



**Politecnico  
di Torino**



**UNIVERSITY OF  
ILLINOIS**  
URBANA - CHAMPAIGN

# Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Aerospaziale  
A.A. 2021/2022  
Sessione di Laurea Marzo/Aprile 2022

## **Optimization of hazardous Near-Earth Objects deflection missions using a kinetic impactor**

Relatori:

*Professor Lorenzo Casalino  
Professor Bruce A Conway*

Candidata:

*Ludovica Malagni*



*Shoot for the Moon. Even if you miss,  
you will land among the stars.*

# Acknowledgements

First of all, I would like to thank my whole family and, in particular, my parents for always being there for me, for always supporting me, both morally and financially, in everything I wanted to do, for raising me and educating me to ambition and dream big and for giving me the possibility and the freedom to follow those dreams and try to make them come true.

My grandparents for all they have done for me. I hope you can be proud of me and this accomplishment of mine, wherever you are. I love you.

My friends for always believing in me, even when I didn't believe in myself, for always pushing me to give more and for always being there in the good times and especially in the bad ones.

Alessia for sharing this amazing experience with me and helping me with this project.

Professor Casalino for his collaboration and help in developing this project.

I would also like to thank our advisor at UIUC, Dr. Bruce A. Conway for giving us the opportunity to live a fantastic experience, for his incredible availability, kindness and his precious teachings. Thank you for making this experience even more unique.

And last but not least I would like to thank myself for always having the strength to get back up after every little fall and disappointment, for not believing that part of me that thought I wasn't enough, and for continuing to daydream, despite everything.

# Ringraziamenti

Vorrei ringraziare in primo luogo tutta la mia famiglia e, in particolare, i miei genitori per essermi sempre stati accanto, per avermi sempre supportata, sia moralmente che economicamente, in tutto ciò che volessi fare, per avermi cresciuto ed educato all'ambizione e al sognare in grande e per avermi dato la possibilità e la libertà di seguirli e tentare di realizzarli quei sogni.

I miei nonni per tutto quello che hanno fatto per me. Spero possiate essere orgogliosi di me e di questo mio traguardo, ovunque voi siate. Vi voglio bene.

Le mie amiche per aver sempre creduto in me, anche quando io stessa non ci credevo, per avermi sempre spronato a dare di più e per esserci sempre state nei momenti belli come e soprattutto in quelli brutti.

Alessia per aver condiviso con me questa fantastica esperienza ed avermi aiutato in questo progetto.

Il professor Casalino per la collaborazione e l'aiuto nello sviluppo di questo elaborato.

Vorrei anche ringraziare il nostro consulente alla UIUC, il Dr. Bruce A. Conway per averci dato l'opportunità di vivere un'esperienza fantastica, per la sua incredibile disponibilità, gentilezza e i suoi preziosi insegnamenti. Grazie per aver reso questa esperienza ancora più unica.

E per ultima ma non per importanza vorrei ringraziare me stessa per aver sempre avuto la forza di rialzarmi dopo ogni piccola caduta e delusione, per non aver creduto a quella parte di me che pensava di non essere abbastanza, per aver continuato e per continuare a sognare ad occhi aperti, nonostante tutto.



# Abstract

In this work a space trajectory is optimized for the case of a kinetic impactor spacecraft sent to collide with a threatening Earth-approaching asteroid, with the objective of maximizing the subsequent miss distance of the asteroid at its closet approach to Earth.

The direction of the thrust vector to direct the low-thrust spacecraft from Earth to the asteroid, the launch and interception dates, and initial Earth  $V$ -infinity departure direction are found by solving an optimal control problem so that the impact maximizes the resulting perigee of the subsequent Earth hyperbolic flyby. Due to the fact that the solution space considered by the optimizer is large and the objective function is complicated, intuition is not sufficient to provide an adequate initial guess for the nonlinear programming problem (NLP) solver used to optimize all aspects of the trajectory. A heuristic algorithm method called particle swarm optimization (PSO) is used to find an approximate solution that can then be used as an initial guess to the NLP optimizer and enable the problem to be treatable.

# Abstract

In questo lavoro viene ottimizzata una traiettoria spaziale per il caso di un veicolo spaziale ad impatto cinetico inviato a collidere con un minaccioso asteroide che si avvicina alla Terra, con l'obiettivo di massimizzare la successiva distanza di mancato impatto dell'asteroide al suo prossimo avvicinamento alla Terra.

La direzione che il vettore di spinta deve avere per dirigere il veicolo spaziale a bassa spinta dalla Terra all'asteroide, le date di lancio e di impatto, e la direzione di partenza iniziale  $V$ -infinito dalla Terra sono trovate risolvendo un problema di controllo ottimale in modo che l'impatto massimizzi il raggio di perigeo risultante del successivo flyby iperbolico della Terra. A causa del fatto che lo spazio di soluzione considerato dall'ottimizzatore è grande e la funzione obiettivo è complicata, l'intuizione non è sufficiente a fornire un'adeguata ipotesi iniziale per il solutore del problema di programmazione non lineare (NLP) utilizzato per ottimizzare tutti gli aspetti della traiettoria. Un metodo di algoritmo euristico chiamato particle swarm optimization (PSO) viene utilizzato per trovare una soluzione approssimativa che può poi essere utilizzata come ipotesi iniziale per l'ottimizzatore NLP e consentire al problema di essere trattabile.

# Table of Contents

List of Figures	I
List of Table	II
Nomenclature	III
<b>CHAPTER 1: Introduction</b>	
1.1 Asteroid Deflection Missions	1
1.2 Optimization Methods for Low-Thrust Spacecraft Trajectory	3
1.2.1 Analytical Solution	4
1.2.2 Numerical-conversion to NLP	5
1.2.3 Evolutionary and Heuristic methods	6
1.3 Objective	7
1.4 Thesis Outline	8
<b>CHAPTER 2: Numerical Optimization Method</b>	
2.1 Optimization of a dynamic system using PSO	9
2.1.1 Examples	12
2.2 Optimization of a dynamic system using NLP	19
2.3 Integration Methods	21
2.3.1 ODE	22
2.3.2 Euler Step	22
2.3.3 3-Step Runge-Kutta Parallel Shooting	24
2.4 Examples	26
2.5 Nonlinear Programming with SNOPT	32
2.5.1 Examples	33
<b>CHAPTER 3: Asteroid Deflection</b>	
3.1 Mission Description	36
3.1.1 Prior Work	37
3.1.2 The Asteroid	38
3.1.3 Description of the simulation	40

3.2	Governing Equations	41
3.2.1	Equation of Motion in Cylindrical Coordinates	41
3.2.2	Equation of Motion in Cartesian Coordinates	43
3.2.3	Units	44
3.3	Objective Function	44
3.3.1	State Transition Matrix	45

**CHAPTER 4: Optimal asteroid mitigation using a low-thrust spacecraft and kinetic impact**

4.1	Description of the problem	48
4.2	Initial Guess	49
4.3	Results	53

**CHAPTER 5: Conclusions**

5.1	Summary	58
5.2	Future Work	58

***References***

# List of Figures

Figure 1.1 - Orbits of all the known Potentially Hazardous Asteroids (PHAs)	2
Figure 2.1 - Illustration of the PSO method	10
Figure 2.2 - Optimal Hohmann Transfer Trajectory	14
Figure 2.3 - Optimal thrust pointing angle time history	17
Figure 2.4 - Optimal Low-Thrust spacecraft trajectory	17
Figure 2.5 - Optimal final energy for varying times	18
Figure 2.6 - Optimal thrust pointing angle time history (Points)	18
Figure 2.7 – Optimal final energy for varying times (Points)	19
Figure 2.8 - Problem structure for DTRK	24
Figure 2.9 - Illustration of the parallel-shooting method	25
Figure 2.10 – Optimal thrust pointing angle time history using fmincon- ODE45	27
Figure 2.11 – Optimal thrust pointing angle time history using fmincon - Euler step	28
Figure 2.12 - Optimal trajectory fictional asteroid interception 2D case	30
Figure 2.13 - Optimal thrust pointing angle time history fictional asteroid interception 2D case	31
Figure 2.14 - SNOPT Optimal trajectory fictional asteroid interception 2D case	34
Figure 2.15 - SNOPT Optimal thrust pointing angle time history fictional asteroid interception 2D case	35
Figure 3.1 - Asteroid 99942 Apophis - radar observations March 8-10, 2021	38
Figure 3.2 - Schematic representation of the simulation	40
Figure 3.3 - Cylindrical Coordinate system representation	42
Figure 3.4 - Schematic representation of the Objective Function	44
Figure 4.1 - In-plane thrust pointing angle initial guess time history	51
Figure 4.2 - Out-of-plane thrust pointing angle initial guess time history	52
Figure 4.3 - Optimal PSO initial guess trajectory	52
Figure 4.4 - Temporal variation of mass and acceleration	54
Figure 4.5 - Optimal Trajectory	55
Figure 4.6 - Optimal in-plane thrust pointing angle time history	55
Figure 4.7 - Optimal out-of-plane thrust pointing angle time history	56
Figure 4.8 - History of the semi-major axis of the interceptor orbits	56
Figure 4.9 - Local view of asteroid's close approach to Earth after deflection	57

# List of Tables

Table 2.1 - Results related with the optimal Hohmann Transfer	14
Table 2.2 - Results related with the Optimal Low-Thrust trajectory	17
Table 2.3 - Results related with the Optimal fictional asteroid interception 2D case	30
Table 2.4 - Results related with the optimal Hohmann Transfer - SNOPT	33
Table 2.5 - Results related with the Optimal Low-Thrust trajectory- SNOPT	34
Table 2.6 - Results related with the Optimal fictional asteroid interception 2D case - SNOPT	35
Table 3.1 - Apophis Orbit Elements at Epoch April 13, 2029	39
Table 4.1 - PSO lower and upper bounds for initial guess	50
Table 4.2 - Optimal values of PSO parameters for initial guess	51

# Nomenclature

$AU$	Astronomical Unit	$(1.496e^8 \text{ km})$
$TU$	Canonical Time Unit	$(58.13 \text{ days or } 5.02e^6 \text{ sec})$
$\vec{r}$	Cartesian position vector	
$\vec{v}$	Cartesian velocity vector	
$\delta \vec{r}$	change in Cartesian position vector relative to a reference	
$\delta \vec{v}$	change in Cartesian velocity vector relative to a reference	
$\Delta \vec{v}$	change in Cartesian velocity vector applied by an outside force	
$r$	distance from Sun	
$\theta$	angle from position vector to first point in Aries, in the ecliptic plane	
$z$	component of position vector normal to ecliptic plane	
$m$	spacecraft mass	
$\alpha$	flight path angle in ecliptic plane	
$\beta$	in-plane thrust pointing angle	
$\gamma$	out-of-plane thrust pointing angle	

$T_{max}$	maximum available thrust
$I_{sp}$	engine specific impulse
$c$	engine exhaust velocity
$t$	general symbol for epoch
$t_{launch}$	epoch of spacecraft launch
$t_{flight}$	flight time
$t_{intercept}$	epoch of spacecraft intercept
$\mu$	gravitational constant
$E$	orbit eccentric anomaly
$a$	orbit semi-major axis
$e$	orbit eccentricity
$i$	orbit inclination
$\omega$	orbit argument of periapse
$\Omega$	orbit longitude of ascending node
$f$	orbit true anomaly

$p$	orbit semi-latus rectum
$N$	number of time segments
$h$	length of each time segment
$u$	control value specified at the boundary between time segments
$v$	control value specified within a time segment
$P$	NLP parameter vector
$f(P)$	set of nonlinear constraints for nonlinear programming problem
$J(P)$	objective function for nonlinear programming problem
$A_L(P)$	set of linear constraints for nonlinear programming problem
$DTRK$	Direct Transcription with 4 <sup>th</sup> order Runge-Kutta integration rule
$PSO$	Particle Swarm Optimization
$NLP$	Nonlinear Programming problem
$RK4$	4 <sup>th</sup> order Runge-Kutta integration rule

Many of the quantities listed above are attached to different bodies in the solar system: the Earth, the Sun, the target asteroid, and the interceptor spacecraft. The following subscripts are used to identify to which body the property refers:

$X_{\odot}$             property  $X$  of the Sun

$X_{\oplus}$             property  $X$  of the Earth

$X_{\otimes}$             property  $X$  of the target asteroid

$X_{s/c}$             property  $X$  of the spacecraft



# Chapter 1: Introduction

## 1.1 Asteroid Deflection Missions

Humans think they live in a safe place and that all the dangers they face in their lives are in front of them, on earth, disregarding the dangers coming from the space in which our planet has been moving for millennia.

The Earth gets hit by asteroids and small debris all the time. Most of them break up in the atmosphere and the probability of a large asteroid hitting the earth, reaching the ground and causing damage to on a regional scale is small but not negligible.

The extreme character of the impacts between the Earth and the asteroids makes it difficult for the general public to understand the true nature of the problem we are facing. The fact that this type of collisions are rare does not mean that they are impossible [16]. In fact, there are evidences both indirect and direct showing that this kind of episodes have occurred in the past and will take place again in the future.

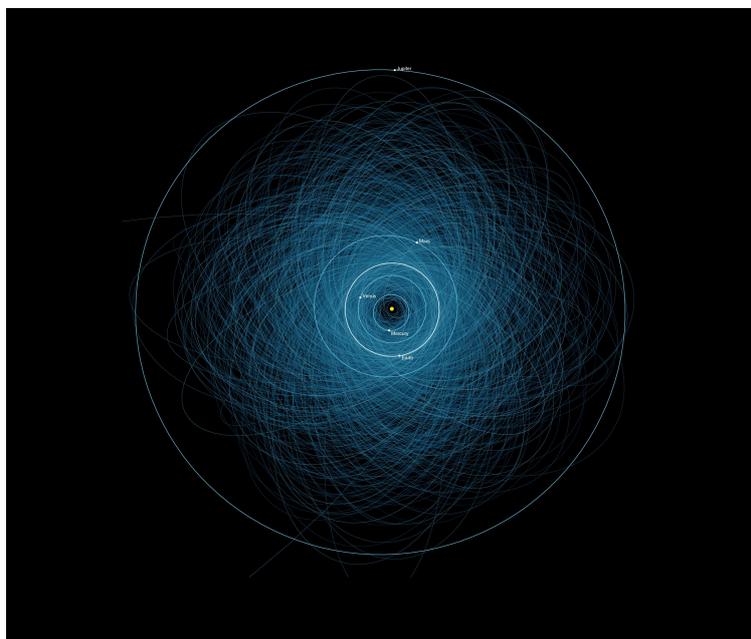
Planetary Defence has started to be taken seriously only in the last two decades, following events that have demonstrated the danger of an asteroid impact, such as the impact of the comet Shoemaker-Levy with Jupiter in July 1994 or the “Tunguska event”, the atmospheric explosion of a stony meteoroid, which took place in Siberia in June 1908.

Among the population of asteroids only the ones with orbits close to Earth's, known as Near Earth Asteroids (NEAs), constitute a serious threat. There are three classes of Near-Earth Asteroids: Atens, Apollos, and Amors. Atens-type asteroids have a semi-major axis smaller than 1 AU and an aphelion greater than 0.983 AU, while Apollo-type asteroids have a semi-major axis greater than 1 AU and a perihelion smaller than 1.017 AU. Hence, Apollo- type and Atens-type asteroids can have Earth-crossing orbits. Amors have orbits that lie completely outside Earth's orbit (perihelial distance between 1.017 and 1.3 AU) but have the potential to be perturbed into Earth-crossing trajectories [21].

There is a number of different measures that can be taken to mitigate this hazard. The most basic measures have a preventative nature and aim at cataloging the whole NEA population. A complete catalog of accurate orbits would allow us to know, well in advance, the asteroids that are on a collision course. The second type of measures, more

selective and targeted for a particular threatening asteroid, include different deflection techniques to avoid its collision with the Earth. Finally, if the previous actions fail, it is still possible to implement a third kind of measures, such as population evacuations before the impact or construction of refuges near the shock area, to mitigate the effects of a certain collision [1].

With these purposes in the 2016 NASA established the Planetary Defense Coordination Office, whose task is indeed to provides early detection of potentially hazardous objects (PHOs), tracks and characterizes PHOs and issues warnings of the possible effects of potential impacts and studies strategies and technologies for mitigating PHO impacts.



*Figure 1.1 - Orbits of all the known Potentially Hazardous Asteroids (PHAs) [Credit: NASA/JPL-Caltech]*

Figure 1.1 shows the orbits of all the known Potentially Hazardous Asteroids (PHAs), numbering over 1,400 as of early 2013. These are the asteroids considered hazardous because they are fairly large (at least 140 meters in size), and because they follow orbits that pass close to the Earth's orbit (within 7.5 million kilometers). But being classified as a PHA does not mean that an asteroid will impact the Earth [22].

In the NASA Planetary Defense division different methods to deflect an asteroid on a course to impact Earth are being studied.

One of these techniques is called a gravity tractor which would deflect another object in space without physically contacting it, using only its gravitational field to transmit the required impulse.

A kinetic impactor, which is the method chosen in this work, is currently the simplest and most technologically mature method available to defend against asteroids. In this technique, a spacecraft is launched and simply slams itself into the asteroid. If this is done far enough in advance then it will only be necessary to change the speed by a millimeter a second or less and then it will miss the Earth entirely. Scientists are testing the kinetic impact technique by the Double-Asteroid Redirect Test mission (DART) on an asteroid system called Didymos.

Nuclear explosive device methods are considered the last resort when it comes to NEO deflection, although they may be the most effective for preventing a cataclysmic event. This is the most effective option when the warning time is short or the asteroid is large. In this method, a nuclear device is detonated a few hundred meters above the surface of the asteroid, emitting energy in the form of X-rays. It is this energy that instantly reaches the surface of the asteroid, causing it to vaporize, which creates a thrust that deflects its trajectory.

## **1.2 Optimization Methods for Low-Thrust Spacecraft Trajectory**

The main purpose of a Near-Earth Object deflection mission through a kinetic impactor is to transfer the maximum amount of momentum to the asteroid.

To achieve this, the spacecraft has to impact its target with the greatest possible mass.

Chemical rocket propelled spacecraft can offer a high thrust, but it has an excessive mass reduction due to high propellant consumption and so this type of spacecraft is not suitable for the type of mission analyzed.

Low-thrust electric motors have a higher specific impulse and greater efficiency and are therefore preferable for the deflection mission. On the other hand electric propulsion produces a very small thrust, with typical spacecraft acceleration on the order of  $10^{-5}$  g, so much so that the thrust is used continuously or almost continuously over the duration of the mission instead of applying the required  $\Delta v$  as an impulse. This can prove to be a complication for trajectory optimization as there is no analytical solution to the general

problem of optimizing a continuous-thrust trajectory. It is therefore necessary to use numerical methods.

The Spacecraft Trajectory Optimization problem may simply be expressed as the determination of a trajectory for a spacecraft that satisfy specific initial and final condition, while minimizing some quantity of importance. The most common missions are those that aim to minimize the propellant required or those where minimizing the time of flight is the important thing or again, problems where these two things are synonymous, as happens in cases of continuous thrust.

In the context of numerical optimization methods for continuous control problems, two types of solutions can be generically identified. On the one hand there are the solutions that use the analytical necessary conditions from the calculus of variations, these are called *Indirect* solutions. These solutions require the addition of the costate variables of the problem, equal in number to the state variables, and their governing equations. This doubles the size of the dynamical system.

On the other hand there are the solutions, called *Direct* solution, which transcribe the continuous optimal control problem into a parameter optimization problem. Satisfaction of the system equations is accomplished by integrating them stepwise using either implicit or explicit rules. The problem is thus converted into a nonlinear programming problem [2].

Significant progress in the direct solution for the optimal control problem has occurred over the past decade. All solutions can be roughly categorized into being analytical or numerical.

### 1.2.1 Analytical Solutions

This is the original approach for spacecraft trajectory optimization. The problem can be described in terms of a dynamical system of differential equations:

$$\dot{x} = f(x, u, t) \tag{1.1}$$

where  $x$  represent an n-dimensional state (vector) and  $u$  represent the m-dimensional control (vector). The state vector is problem dependent and it can be formulated in different way, for the three-body problems are typically used the cartesian coordinates. The control vector is typically a control of thrust magnitude and direction or its equivalent. Some initial and terminal conditions are specified for the problem.

The goal is then to minimize the objective:

$$J = \phi[x(T), T] + \int_0^T L[x, u, t]dt \quad (1.2)$$

subject to constraints on the path  $\vec{x}(t)$  and the controls  $\vec{u}(t)$ .

The problem can be solved by defining a system Hamiltonian. The set of all equations constitutes a two-point-boundar-value problem (TPBVP). For all but the most elementary optimal control problems, the solution of this TPBVP is challenging and numerical solutions are required. The optimal control is instead chosen according to Pontryagin's Minimum Principle [2].

### 1.2.2 Numerical-conversion to NLP

Many method have been developed to solve the TPBVP numerically. These methods seek to reduce the optimal control problems to parameter optimization problems that can be solved by a nonlinear programming (NLP) problem solver.

These methods include: the most obvious and well known *shooting*, the *finite-difference methods* and the *collocation* methods. The long-recognized difficulty of the "indirect" approach to determining the optimal trajectory is that the initial costate variables of the TPBVP are unknown and further that the nonlinearity of the problem means that the vector flow is very sensitive to some or all of these initial costate variables [2].

A variety of direct solution methods have been developed. The most successful approach is called "*direct transcription*". In this method the continuous problem is discretized and state and control variables are known only at discrete times. Satisfaction of the equations of motion is achieved by employing an explicit or implicit numerical integration rule that needs to be satisfied at each step.

This approach is significantly more robust than shooting methods because it eliminates the sequential nature of the shooting solution, with its forward numerical integration, in favor of a solution in which simultaneous changes in all of the discrete state and costate parameters are made in order to satisfy algebraic constraints.

Direct methods has been significantly developed in the last two decades and it would be accurate to say that the grate majority of optimal space trajectories are now determined numerically.

### 1.2.3 Evolutionary and Heuristic methods

Recently, a different approach is becoming increasingly popular: the use of “*evolutionary*” algorithms (EA). The best known of the EAs is the *genetic algorithm* (GA), which model the evolution of a species based on Darwin's principle of survival of the fittest. These numerical optimizers determine, using methods similar to those found in nature, an optimal set of discrete parameters that has been used to characterize the problem solution. Two main advantages can be identified using the EA: no initial “guess” of the solution is required, and they are more likely than other methods to locate a global minimum in the search space rather than be attracted to a local minimum.

The use of EA allows the problem solution to be described by a relatively small set of discrete parameters, compared to the vector of parameters of a nonlinear program. The set of parameters describing the solution is written as a string or sequence of numbers. Each of these sequences if converted to binary form can remember a chromosome. Every sequence can be "decoded" to yield a trajectory whose cost or objective value can be determined.

The first step in the GA is the generation of a “population” of sequences using a random process. Three natural processes are then used to improve the population:

- *Selection*: removes the worst sequences and may also, via elitism, guarantee that the best sequence survives unchanged;
- *Combination*: remaining sequences are used as “parents”, partial sequences from two parents are combined to form new individuals;
- *Mutation*: changes a randomly chosen bit in a small fraction of the population.

The process is then repeated and the cost of every individual in the new generation is determined. Since the best individual from the previous generation was retained, the objective may improve but cannot worsen [2].

Termination of the algorithm is usually done after a fixed number of generations or after the objective has reached a plateau and a minimum has been found.

Similar to EAs are Heuristic methods, in which no solution is discarded but every particle tries to improve its cost. Among these the most famous is method is *particle swarm optimization* (PSO). This particular method uses some number of particles randomly distributed in a N-dimensional decision parameter space. The objective value is determined for the solution vector corresponding to each particle.

It is assumed that the particles can communicate with each other, according to an anthropomorphic view, so that all know the objective value for all the others. Every particle take a step in the parameter space at every time step. This step has three components: the first is an “*inertia*” that causes the particle to move in the direction it had previously been moving, the second is the “*nostalgia*” component that reflects a tendency for the particle to move toward its own most satisfactory position, and the third “*social*” component draws the particle toward the best position found by any of its colleagues.

As already seen with the GA, the process can be terminated after a fixed number of iterations or when the “best” solution has not changed for several iterations. This method has proven quite robust, is also very simple to use, and particularly good in locating global minima when the solution space contains many local minima.

For all these reasons, PSO has been very useful when applied to optimizing space trajectories, especially when used to provide an initial guess for more accurate methods, for example collocation with NLP as we did in this project.

### **1.3 Objective**

Knowledge of numerical solutions for trajectory optimization, both direct transcription and heuristic methods, was developed during the development period of this project.

This has been made possible by studying and solving increasingly sophisticated example problems, starting from the application of the various methods in the solution of the well-known Hohmann transfer with two impulses. Understanding the use of PSO in this easy example, we moved on to the application of this method on low-thrust problems, starting with the problem of maximizing the final energy of a low-thrust trajectory. This problem has been solved both by using PSO and by using the `fmincon` optimizer already present in Matlab. With this example we also tried to replace the Matlab ODE integration, used to integrate the equations of motion, with an Euler step based integration.

Then we approached what was the final goal by starting to solve the problem of intercepting a fictitious asteroid, first in 2D in both cylindrical and Cartesian coordinates and then in the 3D version.

The latter was made progressively more complete with the addition of the mass equation to the set of equations of motion, the initial impulse  $\delta \vec{v}$  that the upper stage of the launch vehicle gives to the spacecraft when it releases it, and the true position and velocity of the Earth.

The knowledge of position and velocity at a certain time of the various celestial bodies involved in the analysis has been possible thanks to the introduction in the code of orbital mechanics tools such as SPICE, a toolkit developed by NASA that can compute many kinds of observation geometry parameters at selected times.

The ultimate goal is the full simulation of optimal asteroid mitigation using a low-thrust spacecraft that through collision brings the asteroid to a maximum deflection of its orbit at the time of its closest approach to the Earth.

Deflection is calculated through the use and evaluation of the State Transition Matrix (STM) which calculates the perturbation in position and velocity of the asteroid as a result of the initial perturbing impulse.

## **1.4 Thesis Outline**

Chapter 2 describes the numerical optimization method including the presentation of some examples, and the different numerical integration methods used during the development of the project. Chapter 3 contains a description of the mission, the derivation of the objective function, the definitions of the coordinate variables and the equations of motion, and a description and justification of the units used. Chapter 4 presents the formulation of the optimal problem and discusses the results. Chapter 5 contains conclusions and recommendations for future work.

# Chapter 2: Numerical Optimization Method

## 2.1 Optimization of a dynamic system using PSO

The determination of optimal (either minimum-time or minimum-propellant-consumption) space trajectories has been pursued for decades with different numerical optimization methods, which in general can be classified as deterministic or stochastic methods. In the last decade, the development of effective stochastic methods has been preferred over deterministic gradient-based methods because the latter have limitations. In fact, gradient-based methods assume the continuity and differentiability of the objective function to be minimized and they are local in nature and require the identification of a suitable first-attempt "solution" in the region of convergence, which is unknown a priori and strongly problem dependent [3].

The stochastic methods are also referred to as evolutionary algorithms and are inspired by natural phenomena. Evolutionary computation techniques exploit a population of individuals, representing possible solutions to the problem of interest. The optimal solution is sought through cooperation and competition among individuals.

Heuristic algorithms, also inspired by natural phenomena, represent an alternative to the previously mentioned evolutionary algorithms.

The particle swarm optimization (PSO) technique was first introduced by Eberhart and Kennedy in 1995 and belongs to the category of swarm intelligence methods [4,5]. It mimics the unpredictable motion of bird flocks while searching for food, taking advantage of the mechanism of information sharing that affects the overall behavior of a swarm.

The initial population that composes the swarm is randomly generated at the first iteration of the process. Each particle is associated with a position vector and a velocity vector at a given iteration. More specifically, the position vector includes the values of the unknown parameters of the problem, whereas the velocity vector determines the position update. Each particle represents a possible solution to the problem and corresponds to a specific value of the objective (or fitness) function. At the end of the process, the best particle (i.e., the best solution with reference to the objective function) is selected. Both the position and the velocity vector are updated in a single iteration.

For each particle, the formula for velocity update includes three terms with stochastic weights; one of these terms is the so-called social component, related to the collective best position ever visited by a portion of the particles that form the swarm.

Let  $x_i(n)$  denote the position of particle  $i$  at the  $n$ th time step. At the next iteration, the particle take a step  $\nu_i(n + 1)$  in the parameter space so that the new position of particle  $i$  becomes

$$x_i(n + 1) = x_i(n) + \nu_i(n + 1) \quad (2.1)$$

with

$$\nu_{ij}(n + 1) = \nu_{ij}(n) + c_1 r_{1j}(n)[y_{ij}(n) - x_{ij}(n)] + c_2 r_{2j}(n)[\hat{y}_j(n) - x_{ij}(n)] \quad (2.2)$$

where  $\nu_{ij}(n)$  is the velocity (step) for component  $j$  of particle  $i$  at time step  $n$ ,  $x_{ij}(n)$  is the  $j^{th}$  component of the position of particle  $i$  at the  $n^{th}$  time step,  $r_{1j}(n)$  and  $r_{2j}(n) \subset U(0,1)$  are random values in the range  $[0, 1]$  sampled from a uniform distribution,  $y_i(n)$  is the “personal best” position, the best position located by the particle since the first time step,  $\hat{y}_j(n)$  is the “global best” position, the best position located by the any particle of the swarm since the first step.

Therefore, three different components can be identified within the step. The first one is the “inertia” component that causes the particle to move in the direction it had previously been moving, the second one is the “nostalgia” component that reflects a tendency for the particle to move toward its own most satisfactory position, and the third one is the “social” component which draws the particle toward the best position found by any of its colleagues. The  $c$ ’s are constants that weight the importance of the three components and the  $r$ ’s provide stochasticity to the system.

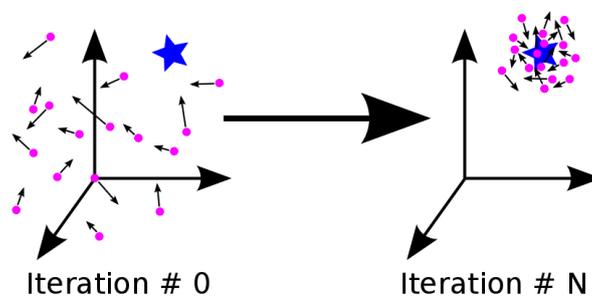


Figure 2.1 - Illustration of the PSO method [23]

Termination of the algorithm can occur after a fixed number of iterations or when the “best” solution has not changed for several iterations.

The basic version of the particle swarm algorithm appears as very intuitive and is extremely easy to program. In addition, this kind of method is well suited for finding the globally optimal solution to an optimization problem and requires only the definition of the search space for the unknown parameters. Although computationally expensive with respect to gradient-based methods, in the scientific literature the particle swarm technique is reported to be more efficient when compared to genetic algorithms, due to a reduced number of function evaluations. This method has thus proven to be quite robust, it is also very simple to use, and particularly good in locating global minima when the solution space contains many local minima.

In the context of space trajectories, the optimization problems of interest usually consist of minimizing a given objective function related with the time evolution of a dynamical system, which can be governed either by differential equations or by algebraic equations. The minimization is achieved by selecting the optimal values of the unknown parameters and time-varying variables. Several methodologies can be employed to translate the optimal control problems that involve continuous time-dependent control variables into parameter optimization problems. If the system dynamics are governed by a set of algebraic (nonlinear) equations the problem reduces to a nonlinear programming problem. Definitely, in both cases - in the presence of optimal control problems or nonlinear programming problems - the optimization process is aimed at finding the optimal values of a set of unknown parameters.

Space trajectory optimization problems must be frequently modeled as constrained optimization problems, which means that they involve equalities and/or inequalities regarding the unknown parameters. To deal with this type of problem, the PSO algorithm must be adapted.

Equality constraints reduce the degree of freedom of the problem according to their number. This leads to increased difficulty in dealing with these types of problems as they considerably narrow the search space where a feasible solution can be found. The most common approach for dealing with these constraints consist in penalizing them by summing additional terms to the objective function.

Inequality constraints are less problematic due to the fact that they reduce the search space of feasible solutions without decreasing the degree of freedom of the problem. For each particle, the simplest way of treating inequality constraints consists of assigning a

fictitious infinite value to the fitness function if the particle violates at least one of them. In addition, the corresponding velocity is set to zero, so that the successive velocity update is affected only by the social term and by the cognitive term. This circumstance statistically leads the particle to a feasible region of the search space.

### 2.1.1 Examples

With the aim of becoming familiar with the use of this optimization algorithm, PSO was applied to several examples, from the easiest to the most complex.

**Hohmann Transfer:** The Hohmann transfer is the most energy efficient two-impulse maneuver for transferring between two coplanar circular orbits sharing a common focus. The Hohmann transfer is an elliptical orbit tangent to both circles at its apse line. The periapse and apoapse of the transfer ellipse are the radii of the inner and outer circles, respectively. Obviously, only one-half of the ellipse is flown during the maneuver, which can occur in either direction, from the inner to the outer circle or vice versa. [6] In this type of orbits having the two  $\Delta V$  parallel to the velocity components of the two circular orbits (thus the two flight path angles are zero) results in minimal misalignment losses.

The problem consists of determining the optimal direction and magnitude of the two impulses and the time taken to perform the transfer between the two orbits.

The initial condition of the spacecraft, referred to the instant immediately before applying the impulse, are as follows:

$$v_r(t_0) = 0 \quad v_\theta(t_0) = \sqrt{\frac{\mu}{R_1}} \quad r(t_0) = R_1 \quad (2.3)$$

where  $\mu$  represents the gravitational parameter of the attracting body,  $v_r$  and  $v_\theta$  denote the radial and the horizontal component of the velocity,  $r$  is the radius.

The following parameters can be used to define the Hohmann transfer trajectory:

$$[\Delta v_1 \quad \gamma_1 \quad T_f \quad \Delta v_2 \quad \gamma_2] \quad (2.4)$$

where  $\Delta v_1$  and  $\Delta v_2$  represent the magnitudes of the two impulsive change of velocity,  $\gamma_1$  and  $\gamma_2$  their respective direction and finally  $T_f$  is the time spent on the journey.

After the first impulse, the velocity components  $v_r$  and  $v_\theta$  change to

$$v_r = v_r(t_0) + \Delta v_1 \sin(\gamma_1) \quad \text{and} \quad v_\theta = v_\theta(t_0) + \Delta v_1 \cos(\gamma_1) \quad (2.5)$$

whereas  $r = r(t_0) = R_1$ .

The optimal spacecraft transfer, which is sought by the PSO algorithm, minimizes the characteristic velocity of the overall orbital maneuver. This means that the objective function for the problem at hand is

$$J = \Delta v_1 + \Delta v_2 \quad (2.6)$$

The Hohmann transfer takes the spacecraft from one circular orbit to entry into another circular orbit. Therefore, the following constraints must be met:

$$v_r(t_f) = 0 \quad v_\theta(t_f) = \sqrt{\frac{\mu}{R_2}} \quad r(t_f) = R_2 \quad (2.7)$$

The orbital maneuver must also satisfy the equations of motion, expressed using polar coordinates and integrated using the Matlab ODE45 function.

For the problem in hand, the PSO algorithm employs 100 particles and is run for a maximum of 200 iterations. Each particle include the values of the five unknown parameters

$$[\Delta v_1 \quad \gamma_1 \quad T_f \quad \Delta v_2 \quad \gamma_2]$$

The problem is solved by employing a normalized set of units using astronomical unit (AU) as a measure of distance and time unit (TU) as a measure of time. This leads us to

consider the gravitational parameter of the Sun  $\mu_\odot = 1 \frac{AU^3}{TU^2}$ , the radius of the initial orbit equal to  $R_1 = 1$  AU and the one of the final orbit equal to  $R_2 = 3$  AU.

Instead, the impulsive changes of velocity are normalized by the circular velocities of the respective orbits.

The search space is defined by the following inequalities:

$$0.1 \frac{AU}{TU} \leq \frac{\Delta v_i}{V_{C_i}} \leq 0.5 \frac{AU}{TU} \quad -0.2 \leq \gamma_i \leq 0.2 \quad 0 \leq T_f \leq 18 \text{ AU} \quad (i = 1,2) \quad (2.8)$$

The table 2.1 collect the results of the optimization process, that are the optimal direction and magnitude of the two impulses and the time taken to perform the transfer between the two orbits. Figure 2.2 illustrates the optimal Hohmann transfer trajectory.

Table 2.1 Results related with the optimal Hohmann Transfer

Impulse	$\Delta v$ (AU/TU)	$\gamma$ (deg)	$T$ (TU)	$J$
1	0.2247	0	0	0.3938
2	0.1691	0	8.9094	

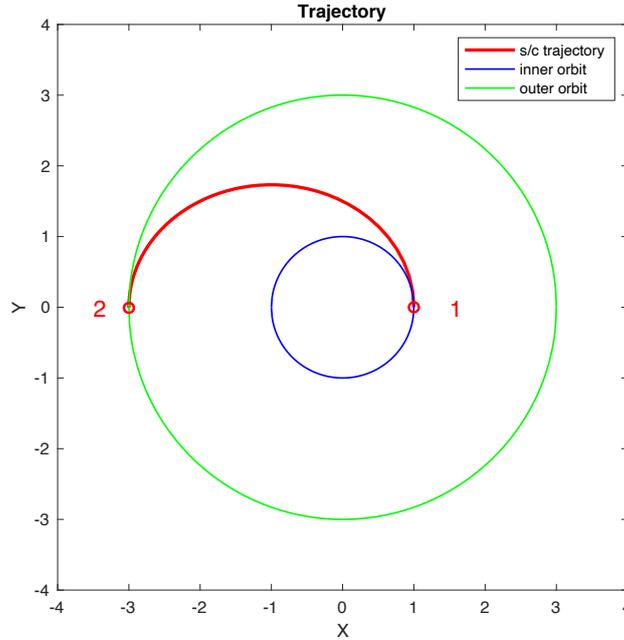


Figure 2.2 Optimal Hohmann Transfer Trajectory

If we compare these results with those below, obtained from the analytical calculation:

$$\Delta V_1 = V_{1_H} - V_{1_C} = \sqrt{\frac{\mu}{R_1}} \left( \sqrt{\frac{2R_2}{R_1 + R_2}} - 1 \right) = 0.2247$$

$$\Delta V_2 = V_{2_C} - V_{1_H} = \sqrt{\frac{\mu}{R_2}} \left( 1 - \sqrt{\frac{2R_1}{R_1 + R_2}} \right) = 0.1691$$

$$J = \Delta V_1 + \Delta V_2 = 0.3938$$

we can see an exact agreement between analytical solution and optimized solution.

***Low-Thrust spacecraft orbit transfer with maximization of the final energy:***

The problem under investigation is a continuous thrust, orbit transfer problem in which a constant acceleration is imparted on the spacecraft from the rocket engine. The main purpose of this orbital transfer is to determine the time history of the thrust pointing angle that maximizes the final energy in a fixed time of flight. Motion is confined to a single plane and, as mentioned in the previous example, the spacecraft's position is described using polar coordinates  $\{ r, \theta \}$  which have their origin located at the center-of-mass of the attracting body. The only control variable is the thrust angle,  $\beta$ , which is measured relative to the local horizontal. Finally the thrust acceleration,  $A$ , is the thrust divided by the mass of the spacecraft and is considered to be of constant magnitude and to have the value  $A = 0.025$ , for this investigation.

The problem is solved by employing a normalized set of units using astronomical unit (AU) as a measure of distance and time unit (TU) as a measure of time. This leads us to consider the gravitational parameter of the Sun  $\mu_{\odot} = 1 \frac{AU^3}{TU^2}$ .

The equations governing the motion of the spacecraft are then expressed as:

$$\begin{cases} \dot{r} = v_r \\ \dot{\theta} = \frac{v_{\theta}}{r} \\ \dot{v}_r = \frac{v_{\theta}^2}{r} - \frac{1}{r^2} + A \sin \beta \\ \dot{v}_{\theta} = -\frac{v_r v_{\theta}}{r} + A \cos \beta \end{cases}$$

where  $\beta$  is the in-plane thrust pointing angle and  $A$  is the magnitude of the thrust acceleration.

The initial condition of the spacecraft, referred to the instant immediately before applying the constant acceleration, are as follows:

$$v_r(t_0) = 0 \quad v_{\theta}(t_0) = \sqrt{\frac{\mu}{R_1}} \quad r(t_0) = R_1 = 1.1 \text{ AU} \quad (2.9)$$

The fixed time of flight is chosen to have the value of  $T_f = 16 \text{ TU}$ .

The optimal Low-Thrust orbital transfer, which is sought by the PSO algorithm, maximize the final energy of the overall orbital maneuver. This means that the objective function for the problem at hand is

$$J = - \left( \frac{V^2}{2} - \frac{1}{R_f} \right) \quad (2.10)$$

where the minus sign in front indicates the maximization of the quantity in parenthesis and  $V = \sqrt{v_r^2 + v_\theta^2}$ .

There are no final condition constraints imposed but the orbital maneuver must satisfy the equations of motion integrated using the Matlab ODE45 function.

For the problem in hand, the PSO algorithm use a population of 100 particles and is run for a maximum of 200 iterations.

The problem under consideration was approached by parameterizing the variable  $\beta$  to be optimized in two different methodologies.

In the first case  $\beta$  was parameterized using the following formulation

$$\beta = (a + bt)\sin(\omega t + \phi) \quad (2.11)$$

Each particle include the values of the four unknown parameters:

$$[a \quad b \quad \omega \quad \phi]$$

and the search space is defined by the following inequalities

$$-0.5 \leq i \leq 0.5 \quad (i = a, b, \omega, \phi) \quad (2.12)$$

The Table 2.2 collect the result of the optimization process, that are the optimal values of the parameters that define  $\beta$ , whose optimal time history is shown in Figure 2.3, and the value of the final energy  $J$ . Figure 2.4 illustrates the optimal low-thrust transfer trajectory and Figure 2.5 show the optimal final energy for varying times.

Table 2.2 Results related with the Optimal Low-Thrust trajectory

$a$	$b$	$\omega$	$\phi$	$J$
0.2003	-0.0318	0.3345	-0.1703	-0.1396

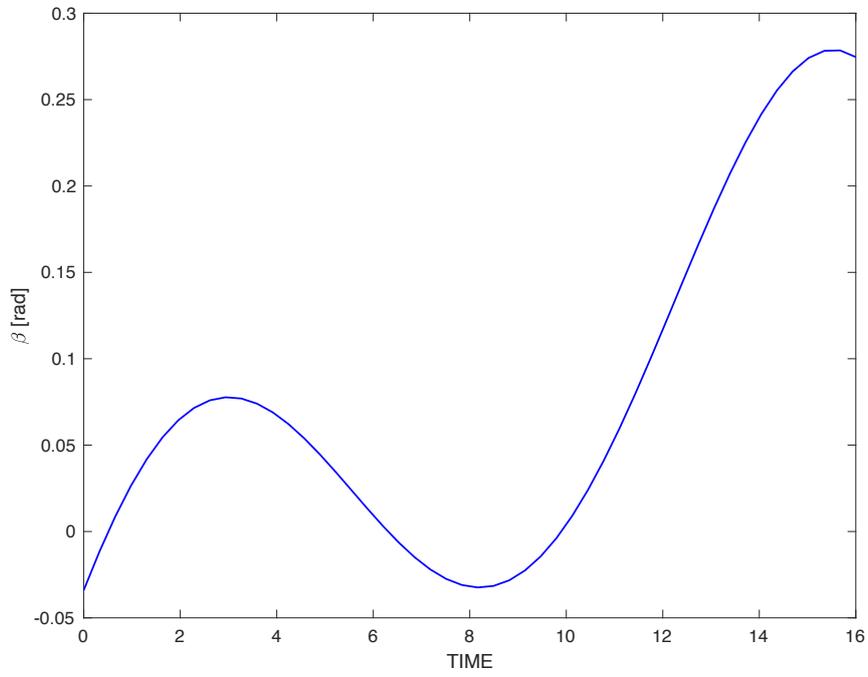


Figure 2.3 Optimal thrust pointing angle time history

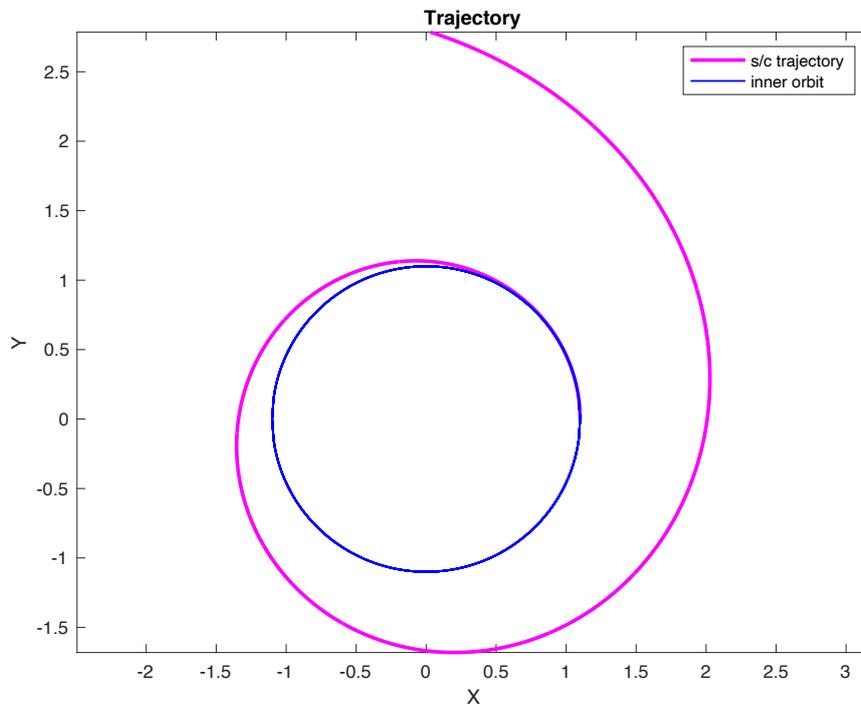


Figure 2.4 Optimal Low-Thrust spacecraft trajectory

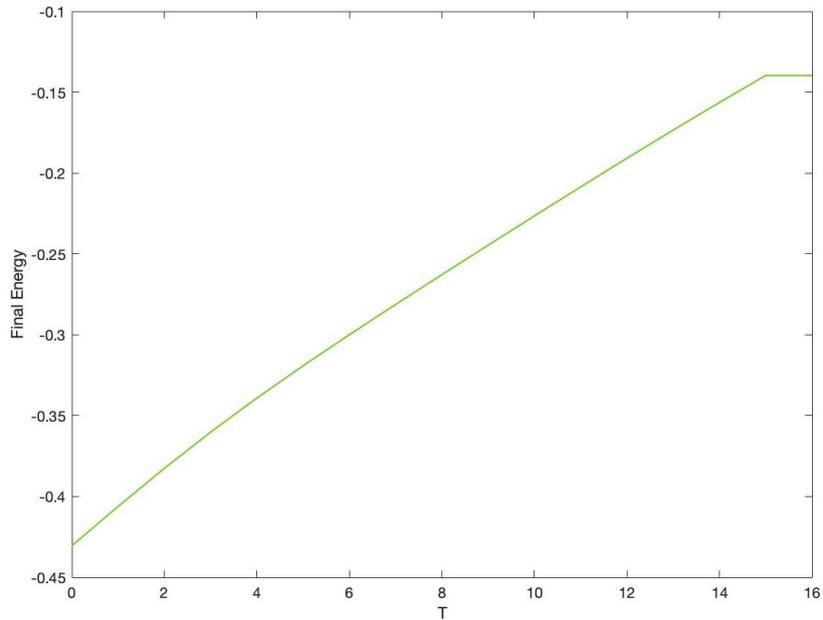


Figure 2.5 Optimal final energy for varying times

In the second case, however,  $\beta$  was parameterized using 17 points, which became the unknown parameters contained in each particle and optimized by the algorithm.

$$[x_1 \ x_2 \ \dots \ x_{16} \ x_{17}]$$

The limits of the search space are defined with the same values used previously.

The Figure 2.6 illustrates the optimal time history of  $\beta$  and the optimal final energy for varying times is shown in Figure 2.7.

In this case we achieve a final energy value  $J$  of -0.14, very close to that obtained in the previous case.

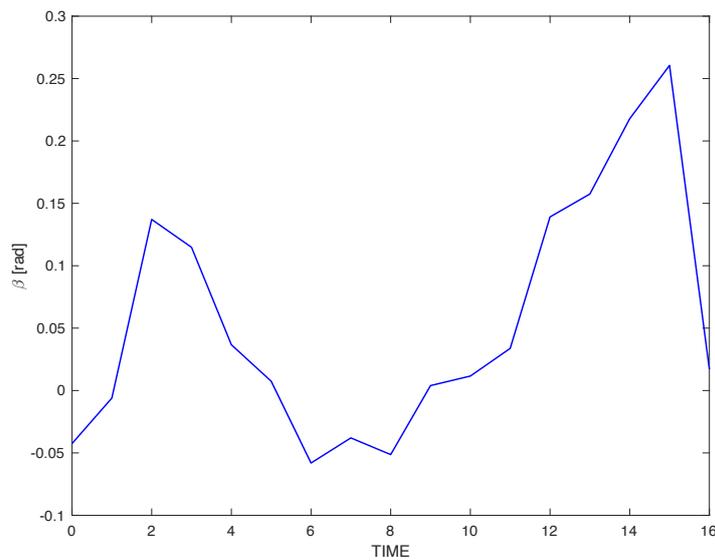


Figure 2.6 Optimal thrust pointing angle time history (Points)

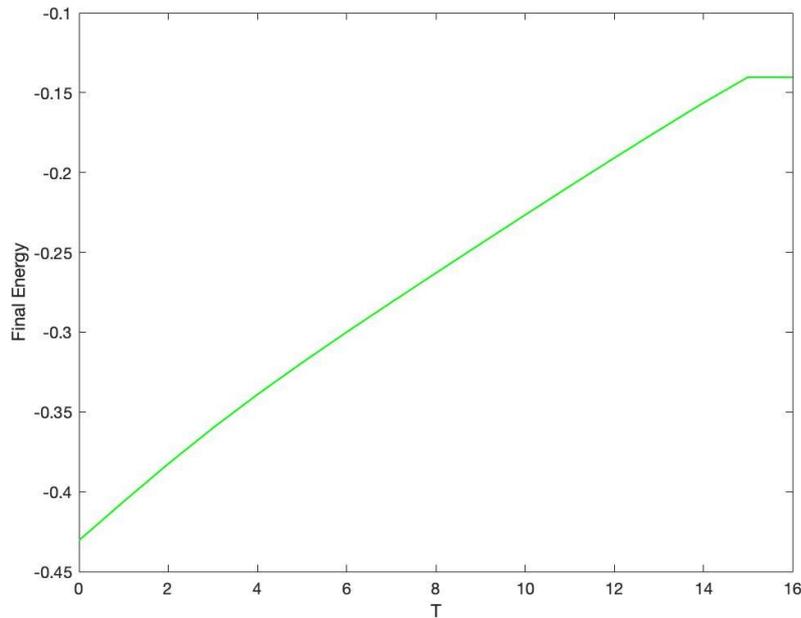


Figure 2.7 Optimal final energy for varying times (Points)

## 2.2 Optimization of a dynamic system using NLP problem solver

A spacecraft in flight is a dynamical system. As far as dynamical systems are concerned, it is relatively simple; the equations of motion are continuous and deterministic, for the unforced case they are essentially integrable, and perturbations, such as attractions of bodies other than the central body, are usually small. Difficulties arise when considering the complete problem, corresponding to a real space mission. For example, a complete interplanetary flight has complicated time-dependent boundary conditions, simple equations of motion but requiring coordinate transformations when the spacecraft transitions from planet-centered to heliocentric flight (and vice versa), and likely discrete changes in the states of the system when the rocket motor is turned on and the spacecraft suddenly changes velocity and mass [9].

If low-thrust electric propulsion is used, the system becomes even more complicated because there are no longer integrable arcs, and the decision variables, which were previously discrete quantities, now include continuous time histories.

Due to the fact that the cost of putting a spacecraft into orbit, which is usually the first step in any trajectory, is so enormous, it is especially important to optimize space trajectories so that a given mission can be accomplished with the lightest possible spacecraft and within the capabilities of existing (or accessible) launch vehicles.

Determining the necessary conditions for the optimization of a continuous, deterministic dynamical system of the type corresponding to a spacecraft in flight is also a straightforward problem, although the result, especially for the common case in which staging or impulsive  $\Delta V$ 's are used, is a sophisticated two-point-boundary-value-problem (TPBVP) with interior point constraints. As described in Chapter 1, the solution of this TPBVP, except for certain special cases, is very difficult.

As previously noted, many methods have been developed to solve the TPBVP numerically. These methods seek to reduce the optimal control problems to parameter optimization problems that can be solved by a nonlinear programming (NLP) problem solver and are solution methods in which all of the free parameters are adjusted contemporaneously. Methods of this type include the *finite-difference methods* and the *collocation* methods, the best known and most implemented direct transcription method. The nonlinear programming (NLP) problem solver is used to enforce the constraints while simultaneously minimizing the problem objective and they are capable of solving the large, sparse NLP problems resulting from the application of the collocation method to sophisticated problems [9].

Direct transcription schemes have a number of advantages over other numerical optimization methods. Since there are no costate variables, the problem size is reduced by a factor of two, and the problematic estimation of initial costates is avoided. Also, to a degree, one does not have to specify a priori the precise structure of the problem. In addition they present a much better robustness, that is showing the ability to converge to an optimal trajectory from poor initial guess, in comparison to other numerical optimization methods.

Nonlinear programs are expressed in the following general formulation:

$$\begin{aligned} & \text{Minimize } J(P) \\ & \text{Subject to } l \leq \begin{pmatrix} P \\ f(P) \\ A_L(P) \end{pmatrix} \leq u \end{aligned} \quad (2.13)$$

where  $l$  and  $u$  are constraint lower and upper bounds,  $J(P)$  is a smooth scalar function,  $A_L$  represents any linear constraints, and  $f(P)$  is the set of smooth nonlinear constraint functions representing the direct transcription constraints and the terminal constraints.

Once the NLP problem is clearly defined, it can be solved by using dense or sparse solvers. The first solver used in this work was *fmincon*.

*Fmincon* is a Nonlinear Programming solver provided in MATLAB's Optimization Toolbox and which performs nonlinear constrained optimization and supports linear and nonlinear constraints.

The NLP problem presented (2.13) is interpreted by *fmincon* in the following way

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub \end{cases} \quad (2.14)$$

where  $c(x)$  are the nonlinear inequalities and  $ceq(x)$  are the nonlinear equalities. *Fmincon* optimizes such that  $c(x) \leq 0$  and  $ceq(x) = 0$ .  $A$  is a real matrix that specifies linear inequality constraints, this matrix is an M-by-N matrix, where M is the number of inequalities, and N is the number of variables.  $Aeq$  is an Me-by-N matrix, where Me is the number of equalities, and N is the number of variables, and it specifies the linear equality constraints.  $lb$  and  $ub$  express the upper and lower bounds, respectively [7].

In code this look like:

$$[x, fval] = \text{fmincon}(\text{fun}, x0, A, b, Aeq, beq, lb, ub, \text{nonlcon}, \text{options}) \quad (2.15)$$

Where  $\text{fun}$  is the function to minimize, specified as a function handle or function name,  $x0$  is the initial point, specified as a real vector or real array, and  $\text{nonlcon}$  represents the nonlinear constraints, specified as a function handle or function name.

In the development of the various problems, the equations of motion describing the trajectory were integrated using various methods.

## 2.3 Integration Methods

As seen in the various examples previously given, the equations of motion have been integrated in a variety of ways. Below are the main characteristics of each of the methods used and the reasons for choosing the one used in the final project.

### 2.3.1 ODE

ODE is a set of functions provided by Matlab [7]. All MATLAB ODE solvers can solve systems of equations of the form  $y' = f(t, y)$ . The solvers all use similar syntaxes. The one used during the development of this project is ODE45, which is based on an explicit Runge-Kutta formula. It is a single-step solver – in computing  $y(t_n)$ , it needs only the solution at the immediately preceding time point,  $y(t_{n-1})$ .

To work it needs to have in input a function handle that defines the functions to be integrated, an interval of integration and initial conditions for each equation defined in the function handle.

ODE45 is not a fixed-step integration algorithm but it is an adaptive method. It is because of this that in the examples presented in this paper in the presence of this method of integration the vectors containing the time history of a given parameter had to be linearly interpolated in order to allow their use in the integration of the equations of motion.

Consider a coast arc in a spacecraft trajectory. If the initial states and the duration of the arc are known, the system ODEs can be integrated forward. There are then nonlinear constraints reflecting that the states resulting from the numerical integration are equal to the states that the NLP solver allocates to the end of the coast arc. These become equations in the *ceq* vector in equation (2.14). The situation is similar for a thrust arc except that some method needs to be provided to specify the control variables (as additional NLP parameters) during the thrust arc.

This integration method is quite accurate and good to use but too time consuming.

### 2.3.2 Euler Step

Euler-step integration method is a numerical method that is able to determine the solution of the differential equation. Using a numerical method we get an approximation of the solution, not the exact solution and the solution is calculated incrementally, step by step.

Euler integration method is one of the simplest integration method, named after the mathematician Leonhard Euler. The Euler method is a first-order method, which means that the local error (error per step) is proportional to the square of the step size, and the global error (error at a given time) is proportional to the step size [25].

The Euler integration method is also an explicit integration method, which means that the state of a system at a later time (next step) is calculated from the state of the system at the current time (current step).

$$y(t + \Delta t) = f(y(t)) \quad (2.16)$$

The Euler integration method is also called the polygonal integration method, because it approximates the solution of a differential equation with a series of connected lines (polygon).

To use this method it is necessary to define some parameters. The size of the integration interval  $[a, b]$  and the number of integration steps  $N$  define the integration step size  $h$ . The smaller the step size, the better the approximation, the smaller the integration error. It is possible to directly define the step size, which will further determine the number of integration steps.

The step size  $h$  is calculated as:

$$h = \frac{b - a}{N} \quad (2.17)$$

The initial conditions  $t_0, y_0$  represent the solution ( $y_0$ ) of the differential equation at a given time ( $t_0$ ). Usually  $t_0$  is equal with the start value of the integration interval  $a$ .

The iterative equation defining the method can be expressed as:

$$t_i = t_0 + h \cdot i \quad (2.18)$$

$$y_i = y_{i-1} + h \cdot f(t_{i-1}, y_{i-1}) \quad (2.19)$$

where  $t_i$  function argument and  $y_i$  is the function approximation.

In each step of the iteration, the Euler approximation calculate the end point of a line. The starting point  $A_0$  is known, it has the coordinates  $(t_0, y_0)$ . The point  $A_1$  is calculated based on the point  $A_0$  and the slope  $f(t, y)$ . The next points  $A_n$  are calculated based on the previous points  $A_{n-1}$  and the slope [25].

The evaluation of equation (2.19) over many steps and for each state variable supplies many nonlinear constraint equations into the *ceq* vector of equation (2.14), resulting in a much larger NLP problem than would be obtained if explicit numerical integration as in Sec 2.3.1 is used.

Euler's method is therefore simple and fast to use. However, the application of Euler's method proved to be too inaccurate to be used in the final project.

### 2.3.3 3-Step Runge-Kutta Parallel Shooting

The method of direct transcription with Runge-Kutta (RK) integration and parallel shooting is used to solve the trajectory optimization problems which are described in this work. The algorithm and study related to this method was developed by my colleague Alessia Speziale.

It is a direct method that also converts the optimization problem into a NLP problem.

In this method, the continuous optimal control problem is discretized into a series of segments  $[1,2,\dots,N]$  each of length  $h$ . The time history of each state and control value is specified at the  $(N + 1)$  node points between segments as shown in Figure 2.8 . There are state variables at each mesh points  $x_i \cong x(t_i)$ . [8]

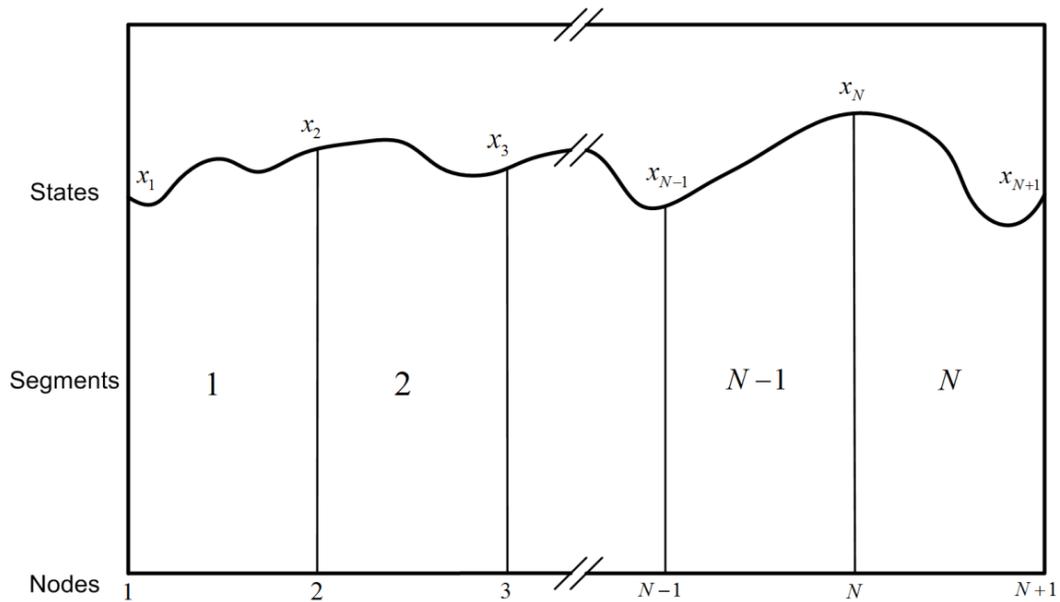


Figure 2.8 Problem structure for DTRK

In addition, the control variables are provided at the mesh points  $t_i$  and also at the centre points  $t_i + \frac{h}{2}$ , as shown in Figure 2.9.

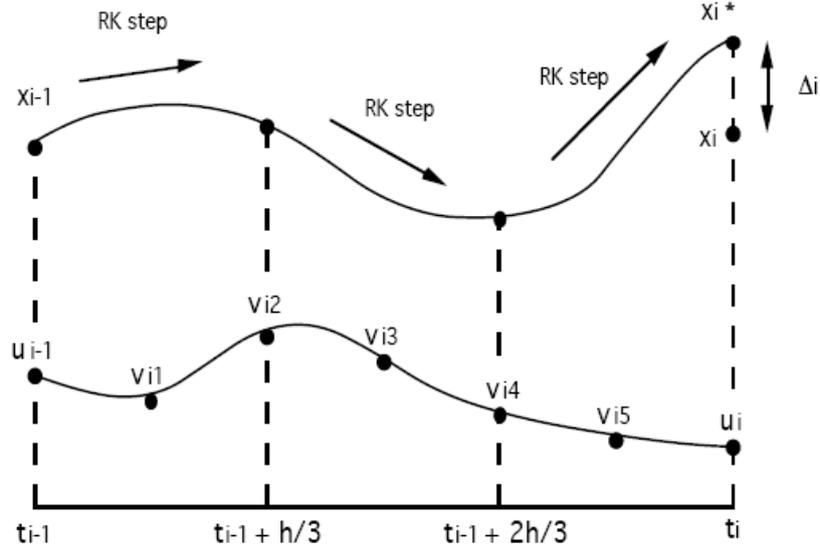


Figure 2.9 Illustration of the parallel-shooting method

From a given mesh point,  $t_{i-1}$ , the equations of motion are integrated forward from  $x_{i-1}$  to the next mesh point  $t_i$  using the control  $u_{i-1}$ ,  $v_i$ , and  $u_i$  by step of a four-stage Runge-Kutta formula:

$$y_i^1 = x_{i-1} + \frac{h}{2} f(x_{i-1}, u_{i-1}) \quad (2.20)$$

$$y_i^2 = x_{i-1} + \frac{h}{2} f(y_i^1, v_i) \quad (2.21)$$

$$y_i^3 = x_{i-1} + h f(y_i^2, v_i) \quad (2.22)$$

$$y_i^4 = x_{i-1} + \frac{h}{6} [f(x_{i-1}, u_{i-1}) + 2f(y_i^1, v_i) + 2f(y_i^2, v_i) + f(y_i^3, u_i)] \quad (2.23)$$

where  $f(x, u)$  refers to the system equations of motion,  $x$  are the state variable and  $u$  are the control variables; and  $y_i^4$  is an estimate of the state at the next mesh point, so for continuity we required that the “Runge-Kutta defects”

$$\Delta_i = y_i^4 - x_i \quad (2.24)$$

The set of the defect equations for all of the states over all of the time segments forms a vector of nonlinear constraints which are zero only if the equations of motion are satisfied. If they are satisfied, then the equations of motion are effectively integrated across the entire problem using the 4<sup>th</sup> order Runge-Kutta method.

The control variables  $u_i$  and  $v_{ik}$  are specified at the five interior points, but the state variables  $x$  are not. This significantly decreases the size of the problem, increasing solution speed without sacrificing accuracy [8].

System control variable parameters are specified much more frequently than system state variables. This is especially beneficial for problems, such as low-thrust trajectory optimization, where the control changes rapidly while the states, for example the orbit elements, change only slowly [9].

## 2.4 Examples

With the aim of becoming familiar with the use of this optimization algorithm, *fmincon* was applied to several examples, from the easiest to the most complex.

### ***Low-Thrust spacecraft orbit transfer with maximization of the final energy:***

The problem considered is the same as the one seen in the previous section describing the PSO method. Therefore, it is a continuous thrust, orbit transfer problem in which the main purpose is to determine the time history of the thrust pointing angle that maximizes the final energy in a fixed time of flight. Motion is confined to a single plane and the spacecraft's position is described using polar coordinates. The problem is solved by employing a normalized set of units.

The only control variable, the thrust angle  $\beta$ , is measured relative to the local horizontal and is parameterized using 100 points, which are the unknown parameters optimized by the optimizer to obtain the maximum value of the objective function

$$J = - \left( \frac{V^2}{2} - \frac{1}{R_f} \right)$$

A vector of 100 components set all to zero is given as the initial guess at *fmincon* for the points constituting  $\beta$ . The vectors that constitute the upper and lower bounds were set to a value of 0.5 and -0.5, respectively.

In the problem under consideration, in contrast to what was previously seen with PSO, the satisfaction of the equations of motion and constraints was obtained using two different methodologies.

In an initial case, as also done in the cases already seen using PSO, the equations were integrated using ODE45, which gave almost the same results as the cases already seen and a trend of  $\beta$  visible in Figure 2.10. The vector points were linearly interpolated for use in the integration.

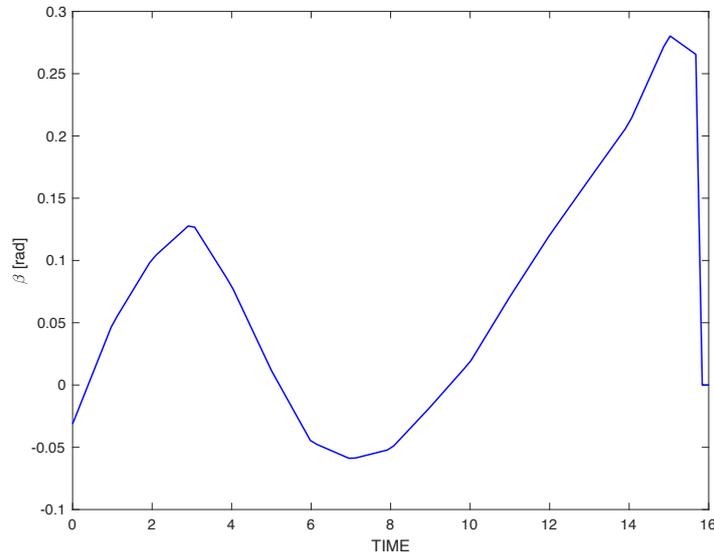


Figure 2.10 Optimal thrust pointing angle time history using *fmincon - ODE45*

In a second case, integration of the equations of motion and constraint satisfaction are obtained using the Euler-step method.

This leads to an increase in the optimized parameters because the optimizer has to work not only on the points that constitute  $\beta$ , as seen above, but also on the parameters that constitute the state vector, which are optimized in order to satisfy the imposed constraints step by step.

Thus, in addition to the initial guess of the angle, it is also necessary to impose an initial guess for the parameters that constitute the state vector, i.e.,

$$X = [r \quad \theta \quad v_r \quad v_\theta].$$

This was given by imposing typical trends for these elements, derived through knowledge of the theory.

The optimal time history of the thrust pointing angle  $\beta$  is shown in Figure 2.11, in which we can see a smoother curve, as compared to the one in Figure 2.10, thanks to the integration method used that does not require a linear interpolation of the vector.

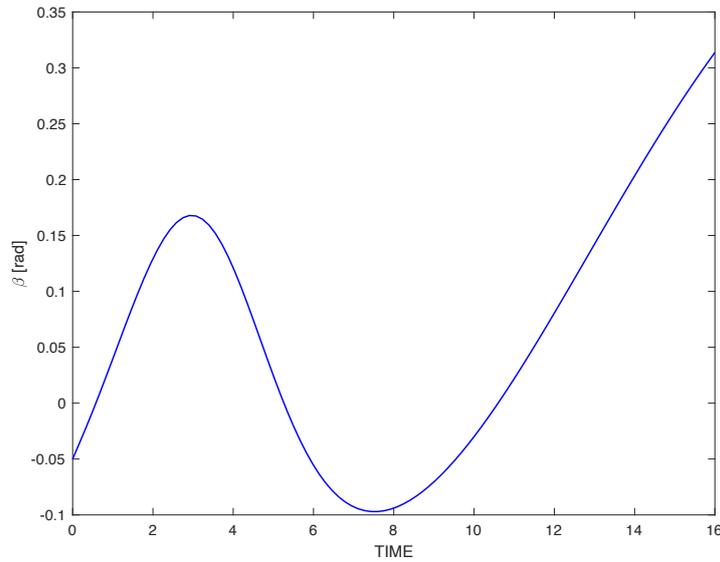


Figure 2.11 Optimal thrust pointing angle time history using *fmincon*  
- Euler Step

**2D Fictional Asteroid Interception:** The next problem began to move us toward what will be then the final project, this is where we started to approach asteroid interception.

The problem under investigation consists in the interception of a fictional asteroid using a low-thrust spacecraft. Motion is confined to a single plane and the spacecraft's position is described by employing a normalized set of units.

Cases have been solved where the equations of motion have been expressed either in cylindrical coordinates, as in all the examples seen so far, as well as in Cartesian coordinates, thus expressing the state vector as:

$$X = [x \quad y \quad v_x \quad v_y].$$

The equations governing the motion of the spacecraft are then expressed as:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{v}_x = -\frac{\mu}{r^3}x - A \sin\beta \\ \dot{v}_y = -\frac{\mu}{r^3}y + A \cos\beta \end{cases}$$

The goal of the problem is to intercept an asteroid using as little flight time as possible. The objective function to be minimized by the optimizer is then defined as:

$$J = T_f - T_i \quad (2.25)$$

In the parameters optimized by *fmincon* two times are added to the well known thrust pointing angle  $\beta$ , always constituted by 100 points, the start time of trajectory  $T_i$  and the end time  $T_f$ , which corresponds to the intercept time with the asteroid.  $\beta$  is chosen such that, starting from an initial circular orbit of radius  $r_0 = 1\text{AU}$ , it reaches the asteroid taking the shortest possible time.

An initial guess for these parameters, from a simple program done with PSO, is given to the optimizer and the lower and upper bounds are established.

The following orbital characteristics are given to the fictitious asteroid to be targeted:

- Semi-major axis,  $a$  : 2.5 AU;
- Periapsis,  $f_0$ :  $30^\circ$ ;
- Eccentricity,  $e$ : 0.2;
- Argument of Periapsis,  $\omega$  :  $90^\circ$ ;

The parameters are optimized to satisfy the equations of motion, integrated as seen above using both ODE45 or Euler step, and the imposed constraints. The nonlinear equalities  $ceq(x)$  that the optimizer must bring to zero consist of the intercept conditions, i.e., the difference between the position of the spacecraft and that of the asteroid both evaluated at  $T_f$ . In fact, for the encounter to occur both problem subjects must be in the same position in space at the final time,

$$ceq = \sqrt{(x_a - x_{s/c})^2 + (y_a - y_{s/c})^2} \quad (2.26)$$

where  $a$  indicates the coordinate concerning the asteroid while S/C indicates the coordinate concerning the spacecraft. If integration with Euler steps is used the  $ceq(x)$  will also include satisfaction of the equations of motion.

The position of the asteroid in time was derived by applying Newton's method to solve Kepler's problem,

$$M = (t - t_0)\sqrt{\frac{\mu}{a^3}} + [E_0 - e\sin(E_0)] \quad (2.27)$$

The state vector  $X$  is added to the optimization variables  $\beta, T_i, T_f$  in the case where the Euler step method is used, as seen in the previous examples.

In the images below we can see: in Figure 2.12 the optimized trajectory for this 2D case solved using Euler step, and in Figure 2.13 the trend of  $\beta$ . Table 2.3 instead shows the main values obtained from the optimization process.

Table 2.3 Results related with the Optimal fictional asteroid interception 2D case

$T_i$	$T_f$	$J$	Feasibility
4.6343	14.2743	9.64	2,174E-07

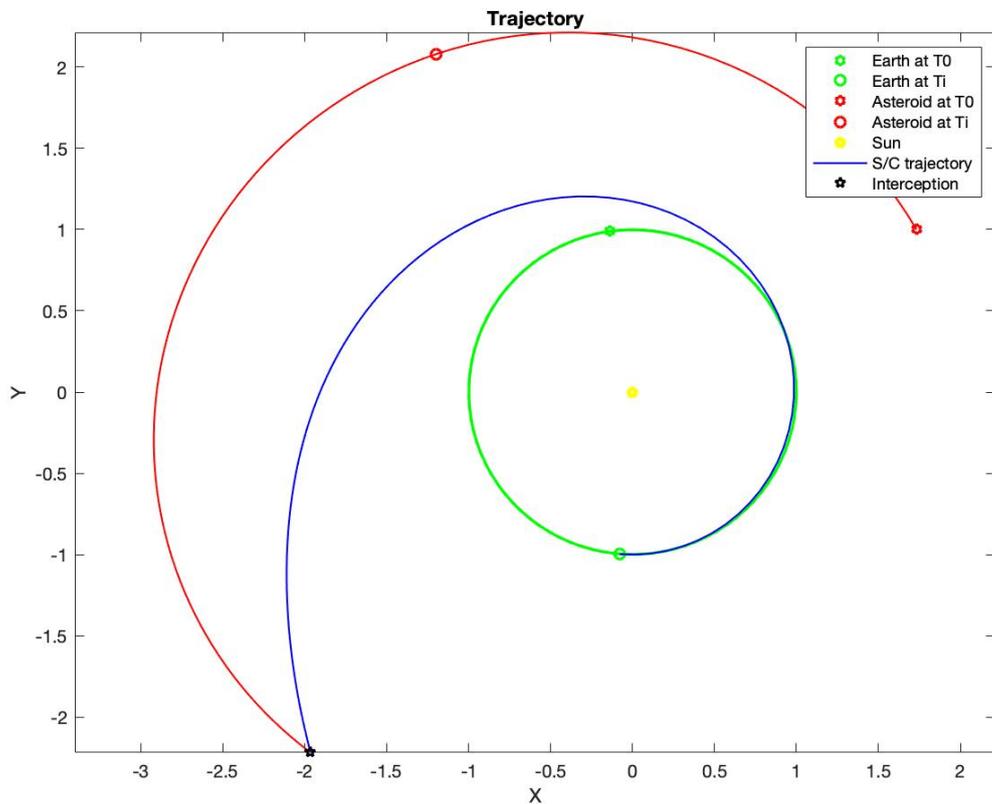


Figure 2.12 Optimal trajectory fictional asteroid interception 2D case

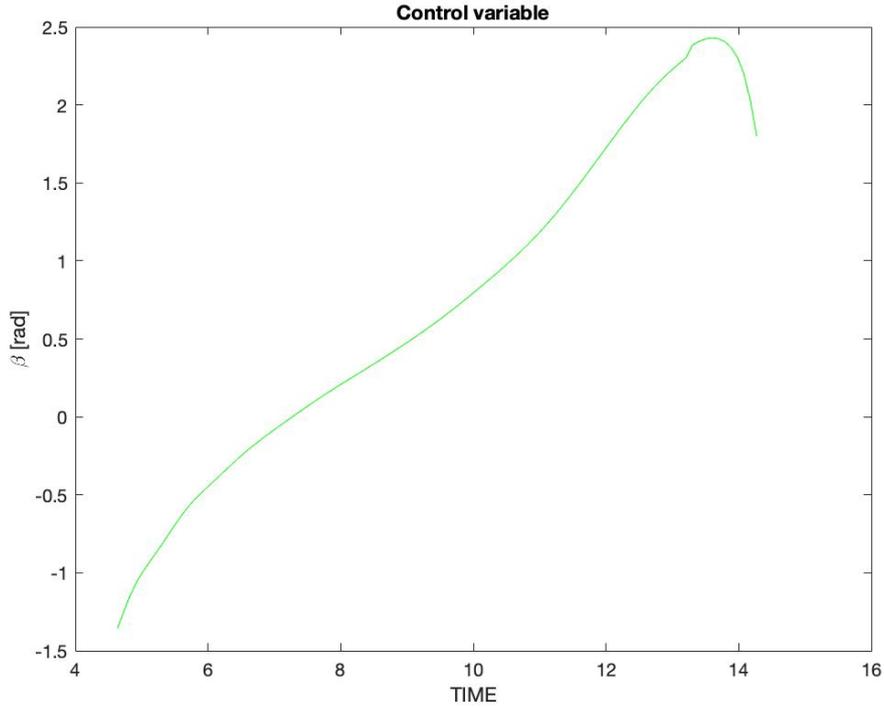


Figure 2.13 Optimal thrust pointing angle time history fictional asteroid interception 2D case

**3D Fictional Asteroid Interception:** From the asteroid interception problem confined to a single plane, we then moved on to address the same problem in three-dimensional space. For both cylindrical and Cartesian coordinate choices, the  $z$  component is then added for both position and velocity, the state vector thus increases from four to six components. To the orbital characteristics of the asteroid already listed is then added an orbital inclination  $i$  equal to  $10^\circ$ .

The problem is solved by employing a normalized set of units.

The goal of the problem is to intercept an asteroid using as little flight time as possible.

The objective function to be minimized by the optimizer is then defined as:

$$J = T_f - T_i$$

The out of plane thrust pointing angle,  $\gamma$ , still composed by 100 points, is added to the parameters optimized by *fmincon* in the 2D case, which were the in-plane thrust pointing angle  $\beta$ , the initial time  $T_i$  and the final time  $T_f$ .

Once again, the constraints imposed on *fmincon* consist of satisfying the impact condition,

$$ceq = \sqrt{(x_a - x_{s/c})^2 + (y_a - y_{s/c})^2 + (z_a - z_{s/c})^2} \quad (2.28)$$

The position of the spacecraft is obtained by integrating the equations of motion, while the one of the asteroid by solving Kepler's problem.

In addition to including the  $z$  component for position and the  $v_z$  component for velocity in the state vector, this example differs from the 2D case in a number of refinements made to make it more complete and similar to reality. In fact, the mass equation,

$$\dot{m} = -\frac{T_{max}}{c} \quad (2.29)$$

where  $T_{max}$  is the maximum value of the thrust magnitude and  $c$  is the exhaust velocity; the  $\Delta v$  of the upper stage launch vehicle, along with the two relative angles, the initial impulse in-plane pointing angle and the initial impulse out-of-plane pointing angle, which values are chosen by the optimizer within the feasible range; and the Earth's position and velocity from SPICE have been added to the original program.

## 2.5 Nonlinear Programming with SNOPT

The problems defined as above become nonlinear programming (NLP) problems. The NLP parameters can be arranged as a single vector  $P$  that collects the entire time history of state and control variables, along with additional parameters for the launch date, the time of flight, and the magnitude and direction of the initial impulse in the heliocentric reference frame.

Once a trajectory optimization problem has been clearly defined, it must then be solved by nonlinear constrained optimization solver. Two of the most popular algorithms are *fmincon*, which has been analyzed in section 2.2, and SNOPT.

It seems that *fmincon* solves the multiple shooting problem by first finding a feasible solution, and then attempting to optimize it. This means that *fmincon* can handle a worse initial guess than SNOPT, but it is a little worse at finding the true optimal solution, because it does not allow for much flexibility in the constraints, and it is limited to modest size problems [10].

Therefore, a more capable NLP solver is needed.

The sparse solvers SNOPT can take advantage of the sparsity present in the constraint Jacobian. [9]

SNOPT (Sparse Nonlinear OPTimizer) is a software package for solving large-scale optimization problems. It minimizes a linear or nonlinear function subject to bounds on the variables and sparse linear or nonlinear constraints. It is suitable for large-scale linear and quadratic programming and for linearly constrained optimization, as well as for general nonlinear programs of the form presented in equation (2.13) [11].

SNOPT requires that the user supply an initial guess of the solution in the form of an NLP parameter vector, a vector of upper and lower bounds for each NLP parameter, a specifications file containing the feasibility and optimality tolerances, and routines to calculate the vector of nonlinear constraints and the value of the objective function.

This is a software developed by Systems Optimization Laboratory in Department of Management Science and Engineering at Stanford University. The source code is re-entrant and suitable for any machine with a Fortran compiler. SNOPT may be called from a driver program in Fortran, Matlab, or C/C++.

In this work the Matlab interface is used. This particular interface is called *snsolve.m* and matches the call sequence of Matlab's *fmincon* function, shown in equation (2.15), as can be seen in the code below:

$$[x, fval] = snsolve(obj, x0, A, b, Aeq, beq, xlow, xupp, nonlcon, options) \quad (2.30)$$

### 2.5.1 Example

With the purpose of getting familiar with the use of this optimization algorithm, it was applied to some of the examples seen so far. The results were then compared to verify the correct behavior before using it in the final version of the project.

#### ***Hohmann Transfer:***

Table 2.4 Results related with the optimal Hohmann Transfer-SNOPT

Impulse	$\Delta v$ (AU/TU)	$\gamma$ (deg)	$T$ (TU)	$J$
1	0.2247	0	0	0.3938
2	0.1691	0	8.8858	

As we can see in Table 2.1, the results are in agreement with what was obtained through PSO. But in this case the optimization ended with a better feasibility than that obtained with PSO,  $1e-10$  versus the previous  $1e-7$ .

***Low-Thrust spacecraft orbit transfer with maximization of the final energy:***

Table 2.5 Results related with the Optimal Low-Thrust trajectory-SNOPT

$a$	$b$	$\omega$	$\phi$	$J$
0.2104	-0.0326	0.3349	-0.1639	-0.1395

As we can see in Table 2.2, the results are in agreement with what was obtained through PSO.

***2D Fictional Asteroid Interception:*** To better understand the use of SNOPT and before using it in the complete project I tried substituting it for *fmincon* in the problem of intercepting a fictitious asteroid in 2D space while using Cylindrical coordinates and Euler step method.

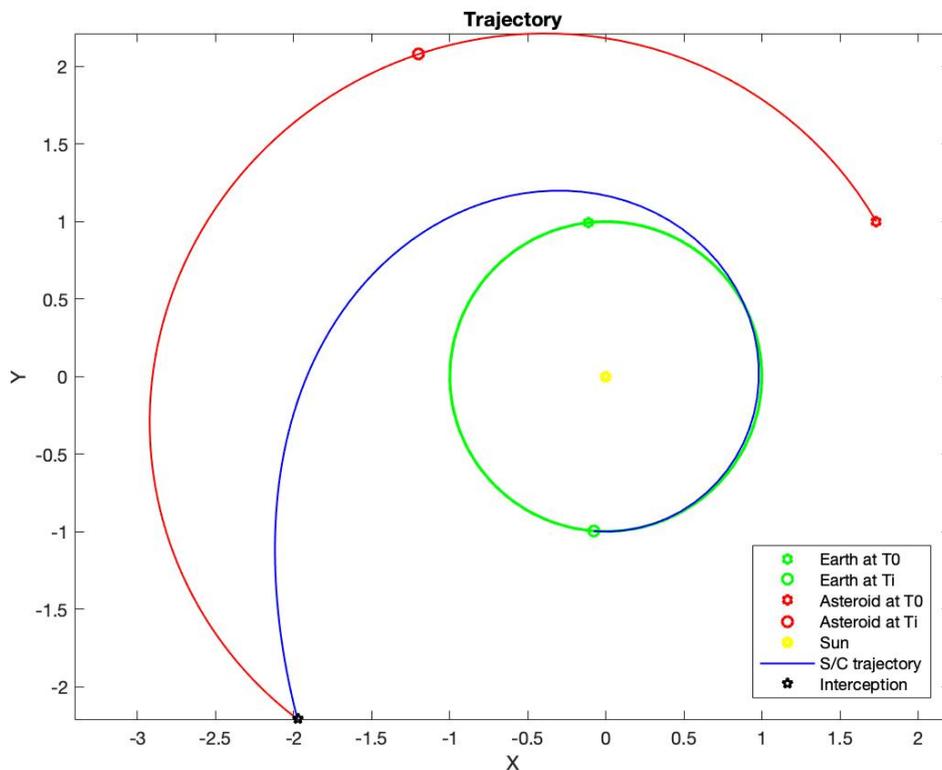


Figure 2.14 SNOPT Optimal trajectory fictional asteroid interception 2D case

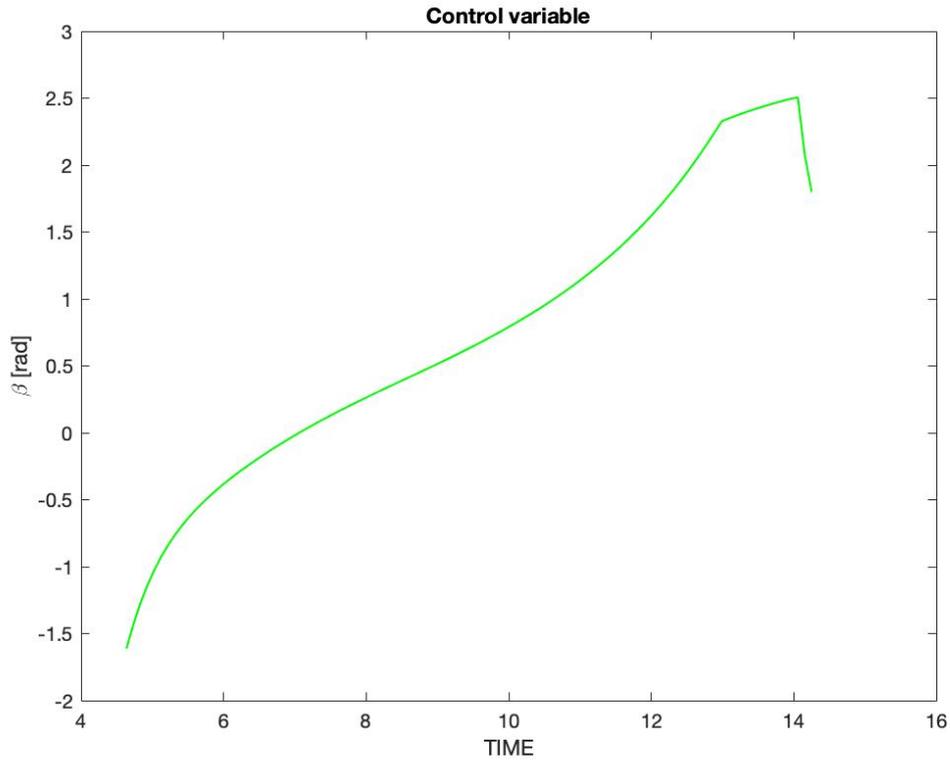


Figure 2.15 SNOPT Optimal thrust pointing angle time history fictional asteroid interception 2D case

Table 2.6 Results related with the Optimal fictional asteroid interception 2D case SNOPT

$T_i$	$T_f$	$J$	Feasibility
4.6358	14.2509	9.6151	1.4 E-06

As we can see from the comparison of the figures and the table above with those reported in the solution with *fmincon* (Figure 2.12, Figure 2.13 and Table 2.3) the results are almost equivalent.

# Chapter 3: Asteroid Interception

## 3.1 Mission Description

Impacts of near-Earth objects (NEOs) onto our planet are natural events where the effects of each single impact mainly depend on NEO size, structure, relative velocity, and impact location. To determine if a newly discovered object might impact on Earth one day, the object's orbit has to be numerically computed into the future. NEOs larger than 150 m in diameter and approaching Earth's orbit closer than 7.5 million *km* (0.05 AU) are called Potentially Hazardous Objects (PHOs). Due to their susceptibility to small orbit disturbances on short timescales they are candidates for future collisions with Earth. In general the NEO impact risk is a "high-consequence – low-probability" level risk. Therefore, it is categorized as a "moderate" level risk which requires certain precautionary actions. The benefits of deflection missions can be defined simply as the costs of the damages that would occur in the event of a NEO impact [12].

Generally, mitigating the impact hazard could be either done by deflecting the threatening object from its collision course or by destroying the object itself.

In case of NEO deflection one can further distinguish between the long-term application of a continuous small thrust, and the sudden application of a large impulsive thrust. In the first case, the NEO would be propelled for a period of several months. Ideally, the force would be applied through the object's center of mass parallel or antiparallel to its velocity vector, which would increase or decrease the semi-major axis of the NEO's orbit, respectively. Thus, its arrival at the intersection point with Earth's orbit would be delayed or advanced, respectively. But, this kind of orbit alteration can be applied only if sufficient warning time is given. If less warning time is available, the orbit has to be changed rapidly. The correction force should be applied via a high-energy interaction in an almost perpendicular direction with respect to the NEO velocity vector.

Destroying the NEO might not guarantee that the resulting fragments will be very small and therefore harmless [12].

Currently, various ideas for the deflection and disruption of hazardous asteroids and comets are under consideration.

Among them are systems that might be technologically feasible at present, such as chemical rocket engines, kinetic energy impacts, and nuclear explosives. Others are currently under development and might be available with some development effort in the near future, such as solar concentrators. But there are also systems that seem to be too far off to be realized in the required dimensions for the task of NEO deflection within the next decades, such as solar sails, laser systems, and the utilization of the Yarkovsky effect. Besides, there are also futuristic technologies such as eaters and the use of antimatter [12].

The technique chosen in this work is the kinetic impactor, which is currently the simplest and most technologically mature method available to defend against asteroids. In this method, a spacecraft is launched and simply slams itself into the asteroid. If this is done far enough in advance then it will only be necessary to change the speed by a millimeter a second or less and then it will miss the Earth entirely.

Low-thrust, high specific impulse propulsion is used because of the significant advantages it provides in propulsive mass required for a given mission. However, a low-thrust departure from Earth would require many revolutions of the Earth, which would consume a lot of time. It seems reasonable then to use an impulsive velocity change for the initial departure, followed by continuous low-thrust propulsion. [13]

### **3.1.1 Prior works**

The problem of asteroid deflection is an issue that has been widely discussed in the past. The first technical approach to define a NEO mitigation system was the students' system project "Project Icarus" at the Massachusetts Institute of Technology (MIT) in 1967. They recommended the use of six nuclear explosive devices each delivered to Icarus by a Saturn V launcher. The first conferences on the problem of asteroid and comet impacts on Earth took place in the early 1980s ("Snowmass conferences") and a total of 288 scientific papers and conference abstracts on NEO mitigation were published between 1967 and 2000, together with an average of about ten publications per year on NEO mitigation strategies and systems [12].

The work presented in this paper is based primarily on the studies done by Professor Bruce A Conway, such as the "Optimal Low-Thrust Interception of Earth-Crossing Asteroids" paper [13], in which he presents deflection work similar to that of this project except that he researches optimal trajectories aimed at minimizing time of flight, or the "Near-Optimal Deflection of Earth-Approaching Asteroids" paper [14], in which

he presents the use of STM for the calculation of deflection, and the thesis work “Optimal strategies for deflecting hazardous Near-Earth Objects via kinetic impactor” done by Jacob Englander during his master's degree studies in Aerospace Engineering at the University of Illinois Urbana-Champaign [15].

### 3.1.2 The Asteroid

The example of Potential Hazardous Asteroid (PHA) that was chosen for the implementation of this work is 99942 Apophis. It had been discovered on June 19<sup>th</sup>, 2004 by Roy Tucker, David Tholen and Fabrizio Bernardi. Apophis has a mass of approximately  $2.699 \times 10^{10}$  kg, it has an elongated shape with a mean diameter of 340 m, and passes close to the Earth every seven years [16].

After its discovery, asteroid 99942 Apophis had been identified as one of the most hazardous asteroids that could impact Earth, in fact the first observations showed that it had a 1 in 38 chance of impacting our planet on April 13<sup>th</sup>, 2029, the highest impact probability ever recorded.

That assessment of the impact changed when astronomers tracked Apophis and its orbit was better determined. The risk of an impact in 2029 was later ruled out, as was the potential impact risk posed by another close approach in 2036. Until March, 2021, however, a small chance of impact in 2068 still remained. When Apophis made a distant flyby of Earth around March 5<sup>th</sup>, 2021, astronomers took the opportunity to use powerful radar observations to refine the estimate of its orbit around the Sun with extreme precision, enabling them to confidently rule out any impact risk in 2068 and long after. So, the results from a new radar observation campaign combined with precise orbit analysis have helped astronomers conclude that there is no risk of Apophis impacting our planet for at least a century [17].

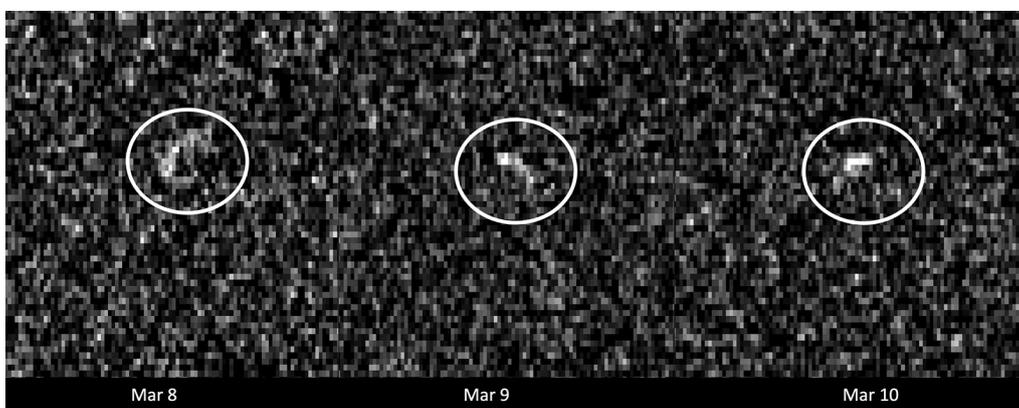


Figure 3.1 - Asteroid 99942 Apophis – radar observations March 8–10, 2021 [Credits: NASA/JPL-Caltech and NSF/AUI/GBO]

Apophis is now expected to pass less than 32000 kilometers from our planet’s surface – closer than the distance of geosynchronous satellites [18]. During that 2029 close approach, Apophis will be visible to observers on the ground in the Eastern Hemisphere without the aid of a telescope or binocular.

Although Apophis is no longer a likely threat to the Earth, it is a valuable example of a potentially hazardous asteroid and is worthy of study. This work considers a hypothetical asteroid based on Apophis, but which impacts the Earth on April 13<sup>th</sup>, 2029 instead of merely passing close by.

Apophis’s orbit elements for the date of close approach to the Earth are given in Table 3.1 [19].

*Table 3.1 Apophis Orbit Elements at Epoch April 13, 2029*

Symbol	Description	Value
$a$	semi-major axis	0.9227 AU
$e$	eccentricity	0.1914
$i$	inclination	3.34°
$\omega$	argument of periapse	126.60°
$\Omega$	longitude of ascending node	203.96°
$f$	true anomaly	232.58°



The spacecraft then reaches the asteroid and the impact occurs at the time identified as  $T_I$ . In order for this to happen, the fulfillment of the intercept condition is imposed. Both bodies, either spacecraft and asteroid, must be at time  $T_I$  at the same point in space. So this is like imposing that  $\vec{r}_{sc}(T_I) = \vec{r}_{\otimes}(T_I)$ , where  $\vec{r}_{\otimes}(T_I)$  is the position of the asteroid. The final velocity at interception is used to determine the impulsive perturbation on the asteroid, using conservation of momentum and assuming an inelastic collision,  $\delta \vec{v}_I$ . This work assumes that the impact of the spacecraft against the asteroid is a totally inelastic collision. In reality, the impact may knock off pieces of the asteroid and send them flying in the opposite direction, causing an increase in the momentum imparted to the asteroid.

Once the impact has occurred we move on to the second part of the simulation represented by the motion of the asteroid toward the Earth, colored orange in the figure. The motion of the asteroid is propagated to the boundary with the Earth's sphere of influence (SOI). Once it arrives at the Earth's sphere of influence, the position and velocity of the asteroid is used together with the impulse received from the impact and through the use of the state transition matrix (STM) the deflection, i.e. the displacement from the point at which the asteroid would have entered the sphere of influence absent the kinetic impact, is calculated.

After entering the Earth's sphere of influence we have a change of coordinates, from heliocentric to geocentric, and the motion of the asteroid is propagated further on a hyperbolic flyby trajectory, until finding the point of closest approach to the Earth. This distance will constitute the objective function  $J$  to be maximized in the project.

## 3.2 Governing Equations

The choice of a coordinate system on which to express the trajectory optimization problem is fundamental and strongly influences the solution process. The main choice is whether to use coordinates, for example Cartesian or polar coordinates [9].

### 3.2.1 Equation of Motion in Cylindrical Coordinates

Using this coordinate system has advantages when combined with the use of an NLP solver. In this type of coordinate system the position and velocity state variables do not change rapidly. The radius will always be positive and change only slowly.

Angular position coordinates also change either slowly or rapidly and predictably. In case of no retrograde motion, the angular velocities are also generally one sign and do not change rapidly. The system control parameters are thus in a form that improves or maintains the robustness of the solution using NLP [9].

In the cylindrical coordinates system, a point in space is represented by the ordered triple  $(r, \theta, z)$ , where  $(r, \theta)$  are the polar coordinates of the point's projection in the  $xy$ -plane and  $z$  is the usual  $z$ -coordinate in the Cartesian coordinate system, as we can see in Figure 3.3. The  $x - y$  plane is the ecliptic plane, and  $x$  points towards the first point in Aries.

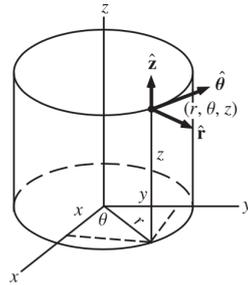


Figure 3.3 - Cylindrical Coordinate system representation [24]

The motion of the spacecraft is governed, in the 3D space, by:

$$\begin{cases} \dot{r} = v_r \\ \dot{\theta} = \frac{v_\theta}{r} \\ \dot{z} = v_z \\ \dot{v}_r = \frac{v_\theta^2}{r} - \frac{r}{(r^2 + z^2)^{\frac{3}{2}}} + A \sin \beta \cos \gamma \\ \dot{v}_\theta = -\frac{v_r v_\theta}{r} + A \cos \beta \cos \gamma \\ \dot{v}_z = -\frac{z}{(r^2 + z^2)^{\frac{3}{2}}} + A \sin \gamma \end{cases} \quad (3.1)$$

where  $\beta$  is the in-plane thrust pointing angle,  $\gamma$  represents the out-of-plane thrust pointing angle and  $A$  is the magnitude of the thrust acceleration.

### 3.2.2 Equation of Motion in Cartesian Coordinates

Using Cartesian coordinates is the simplest choice. It is a particularly natural choice for the three-body-problem as the Cartesian form of the system equations, with the system origin at the center of mass of the two primary bodies, is the most well known. One downside of using this type of coordinate system is that all the position and velocity state variables change rapidly, which can cause problems in the efficiency and robustness of the NLP solver [9]. Despite this, we decided to use this type of coordinates in the final project in order to have a correspondence with the different tools used, such as SPICE, which provide results in Cartesian coordinates.

Cartesian coordinates are therefore used to model the motion of the spacecraft in three dimensions:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = -\frac{\mu}{r^3}x + \frac{T_x}{m} + a_x \\ \dot{v}_y = -\frac{\mu}{r^3}y + \frac{T_y}{m} + a_y \\ \dot{v}_z = -\frac{\mu}{r^3}z + \frac{T_z}{m} + a_z \end{cases} \quad (3.2)$$

where  $\mu$  is the standard gravitational parameter of the central body,  $m = -\frac{T}{I_{sp}g_0}$  the mass, with  $T = \sqrt{T_x^2 + T_y^2 + T_z^2}$ , and  $r = \sqrt{x^2 + y^2 + z^2}$  the radius magnitude.

$a_i$  ( with  $i = x, y, z$  ) represent the sum of all the disturbing acceleration for each planetary body. For our analysis we took into account the effects of disturbances from Venus, Earth-Moon, Mars, and Jupiter. The perturbing acceleration vector is formulated as follows:

$$\vec{a} = -\mu \left[ \frac{\vec{y} - \vec{r}}{|\vec{y} - \vec{r}|^3} + \frac{\vec{r}}{r^3} \right] \quad (3.3)$$

where  $\mu$  is the standard gravitational parameter of the third body under consideration,  $\vec{y}$  is the spacecraft position vector and  $\vec{r}$  represents planetary position vector of the third body found via SPICE.

It is then necessary to add the mass variation equation to the system equation (3.2):

$$\dot{m} = -\frac{T_{max}}{c} \quad (3.4)$$

where  $T_{max}$  is the maximum thrust magnitude and  $c$  is the effective exhaust velocity.

### 3.2.3 Units

In the final project all calculations in the heliocentric reference frame, including the motion of the spacecraft, are performed in normalized units such that  $\mu_{\odot} = 1$ , and an object in a circular orbit of radius 1 astronomical unit from the sun has an orbital period of  $2\pi$  time units. The asteroid-Earth encounter and the computation of the state transition matrix are computed in non-normalized units (kilometers, seconds).

## 3.3 Objective Function

The goal of this mission is to apply an impulse to the target asteroid such that  $r_{miss}$ , the distance of closest approach between the asteroid and the Earth, is maximized. This distance was translated into the project code as the difference between the asteroid's actual closest approach distance in 2029  $r_{miss0}$ , about 32,000 km, and the closest approach distance obtained with the deflection given by the impact,  $r_{miss}$ . This results in the formation of the objective function as follows:

$$J = r_{miss0} - r_{miss}$$

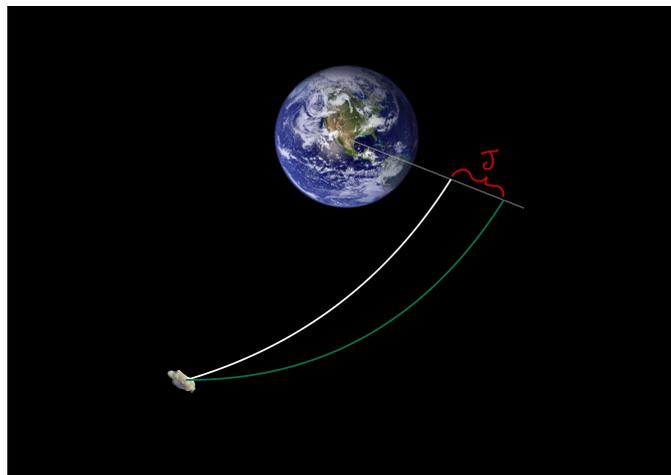


Figure 3.4 - Schematic representation of the Objective Function

As can be seen in the figure above, the white line represents the trajectory of the undeflected asteroid, while the green line represents the deviated trajectory.  $J$  is the difference of the two distances calculated at the closest approach point.

Maximizing this amount then allows the asteroid to move away from Earth.

$r_{miss_0}$  is calculated by propagating the motion of the asteroid within the Earth's SOI in geocentric coordinates and finding the minimum radius in its motion.  $r_{miss}$  is found in the same way except that the propagation in this case takes deflection into account.

### 3.3.1 State Transition Matrix

The impulse is provided by a spacecraft that is launched from Earth at an initial time  $t_0$  and impacts the asteroid at the time of interception,  $t_i$ . At any time  $t$  after  $t_i$ , the system state transition matrix  $\Phi(t, t_i)$  determines the perturbation in position ( $\delta \vec{r}$ ) and velocity ( $\delta \vec{v}$ ) of the asteroid as a result of the initial perturbing impulse [14].

At any time  $t$  later than  $t_0$ , the position and velocity vectors will be a function, not only of time, but also of the position and velocity that the vehicle had at the earlier time  $t_0$ .

Thus, we may expand  $r[t, r(t_0), v(t_0)]$  in a Taylor series about the reference quantities to obtain, in a vector-matrix notation [20] :

$$\begin{bmatrix} \delta \vec{r}_{\otimes} \\ \delta \vec{v}_{\otimes} \end{bmatrix} = \Phi(t, t_i) \begin{bmatrix} \delta \vec{r}_0 \\ \delta \vec{v}_0 \end{bmatrix} = \begin{bmatrix} \tilde{R} & R \\ \tilde{V} & V \end{bmatrix} \begin{bmatrix} \delta \vec{r}_0 \\ \delta \vec{v}_0 \end{bmatrix} \quad (3.5)$$

Where, since the change in position at impact is considered negligible, and therefore  $\delta \vec{r}_0 = 0$ ,

$$\delta \vec{r}_{\otimes}(t) = [R] \delta \vec{v}_0(t_i) \quad (3.6)$$

$$\delta \vec{v}_{\otimes}(t) = [V] \delta \vec{v}_0(t_i) \quad (3.7)$$

with

$$[R] = \frac{r_0}{\mu_{\odot}}(1 - F)[(\vec{r} - \vec{r}_0)\vec{v}_0^T - (\vec{v} - \vec{v}_0)\vec{r}_0^T] + \frac{C}{\mu_{\odot}}\vec{v}\vec{v}_0^T + G[I_3] \quad (3.8)$$

$$[V] = \frac{r_0}{\mu_{\odot}}\|\vec{v} - \vec{v}_0\|^2 + \frac{1}{r^3}[r_0(1 - F)(\vec{r}\vec{r}_0^T) - C\vec{r}\vec{v}_0^T] + G_t[I_3] \quad (3.9)$$

$$F = 1 - \frac{u_2}{r_0} \quad (3.10)$$

$$G = t - t_0 - \frac{u_3}{\sqrt{\mu_\odot}} \quad (3.11)$$

$$G_t = 1 - \frac{u_2}{r} \quad (3.12)$$

$$C = \frac{1}{\sqrt{\mu_\odot}}(3u_5 - \chi u_4) - (t - t_0)u_2 \quad (3.13)$$

$$\chi = \sqrt{a}(E - E_0) \quad (3.14)$$

$$\alpha = \frac{1}{a} \quad (3.15)$$

$a$  is its semi-major axis,  $\mu_\odot$  is the gravitational parameter of the Sun,  $E$  is the eccentric anomaly found by solving Kepler's problem, and  $u_k(\chi, \alpha)$  are the universal functions described by Battin [20]:

$$u = \frac{U_1(\frac{1}{2}\chi; \alpha)}{U_0(\frac{1}{2}\chi; \alpha)} = \frac{\frac{1}{2}\chi}{1 - \frac{\alpha(\frac{1}{2}\chi)^2}{3 - \frac{\alpha(\frac{1}{2}\chi)^2}{5 - \frac{\alpha(\frac{1}{2}\chi)^2}{7 - \frac{\alpha(\frac{1}{2}\chi)^2}{9}}}} \quad (3.16)$$

$$u_1(\chi; \alpha) = \frac{2u}{1 + \alpha u^2} \quad (3.17)$$

$$u_2(\chi; \alpha) = \frac{2u^2}{1 + \alpha u^2} \quad (3.18)$$

$$q = \frac{\alpha u^2}{1 + \alpha u^2} \quad (3.19)$$

$$u_3(2\chi) = \frac{\frac{4}{3}u_1^3}{1 - \frac{\frac{6}{5}q}{1 + \frac{2}{35}q}} \quad (3.20)$$

$$u_3(\chi) = \frac{1}{2}u_3(2\chi) - u_1(\chi)u_2(\chi) \quad (3.21)$$

$$u_4(\chi) = u_1(\chi)u_3(\chi) - \frac{1}{2}[u_2^2(\chi) - \alpha u_3^2(\chi)] \quad (3.22)$$

$$u_5(2\chi) = \frac{4}{3}[u_1^2(\chi)u_3(\chi) + u_1^3(\chi)u_2(\chi) + \chi u_2^2(\chi)] + \frac{8}{3}\chi u_4(\chi) - \frac{\frac{16}{15}u_1^5(\chi)}{1 - \frac{\frac{10}{7}q}{1 + \frac{2}{21}q}} \quad (3.23)$$

$$u_5(\chi) = \frac{1}{2}u_5(2\chi) - u_1(\chi)u_4(\chi) - \frac{1}{2}\chi^2 u_3(\chi) \quad (3.24)$$

The value of  $\delta \vec{v}_0$  is derived from the impact, using conservation of momentum and assuming an inelastic collision,

$$\delta \vec{v}_0 = \frac{m_{sc}(t_i)(\vec{v}_{sc}(t_i) - \vec{v}_{\otimes}(t_i))}{m_{\otimes} + m_{sc}(t_i)} \quad (3.25)$$

Consequently, for any initial pulse  $\delta \vec{v}_0$  and time lapse  $t_f - t_i$ , we can find

$$\begin{bmatrix} \vec{r}(t_f) \\ \vec{v}(t_f) \end{bmatrix} = \Phi(t_f, t_i) \begin{bmatrix} 0 \\ \delta \vec{v}_0 \end{bmatrix} + \begin{bmatrix} \vec{r}_{reference}(t_f) \\ \vec{v}_{reference}(t_f) \end{bmatrix} \quad (3.26)$$

with a semi-analytical method that requires only a solution of Kepler's equation. This method determines the position and velocity of the asteroid at any future time  $t_f$  as a result of the sun's gravity alone. Note that equation (3.26) is time dependent, indicating that the earlier the perturbing impulse  $\delta \vec{v}_0$  is applied, the farther the asteroid will be moved from its reference position. In order to accurately model the asteroid's approach to Earth, however, Earth's gravity must also be considered.

The choice of applying this method to calculate the deflection in the project is dictated by the fact that we can use SPICE to get the precise values of  $\vec{r}$  and  $\vec{v}$  of the asteroid once it reaches the Earth's sphere of influence and it is much simpler and faster than time-consuming numerical integration with planetary perturbations performed by ODE.

# Chapter 4: Optimal asteroid mitigation using a low-thrust spacecraft and kinetic impactor

## 4.1 Description of the problem

The spacecraft is permitted to launch as early as January 1<sup>st</sup>, 2026 and must deflect the asteroid before it impacts the Earth on April 13<sup>th</sup>, 2029. The optimizer may choose the point in the initial low Earth orbit from which to begin the departure. It may also choose the direction in which the departure impulse is applied via in-plane and out-of-plane  $V_{inf}$  departure angles. The Earth real orbit about the sun, and its true longitude on the departure date are found using the ephemeris program SPICE from the The Navigation and Ancillary Information Facility (NAIF), acting under the directions of NASA's Planetary Science Division [15]. Specific impulse for the low-thrust motor is chosen from a range of values (2000–4000 s) representative of current technology.

A spacecraft with an initial mass of 10000 kg is launched from Earth. The upper stage of the launch vehicle burns all of its fuel in the initial impulse to propel the spacecraft out of low Earth orbit and onto the first leg of the mission. The spacecraft then switches to low-thrust electric propulsion to travel to the asteroid.

For the first part already described in the previous section 3.1.3 the trajectory of the spacecraft is described by the vector of states, which is one of the parameters to be optimized. This vector, in fact, contains the parameters that compose the equations of motion in Cartesian form,  $X = [x, y, z, v_x, v_y, v_z, m]$ , as seen in section 3.2.2.

The second part consists of forward integration of the asteroid motion from real data obtained from SPICE, the code of this section was obtained from Master's Thesis Research work of Andrew Koehler at University of Illinois Urbana-Champaign.

Once we arrive at the entrance of the Earth's sphere of influence, the position of the asteroid is updated considering the deflection calculated through the STM as seen in section 3.3.1. After crossing the SOI boundary a change of coordinates from heliocentric to geocentric is necessary.

When the spacecraft leaves the sphere of influence of one body and enters that of another the equations of motion, (3.2), become extraordinarily sensitive to changes in some or all of the state variables, which is very disadvantageous for a numerical solution using direct transcription and NLP. The obvious solution is to switch from one set of coordinates, centered on the body the spacecraft is departing, to a set centered on the body to which the spacecraft is arriving, with the switch occurring at the boundary of this body's sphere of influence [9].

After the coordinates change occurs, the propagation of the asteroid's motion continues to the point of closest approach with the Earth and the objective function is calculated.

## 4.2 Initial Guess

Solution via the method of direct transcription with NLP requires that the NLP problem solver be given an initial guess of the vector of NLP parameters. This vector contains the discrete time history of the state and control parameters, which normally number in the hundreds or few thousands, and a small number of additional parameters, for example, times of certain events and possibly the final time. While modern NLP solvers are typically quite robust, it has nonetheless been our experience that a "reasonable" initial guess needs to be provided, especially for large problems.

Of course "reasonable" is not a very precise term. In our experience, initial guesses, that is, approximate candidate optimal trajectories, are more "reasonable" to the extent that they:

- (1) satisfy, at least at the 0th iteration, the system EOM, so that initially all of the nonlinear "defects" are very small;
- (2) satisfy any specified initial and terminal constraints;
- (3) satisfy the boundary conditions given to the NLP problem solver for the upper and lower bounds for all of the parameters.

Creating an initial guess that does all three of these things would be very difficult in most cases; fortunately that is seldom necessary. There are several approaches for the generation of a satisfactory initial guess [9], such as the use of genetic or heuristic algorithms.

The experience previously made on example problems, presented in section 2.1.1, allowed us to be able to use particle swarm optimization (PSO) algorithm as a possible approach to finding optimal spacecraft trajectories.

Because of the limitations of PSO, these trajectories will necessarily be more inaccurate than trajectories found using direct transcription with NLP or even indirect methods, for example, methods based on shooting. However, for the purpose of providing an initial guess for a direct solution, the suboptimal trajectory found using PSO may be completely satisfactory.

It has also been seen that once an initial guess, even one that has very different initial and/or terminal conditions from the desired case, allows the optimizer to converge to an optimal trajectory, new optimal trajectories can be obtained using the converged solution as the new initial guess. Nevertheless, the initial guess we used to obtain the optimal solution was derived directly from a PSO algorithm.

The task of providing a correct initial guess using this method was carried out by Alessia Speziale. She obtained an Initial Guess using a PSO algorithm that use a population of 30 particles and is run for a maximum of 50-100 iterations. The PSO completed the run with a corresponding deviation from the closest approach point of 316.6405 *km*.

The lower and upper bounds for initial guess and the optimal values of PSO parameters for 3D initial guess are shown respectively in Table 4.1 and Table 4.2.

*Table 4.1 PSO lower and upper bounds for initial guess*

Parameter	Lower Bound	Upper Bound
Launch Time	163.4543 TU 1 Jan 2026	179.4543 TU 19 Jul 2028
Initial impulse in- plane pointing angle (radians)	$-\pi$	$\pi$
Initial impulse out-of-plane pointing angle (radians)	$-\pi$	$\pi$
In-plane thrust pointing angle (radians)	$-\pi$	$\pi$
Out-of-plane thrust pointing angle (radians)	$-\pi$	$\pi$
Interception Time	163.4543 TU 1 Jan 2026	183.4543 TU 8 Mar 2029

Table 4.2 Optimal values of PSO parameters for initial guess

Parameter	Optimal Value
Launch Time	172.9019 TU 4 Jul 2027
Initial impulse in- plane pointing angle (radians)	-0.9197
Initial impulse out-of- plane pointing angle (radians)	0.4439
Interception Time	176.4785 TU 28 Jan 2028

A time history of the in-plane control angle  $\beta$  , and the out-of-plane control angle  $\gamma$  was generated. The angles time history and the initial guess trajectory can be found in Figures 4.1, 4.2, and 4.3, respectively.

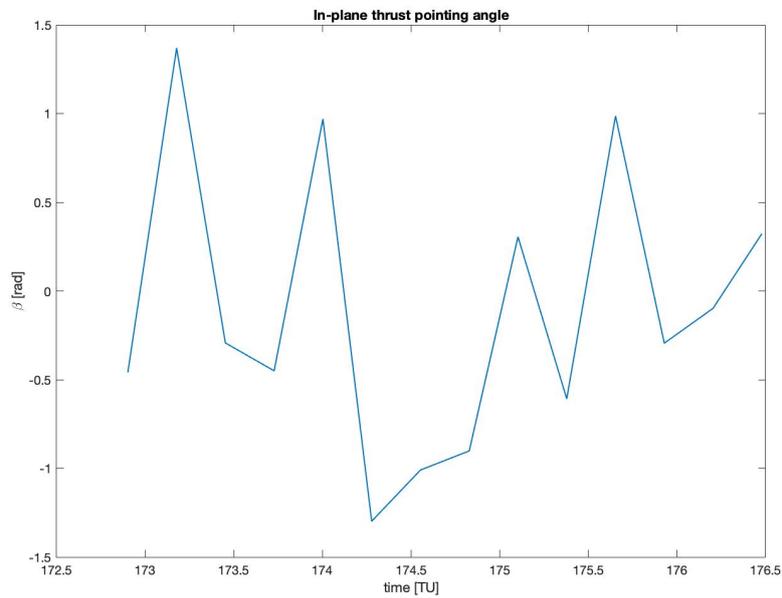


Figure 4.1 - In-plane thrust pointing angle initial guess time history

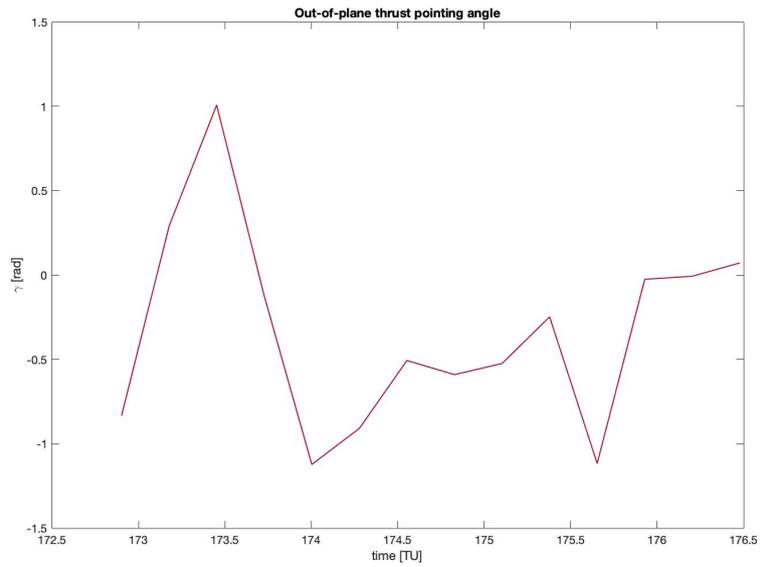


Figure 4.2 - Out-of-plane thrust pointing angle initial guess time history

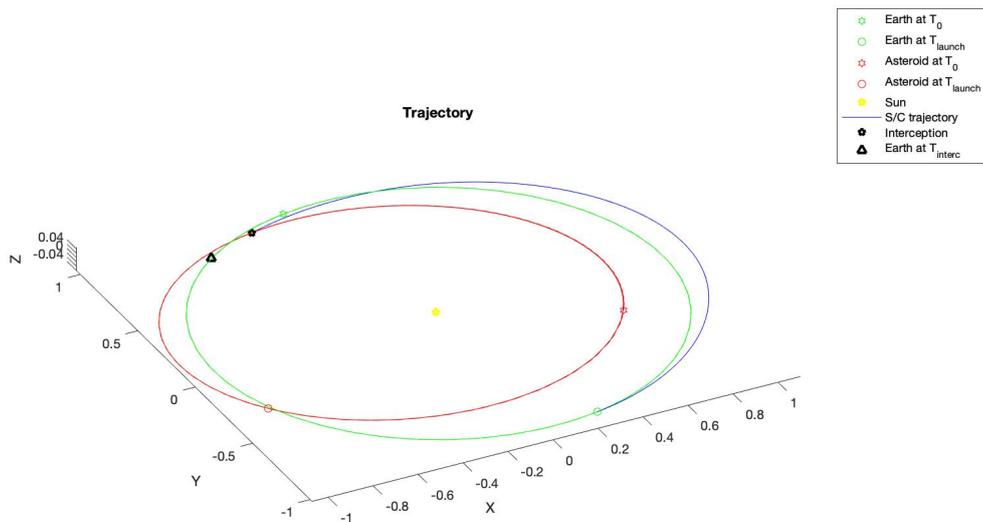


Figure 4.3 - Optimal PSO initial guess trajectory

The resulting launch date, interception time, initial impulse magnitude and direction, time history of the thrust pointing angles, and the state vector were used as an initial guess for the DTRK solution.

### 4.3 Results

The NLP problem is solved using *fmincon*. For the problem under investigation, 29 segments were used to discretize the trajectory, yielding an NLP problem with 564 variables.

After several hours of CPU time, the DTRK solver converged to a solution with a deviation from the closest approach point of 599.4544 *km* and with a satisfaction of the constraints at the intercept,

$$ceq = \sqrt{(x_a - x_{s/c})^2 + (y_a - y_{s/c})^2 + (z_a - z_{s/c})^2},$$

equal to  $2.8703 \cdot 10^{-8}$ .

The optimizer is free to choose two parameters that do not explicitly appear in the variational equations; two pointing angles, in-plane ( $\beta_L$ ) and out-of-(ecliptic)-plane ( $\gamma_L$ ) pointing angles that describe the direction of  $\vec{v}_\infty$  following the impulsive  $\Delta v$ , which allows the vehicle to escape from low-Earth orbit.

In the optimal solution, the spacecraft launches from Earth on April 27<sup>th</sup>, 2027 and applies a  $\Delta v$  of 1.788 km/s in low Earth orbit, with an initial impulse in-plane pointing angle  $\beta_L$  of -1.9724 *rad* and an out-of-plane pointing angle  $\gamma_L$  equal to 0.6313 *rad*. The spacecraft then travels via electric propulsion to the asteroid, intercepting it on December 29<sup>th</sup>, 2027 and imparting a  $\Delta v$  of 1.8878 mm/s to the asteroid. The spacecraft burns 2126 kg of propellant along its trajectory, as we can see in Figure 4.4 and according to the mass equation (3.4), leaving a 7884 kg spacecraft to impact the asteroid. In this equation the maximum thrust  $T_{max}$  has been set equal to  $0.05 \cdot m_0$ , where  $m_0$  is set equal to 1, as is the effective exhaust velocity  $c$ . All expressed in normalized units.

The decrease in mass leads to a consequent, even if limited, increase in acceleration.

The asteroid then coasts for 465 days until its close encounter with the Earth on April 13<sup>th</sup>, 2029. The asteroid is deflected such that it passes 599.4544 *km* from the expected closest approach point. So with a safe distance of about 32600 *km* from the center of the Earth.

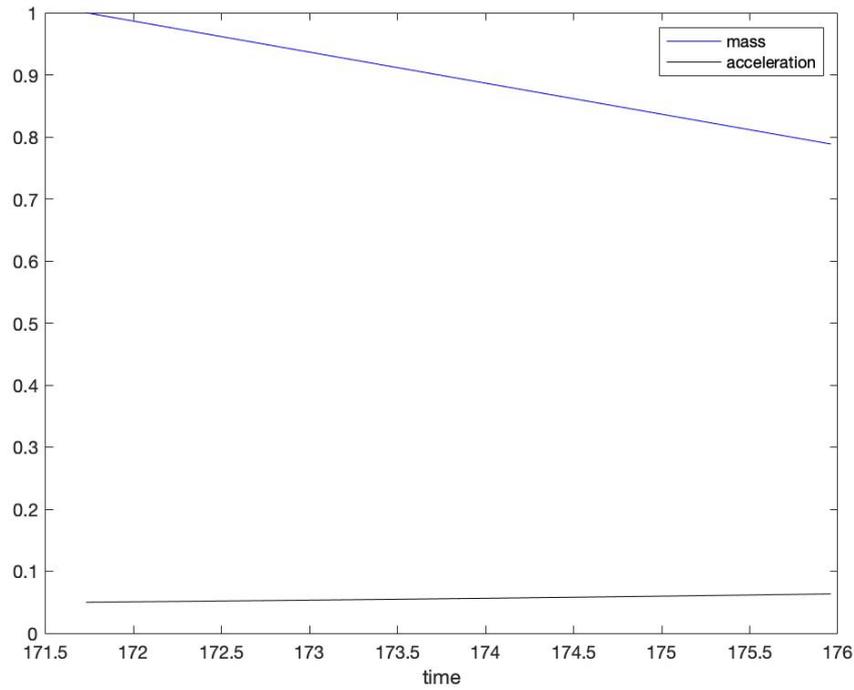


Figure 4.4 - Temporal variation of mass and acceleration

The control variables in the problem are the thrust pointing angles  $\beta$  and  $\gamma$ . Angle  $\beta$  is the in-plane thrust pointing angle; it is measured from the normal to the radius vector and is a positive angle if it yields a component of thrust pointing radially outward. Angle  $\gamma$  is the out- of-plane thrust pointing angle. It is positive if it yields a component of thrust in the direction of the orbital angular momentum. The time history of the thrust pointing angles  $\beta$  and  $\gamma$  are chosen by the problem to maximize the performance index, subject to satisfaction of the system equation of motion, and the system initial condition constraints and the terminal constraint (of interception).

The optimal trajectory and angles time history are given in Figures 4.5, 4.6, and 4.7, respectively. In Figure 4.8 can be seen time history of the semi-major axis  $a$  of the interceptor orbit. The shape of the trajectory in the initial guess in Figure 4.3 is similar to the optimal trajectory in Figure 4.5, although the endpoints are different because the NLP solver modified the launch and intercept dates.

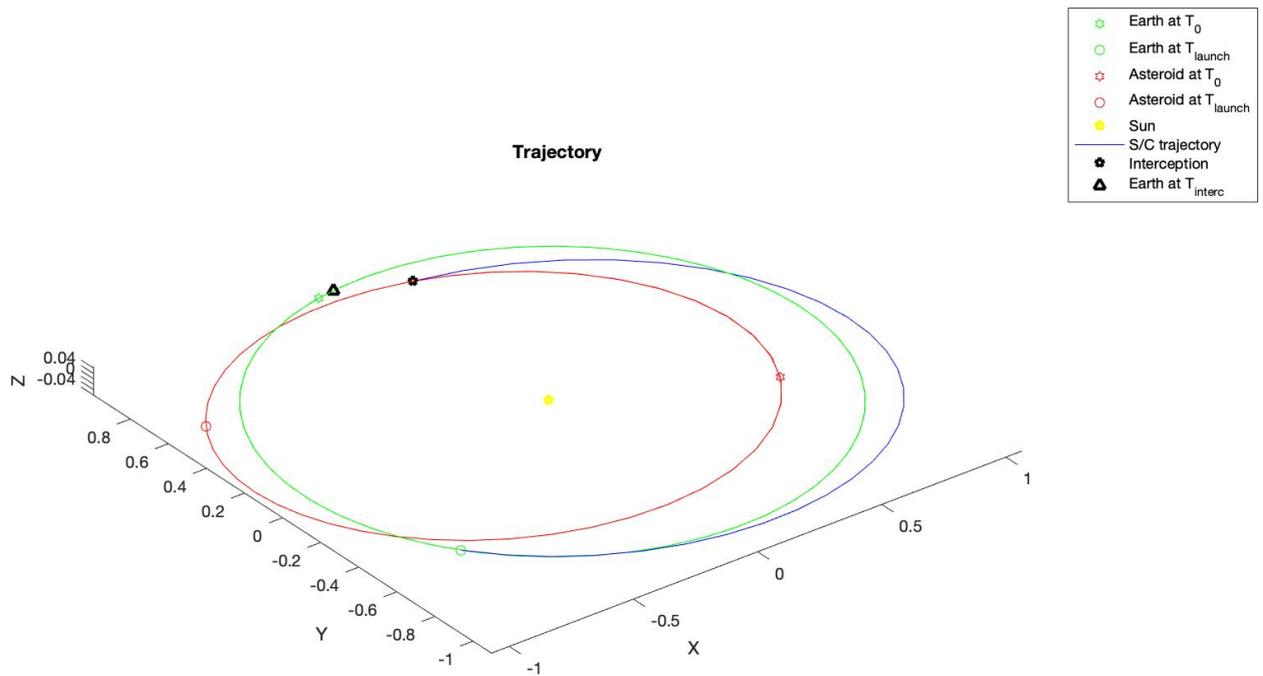


Figure 4.5 - Optimal Trajectory

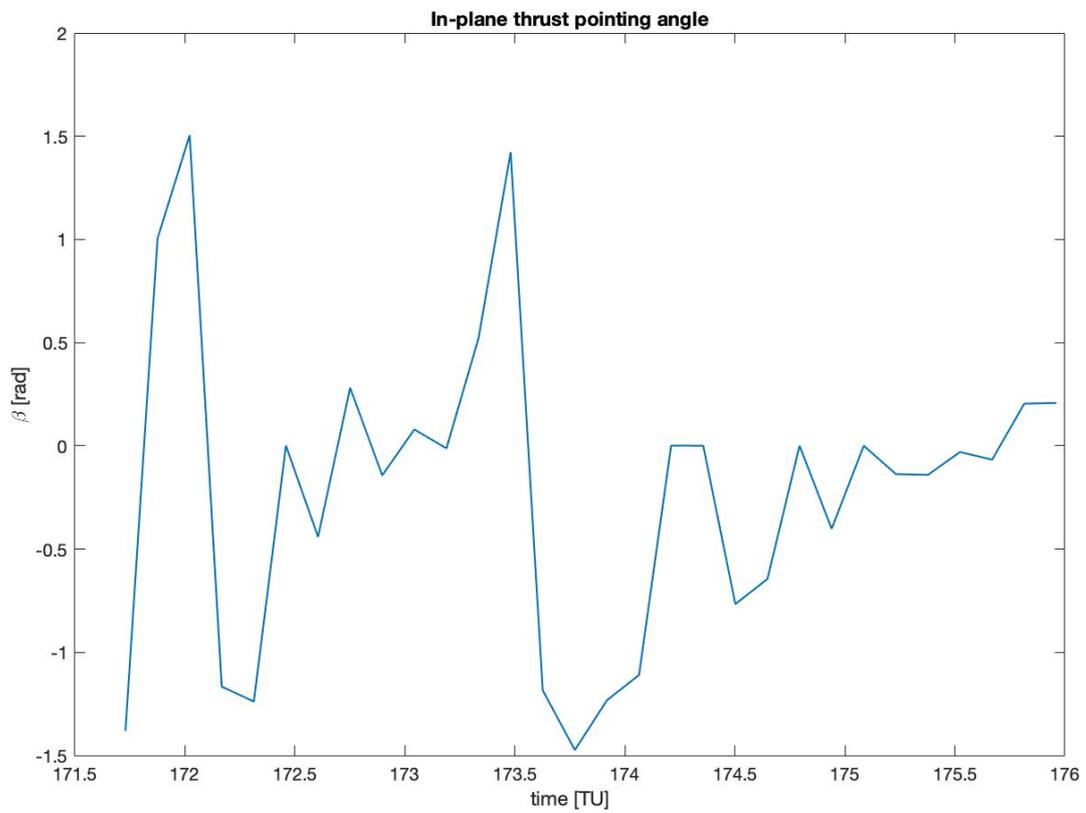


Figure 4.6 - Optimal in-plane thrust pointing angle time history

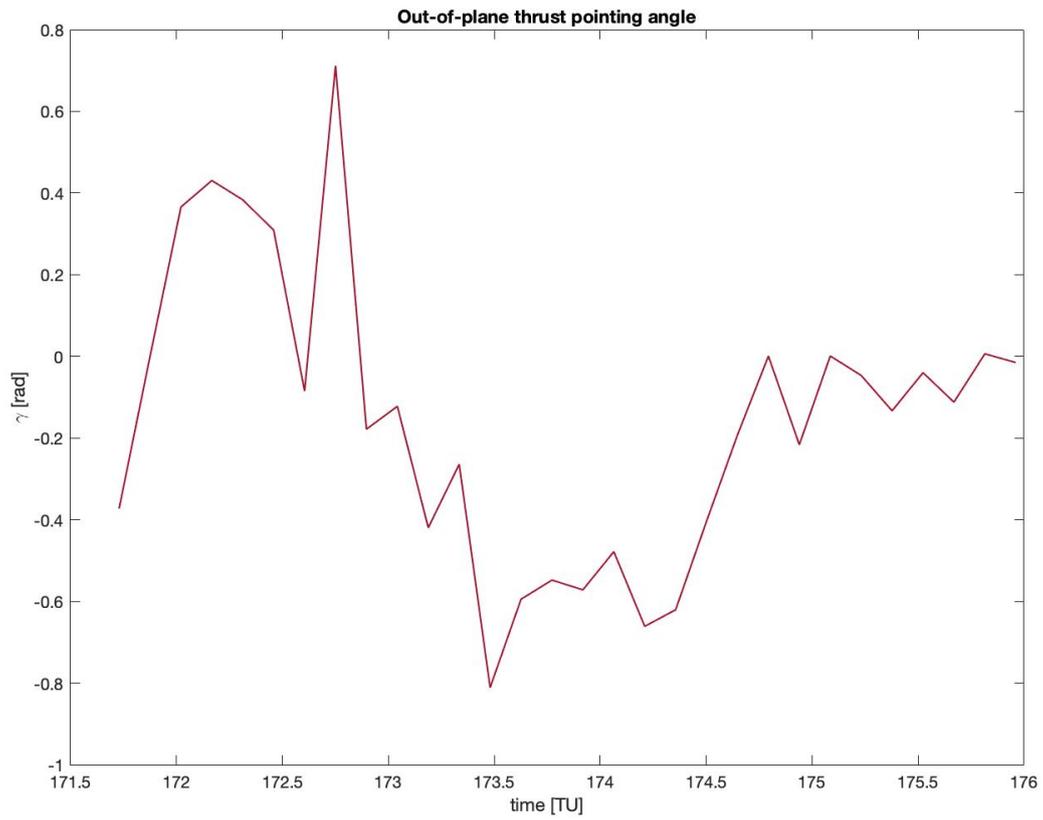


Figure 4.7 - Optimal out-of-plane thrust pointing angle time history

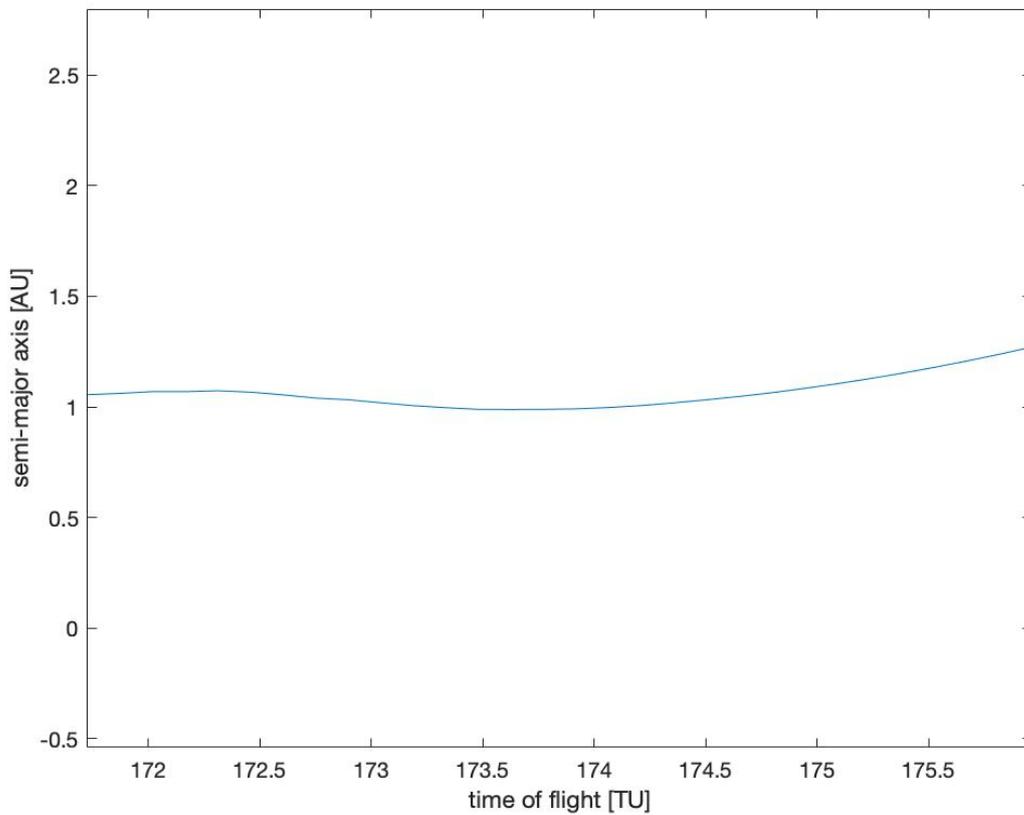


Figure 4.8- History of the semi-major axis of the interceptor orbit

Note that the optimizer chose to wait about 480 days after the opening of the mission window to launch the spacecraft (January 1<sup>st</sup>, 2026). This seems counter to intuition; since the state transition matrix given in equation (3.26) is time dependent, intercepting the asteroid as early as possible would appear to be preferable. In this case, however, it is optimal to wait more than a year and then launch the spacecraft, sacrificing time in order to intercept the asteroid from a more favorable angle and thus impart a larger  $\Delta v$ . A local picture of the asteroid's close encounter with the Earth is given in Figure 4.9.

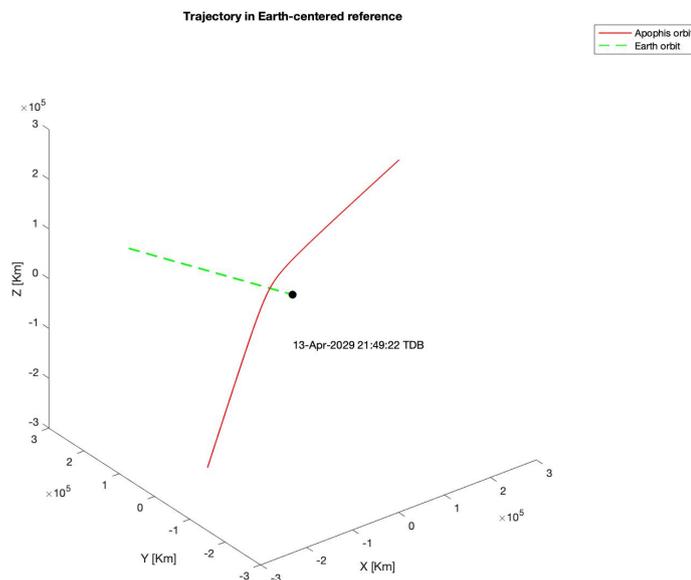


Figure 4.9 - Local view of asteroid's close approach to Earth after deflection

The Earth's gravity altered the orbit of the asteroid during its closest approach to the Earth. Thanks to the deflection and this alteration the asteroid is now on a new orbit and its potential hazard must be re-evaluated.

Taking as initial conditions for integration of the equations of motion the position and velocity of the asteroid as it exited the Earth's sphere of influence following the closest approach, the motion of the asteroid has been propagated and a possible further close encounter with the Earth has been recalculated. This analysis showed that for at least the next 10 years the asteroid will maintain a distance from Earth greater than  $10^6 km$ . Therefore it will not represent a threat to the Earth for at least this period of time.

# Chapter 5: Conclusion

## 5.1 Summary

An analytical model for the optimization of hazardous Near-Earth Objects deflection missions using a kinetic impactor has been developed and tested. In contrast to previous work in this area, this analytical model has been designed for maximizing the "real" objective in asteroid hazard mitigation, the distance by which the asteroid misses Earth impact. A spacecraft using existing technology was found capable of deflecting an asteroid having the same mass and orbit as the asteroid Apophis 599.4544 *km* from the expected closest approach point to Earth. Thanks to this impact, the asteroid does not become a threat again for at least 10 years.

Assuming a launch window opening time, the spacecraft is found to depart more than a year later intercepting the asteroid in the following months.

Many of the strategies for amelioration of the danger of an asteroid's collision with the Earth involve, at the time of interception, applying a small impulsive velocity change to the asteroid. In this work we show how, using the asteroid orbit state transition matrix, this impulse should be applied to maximize the deflection of the asteroid at the time of close approach.

## 5.2 Future work

Future research should focus on four main improvements that need to be made: changing some assigned parameters in the model, improving the fidelity of the model, increasing the maneuvering options available to the spacecraft, and testing it on other targets.

Currently the model contains some initially imposed parameters such as the maximum thrust value  $T_{max}$ , the launch window opening date  $T_0$  or the value of the initial impulse magnitude provided by the upper stage of the launch vehicle  $\Delta v$ . It would be interesting to study the evolution of the solution by changing the value of one or more of these parameters.

The fidelity of the model can be improved by adding to the existing planetary perturbations some of the larger and easier to model perturbations to the asteroid orbit, such as the effect of solar radiation pressure and the solar wind. Or even providing additional details regarding the spacecraft or the asteroid itself.

Increasing the maneuver options available to the spacecraft is also another improvement that needs to be made. At present, the spacecraft can travel propelled by its low-thrust engine. Future work should incorporate planetary flybys so that the spacecraft can choose to make a close approach to Venus or Mars to gain speed and change its course. A planetary flyby would allow the spacecraft to significantly change its trajectory without expending any propellant. New and improved intercept geometries could be available with this method.

Finally, another action to try is definitely to test the model, in its current form or some evolved form taking into account the previous suggestions, against other targets such as asteroids with high inclination orbits and short and long period comets. Although low-inclination asteroids with Earth-like orbits, such as Apophis, are the most likely threat due to the fact that they spend a lot of time near Earth, they are by no means the only danger. These common threats, in fact, since they would impact the Earth at a lower relative velocity would do significantly less damage than a high inclination asteroid or a short or long period comet, which due to a higher relative velocity and therefore a higher release of energy on impact are more lethal. Although the model was not developed with such goals in mind, it should still prove useful.

## References

- [1] NASA [Online] <https://www.nasa.gov/planetarydefense/did-you-know>
- [2] *Spacecraft Trajectory Optimization*, ed Conway, Bruce A. Published by Cambridge University Press. ©Cambridge University Press 2010. pp 1-13.
- [3] *Spacecraft Trajectory Optimization*, ed Conway, Bruce A. Published by Cambridge University Press. ©Cambridge University Press 2010. pp 263-295
- [4] Eberhart, R., and Kennedy, J. (1995) *A New Optimizer Using Particle Swarm Theory*, Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan.
- [5] Kennedy, J., and Eberhart, R. (1995) *Particle swarm optimization*, Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ.
- [6] Orbital Mechanics for engineering students (sez. 6.3) - Howard Curtis
- [7] MATLAB. [Online] 2008. <http://www.mathworks.com>.
- [8] Enright, Paul J. and Conway, Bruce A. Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming. 4, July-Aug. 1992, Journal of Guidance, Control, and Dynamics, Vol. 15, pp. 994-1002
- [9] *Spacecraft Trajectory Optimization*, ed Conway, Bruce A. Published by Cambridge University Press. ©Cambridge University Press 2010. pp 37-76.
- [10] Kelly, M. P., 2015, *Transcription Methods for Trajectory Optimization*, Tutorial, Cornell University, Ithaca, New York.
- [11] P. E. Gill, E. Wong, W. Murray, M.A.Saunders, *User's Guide for SNOPT Version 7.7: Software for Large-Scale Nonlinear Programming*, Stanford, CA : Stanford University, March 2021
- [12] *Mitigation of Hazardous Comets and Asteroids*, ed. M. J. S. Belton, T. H. Morgan, N. H. Samarasinha, and D. K. Yeomans. Published by Cambridge University Press. ©Cambridge University Press 2004. pp.167-200
- [13] Conway, Bruce A. *Optimal Low-Thrust Interception of Earth-Crossing Asteroids*. 5, Sep.-Oct. 1997, Journal of Guidance, Control, and Dynamics, Vol. 20, pp. 995-1002.
- [14] Conway, Bruce A. *Near-Optimal Deflection of Earth-Approaching Asteroids*. 14, Apr. 2001, Journal of Guidance, Control, and Dynamics, Vol. 24, NO 5, pp. 1035-1037.

- [15] J.Englander, *Optimal strategies for deflecting hazardous Near-Earth Objects via kinetic impactor*, Master's Thesis, Graduate College of the University of Illinois at Urbana-Champaign, 2008, adviser Professor Bruce A Conway.
- [16] Sansaturio, Maria E. and Arratia, Oscar. *Apophis: the Story Behind the Scenes*. 1-4, June 2008, Earth, Moon, and Planets, Vol. 102, pp. 425-434.
- [17] NASA [Online] <https://www.nasa.gov/feature/jpl/nasa-analysis-earth-is-safe-from-asteroid-apophis-for-100-plus-years>
- [18] Giorgini, Jon D., et al. *Predicting the Earth encounters of (99942) Apophis*. 193, 2008, Icarus, pp. 1-19.
- [19] JPL HORIZONS Database. *JPL Solar System Dynamics*. [Online] 2008. <http://ssd.jpl.nasa.gov/>.
- [20] Battin, Richard H. *An Introduction to the Mathematics and Methods of Astrodynamics*. Reston, VA : AIAA, 1987, pp. 187-190.
- [21] Park S-Y, Mazanek D. D. *Mission Functionality for Deflecting Earth-Crossing Asteroids/Comets*. 5, Sep.-Oct. 2003, Journal of Guidance, Control, and Dynamics, Vol. 26, No. 5.
- [22] NASA [Online] <https://www.nasa.gov/image-feature/orbits-of-potentially-hazardous-asteroids-phas>
- [23] Online - <https://medium.com/@iamterryclark/swarm-intelli-eb5e46eda0c3>
- [24] Online - <https://mathworld.wolfram.com/CylindricalCoordinates.html>
- [25] Online - <https://x-engineer.org/euler-integration/>