case when $J(x_k)$ is singular. To use a trust-region strategy, a merit function is needed to decide if x_{k+1} is better or worse than x_k. A possible choice is

$$\min_d f(d) = \frac{1}{2} F(x_k + d)^T F(x_k + d)$$

But a minimum of $f(d)$ is not necessarily a root of $F(x)$. The Newton step d_k is a root of

$$M(x_k + d) = F(x_k) + J(x_k)d$$

and so it is also a minimum of m(d), where

$$\begin{aligned} \underbrace{\min}_d m(d) &= \frac{1}{2}\|M(x_k + d)\|_2^2 = \frac{1}{2}\|F(x_k) + J(x_k)d\|_2^2 \\ &= \frac{1}{2} F(x_k)^T F(x_k) + d^T J(x_k)^T F(x_k) + \frac{1}{2} d^T J(x_k)^T J(x_k)d \end{aligned}$$

Then m(d) is a better choice of merit function than f(d), and so the trust-region subproblem is

$$\underbrace{\min}_d m(d) = \left[\frac{1}{2} F(x_k)^T F(x_k) + d^T J(x_k)^T F(x_k) + \frac{1}{2} d^T J(x_k)^T J(x_k)d \right]$$

such that $\|D.d\| \le \Delta$. This subproblem can be efficiently solved using a dogleg strategy (see Nocedal [4]). For extra information on algorithms refer to Matlab documentation. The Algorithm option specifies a preference for which algorithm to use. It is only a preference because for the trust-region-reflective algorithm, the nonlinear system of equations cannot be underdetermined; that is, the number of equations must be at least as many as the number of unknowns. Similarly, for the trust-region-dogleg algorithm, the number of equations must be the same as the number of unknowns. The algorithm ‘trust-region-dogleg’ is the only algorithm that is specially designed to solve nonlinear equations and it is mentioned that for the 'trust-region-dogleg' algorithm, the number of equations must be the same as the number of unknowns. The other algorithms attempt to minimize the sum of squares of the function. In order to have exact solution we should have a square system, i.e. number of equations be equal to the number of unknowns. Assuming that there are m discretization points we can construct m equations. For the unknowns, we have $2(n_r + n_\theta) - 6$ unknowns from r and θ equations that are to be determined. If we want to have a square system we should have $m = 2(n_r + n_\theta) - 6$. The number of points is simply equal to the number of unknowns in Fourier series and is not a good point as we want to consider many points in covering the whole domain of time to fully capture the motion dynamics. So if we want to include more points the system is over determined and the 'Levenberg-marquardt' algorithm seems to be the best amongst the most efficient algorithms. So in the unconstrained version we would solve the following optimization problem:

$$J = \min \sum_{j=1}^{m} f_j^2$$

3.4 Constrained problem and Algorithms

The constrained problem is the same objective function subject to a constraint on thrust acceleration value at some points:

$$\frac{2\dot{r}\dot{\theta} + r\ddot{\theta}}{\cos(\alpha)} = T_a \rightarrow \boxed{T_a(r, \dot{r}, \dot{\theta}, \ddot{\theta}, t) = 0}$$

$$\left(\frac{T_a}{T_{a,\max}}\right)^2 \leq 1$$

which again is a function of the unknown coefficients:

$$C\left(a_0, a_1 \cdots a_{n_r}, b_1 \cdots b_{n_r}, c_0, c_1 \cdots a_{n_\theta}, d_1 \cdots d_{n_\theta}; t\right) \leq 0$$

$$J = \min \sum_{j=1}^{m} f_j^2 \quad ; sub. \;\; C \leq 0$$

For the constrained problem we would use fmincon function with the following algorithms:

- SQP
- Interior-point
- Active-set
- Trust-Region-Reflective

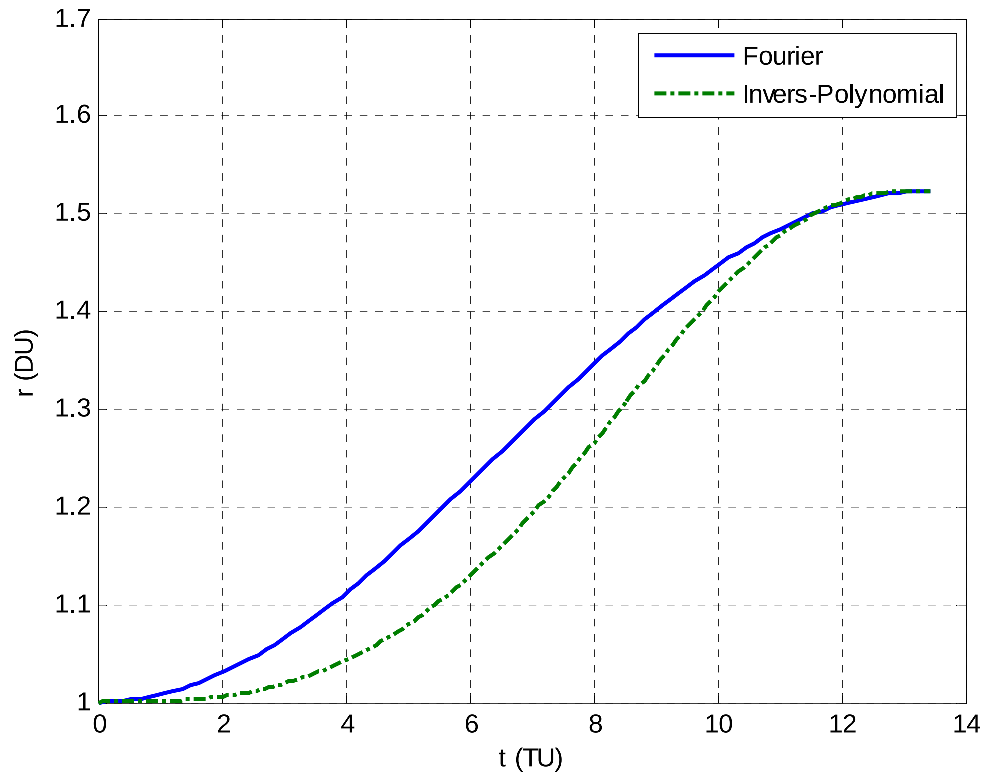
fmincon uses a sequential quadratic programming (SQP) method. In this method, the function solves a quadratic programming (QP) subproblem at each iteration. fmincon updates an estimate of the Hessian of the Lagrangian at each iteration using the ***BFGS*** formula and performs a ***line search***. The QP subproblem is solved using an ***active set strategy***. Trust-region-reflective is a subspace trust-region method and is based on the interior-reflective Newton method. Each iteration involves the approximate solution of a large linear system using the method of preconditioned conjugate gradients (PCG).

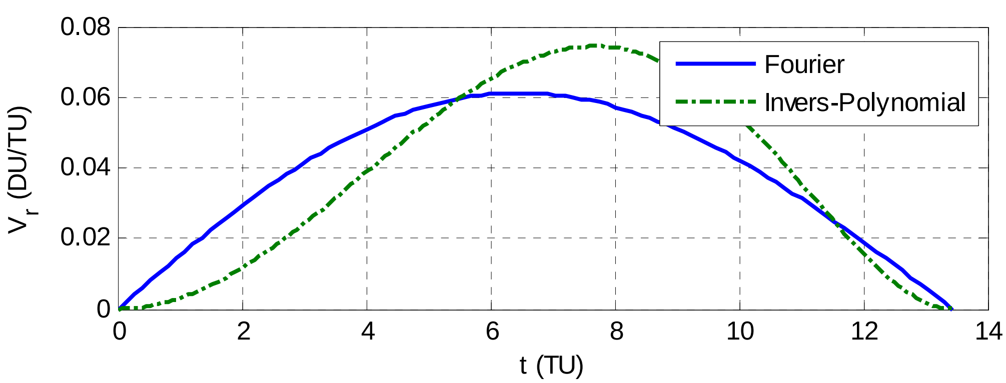
4. Earth-to-Mars transfer

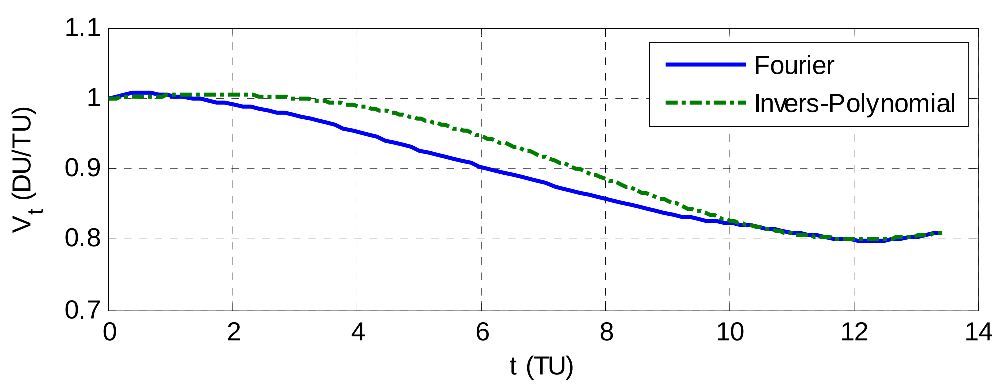
As it was stated previously, this method is developed to provide a rapid tool for the preliminary assessment of the trajectory feasibility. So, for this project we would consider a feasible trajectory and try to answer some questions considering the capability of finite Fourier series. As an example, Earth-Mars transfer is considered according to Ref.1. The information used for numerical evaluations are given in Table.1 using the canonical units (see Ref.3). Subscripts 1 and 2 refer to Earth and Mars respectively. For the sake of comparison we provided the results of the inverse polynomial method also. We assumed 22 points that include terminal points for discretization. For the constrained problem we set $T_{a_{\max}} = 0.02\left({}^{DU}\!/\!{}_{TU^2} \right)$.

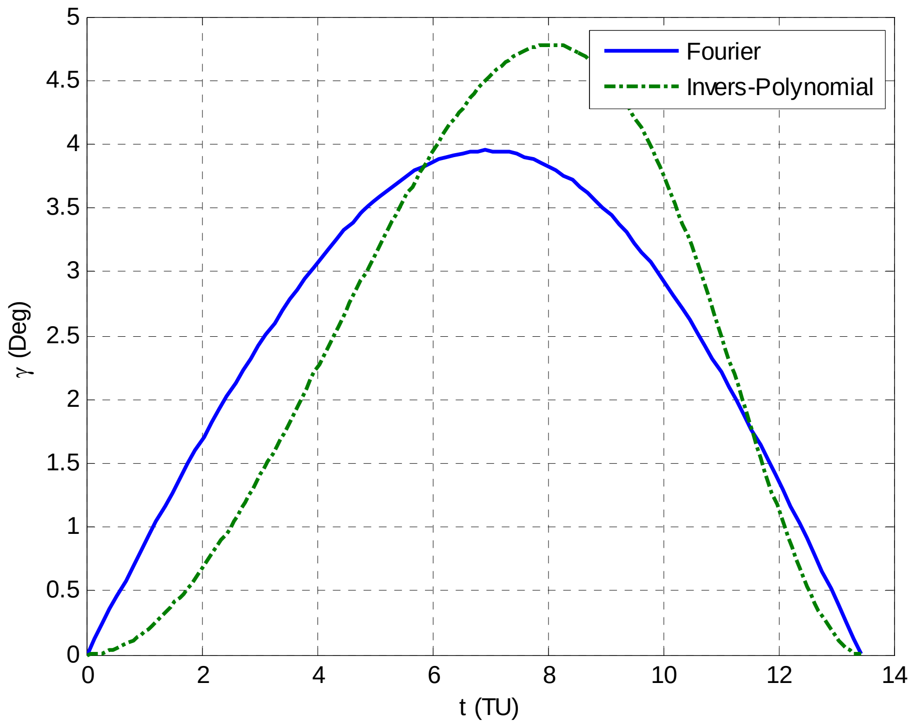
Table.1

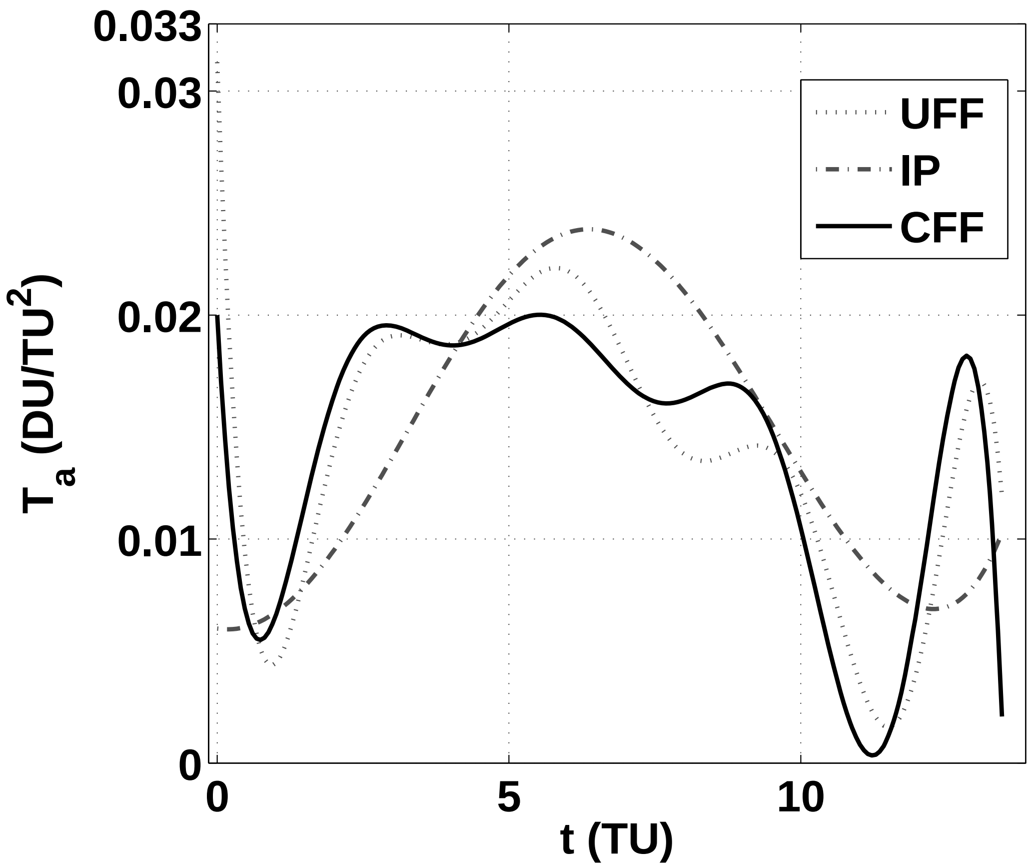
Parameter	Value
r_1	1
r_2	1.5234 (DU)
$\dot{r}_1$	0 (DU/TU)
$\dot{r}_2$	0 (DU/TU)
θ_1	0 (rad)
θ_2	9.8310 (rad)
$\dot{\theta}_1$	1(rad/sec)
$\dot{\theta}_2$	0.5318(rad/sec)
μ	1
T	13.447 (TU) = 781.73 Days

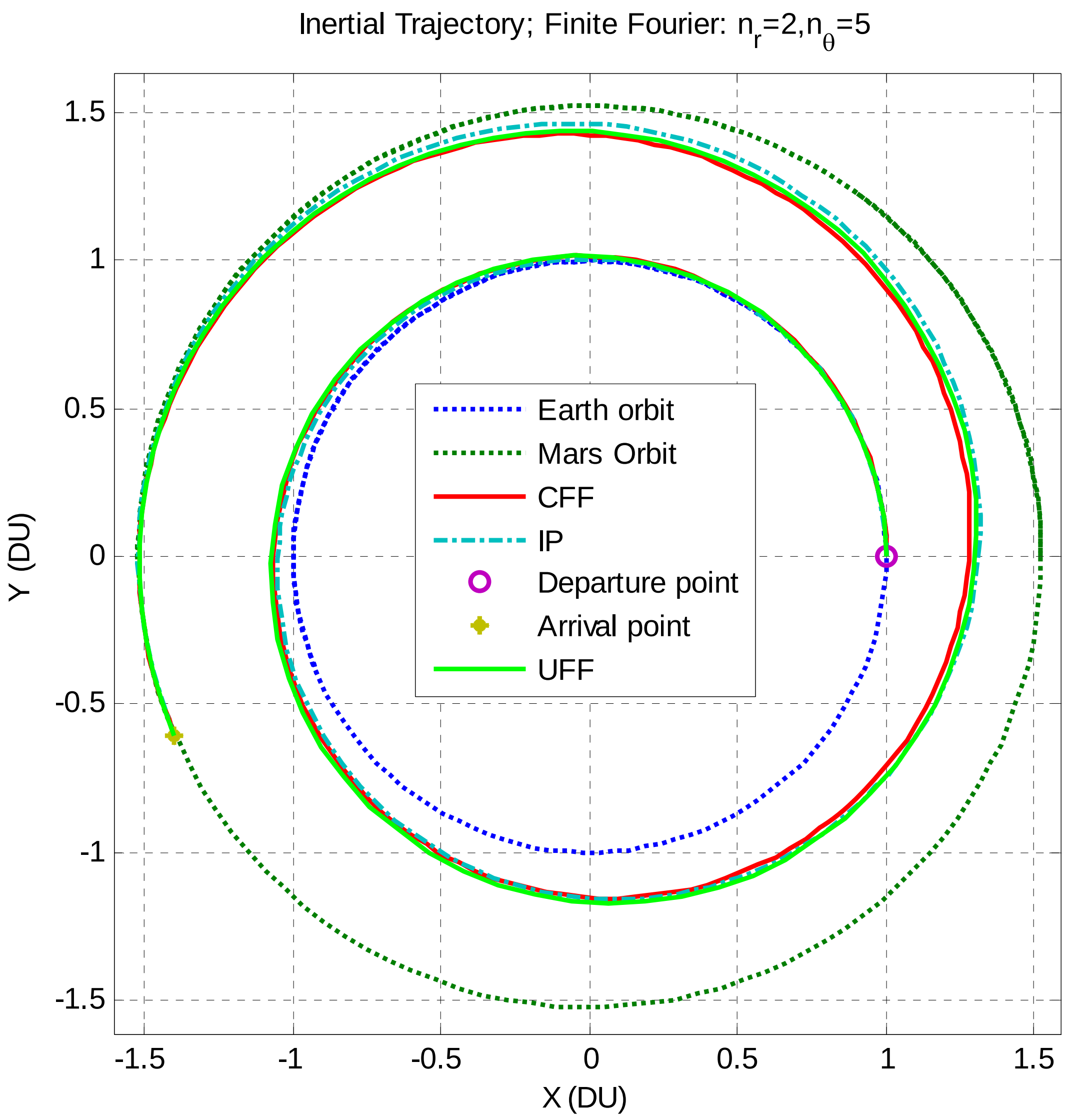

1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
r (DU)
0
2
4
6
8
10
12
14
t (TU)
Fourier
Invers-Polynomial


0.08
0.06
0.04
0.02
0
V r (DU/TU)
0
2
4
6
8
10
12
14
t (TU)
Fourier
Invers-Polynomial

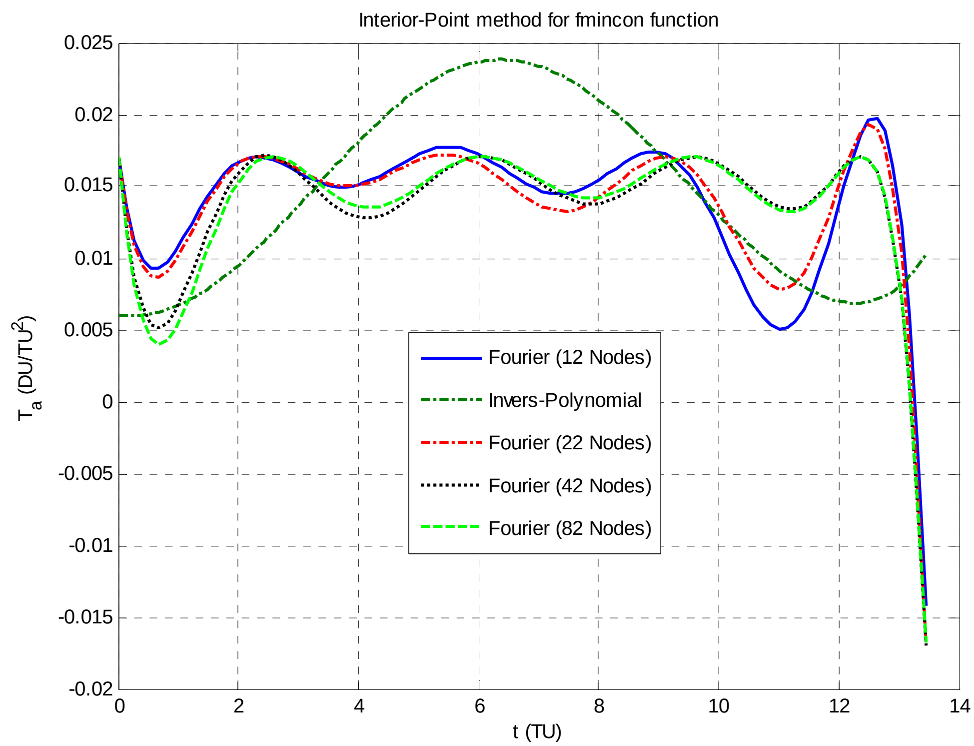

1.1
1
0.9
0.8
0.7
V t (DU/TU)
0
2
4
6
8
10
12
14
t (TU)
Fourier
Invers-Polynomial


Fourier
Invers-Polynomial
γ (Deg)
t (TU)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
2
4
6
8
10
12
14

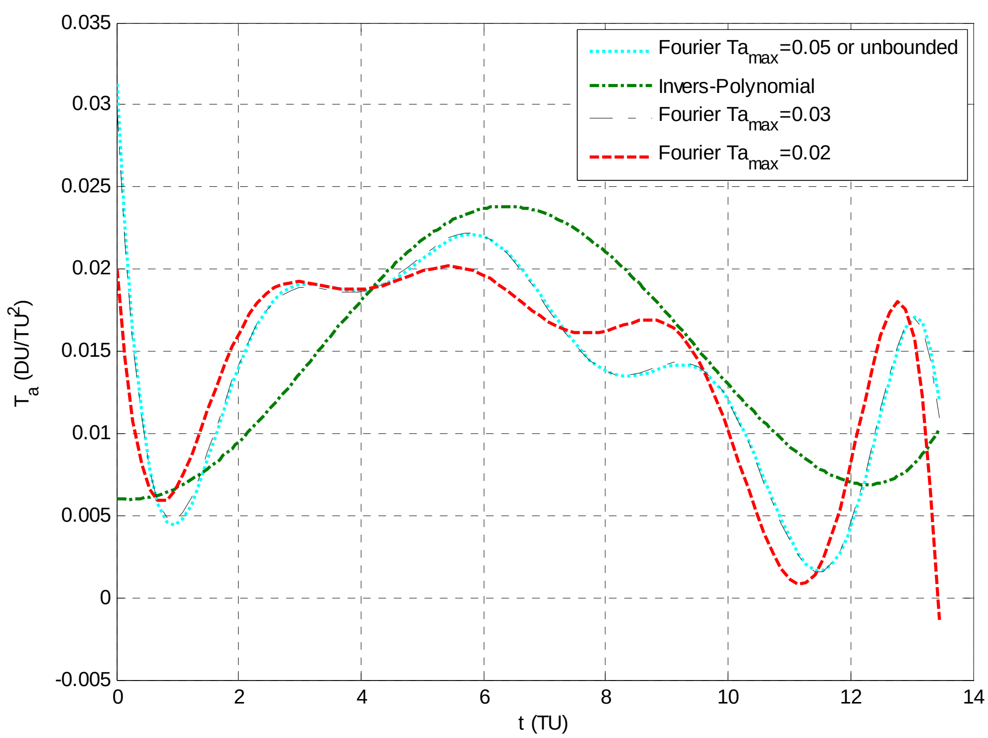

UFF
IP
CFF
T_a (DU/TU^2)
t (TU)
0.033
0.03
0.02
0.01
0
0
5
10



The final mass (m_f) of the Fourier-Expansion method is 3305.5 kg and using the inverse polynomial it is 3311.18 kg. It should be noted that this trajectory is completely a new one and we do not expect to get similar trend of variation for parameters such as thrust acceleration. In the next figure we set different number for discretization points and different cases for thrust acceleration limit.



For comparison, the first three cases are plotted together and represented in the following figure.



Computational time of each method in denoted in the following table using tic-toc command:

Method	CPU-time (sec)
Inverse Polynomial	0.11
Finite Fourier Series (SQP)	0.12
Finite Fourier Series (Interior-point)	0.36
Finite Fourier Series (Active-set)	0.15
Finite Fourier Series (Trust-Region-Reflective)	0.15

5. Conclusion

1- The presented method shows comparable performance with respect to the other existing methods.
2- The SQP method provides the best solution in terms of the performance time while all the algorithms satisfied the same tolerance put on optimality conditions.
3- Contrary to what we expected, the interior method did not show a fast performance in this problem.

6. References

[1]- A. Miele, T. Wang, P. N. Williams, "*Optimal Interplanetary Orbital Transfers*", journal of optimization theory and applications: Vol. 127, No. 3, pp. 605–625, December 2005, DOI: 10.1007/s10957-005-7506-9.
[2]- Wall, B.J., Conway, B.A., " *Shape-based approach to low-thrust rendezvous trajectory design*", JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS Vol. 32, No. 1, January–February 2009.
[3]- Stuart A. Stanton, "*Optimal Orbital Transfer Using a Legendre Pseudo-spectral Method*", Master of Science Thesis in Aeronautics and Astronautics, May 23, 2003.
[4]- Nocedal, J. and S. J. Wright. Numerical Optimization, Second Edition. Springer Series in Operations Research, Springer Verlag, 2006.