A Rapid Grid-Search Technique for KBO Exploration Trajectories

Miguel Benayas Penas

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

A Rapid Grid-Search Technique for KBO Exploration Trajectories.

by

Miguel Benayas Penas

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday January 7, 2020 at 15:00.

Student number:	4503228	
Project duration:	June 29, 2018 – November 20, 2019	
Thesis committee:	MBA ir. K.J Cowan	TU Delft, supervisor
	Dr. ir. K.M. Hughes,	NASA, supervisor
	Prof. dr. ir. P. Visser,	TU Delft
	Dr. ir. A. Cervone,	TU Delft

Title page image copyright: NASA/JHU APL/SwRI/Steve Gribben

TU Delft

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisors Kyle Hughes and Kevin Cowan for all their kindness, patience, time and technical support. This thesis would have not been possible without them.

Secondly, I would like to thank all the NASA Goddard Space Flight Center colleagues, it has been a privilege both personally and professionally speaking.

I also want to thank my dear "Bombers" Livio, Giovanni P., Giovanni G., Alex, Mattia, Francesco... without you guys life in Delft would have been much harder. I could not forget about Skarleth, Jorge, Bea, and "La Tríada” thank you all for your friendship despite distance.

Finally, I would like to dedicate these last lines to my parents and brother for their unconditional love and support.

Abstract

The Kuiper Belt is considered to be formed by remnants of the original Solar System, that is why exploration missions to that region are susceptible of having an enormous scientific impact. Nonetheless, it is far away from Earth, missions to KBO targets a significant challenge. Additionally, current methods to identify trajectories that encounter multiple KBOs are computationally intensive with impractically long run times on the order of months.

The present work deals with applying a rapid low-fidelity technique to identify candidate preliminary KBO sequences, to be used as a starting point for future mission designers to identify trajectories to multiple KBOs. This pathfinding approach uses two consecutive grid search types.

First, a well-tuned grid search is implemented using Lambert arcs to reach a first KBO, including the introduction of a new, systematic approach to identify the step sizes in target-body arrival date for the Lambert-based grid search. Then, second and third KBO encounters are assessed from the results of the first grid search and second KBO search respectively. Both second and third KBO searches are performed with a new algorithm consisting of a time grid along with STM linear propagation for maneuver calculation. Finally, an example trajectory resulting from this technique that encounters two KBOs is given as a potential flyable route.

Contents

Appendices

List of Figures

List of Tables

Nomenclature

ABBREVIATIONS

AU	Astronomical Unit
COOP	Cooperative Algorithm of DE and PSO
DE	Differential Evolution
DSM	Deep Space Maneuver
EA	Evolutionary Algorithm
EMTG	Evolutionary Mission Trajectory Generator
GA	Genetic Algorithm
GTOP	Global Trajectory Optimisation Problems
HF	Hybrid Formulation
HTIT	High Thrust Interplanetary Trajectory
ICRF	International Celestial Reference Frame
Isp	Specific Impulse
JD	Julian Date
JPL SSD	Jet Propulsion Laboratory Solar System Database
KBO	Kuiper Belt Object
Launch C_3	Launch Energy
LRTS	Lazy Race Tree Search
MGA	Multiple Gravity Assist
MCTS	Monte Carlo Tree Search
NKBO	Nominal Trajectory KBO
NKSA	N^{th} KBO Search Algorithm
NT	Nominal Trajectory
OCT	Optimal Control Theory
PF	Position Formulation
PSO	Particle Swarm Optimization
RQ	Research Questions
RO	Research Object
SA	Search Algorithm
SAF	Semi-Analytic Formulations
SC	Spacecraft
SKSA	Second KBO Search Algorithm
SP	Search Period
STM	State Transition Matrix
STOUR	Satellite Tour Design Program
TOF	Time Of Flight
VF	Velocity Formulation

SYMBOLS

a	Semimajor axis
E	Energy
e	Eccentricity
i	Inclination
k	Local Curvature
P	Orbital Period
r_p	Orbital Periapse Position Vector
R	Local Radius of Curvature
μ	Standard Gravitational Parameter
V_∞	Excess Velocity

1 Introduction

This first chapter addresses the motivation of this work, the research goal and questions, the theoretical background necessary to easily read the thesis, and the thesis structure. These four blocks are represented in the chapter's four sections. The first section presents what has been researched and why. The second section narrows down the thesis motivation, setting the research goal and questions to tackle. The third section contains the key theoretical concepts used to design the proposed KBO pathfinding technique. Finally, the last section of this chapter consists of a brief description on how the research has been organized in order to effectively assess the research goal and the research questions.

1.1. Motivation

The Kuiper Belt, also referred to as the Edgeworth–Kuiper Belt, is a circumstellar disk made out of small bodies located in the outer Solar System, with semi-major axes ranging from 30 Astronomical Units (AU), slightly beyond Neptune's orbit, to approximately 50 AU from the Sun [39]. It is deemed that it was formed from the outer region detachment of the Solar Nebula, as a result of its gravitational collapse and the angular momentum conservation law [41]. That is why these bodies are considered to be remnants of the original Solar System, despite having being exposed to effects such as radiation-reddening or impact-graying over time [41].

In spite of its prominent scientific relevance, the Kuiper Belt is still a mysterious region. So far, more than 2,000 KBOs have been cataloged by observers, representing only a small fraction of the total number of objects that are postulated to exist [11]. In addition, only one mission (New Horizons) has explored bodies in the Kuiper Belt so far, Pluto-Charon system and Ultima Thule [28][48].

Therefore, a mission focused on visiting the Kuiper Belt would be highly desirable in order to expand our knowledge of KBOs and, consequently, the primitive Solar System.

Unfortunately, mission design is a non-trivial task. In fact, it is an iterative process where a body encounter sequence is firstly chosen and, then, trajectories visiting all the bodies in that selected sequence are computed and evaluated. These two distinct stages of the mission design problem are sometimes known as pathsolving and pathfinding respectively in literature [22][30][38].

The non-triviality issue is aggravated in this thesis due to the immense number of potential body encounter sequences to evaluate, compared with planetary missions. In this thesis, on top of conducting planetary flybys, KBO sequences have also to be included in pathfinding.

Due to the usual mission design complexity, the approach followed by several world-class institutions (Figure 1.1) consists of a tool sequence of increasing fidelity. The reason is simple, instead of directly applying a high-fidelity tool[1], a low-fidelity one is applied first to detect the regions of interest and then perform local searches in those regions with tools of increasing mathematical model complexity. Search space is narrowed

[1] Which is expensive in terms of computational time and would make the search space scouting intractable.

down after each tool. This approach sharply reduces the required computational time to get (near) optimal solution.

Therefore, preliminary mission analyses[2] are used to provide suboptimal guesses for a later optimization in tools with higher fidelity[3] such as the General Mission Analysis Toolkit (GMAT) [32], the Systems Tool Kit (STK) [13], FreeFlyer [5], Copernicus [43], MONTE [9] or NASA's Evolutionary Mission Trajectory Generator (EMTG) [55] for instance.

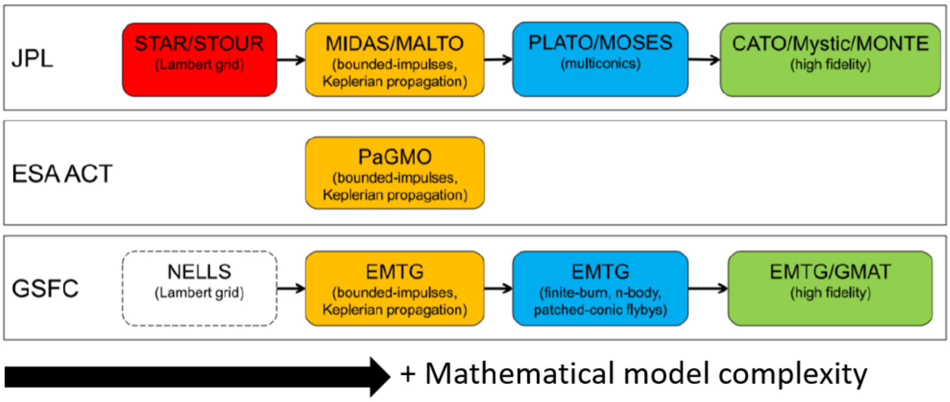

JPL
STAR/STOUR
(Lambert grid)
MIDAS/MALTO
(bounded-impulses, Keplerian propagation)
PLATO/MOSES
(multiconics)
CATO/Mystic/MONTE
(high fidelity)
ESA ACT
PaGMO
(bounded-impulses, Keplerian propagation)
GSFC
NELLS
(Lambert grid)
EMTG
(bounded-impulses, Keplerian propagation)
EMTG
(finite-burn, n-body, patched-conic flybys)
EMTG/GMAT
(high fidelity)
+ Mathematical model complexity

Figure 1.1: Mission design workflow paradigm for different organizations [24]. Colors/columns refer to four different levels fidelity or, in other words, model complexity.

This thesis deals with the development of a rapid low-fidelity technique to help future mission designers plan Kuiper Belt exploration missions. In particular, the technique is to be used as a pathfinding solver to seed[4] EMTG[5], a higher fidelity tool, for pathsolving.

The technique, hereinafter referred as SA for Search Algorithm, consists of two types of grid search run in a sequential way. In other words, the SA **consists of two phases** written in Python 3.6, each one associated with a different type of grid search. The first one is a ballistic trajectory grid search for a single KBO performed with the NASA Exhaustive Lambert Lattice Search (NELLS) low fidelity tool, still under development [24].

NELLS is based on the Satellite Tour Design Program (STOUR), developed at the Jet Propulsion Laboratory (JPL) for the Galileo mission to Jupiter [47]. STOUR originally consisted of approximating mission trajectories as two-body conic sections connected with patched-conic flybys for a given path of targets, operated only through an interactive mode. Over the years, updates to the initial STOUR concept such as providing an automate seek mode [58] or a maneuver package [44] were added. NELLS requires as input the body encounter sequence to calculate trajectories from, as may be seen in Appendix A.

The second grid search type seeks KBOs in the vicinity of the trajectories attained in NELLS since (a priori) They can be visited by applying small maneuvers. Due to time constraints, this second grid search type is a simplified version of a non-ballistic NELLS grid search . The consequence is the total delta-V of the attained trajectories can not be regarded as good approximations of the actual values, instead, total delta-V will be deemed as a KBO alignment[6] metric.

1.2. Research objective and questions

Based on section 1.1 considerations, the thesis research objective along with its research questions are hereunder provided.

[2] With simplified mathematical models.

[3] Which use more realistic mathematical models.

[4] To be used as first guess.

[5] The choice of EMTG is not arbitrary, it is due to the fact that the research was conducted at one NASA Goddard Space Flight Center (GSFC) group, being EMTG its usual medium-high fidelity tool.

[6] KBO alignment shall be understood in this thesis as, for a given trajectory, the KBOs are forming a smooth trajectory with respect to themselves and the last planet flyby.

The research objective is to develop a rapid low-fidelity technique (SA) for pathfinding in EMTG. SA will perform a grid search in NELLS for trajectories[7] to all the cataloged KBOs in the JPL's Solar System Database (JPL SSD) [7], followed by a grid search in time to modify each of the attained trajectories in order to add at least one more KBO flyby.

Wherein "low-fidelity" is attributed to the SA due to its reliance on the assumptions explained in subsection 1.2.3, which decrease the results accuracy at the expense of needing lower computational times with respect to the non-simplified problem of trajectory optimization.

In order to satisfy the research objective, the following research questions have to be answered.

- **RQ1**. What are the proper step size values for discretizing the target body locations for the Lambert grid-search[8]?
- **RQ1.1**. Do intermediate flybys affect target body step sizes?
- **RQ2**. Are there any trajectories which encounter one KBO, fulfilling the mission constraints discussed in subsection 1.2.2?
- **RQ3**. Are there any trajectories which encounter at least two KBOs with a delta-V no higher than 2 km/s, fulfilling the mission constraints discussed in subsection 1.2.2?
- **RQ3.1** What is the proper time step size for the time grid?
- **RQ3.2** Is it possible to apply a KBO filter, in order to avoid conducting a second grid search for unreachable KBOs?

There exits a remarkable conceptual leap from the Motivation section, where the RO was simply outlined at the end of that section 1.1, with respect to the aforementioned RO and RQs. The following subsections 1.2.1, 1.2.2, and 1.2.3 aim to clarify the choice for this RO and these RQs.

1.2.1. Why grid search?

A wide range of methods has been utilized to tackle the interplanetary trajectory design problem, both in its combinatorial subproblem, finding the body encounter sequence to follow, and in its trajectory optimization subproblem with the body sequence already fixed.

Methods to solve these problems may be grouped in the following families.

1. Nature-Inspired Methods such as Evolutionary Algorithms[9], Particle Swarm Optimization or Simulated Annealing.
2. Repeated Local Searches like Multi-Start or Monotonic Basin Hopping.
3. Enumerative Searches like Grid Search.
4. Composite Algorithms based on combinations of other methods.

Besides, there exist other methods that can be added on top of the previously listed ones such as Branching, Parallel Computation or Self-Adaptation in order to decrease the total number of cost function evaluations before (approximately) reaching the optimum solution.

A summary of the behavior of these algorithms along with performance comparisons for several high-thrust interplanetary trajectory problems may be found in the author's literature study [18].

There are three main factors shown in that document which are important for this thesis as well, since they sway some key decisions. These three factors are as follows.

[7] Meeting the mission constraints listed on subsection 1.2.2 and subject to the section 1.2.3 assumptions.

[8] As will be shown in the thesis, the first grid search core theoretical background relies on the patched-conics approach along with the Lambert's problem. Therefore a fine bodies trajectory discretization represents an essential part to attain good results with a tractable computational time.

[9] In this thesis the terms algorithm and method are considered synonyms and, thus, may be used interchangeably.

1. Firstly, all the aforementioned methods must go through a parameter tuning phase before seeking the problem solution. This is done in order to increase their performance in terms of the number of cost function[10] evaluations required to get at least a 95% chance of finding the solution.

2. Secondly, one cannot assure whether the best found solution is actually the optimum one when using a well-tuned algorithm or a combination of the Nature-Inspired, Repeated Local Search families. The reason is those methods have a random component and, thus, are not deterministic regarding search space mapping.

3. Conversely to the second conclusion, A fine-tuned parameter space grid search is itself theoretically complete albeit is computationally expensive, with respect to the other methods, for large search spaces. Interplanetary trajectory optimization problems usually present large search spaces even when using simpler mathematical models like the patched-conics approach[11], especially if Deep Space Maneuvers (DSMs) are allowed.

Based on these three ideas, the grid search method was selected for the SA because, unlike the other methods, one can be sure of having mapped the whole search space with low chances of missing out regions of interest if a proper grid resolution[12] is used. It is therefore essential to find a grid resolution fine enough to avoid ignoring many potential minimum basins but coarse enough to avoid intractable computational times. In fact, the reason why a second grid search type[13] was implemented to seek extra KBO flybys was due to time constraints, since finding extra KBO to flyby represents an much larger combinatorial space than reaching the first one.

It is important to recall SA is a pathfinding technique to seed EMTG, a higher fidelity tool. Consequently, a loss of accuracy as result of increasing the grid size or linearizing the equations of motions to look for near KBOs for each NELLS trajectory may seem reasonable.

1.2.2. Mission constraints

The constraints chosen for this work in order to limit the search space in the grid searches are the following ones.

1. Departure from Earth, whether as launch or continuation from an earlier gravity-assist sequence. Start date is January 1, 2026 (2461041.5 in JD) through December 31, 2041. These were supposed to just be useful dates for future mission concepts, since they roughly align with the next planetary decadal survey (led by the United States National Academy of Sciences), which directs all NASA missions for the decade of 2023-2032 (and possibly beyond) [10].

2. Maximum time-of-flight (TOF) to first KBO of 15 years and maximum total mission TOF of 20 years. These values represent a conservative upper bound based on benchmark missions like New Horizons [28], Voyager 2 [25][36][37] or preliminary mission proposals present on a NASA decadal surveys [40]

3. Maximum delta-V for maneuvers of 2 km^2/s^2.

4. Launch C_3: Maximum value of 500 km^2/s^2 (far beyond Cassini's last Earth flyby of 361.5 km^2/s^2 [46], the largest Earth C_3 ever) and minimum value of 49 km^2/s^2, based on Tisserand graph analyses (see Appendix B).

Where the "earlier gravity-assist sequence" mentioned in the earliest Earth launch date constraint, makes reference to a possible flyby sequence that a spacecraft could perform before conducting a final Earth flyby, where the trajectories in this study begin. This initial phase would have the goal of decreasing the required launch C_3 to be provided by the launcher. Due to the time constraints, none of that family of sequences is evaluated in this thesis, representing a worthwhile avenue for future research.

[10] The function to maximize or minimize, depending on the problem.

[11] Simplified problem wherein spacecraft motion is only affected by solar Newtonian gravity, body flybys, and DSMs. Both body flybys and DSMs are only assumed to produce an instantaneous delta in spacecraft velocity, not in position. See [54] for more details.

[12] The sampling frequency, that is, the grid size.

[13] which constitutes a simplification of the NELLS maneuver implementation, a linearization of equation of motion in particular.

The earliest **Earth departure date** was set to January 1st 2026, in order to seek potential KBO sequences that could be contemplated in missions for the next NASA Planetary Decadal Survey, as the current one spans from 2013 to 2022[14]. Nonetheless, previous years (for instance 2024 or 2025) were discarded for first possible launch dates due to New Horizons mission took 5 years from conceptual design to its launch [52].

The maximum permitted **delta-V** value was set to 2 km^2/s^2, a high value compared to the benchmark value of 500 m/s taken from averaging interplanetary mission delta-V budgets[15] such as Voyager, Messenger, Cassini, Galileo, or New Horizons [21][23][28][45]. This conservative value of 2 km/s was chosen in order to reduce the risk of losing candidate trajectories that could lead to low delta-V trajectories using higher fidelity tools. That is, it could happen that some SA trajectories with high mission delta-V values are in the vicinity of trajectories with small delta-V budget, which could be discovered in a later local optimization phase running those SA trajectories as first guesses in a pathsolving tool.

1.2.3. Assumptions made in the research objective and questions

The following assumptions have been made in order to ensure the RO feasibility within the time schedule and computational resources available, taking into account that grid search is an expensive method, at the expense of decreasing the accuracy of the SA results.

Nonetheless, this does not suppose a problem as the aim of the thesis is to produce a rapid low-fidelity technique for pathsolving. Such a technique will provide initial guesses for a later local optimization tool like EMTG.

The assumptions made for the **first grid search type** (i.e., the first of the two SA phases) are the following ones.

A1.1. The adopted mathematical model is the patched-conics approach, a widely used model in preliminary mission analysis [54]. This approach considers the spacecraft interplanetary motion is only affected by the Sun gravity, DSMs and (un)powered flybys. Besides, this approach also assumes a zero-dimension sphere of influence and (if any) instantaneous impulses in the heliocentric perspective. Therefore, both flybys and DSMs influence are modelled as an impulse in velocity with position unchanged. Motion during flybys is modelled as hyperbolas.

A1.2. No revolutions[16] are considered. The reason is adding the possibility of performing revolutions in transfer orbits sharply increases the required computational time while (in addition) only a small subset of trajectories was assumed to be affected for that decision.

Taking into account that a Hohmann transfer from Earth to Jupiter [17] is approximately 5.5 years, a multi-rev solution to Jupiter (without a DSM or additional flyby) would be at about 16.5 years (one full rev plus E-J Hohmann leg, exceeding the 15-year limit.

A1.3. Only ballistic trajectories to the first KBO encounter are considered for two reasons.

Firstly, performing a grid search for all the DSM parameters, both angles and magnitude of delta-V, would sharply increase the search space complexity, both adding more local minima clusters around the local optimum basins and creating new local optimum basins [42][14]. These two effects of including DSMs make very hard for the thesis time constraints to tune and then to conduct a well-tuned grid search for all the KBOs listed in the JPL SSD.

Secondly, the Jupiter gravity-assist has proven to be enough to reach the Kuiper Belt region, as may be seen whether in the Tisserand Graph (Figure B.1) wherein the Jupiter's curves intersect with Pluto's or in New Horizon's trajectory Figure 1.2.

[14] The current planetary decadal survey may be found in the following link, `https://science.nasa.gov/about-us/science-strategy/decadal-surveys`

[15] Only the delta-Vs related to interplanetary maneuvers or trajectory corrections before arriving the target planet were regarded to compute that benchmark value. For instance, maneuvers for orbit insertion, to raise the periapse, or during launch were excluded.

[16] The spacecraft motion in a complete orbital period In circular or elliptical orbits. That is, the spacecraft motion for a true anomaly increment of 2π radians.

[17] Which may be considered a benchmark for those potential orbits to perform a revolution.

A1.4. This last assumption addresses the issue of selecting which planet flyby sequences are to be studied in the first grid search. Gravity-assists play a major role especially in missions aiming distant targets as they give rise to important changes of energy and/or inclination in spacecraft orbits.

Only the following planet sequences are evaluated in the first grid search: E-J-KBO, E-J-U-KBO and E-J-N-KBO or just EJ,EJU and EJN. Where E,J,U, and N stand for Earth, Jupiter, Uranus, and Neptune respectively.

This decision may be explained taking into account that Mars does not provide an important flyby impulse and Saturn is relatively close to Jupiter with respect to the distance between Earth and Kuiper Belt. Therefore, a Saturn flyby effect during trajectory search should be possible to approximate with an EJ at a certain launch date. Finally, Uranus is misaligned with respect to Neptune for the time of interest [30] and, hence, it is not possible to connect Uranus and Neptune in the same flyby sequence.

One could argue why it was deemed important to also consider EJU and EJN if Jupiter is the most massive planet in the Solar System[18] and the launch window is larger than Jupiter's orbital period. In addition, as said before, EJ should be enough to search for trajectories in all directions. The reason is (see Figure 1.2) both Neptune and Uranus could bend the incoming trajectory, fostering the modified trajectory can stay longer Kuiper Belt and, therefore, increasing the chances of encountering more KBOs to flyby around its vicinity[19].

The idea of bending the trajectory to enlarge its visit of the Kuiper Belt region was inspired in the hyperbolic trajectory of New Horizons, illustrated in Figure 1.3 and depicted as the black trajectory displayed in Figure 1.2.

On top of the assumption **A1.2**, three more assumptions were made in the **second grid search type**.

A2.1. Orbit propagation is performed under the patched-conics approach. Nonetheless, maneuvers to reach secondary KBOs are calculated using instead a linear propagation of the State Transition Matrix (STM).

Using linear approximation sharply reduces required computational time at expenses of decreasing the accuracy of delta-V calculation. Such accuracy decreases the more the STM propagation deviates from the nominal trajectory. Therefore, the linear approximation will only yield similar results than the real solution for secondary KBOs in the vicinity of the first grid search's resulting trajectories. There is no guarantee the delta-V is close to the actual value for more distant secondary KBOs, further details are given in Chapter 4.

A2.2. Only high-thrust maneuvers are considered in this thesis.

A2.3. KBOs do not perturb trajectories during flyby.

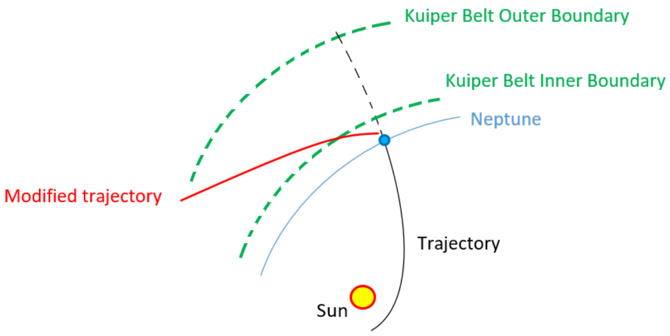


Figure 1.2: Conceptual design on the advantages of including the outermost Solar System planets. Flyby about Neptune bends the black trajectory turning it into the 'modified trajectory', which crosses a longer distance in the Kuiper Belt region than the black trajectory.

[18] In fact, New Horizons only needed to perform a flyby in Jupiter to reach Kuiper Belt.
[19] This could be crucial taking into account only a small fraction of KBOs are catalogued in the JPL SSD.

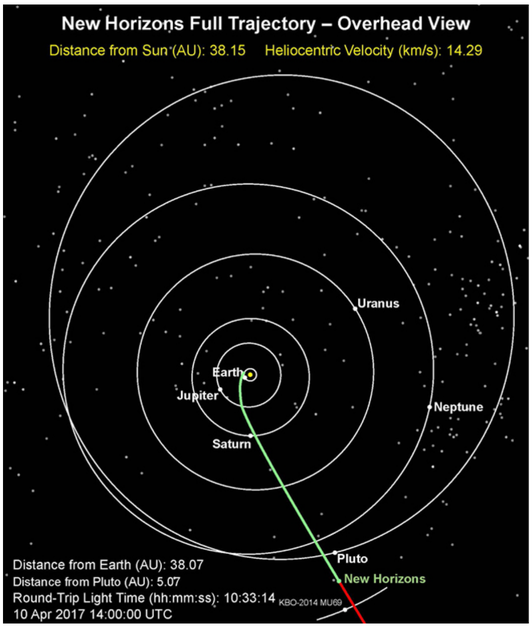

New Horizons Full Trajectory – Overhead View
Distance from Sun (AU): 38.15
Heliocentric Velocity (km/s): 14.29
Uranus
Earth
Jupiter
Neptune
Saturn
Pluto
New Horizons
KBO-2014 MU69
Distance from Earth (AU): 38.07
Distance from Pluto (AU): 5.07
Round-Trip Light Time (hh:mm:ss): 10:33:14
10 Apr 2017 14:00:00 UTC

Figure 1.3: New Horizons hyperbolic trajectory, analogous to the black trajectory given in Figure 1.2 [2].

1.3. Methods used in the SA

The key methods used in the SA are described in this section. The first three methods are used in the SA's first grid search type whereas the fourth one is employed for the second grid search type.

1.3.1. C_3 matching

Low-fidelity grid search tools for ballistic multigravity-assist trajectories (MGAs), approximate trajectories as successions of conic sections (resulting from the two-body problem) connected for a certain body encounter sequence like (for instance) the one depicted in Figure 1.4. In other words, MGAs consist of solving a sequence of Lambert's problems for a certain body sequence. Two critical questions arise, the election of fast and robust Lambert's problem solver[20] as well as an algorithm to link the inbound and outbound trajectories in every flyby, meeting the swingby constraints inherent to the patched-conics model. The implemented NELLS algorithm to connect two conic sections with an in-between flyby is based on STOUR C_3 matching algorithm, hereafter explained.

There are two constraints a swingby must fulfil, normally expressed in the flyby-body centered frame. Those constraints are Equations (1.1) (1.2). Equation (1.1) refers to the energy conservation law in the hyperbolic trajectory about the flyby body[21], simplified in the patched-conics model as a zero-size sphere of influence in the heliocentric perspective. Equation (1.2) express the minimum spacecraft flyby altitude shall be higher than a certain safe altitude with respect to the flyby body, that safe altitude value is set by mission designer.

$$c_{v_\infty} = v_\infty^+ - v_\infty^- = 0 \tag{1.1}$$

$$\begin{aligned} c_{flyby-altitude} &= r_{periapse} - \left(r_{body} + h_{safe}\right) \geqslant 0 \\ &= \frac{\mu_{body}}{{v_\infty^+}^2}\left[\frac{1}{\sin\left({}^{\delta}/_{2}\right)} - 1\right] - \left(r_{body} + h_{safe}\right) \geqslant 0 \end{aligned} \tag{1.2}$$

Being v_∞^- and v_∞^+ the inbound and outbound hyperbolic excess velocities, $r_{periapse}$ and r_{body} the distance of the periapse and flyby body surface to the center of the flyby body respectively, μ_{body} the standard gravitational parameter of the flyby body, h_{safe} the minimum allowed difference between $r_{periapse}$ and r_{body}

[20] See subsection 1.3.2

[21] The energy in hyperbolic trajectories is proportional to the square of the velocity as may be seen from the vis-viva equation. Therefore, the conservation law in hyperbolic trajectories may be expressed with Equation (1.1).

considered, and δ the turn angle of the swingby. NELLS Input files have the *maxC3miss* parameter to indicate the maximum allowable tolerance for constraint (1.1), since it is virtually impossible any computed trajectory can comply with it. Nonetheless, this does not suppose a problem since NELLS results constitute low-fidelity solutions to be optimized and the tolerance value is low, 0.01 km^2/s^2, as may be seen in appendix A.

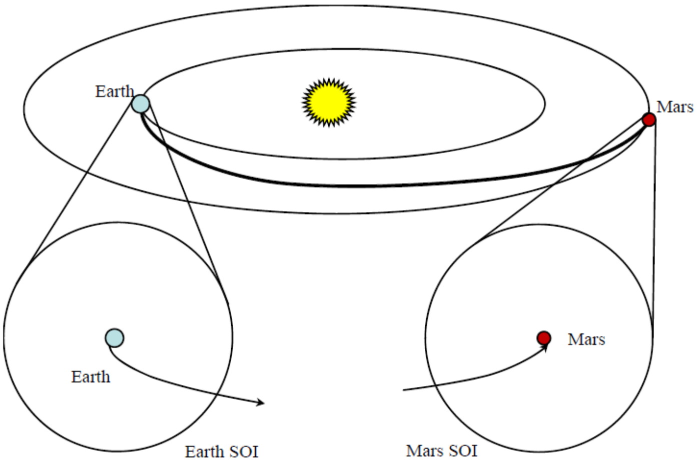


Figure 1.4: Concept of a patched-conic Earth-Mars transfer. The figure is not to scale. The transfer includes a hyperbolic Earth departure, an elliptical heliocentric interplanetary transfer, and a hyperbolic Mars arrival. The Sphere of Influence (SOI) is the region that depicts the extent of a central body gravitational dominance in the spacecraft motion with respect to other bodies. [54]

The C_3 matching algorithm is given in the following steps, which have been directly taken from [24].

1. Solve Lambert's problem for the current geometry and TOF: $\boldsymbol{r}_1(t_1)$, $\boldsymbol{r}_2(t_2)$ to get the outbound C_{30} for the flyby at $\boldsymbol{r}_1(t_1)$.

2. If Lambert solution is found, check if Equation (1.1) is satisfied, i.e. C_{30} - $C_{3target} = 0$, where $C_{3target}$ is the incoming C_3 at $\boldsymbol{r}_1(t_1)$.

3. If the C_3 magnitude matches, then proceed to step 7.

4. Advance the true anomaly grid. If the new TOF is greater than the maximum allowed flight time, then terminate C_3 matching.

5. Solve the new Lambert's problem to get the new outbound C_{31}

6. If the C_3 magnitude matches, then proceed to step 7 otherwise, if sgn(C_{30} - $C_{3target}$) $\neq$ sgn(C_{31} - $C_{3target}$), then a solution was missed and a bisection search is initiated to locate it. Otherwise, a match was not found, return to step 4.

7. Check if Equation (1.2) is satisfied. If the turn angle is valid, then a flyby can be used to connect the two Lambert legs.

8. Repeat steps 4-7.

1.3.2. Lambert's problem solver

It is crucial to have a robust algorithm to solve Lambert's problem. The reason is that algorithm is continuously called during grid search and, thus, will have a tremendous impact in terms of tool time performance.

The Lambert's problem consists of determining a trajectory arc, connecting two known position vectors for a specified transfer time. Conversely, Kepler's problem (which is an initial value problem) Lambert represents a boundary value problem. Besides, while Kepler's problem spawns an only solution, Lambert's problem yields $2N_P+1$ prograde orbits and $2N_R+1$ retrograde orbits, being N_P and N_R the maximum number allowed of revolutions for the prograde and retrogade case respectively [15]. An example of the prograde family of solutions is displayed in Figure 1.5.

As highlighted in [15], there exists a wide range of algorithms used to tackle the problem. In addition, a new method is introduced in that paper which outperformed the rest in terms of required computational time while having at the same time an accuracy similar to Gooding's method, one of the most accurate solvers available [26][35].

Another advantage which caused the election of this new algorithm as the Lambert's problem solver for NELLS is this method relies on an universal variable k based on the the classic Sundman transformation [16]. Consequently, there is no need to break down Lambert's problem in the three - two in practice - possible types of trajectories, parabola, ellipse or hyperbola.

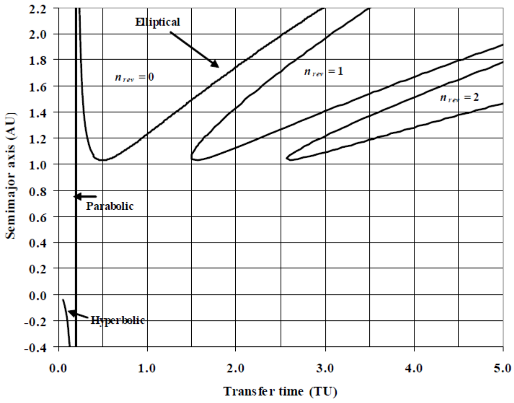


Figure 1.5: Lambert's problem prograde solutions for an Earth-Mars mission whose transfer angle is 75º [54]. The n_{rev} parameter stands for number of complete orbit revolutions made before reaching target body. TU represents time unit, being the required time to travel one radian in an orbit skimming the surface of a hypothetical spherical Earth of radius $R_{\oplus}$.

1.3.3. NELLS tree search algorithm

It should not seem strange that the MGA pathsolving problem may also be seen as a forest data structure [58][24] (Figure 1.6) where the maximum level of depth is already given for a fixed body sequence. In this standpoint, each tree of the forest represents a set of trajectories with the same launch date, the tree's root node, which a zero-depth level is assigned to. From every root, a net of nodes are sequentially expanded through depth levels during the trajectory search, where every depth level corresponds to another body from the provided body encounter sequence. Every node with a non-zero depth level has associated certain parameters like arrival date, that is, a parameter set enough to reconstruct the entire trajectory.

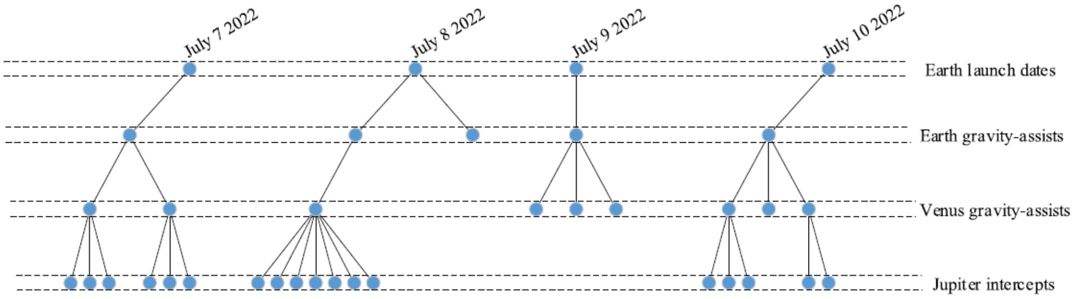


Figure 1.6: Example trajectory forest given for an E-EV-J gravity-assist sequence [24]

Due to the immensity of search space in trajectory optimization problems, a tree search algorithm is mandatory to implement in order to early detect and discard branches that are not going to yield acceptable results, making sure at the same time no potential solutions are overlooked in advance. This is often referred as the exploration-exploitation trade-off [27].

The only available NELLS tree search strategy is beam search because of its straightforwardness to be implemented, although other strategies like (for instance) Monte Carlo Tree Search (MCTS) [29] or the Lazy Race Tree Search (LRTS) [31] could be considered to be implemented in the future, due to the inherent greediness[22] related to the Beam Search algorithm [24].

Beam Search strategy consists of selecting the most promising nodes, discarding the rest of nodes, at each depth level according to some selection criteria until some termination criterion is met. Consequently, the Beam Search algorithm is useful RAM-limited problems where only a fraction of the search space can be explored, at expenses of risking to discard branches prematurely that could yield realistic solutions. The risk of missing out acceptable solutions directly depends on the implemented partial cost function.

There are currently two selection criteria in the NELLS Beam Search [24].

1. Only 20000 trajectories are branched at every new level (flyby) in every tree.
2. The terminal maximum flyby bend angle between sibling leaf nodes, since it can be regarded as a metric to assess how effective a particular transfer will be at accessing a larger number of potential Lambert transfers during the next branching phase.

The termination criteria are given in Appendix A.

1.3.4. State Transition Matrix linearization

The STM linearization used to rapidly identify trajectories to more than one KBO (Chapter 3) is computed as follows.

Position vector **r** at epoch t for a certain trajectory may be linearly approximated from a reference trajectory as Equation (1.3) depicts

$$\mathbf{r}\left[t,\mathbf{r}(t_0),\mathbf{v}(t_0)\right] = \mathbf{r}_{ref}\left[t,\mathbf{r}_{ref}(t_0),\mathbf{v}_{ref}(t_0)\right] + \left.\frac{\partial \mathbf{r}(t)}{\partial \mathbf{r}(t_0)}\right|_{ref}\left[\mathbf{r}(t_0)-\mathbf{r}_{ref}(t_0)\right] + \left.\frac{\partial \mathbf{r}(t)}{\partial \mathbf{v}(t_0)}\right|_{ref}\left[\mathbf{v}(t_0)-\mathbf{v}_{ref}(t_0)\right] + \dots \tag{1.3}$$

Where the characters in bold and the suspension points in Equation (1.3) stand for vectors and higher order terms respectively. Similarly, the velocity vector may be linearly approximated as follows

$$\mathbf{v}\left[t,\mathbf{r}(t_0),\mathbf{v}(t_0)\right] = \mathbf{v}_{ref}\left[t,\mathbf{r}_{ref}(t_0),\mathbf{v}_{ref}(t_0)\right] + \left.\frac{\partial \mathbf{v}(t)}{\partial \mathbf{r}(t_0)}\right|_{ref}\left[\mathbf{r}(t_0)-\mathbf{r}_{ref}(t_0)\right] + \left.\frac{\partial \mathbf{v}(t)}{\partial \mathbf{v}(t_0)}\right|_{ref}\left[\mathbf{v}(t_0)-\mathbf{v}_{ref}(t_0)\right] + \dots \tag{1.4}$$

Ignoring the higher order terms, these two expressions may be written more briefly as

$$\begin{aligned}\mathbf{r}(t) &= \mathbf{r}_{ref}(t) + \left.\frac{\partial \mathbf{r}}{\partial \mathbf{r}_0}\right|_{ref}\delta\mathbf{r}_0 + \left.\frac{\partial \mathbf{r}}{\partial \mathbf{v}_0}\right|_{ref}\delta\mathbf{v}_0\\ \mathbf{v}(t) &= \mathbf{v}_{ref}(t) + \left.\frac{\partial \mathbf{v}}{\partial \mathbf{r}_0}\right|_{ref}\delta\mathbf{r}_0 + \left.\frac{\partial \mathbf{v}}{\partial \mathbf{v}_0}\right|_{ref}\delta\mathbf{v}_0\end{aligned} \tag{1.5}$$

Or in a matrix form like Equation (1.6)

$$\begin{bmatrix}\delta\mathbf{r}\\ \delta\mathbf{v}\end{bmatrix} = \Phi(t,t_0)\begin{bmatrix}\delta\mathbf{r}_0\\ \delta\mathbf{v}_0\end{bmatrix}; \qquad \Phi(t,t_0) = \begin{bmatrix}\frac{\partial \mathbf{r}}{\partial \mathbf{r}_0} & \frac{\partial \mathbf{r}}{\partial \mathbf{v}_0}\\ \frac{\partial \mathbf{v}}{\partial \mathbf{r}_0} & \frac{\partial \mathbf{v}}{\partial \mathbf{v}_0}\end{bmatrix}_{ref} = \begin{pmatrix}\widetilde{\mathbf{R}} & \mathbf{R}\\ \widetilde{\mathbf{V}} & \mathbf{V}\end{pmatrix} \tag{1.6}$$

Being $\Phi(t,t_0)$ the State Transition Matrix (STM) whose components are the following ones [17]

$$\frac{\partial \mathbf{r}}{\partial \mathbf{r}_0} = \widetilde{\mathbf{R}} = \frac{r}{\mu}(\mathbf{v}-\mathbf{v}_0)(\mathbf{v}-\mathbf{v}_0)^T + \frac{1}{r^3}[r_0(1-F)\mathbf{r}\mathbf{r}_0{}^T + C\mathbf{v}\mathbf{r}_0{}^T] + F\mathbf{I} \tag{1.7}$$

[22] The fact of making the locally optimal choice at each stage.

$$\frac{\partial \mathbf{r}}{\partial \mathbf{v}_0} = \mathbf{R} = \frac{r_0}{\mu}(1-F)[(\mathbf{r}-\mathbf{r}_0)\mathbf{v}_0{}^T] \tag{1.8}$$

$$\frac{\partial \mathbf{v}}{\partial \mathbf{r}_0} = \widetilde{\mathbf{V}} = -\frac{1}{r_0{}^2}(\mathbf{v}-\mathbf{v}_0)\mathbf{r}_0{}^T - \frac{1}{r^2}\mathbf{r}(\mathbf{v}-\mathbf{v}_0)^T + F_t\Big(\mathbf{I} - \frac{1}{r^2}\mathbf{r}\mathbf{r}^T + \frac{1}{\mu r}(\mathbf{r}\mathbf{v}^T - \mathbf{v}\mathbf{r}^T)\mathbf{r}(\mathbf{v}-\mathbf{v}_0)^T\Big) - \frac{\mu C}{r^3 r_0{}^3}\mathbf{r}^3\mathbf{r}_0{}^3 \tag{1.9}$$

$$\frac{\partial \mathbf{v}}{\partial \mathbf{v}_0} = \mathbf{V} = \frac{r_0}{\mu}(\mathbf{v}-\mathbf{v}_0)(\mathbf{v}-\mathbf{v}_0)^T + \frac{1}{r^3}[r_0(1-F)\mathbf{r}\mathbf{r}_0{}^T - C\mathbf{r}\mathbf{v}_0{}^T] + G_t\mathbf{I} \tag{1.10}$$

Being F, C, F_t, and G_t The Lagrangian coefficients.

1.4. Report content and structure

The structure of this thesis is compounded of five chapters, concerned with the different aspects of the performed research. Besides this introductory part, Chapter 2 deals with the implementation first grid search type of the SA. Chapter 3 tackles the implementation of the second grid search type of the SA. An end-to-end SA trajectory example is discussed in Chapter 4. The SA validation is provided in Chapter 5. Finally, conclusions and future work recommendations are presented in Chapter 6.

2

First KBO search

As said in Chapter 1, the SA developed in this thesis to perform a rapid preliminary search for sequences of KBO encounters is compounded by two phases. The first phase consists of performing a patched-conics grid search for trajectories that encounter a first KBO of the JPL SSD. Then, a second grid search with a linear propagation of the STM is conducted to look for secondary KBOs in the vicinity of the trajectories found in the first phase.

This chapter addresses the first phase of the proposed SA, which was done by the NASA's GSFC tool called NELLS whose theoretical background was discussed in section 1.3. Firstly, a description on the NELLS workflow is provided. Secondly, NELLS is validated as a tool. Thirdly, a parameter tuning phase prior to the final run is designed and implemented. Finally, the results from running this first grid search are shown.

In consonance with section 1.2.3, the assumptions made in this chapter have been as follows.

1. Use of the patched-conics mathematical model.

2. Seek only ballistic trajectories with no revolutions.

3. Evaluate only three planet sequences (EJ, EJU, and EJN).

2.1. NELLS Workflow

The NELLS workflow is quite simple, an input file containing all the necessary information to run a scenario is opened in an executable file to start a simulation. The simulation results are given in a folder containing a log file with a rundown of the input file parameter settings, along with a csv file showing all (if any) found trajectories. An input file template is given in Appendix A.

There are two aspects on the input file settings worth highlighting for future applications.

1. The whole body sequence to follow shall be specified, including departure and destination bodies.

2. There are two body position discretization types, time and true anomaly. Both discretization types define the number of grid nodes per orbit[1] for any body. The true anomaly discretization was implemented for the sake of user-friendliness, as it is possible to check how fine an orbit grid is just with a rapid check of the number provided, without considering extra parameters like planet period when discretizing in time. The time discretization is usually used for launch whereas true anomaly is normally chosen for the rest of bodies.

[1] That is, for one revolution.

2.2. NELLS output validation

The NELLS validation was conducted by trying to approximate the Voyager 2 trajectory. The grid search parameter boundaries used in the simulation are illustrated in Tables 2.1 and 2.2. Those values are based on the actual Voyager 2 trajectory [36][37].

The values used to discretize the trajectories of the planets are given in Table 2.2. Those values were found through an iterative process until a similar trajectory to Voyager 2's was found by the grid search. The first tested values of 50, 100, and 500 for the number of grid nodes per orbit yielded no close solutions to the Voyager 2 trajectory. For those values, the attained trajectories had a difference in launch dates higher than 80 days with respect to Voyager 2. Finally, a close solution to the Voyager 2 trajectory was found when the number grid nodes per orbit was raised to 1000.

Parameter	Value
Earth Launch	April 7, 1976
Maximum Relative Launch C3 [km2/s2]	160
Minimum Relative Launch C3 [km2/s2]	10
Maximum Relative Arrival C3 [km2/s2]	400
Minimum Relative Arrival C3 [km2/s2]	0
Launch Window [Days]	500
Maximum TOF [yrs]	15
Number of Revolutions allowed per leg	0
Delta-V allowed	No

Table 2.1: NELLS grid search boundaries for Voyager 2 mission simulation.

Parameter	Value
Launch Step Size [Days]	1
Jupiter Orbit Number of Grid Nodes	1000
Saturn Orbit Number of Grid Nodes	1000
Uranus Orbit Number of Grid Nodes	1000
Neptune Orbit Number of Grid Nodes	1000

Table 2.2: Orbit discretization values for the body encounter sequence of the Voyager 2 mission.

The computed trajectory after having used the values of Tables 2.1 and 2.2 as inputs is compared with Voyager 2's in Table 2.3. As may be seen, NELLS yields a similar result to Voyager 2 trajectory.

Trajectory	Departure epoch	Jupiter flyby epoch	Saturn flyby	Uranus flyby epoch	Neptune flyby epoch
NELLS trajectory	August 17 1977	August 14 1979	October 24 1981	May 1 1986	January 6 1990
Voyager 2 trajectory	August 20 1977	July 9 1979	August 25 1981	January 24 1986	August 25 1989

Table 2.3: Orbit discretization values for the body encounter sequence of the Voyager 2 mission. Both trajectories are ballistic.

2.3. Systematic Approach to Define the Arrival- Date Grid

As said in subsection 1.2.1, parameter tuning supposes an indispensable intermediate step in order to obtain a fine grid before conducting simulations to tackle the actual problem. A coarse grid will likely produce a significant loss of solutions, whereas a very fine grid will sharply increase computational time.

The solution for that compromise in the STOUR-inspired tools which utilize the C_3 matching algorithm[2] is translated into finding the maximum step size in TOF that does not miss solutions in any (launch) C_3-TOF curve for a given scenario (EJ, EJU, and EJN in this thesis).

In order to maximize the step size, the flight time needs to somehow be related to the behavior of C_3-TOF curve. In this thesis, the approach taken is to characterize the local curvatures of each C_3-TOF curve[3] and then choosing the TOF step size based on those computed curvature values.

[2] Like NELLS.

[3] That is, to calculate the local curvature of each point for every computed C_3-TOF curve.

As said before, a coarse curve discretization could make the algorithm miss solutions in the step 6 of the aforementioned C_3 matching algorithm, being the worst case scenario when the target line is tangent with a curve's extremum (minimum or maximum) in a narrow basin . This may be appreciated more clearly in Figure 2.1.

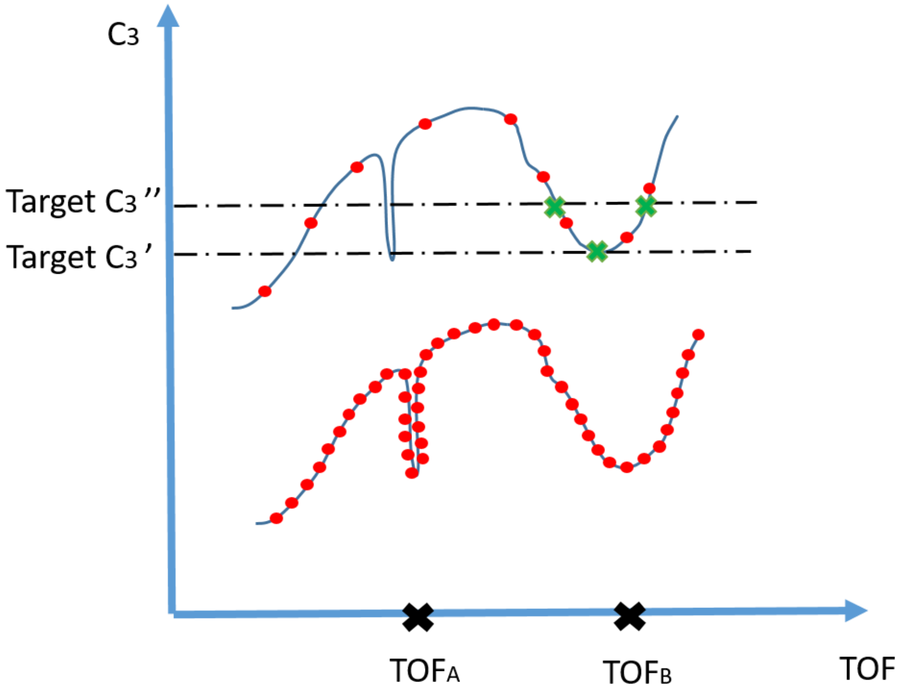


Figure 2.1: Explanation of the discretization influence in solution finding. The upper curve corresponds to a coarse discretization whereas the lower curve makes reference to a fine discretization.

Two effects may be appreciated in Figure 2.1. First, contrary to the fine grid, the coarse discretization will be unable to detect any solution around the TOF_A vicinity. Secondly, contrary to the Target C_3' case, the C_3 matching algorithm's step 6 will be able to detect the two solutions for Target C_3" in the coarse discretization. Those solutions are the intersection points depicted as green crosses in the vicinity of TOF_B, which can be detected since both solutions are in-between two grid points with higher and lower C_3 values allowing to apply the bisection technique.

Furthermore, it is important to note that (in ballistic flights) the C_3-TOF curve shape is affected by the relative position of the flyby and target bodies, which is defined in turn by time. Therefore, it is necessary to seek a figure of merit able to consider these shape changes in each scenario.

The curvature of a C_3-TOF curve is characterized in this work by its minimum local radius of curvature, which is by definition curve-geometry dependent as well as able to characterize the extrema-basin narrowness. By definition, the local radius of curvature (R) of a point P is the radius of the osculating circle at P [19], as Figure 2.2 depicts. R is the inverse of the local curvature (k) in absolute value whose expression, particularizing for C_3-TOF curves, is given by Equation (2.1) [20] [34].

$$R = \frac{1}{|k|} = \frac{|\ddot{C}_3|}{\left(1 + C_3^2\right)^{3/2}} \; ; \quad C_3 = C_3(TOF) \tag{2.1}$$

Therefore, for each C_3-TOF curve, the minimum local radius of curvature in absolute value (minR) is to be calculated, resulting in a close approximation of the actual value as, obviously, only finite precision may be attained with computers. If a C_3-TOF curve is discretized in TOF by its corresponding calculated minR value, then at least one grid point will be close[4] to the the local extrema with smallest R and, hence, only few solutions will be missed under a certain uncertainty threshold, as illustrated in Figure 2.3.

[4] Provided that the computed minR is small enough.

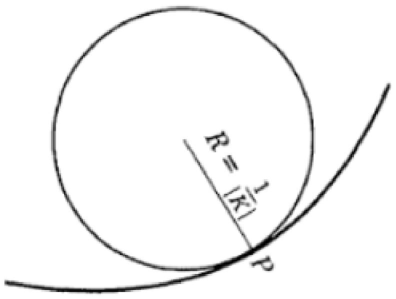


Figure 2.2: Local radius of curvature and osculating circle at point P for a certain curve [34].

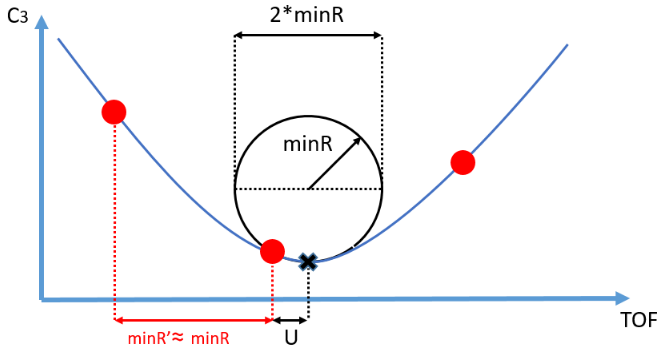


Figure 2.3: Conceptual close-up image around the local extrema with smallest R (minR) in C_3-TOF curve. Red points are the grid points separated by a ΔTOF value of minR'. minR parameter stands for the actual value, whereas minR' represents the computed (approximate) value. It is assumed minR'≈minR.

Where the U parameter in Figure 2.3 is the grid search uncertainty threshold and stands for the difference of the TOF values between the computed minR (minR') and the actual minR. The target C_3 values whose corresponding TOF is within an U range from the actual minimum are to be missed by the C_3 matching algorithm. It may be inferred the following constraint for U from the previous paragraph, such a constraint depicted in Equation (2.2).

$$U < minR' \approx minR \tag{2.2}$$

Therefore, the absolute minimum R value (absminR) from all C3-TOF curves stemming from a whole synodic period for a certain flyby-body target-body pair is the key parameter to be determined. Its value will be the orbit time step to discretize the target body orbit for any scenario.

Regarding NELLS, as said in section 2.1, the body encounter sequences (except the initial body) are gridded in true anomaly, expressed in number of grid nodes per orbit (revolution). Thus, it is necessary in this tool to translate orbit time step (i.e., absminR) values in grid nodes per orbit (g). Such transformation is done with the Equation (2.3).

$$g = \frac{P}{\Delta t} \tag{2.3}$$

Being P the flyby/destination body period and Δt its corresponding absminR value.

One may argue that applying this method is feasible for Jupiter, Uranus and Neptune which are the flyby bodies in the attempt to reach KBOs. Nonetheless the following two key questions arise.

1. Should this parameter tuning method also be applied for every KBO?
2. Do minR values for targets remain the same regardless the chosen flyby body?

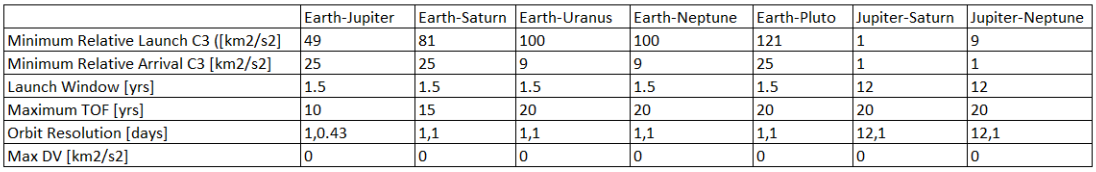
The answer for question one is no, the parameter tuning phase would turn out to be extremely time consuming otherwise. Instead, the parameter tuning is done for Earth-Jupiter, Earth-Saturn, Earth-Neptune, and Earth-Pluto (or simply EJ, ES, EN, and EP respectively) in order to look for a trend between orbit time step size and its semi-major axis that can be extrapolated to any KBO.

Question two evaluates whether the departure body influences what the minimum radius of curvature is for a given target body. For instance, whether or not the ES and JS scenarios yield similar values of absminR. This hypothesis in the thesis will be referred as the **equality condition**.

A negative answer for the equality condition would suppose a limitation for the grid search implementation as more body encounter sequences would have to be run in the parameter tuning phase. For instance, if a grand tour mission is aimed, the following scenarios could be considered: EJ, EJS, EJU, EJN, EJSU, EJSN. If equality condition is true, then parameter tuning phase could be done by assessing EJ, ES, EU, and EN scenarios. Nonetheless, if equality condition turns out to be false, then (on top of the EJ, ES, EU, and EN scenarios) JS, JU, JN, SU and SN must be evaluated as well to make sure the absminR values attained for Jupiter, Saturn, Uranus, and Neptune are the actual minima.

In order to corroborate the equality condition (aside from EJ, ES, EN, and EP) two scenarios JN and JS are to be run and compared with EN and ES respectively. These two scenarios were chosen to get a sample of a long relative distance (Jupiter to Neptune) and a shorter relative distance (Jupiter to Saturn).

The final NELLS input file parameters required to perform the parameter tuning phase on the aforementioned scenarios are shown in Figure 2.4.



	Earth-Jupiter	Earth-Saturn	Earth-Uranus	Earth-Neptune	Earth-Pluto	Jupiter-Saturn	Jupiter-Neptune
Minimum Relative Launch C3 ([km2/s2]	49	81	100	100	121	1	9
Minimum Relative Arrival C3 [km2/s2]	25	25	9	9	25	1	1
Launch Window [yrs]	1.5	1.5	1.5	1.5	1.5	12	12
Maximum TOF [yrs]	10	15	20	20	20	20	20
Orbit Resolution [days]	1,0.43	1,1	1,1	1,1	1,1	12,1	12,1
Max DV [km2/s2]	0	0	0	0	0	0	0

Figure 2.4: Scenario parameters for parameter tuning. Days refer to Earth days.

Where C_3 values were set with the aid of the Tisserand graph of Figure B.1. Launch window values correspond to the synodic periods attained with Equation (2.4) [54], conservatively rounded.

$$\frac{1}{T_{syn}} = \left| \frac{1}{T_d} - \frac{1}{T_t} \right| \tag{2.4}$$

Being T_{syn}, T_d, and T_t the synodic period, departure body period, and target body period respectively.

The maximum TOF values were based on previous interplanetary missions such as Voyager, Messenger, Cassini, Galileo, or New Horizons [21][23][28][45].

The 1-day value election for Earth orbit resolution was made after performing a trade study between computational time and orbit discretization degree, assessed in C_3-TOF curves. This value is conservative as not only it is far lower than the launch opportunity periods in interplanetary trajectories, which are related to synodic periods [28], but also it is in the order of magnitude of the daily launch windows that occur within each launch opportunity in Solar System exploration missions [4][33][49][50][51].

Likewise, the choice for orbit resolution for target bodies was made in an iterative way, consisting of (again) a compromise between computational time and degree of discretization attained in the C_3-TOF curves. The Jupiter time step in the EJ scenario is 0.43 days instead of 1 days since it is much closer to Earth than the rest of bodies and, thus, keeping the same values would lead to a much coarser discretization with respect to the other planets.

Finally, Jupiter's 12-day interval discretization was conducted in JS and JN scenarios in order to have a fair comparison with respect to ES and EJ since, in this case, both departure bodies will present the same number of grid points per orbit[5].

In order to validate the Figure 2.4's parameter settings, tens of TOF-launch C_3 and launch date-TOF plots for each scenario were generated. An example of each plot set is given in Figure 2.5 and Figure 2.6.

The generated plots yielded the following insights.

1. As depicted in Figure 2.5, the whole synodic period is covered with the chosen parameter settings.
2. As depicted in Figure 2.6, the chosen discretizations for the C_3-TOF curves were able to properly characterize them, even in the region close to the ridge wherein largest curvature values are usually attained. Therefore, this is a confirmation that Figure 2.4 parameters were adequate. The ridge in that image is produced when the heliocentric transfer angle approaches 180 degrees since the non-coplanar nature of the planets makes the transfer plane to go far out of the ecliptic, with its consequent rise in the required launch C_3 from Earth [54].

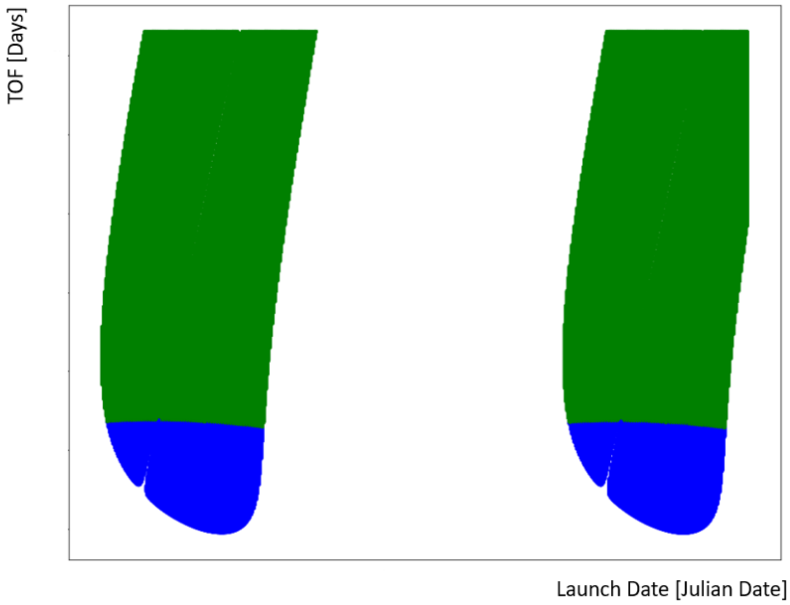


Figure 2.5: Porkchop plot for an Earth-Uranus scenario, for a 15 year launch date range with TOFs between 2.7 and 15 years. Blue points correspond to hyperbolic trajectories. Green points correspond to elliptical trajectories.

2.3.1. R parameter calculation

As Equation (2.1) depicts, it is necessary to compute the launch C_3 second derivative values in order to attain the local radii of curvature for any C_3-TOF curve. Two methods were used to get $\ddot{C}_3$, finite differences and curve fitting.

Finite differencing turned out to be unreliable to compute an accurate R value even after trying high order expressions, using five and seven points, to compute C_3 second derivatives. The reason for the inaccuracy is because of a slight point misalignment in the C_3-TOF curves made from NELLS results.

[5] As mentioned before in the thesis, Jupiter orbital period is approximately 12 years.

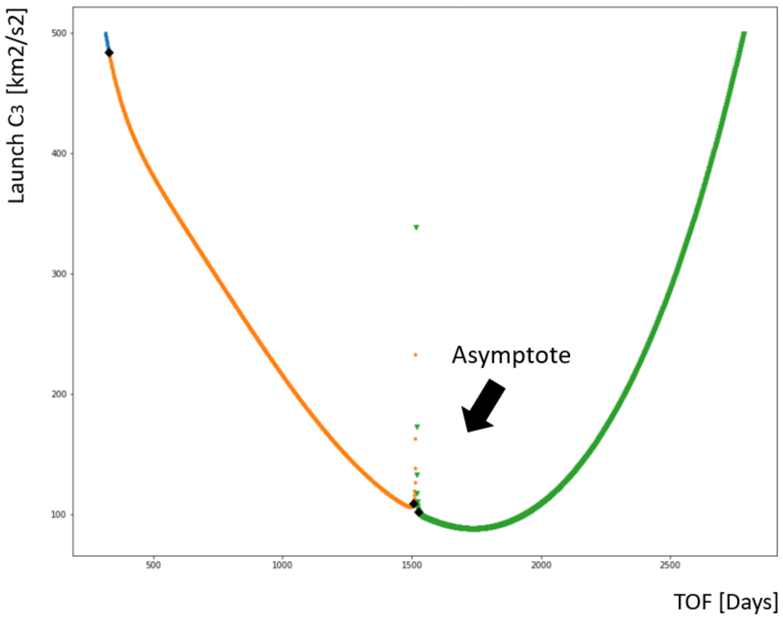


Figure 2.6: Launch C3-TOF plot for an Earth-Jupiter scenario with 2459395.5 as launch date in Julian Date (JD). Blue points correspond to hyperbolic trajectories, while orange and green points refer to elliptical orbits whose transfer angle is below and over 180 degrees respectively. Black points corresponds to the highest curvature values attained in each of the three trajectory types.

Due to the existence of these misalignments, a curve-fitting base method was employed. A 10-point, 5^{th}-order curve fitting was used, centered around the point where R is to be computed. At the boundaries, a 10-point, 5^{th}- order curve fitting was used instead.

Those numbers for the polynomial order the number of points were chosen after a thorough testing, where ten launch dates for every scenario were randomly picked. The maximum difference between a C_3 value of a real point and its value after curve fitting differed by less than 1% in all cases.

As expected, it was noticed during that testing that increasing the point number in curve fitting soared chances of skipping some curves with a small number of samples, as seen when plotting several launch dates. Regarding polynomial order, in turn, lower orders yielded coarse approximations specially in the asymptotic region (the zone where the transfer angle has values in the neighborhood of 180 degrees) wherein the minR values are usually found.

2.3.2. Output Data Sorting

In order to calculate all the local radii of curvature for every curve, for every launch date, and for every scenario; it is necessary first to classify all the different types of trajectories and compute their R values. This is due to two factors.

1. The existence of asymptotes in many C_3-TOF curves, which divide C_3-TOF curves into several pieces[6] that represent the different types of trajectories found. For instance, there are two asymptotes in Figure 2.6 that separate the C_3-TOF curve in three different pieces, each one associated with a different color.
2. As illustrated in subsection 2.3.1, a curve-fitting method is applied in the neighborhood of each point to calculate its curvature value. If two curve pieces from both sides of an asymptote are not evaluated independently when applying the curve-fitting method, then some points within the asymptote vicinity are evaluated with points of the other side of the asymptote. If two pieces are very close, this could give rise to artificial low R values. This undesired effect is depicted in Figure 2.7.

In Figure 2.7, if there is not a sorting algorithm for trajectory types, then the curve-fitting would be applied with those five[7] points of the red and blue pieces giving rise to a curvature that could be even higher than the values attained in the vicinity of TOF_A or TOF_B.

[6] Subsets of the C_3-TOF curves curves, subcurves.

[7] In that example used as illustration of the phenomenon.

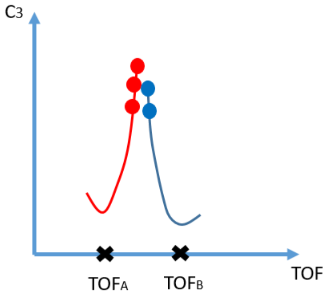


Figure 2.7: Undesired effect of the implemented curve-fitting algorithm (assuming 5-point fitting instead of the actual 10-point fitting) to calculate R values if no trajectory type characterization is applied. The points represent the curve samples attained after running simulations in NELLS with Figure 2.4 parameters, whereas the blue and red curve represent two pieces of a certain C_3-TOF curve.

This phenomenon was rapidly noticed and, for every scenario, all trajectories where classified in different pieces to calculate the Rs with the parameters shown in Table 2.4. The sorting process for a given scenario is as follows.

1. Classify by launch date.

2. For each launch date, sort curves using the prograde/retrograde condition yielding curve subsets like the ones depicted in Figure 1.5.

3. Classify by number of revolutions. This step is skipped in this thesis due to the fifth assumption made in subsection 1.2.3.

4. With the aid of the orbital specific energy, classify trajectory sets in hyperbolas, parabolas or ellipses.

5. Classify by high/low energy. For instance, as may be seen in Figure 1.5, the curve formed out of prograde zero-revolution elliptical trajectories has two solutions for a fixed TOF value with two different semi-major axes. Therefore, this filters divides curves like n_{rev}=0 into two branches, one with high semi-major axes and the other with the low values.

6. For any branch resulting from the previous step, separate between types I and II by checking the transfer angle.

Sorting Criterion	Key Parameter
Launch Date	Launch Date
Type of Conic	Specific Energy
Number of Revolutions	TOF/Orbit Period
Prograde/Retrograde	Cross Product Sign of **r**,**v**
High/Low Energy	k Parameter
Type I or II	Transfer Angle Over/Below 180 degrees

Table 2.4: Curve sorting criteria, k parameter is an universal variable defined in [15]

In addition, as may be seen in Figure 2.6, the asymptote existence induces very high curvature values within its vicinity. Adopting such values would require an enormous computational effort to perform a grid search for any mission to an outer planet. Asymptotes appear (as expected) in the neighborhood of transfer angles of 180 degrees, where the plane determined by Sun, target planet, and departure planet will (generally) be highly inclined to the ecliptic, as a result of the planetary inclination. The usual approach to reduce the overall delta-V is to resort to a mid-course maneuver as Figure 2.8 depicts.

The workaround to this issue has been to filter out solutions whose transfer angle is within 180 ± 2 degrees. That is, a small percentage of the generated C_3-TOF curves has being ignored to compute curvature values in this parameter tuning phase. This decision decreases maximum curvature values attained, getting a larger discretization time steps, and thus reducing the required computational to time perform the first KBO grid search.

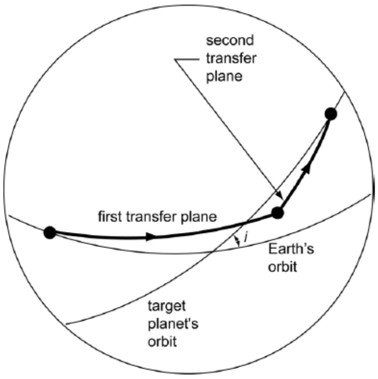


Figure 2.8: Mid-course maneuver giving rise to a broken-plane transfer trajectory for transfer angles close to 180º [56].

Applying this 2-degree constraint will lead to a loss of solutions in the grid search, in the vicinity of the asymptotic regions. Nonetheless, this has been assumed as a non-problematic consequence since it is expected that some solutions yielded by SA will be in the vicinity of those missed solutions (recall only a small part of the curves is ignored) and, hence, they can be found when the SA results are run as first guesses in a higher fidelity tool like EMTG. EMTG does not need a first guess to find solutions but its execution time is reduced when it is provided.

2.3.3. Parameter Tuning Results

Finally, simulations were run for the scenarios described in Figure 1.6, accounting for subsection 2.3.1 and 2.3.2 considerations. Results may be seen in Table 2.5.

	Jupiter	Saturn	Uranus	Neptune	Pluto
Orbit Period [Years]	11.86	29.45	84.02	164.80	247.74
Orbit Period [Days]	4333	10756	30687	60190	90487
Semi-major Axis [AU]	5.20	9.54	19.19	30.07	39.49
Step Size [Days]	7.04	39.58	79.02	371.02	1297.09
Grid Points Per Orbit	615.46	271.75	388.35	162.23	69.76
Log Step Size	0.84	1.59	1.90	2.57	3.11

Table 2.5: Results attained from the EJ, ES, EU, EN and EP scenarios in the parameter tuning phase.

It is important to note that the attained absminR values, the *step sizes [Days]* shown in Table 2.5, are at least one order of magnitude higher than its Table 2.4 counterparts[8] which were used as reference to tune the grid search parameters, before conducting the first KBO grid search in section 2.4. To compute a representative R value, the chosen resolution of the C_3-TOF plots must be computed with much smaller step sizes.

Therefore, having a higher order of magnitude implies that *step sizes [Days]* have been precisely determined and Table 2.4 values have proven to yield a sufficiently fine grid. If it was not the case, then lower values have to be assigned to Table 2.4 *Orbit Resolution [Days]* until their order of magnitude is lower than their resulting absminR values.

Based on the data displayed on Table 2.5, an empirical law (Figure 2.9) has been established in order to extrapolate and get a fine-tuned orbit time step to discretize KBOs orbits.

[8] The *Orbit Resolution [Days]* values.

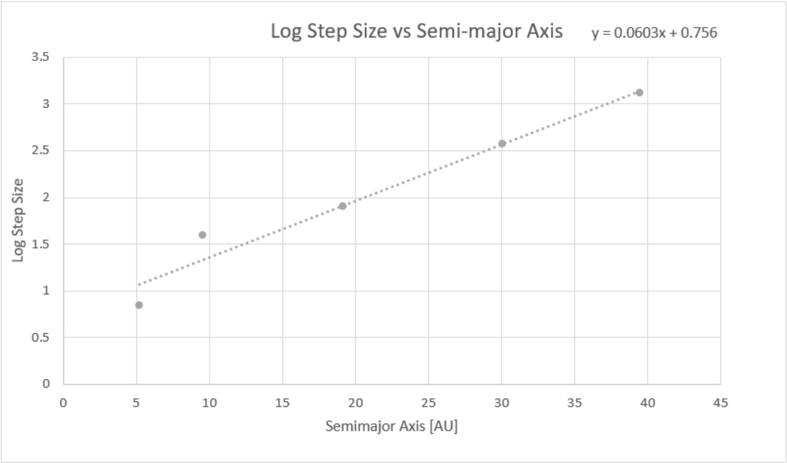


Figure 2.9: Body orbit semi-major axis vs logarithm of orbits' time step sizes resulting from the parameter tuning phase.

On the aforementioned **equality condition**, the computed time step sizes for Jupiter-Saturn and Jupiter-Neptune scenarios were 40.40 and 378.87 days respectively, quite close to their EJ and EN counterparts of Table 2.5. Therefore, equality condition is assumed to be correct and, therefore, no parameter tuning should be necessary for intermediate flybys.

Consequently, for every KBO, only its orbit semi-major axis must be known in order to determine its pertinent well-tuned number of grid nodes per orbit, the essential parameter in the NELLS input file. The grid node numbers can be computed from the time step sizes of Table 2.5 by applying Equation (2.3), being P the orbital period. As a final remark, all KBOs semi-major axes, along with all the other orbital parameters, may be pulled from JPL's SSD [7].

Noteworthy, running simulations with larger step sizes do not mean at all their results are worse than other fine-tuned simulations. The only consequence is the risk of missing solutions will be higher in simulations with coarser step sizes. An example is given in section 2.2, where Voyager 2 simulation was approximated in NELLS. As was stated there, coarser planet orbit discretizations yielded no results in the vicinity of the actual solution until the number of grid nodes per orbit increased to 1000, which is higher than the computed values of Table 2.5.

2.4. First Grid Search Results

The 1-day Earth launch/last flyby date time step along with time discretization values for the rest of bodies (whether intermediate or target bodies) given from Figure 2.9, were finally chosen to seek trajectories that reach a KBO in the JPL SSD. The attained time step values from Figure 2.9 may be converted to their corresponding true anomaly steps with Equation (2.3).

The rest of parameters required for all the NELLS input files to be run may be found in Table 2.6, observing the mission constraints introduced in subsection 1.2.2.

The way each scenario[9] is computed is straightforward. First, all the input files[10] are created from a template wherein only the number of grid nodes of the target KBO and simulation name ("missionName" in the input file) are modified, leading to around 2000 input files per scenario with (obviously) different time step sizes for each target KBO. Then, those generated NELLS input files are run in parallel to minimize computational time on NASA servers. Every time one scenario of the batch is done, another occupies its place, so that the number of active simulation remains the same.

The order of magnitude of the number of solutions found for the three considered scenarios were tens for Earth-Jupiter-KBO, thousands for Earth-Uranus-KBO and hundreds for Earth-Neptune-KBO, distributed in hundreds of csv files.

[9] EJ, EJU, and EJN.

[10] Each one associated with a different target KBO.

Parameter	Value
Earth Launch/Last Flyby Date	January 1, 2026
Maximum Relative Launch C3 $[km^2/s^2]$	500
Minimum Relative Launch C3 $[km^2/s^2]$	0
Maximum Relative Arrival C3 $[km^2/s^2]$	500
Minimum Relative Arrival C3 $[km^2/s^2]$	0
Launch Window [yrs]	15
Maximum TOF [yrs]	15
Number of Revolutions allowed per leg	0
Delta-V allowed	No

Table 2.6: NELLS input file parameters. The minimum relative arrival C_3 value is set to 500 km^2/s^2 in order to avoid discarding any possible trajectory.

A subset of the solutions found are illustrated in Figure 2.10 and 2.11 for the sake of clarity in the plots. All the ignored solutions lie in the vicinity[11] of the trajectories, depicted as points in those two images.

The solutions yielded from EJU and EJN scenarios, especially the ones with low C_3, could already be considered to have scientific interest due to two arguments.

1. They visit two relatively unknown planets like Uranus and Pluto. To date, they only have been visited once, that encounter was made by the Voyager 2 mission.
2. They already visit one KBO and, as was done in New Horizons, extra target KBOs could be discovered in the vicinity of these trajectories before and after the launch, provided that some propellant is stored in the spacecraft to maneuver towards those extra KBOs.

As an example, a couple of trajectories with low launch C_3 and moderate TOF compared to the rest of solutions found are shown in Figure 2.12 and 2.13 and summarized in Table 2.7. It may be seen a high degree of alignment between the outer planets and the KBO in each image.

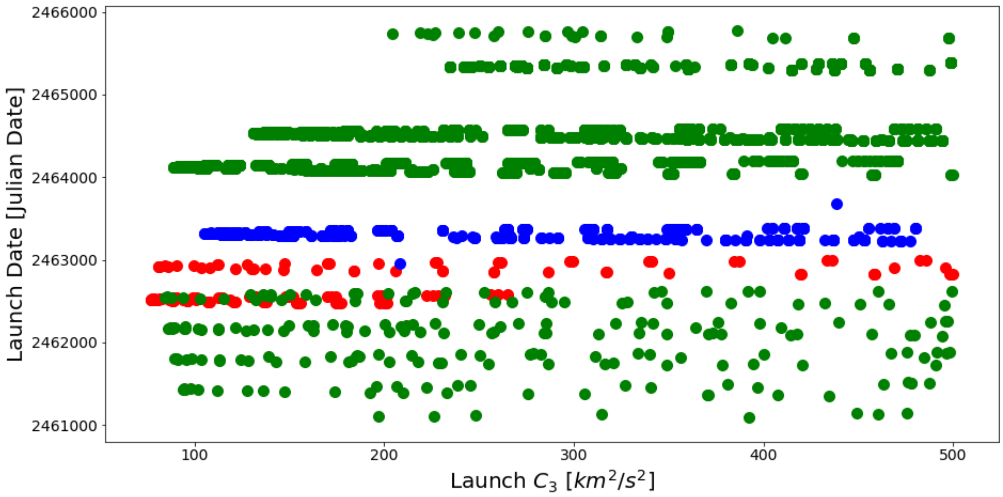


Figure 2.10: Launch C_3 versus launch date plot for the first grid search results. Red points stand for EJ trajectories, while green and blue points represent EJU and EJN scenarios respectively.

[11] The neighborhood boundaries are defined by launch dates of 10 days and TOFs of 1 year.

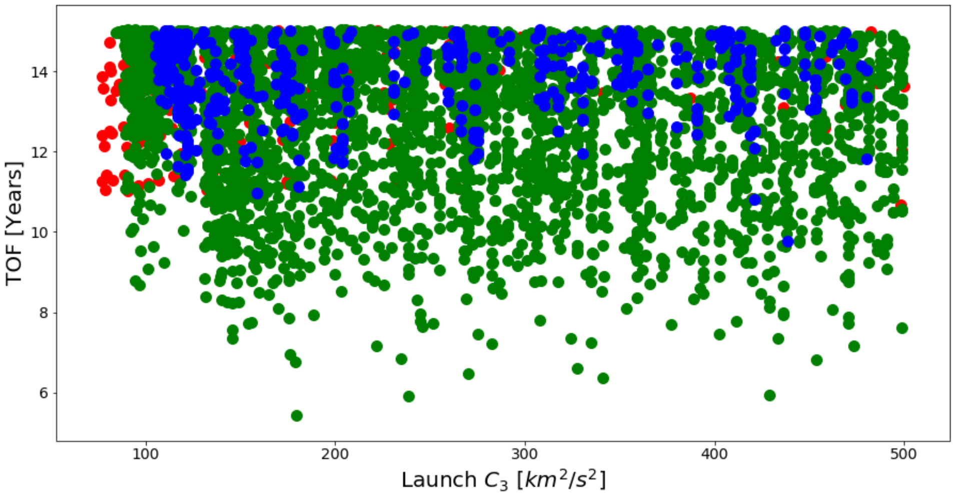


Figure 2.11: Launch C_3 versus TOF for the first grid search results. Red points stand for EJ trajectories, while green and blue points represent EJU and EJN scenarios respectively.

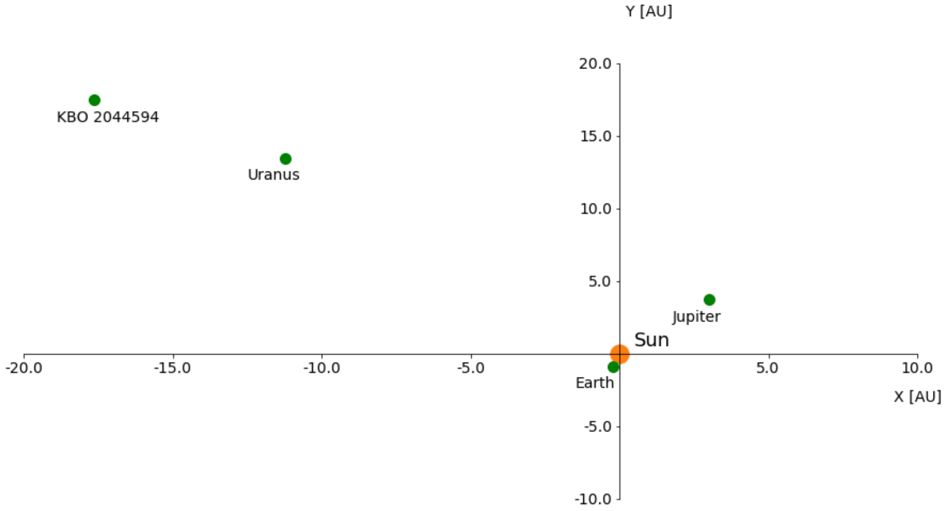


Figure 2.12: Representation of the planetary encounters of the chosen EJU trajectory, projected on XY plane in the International Celestial Reference Frame (ICRF) [6]. Launch C_3 92.17 km^2/s^2, TOF 10.02 years. Each point represent the encounter epoch between body and spacecraft.

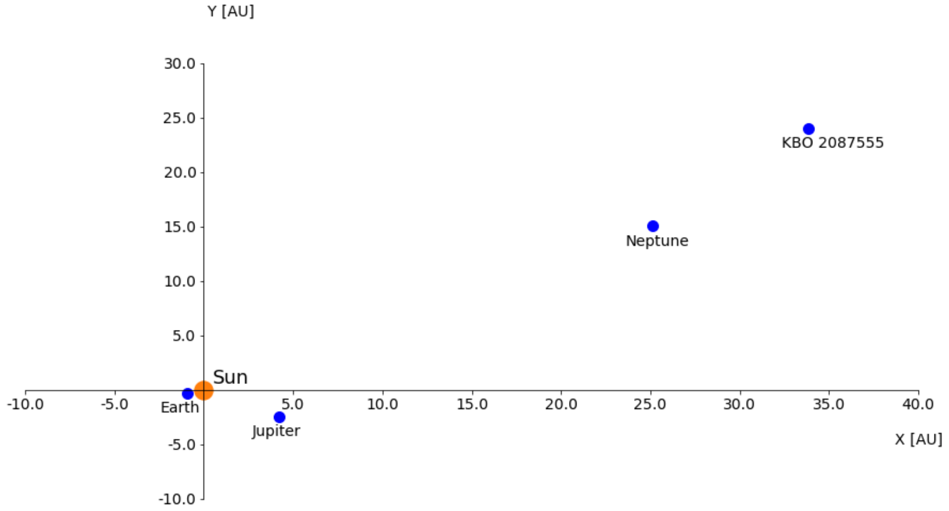


Figure 2.13: Representation of the planetary encounters of the chosen EJN trajectory, projected on XY plane in ICRF. Launch C_3 123.31 km^2/s^2, TOF 12.05 years. Each point represent the encounter epoch between body and spacecraft.

Trajectory	TOF	Launch C_3	Departure body	Epoch	Planet flyby	Epoch	Planet flyby	Epoch	Target body	Epoch
EJU	10.02	92.17	Earth	2464121.5	Jupiter	2464724.36	Uranus	2466692.67	2044594	2467780.87
EJN	12.05	123.31	Earth	2463331.5	Jupiter	2463777.46	Neptune	2466227.15	2087555	2467733.20

Table 2.7: Summary of the planetary planetary encounters conducted in two example trajectories shown on Figure 2.12 and 2.13. Launch C_3s and epochs are expressed in km^2/s^2 and Julian Dates respectively. TOF values are expressed in years. Numbers are rounded to the second decimal.

3
Secondary KBO Search

Chapter 2 addressed the first part of the SA, which consisted of searching for trajectories encountering (each of them) one KBO of the JPL SSD. This chapter deals with seeking trajectories which visit more KBOs from the trajectories presented in section 2.4, utilizing STM linear propagation along with a grid search in time. The trajectories attained in section 2.4 will be referred to as nominal trajectories (NT) in this chapter.

First, the motivation to opt for this new grid search type, distinct from the one described in Chapter 2, is explained in section 3.1. Then, the Second KBO Search Algorithm (SKSA) is discussed in section 3.2. SKSA results are provided in section 3.3. The SKSA generalization to seek n KBO encounters, the n^{th} KBO Search Algorithm (NKSA), is given in section 3.4. Finally, NKSA results are presented in section 3.5.

In consonance with section 1.2.3, the assumptions made in this chapter have been as follows.

1. Use of the patched-conics mathematical model for orbit propagation.
2. Maneuvers are calculated with the STM linearization.
3. Trajectories perform no revolutions.
4. KBOs do not perturb trajectories during flyby.

3.1. Motivation for using linear propagation

The fine grid search results of section 2.4 took around five weeks to compute. Therefore, it would require an enormous amount of computational time applying again that grid search to secondary KBO searches for every trajectory attained in section 2.4.

Since that length of time was impractical for the thesis time constraints, it was decided to approximate maneuver calculation for transfer orbits between two bodies by linearizing the Kepler's problem STM[1], instead of solving Lambert's problem and applying the C_3-matching algorithm, as it was done in the first grid search. This linearization made simulations took days instead of weeks at the expense of losing accuracy. Nonetheless, such accuracy loss is lower when KBOs are in the vicinity of NTs, which are also the ones requiring the lowest delta-V values to be visited.

Therefore, the SA workflow is a NELLS grid search plus SKSA, later updated to a NELLS grid search plus NKSA.

[1] See section 1.3.4 for more details.

3.2. Second KBO Search Algorithm

SKSA consists of several nested loops that go through every NT, after being parsed from the NELLS csv output files. For each NT, a Python script uses the following inputs to compute trajectories to reach extra KBOs.

1. Earth launch date, t_0.
2. The date (t_{lpf}) and its corresponding state vector for the spacecraft just after its last planet flyby.
3. The epoch (t_{flyby}) and state vector in the NT KBO encounter.

Besides, for every NELLS csv output file, another file is created wherein all the SKSA resulting trajectories are to be written.

The SKSA pseudocode may be seen in Figure 3.4 and its main blocks are explained down below.

1. Definition of the final time $t_f = t_0 + 20$ yrs. Consequently, the search period in this second phase is defined by $t_f - t_{lpf}$. The search period is discretized in 60-day steps, while a epoch dictionary and NT state-vector dictionary are created to store the calculated values for each step during a NT propagation phase. More details on how the state vector is propagated are given in subsection 3.2.2.

   There are two root state vectors from which every later state vector is propagated, the NT last-planet flyby and NT KBO-flyby epochs. This was done in order to reduce error propagation, taking advantage of the fact that orbit propagation is faster than solving Lambert's problem.

   Finally, a loop is applied for every KBO present in JPL's SSD and a KBOs dictionary is created to contain their state vectors for every epoch of the epoch dictionary.
2. Apply a filter to discard the subset of unreachable KBOs in order to save computational time, creating a reduced KBO dictionary with the reachable ones.
3. Second KBO linear propagation.
4. Write the updated trajectory, containing the second KBO encounter, in the aforementioned file created to store the attained solutions in its pertinent folder.

The 60-day step size was chosen after performing a trade study between computational time and accuracy. To this purpose, the true anomaly increments resulting from those step sizes were compared with the total true anomaly, corresponding to the total time of the period of interest[2]. The comparison was done by calculating their relative percentages. The vast majority of such increments were around 2-3 % of the total, although some outliers reached values even slightly higher than 10 %. This is due to some KBOs have high eccentricity values and being near perihelion when those percentages were calculated. Nonetheless, all of these cases are excluded from the solution space as they all violate both TOF and delta-V constraints, that is, section 1.2.2 mission constraints 2 and 3.

The KBO filter, second KBO linear propagation, and delta-V calculation blocks are further explained in the following three subsections.

3.2.1. KBO filter

A KBO filter was implemented in order to get rid of all the unreachable bodies before conducting the second KBO search, for every NT of the NELLS output csv files. The filter consists of creating an envelope of orbital element parameters for each NT that determines the reachable region, assuming conservative values for the parameters which dictate its shape.

In order to get such an envelope, a velocity vector representing a maneuver impulse is added to the velocity components of every state vector, present in the already created NT state vector dictionary. Such velocity vector has a fixed norm of 5 km/s whereas two parameter sweeps are done for both the elevation and azimuth angles, with 30-degree step sizes. The angles definition may be seen in Figure 3.1.

In other words, an impulse set consisting of a gridded 5-km/s spheroid has been applied for every epoch included in the state vector dictionary, in order to set the reachability boundaries from each NT.

[2] From 01-01-2026 to 01-01-2026 plus the 15 years of the launch period plus the 20 years of the maximum TOF allowed.

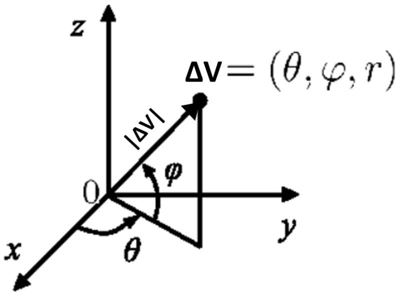


Figure 3.1: Spherical coordinate system used [12]. Angles ϕ and θ are the elevation and azimuth angles respectively, the radial coordinate is denoted by $|\Delta v|$.

Those envelope boundaries in each NT are determined by the maximum distance to Sun or r_{max} and the ϕ_{max}, ϕ_{min}, θ_{max}, and θ_{min} values. In accordance with subsection 2.3.3, there is no need to calculate r_{min} as its value is already known. Such a value is 10 AU approximately.

These parameter values are attained by calculating the two "piercing points" of the new trajectories with the Kuiper Belt lower boundary (i.e., the two points whose Sun distance is 30 AU) as well as their aphelions. Therefore, for n trajectories resulting from applying impulsive maneuvers for every epoch in a certain NT, 3n positions are to be computed.

KBO evaluation is done once the envelope has been determined. Position vectors are retrieved for every KBO from the JPL SSD for every epoch defined in the epoch dictionary created in SKSA step 1. A KBO is considered reachable if has at least one of its retrieved locations within the boundaries of the reachable region, this selection criterion is illustrated in Figure 3.2. Finally, a reduced KBO dictionary is created from the original KBO dictionary, only containing the reachable KBOs.

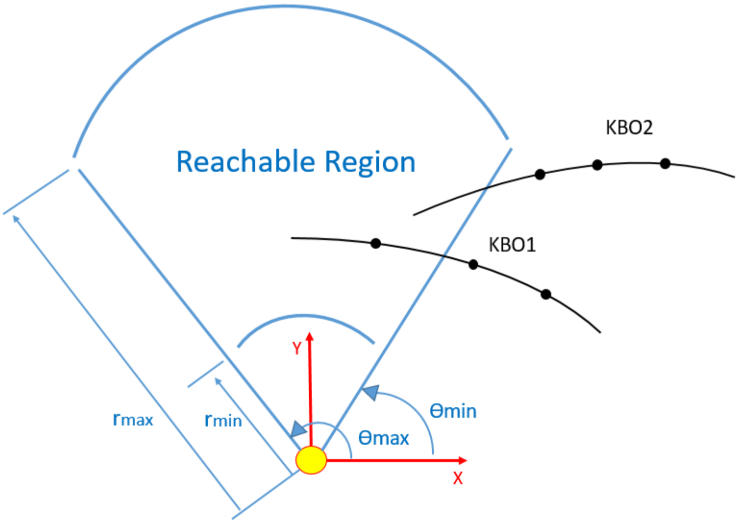


Figure 3.2: KBO filter concept, assuming a 2-dimensional problem, where three epochs of two KBOs have been retrieved from the JPL SSD. KBO1 has one location within the reachable area and, thus, it is assumed to be reachable in SKSA. Conversely, none of the KBO2 locations are within such an area and, therefore, it is discarded for the SKSA. Reachable area boundaries are defined by rmin, rmax, θmax, and θmin in this case.

3.2.2. Second KBO linear propagation

Once a subset of KBOs has been determined as potentially reachable after applying the KBO filter, it is time to perform the second KBO grid search. This second grid search is conducted by looping over every body of the reduced KBO dictionary.

The procedure is outlined in Figure 3.3, where LPF represents the last planet flyby and NKBO, that is, the KBO that NT encounters ("nominal KBO"). FMi and SMi stand for the first maneuver and second maneuver in the i^{th} NT deviation.

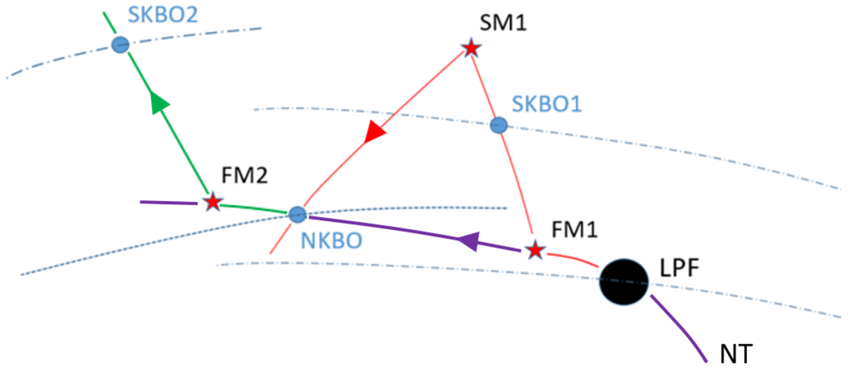


Figure 3.3: Second KBO grid search concept. Purple trajectory stands for the NT. There are two possible deviations of the NT to encounter the second KBO, represented in red (first deviation) and green (second deviation).

Where the numbers in the events refer to the two possible deviations, 1 for red and 2 for green. $SKBO_1$ stands for a second KBO encountered in the first deviation, while $SKBO_2$ corresponds to a second KBO visited in the second deviation. The arrows and the stars indicate the motion direction and maneuver points respectively. It is important to note that the NT overlaps with the green deviation between the nominal KBO arrival and FM2 events in the second deviation.

As may be seen in Figure 3.3, there exist two different alternatives (i.e., the two deviations from the NT) when looking for a second KBO. These two alternatives differ on whether the spacecraft visits the second KBO before or after encountering the nominal KBO.

The second KBO search process is straightforward. It consists of storing the (if any) found trajectories coming from two independent nested loops, corresponding to the two aforementioned deviations. In both cases the outer loop iteration variable t_1 which stands for an epoch of the NT epoch dictionary, used to pull its corresponding state vector from the NT state vector dictionary. The iteration variable t_1 is identified as the FMi[3] epoch, wherein a maneuver to reach a second KBO takes place, more details on delta-V calculation may be found in subsection 3.2.3.

The FM1 epoch search ranges from the last planet's flyby to the nominal KBO arrival where (from that epoch onward) the spacecraft follows a different trajectory than the NT's, depicted in red in Figure 3.3. In other words, FM1 can occur from LPF to the nominal KBO arrival epoch. Likewise, regarding the second possible deviation, FM2 can be applied from the nominal KBO flyby to t_f , that is, when the search period $t_f - t_{lpf}$ ends.

Once this is done, then an inner loop takes place[4] defining another time (t_2) as the iteration variable, ranging from FMi to the last epoch in the NT epoch dictionary (t_f) in case of dealing with the second deviation or the nominal KBO flyby epoch, t_{flyby}, in the first deviation. For every inner-loop iteration, the state vector of the aiming secondary KBO is pulled from the reduced KBO dictionary, using t_2 as the dictionary key. Finally, the required delta-V is calculated to get there from FMi.

The first deviation has another second nested loop with a third time variable t_3 for iteration, ranging from SKBO1 to the t_{flyby} epoch. The required delta-V to be applied at t_3 epoch to encounter the nominal KBO is

[3] Notation, it represents both deviation types.

[4] In both outer loops.

calculated. In order to calculate delta-V, the spacecraft state vector is propagated from SKBO1[5] to t_3, depicted as SM1 in the SKSA algorithm developed in this thesis (Figure 3.3).

All these time variables used the loop variables are summarized in Table 3.1.

# Deviation type	Time variable	Nested loop depth	Meaning	Epoch range
1	t_1	1	First maneuver epoch	From LPF to Nominal KBO arrival
	t_2	2	Second KBO encounter	From FM_1 (t_1) to Nominal KBO arrival
	t_3	3	Second maneuver epoch	From SM_1 (t_2) to Nominal KBO arrival
2	t_1	1	First maneuver epoch	From Nominal KBO arrival to t_f
	t_2	2	Second KBO encounter	From t_1 to t_f

Table 3.1: Summary of the iteration variables (times) used in the nested loops to search for a second KBO encounter. The one-level depth represents the outer loop.

Noteworthy, a delta-V filter is applied after every impulse calculated to prune out impossible candidate trajectories if delta-V value exceeds the very conservative 5-km/s threshold, in order to save computational time. Finally, canonical units were used in both orbit propagation and delta-V calculation with the aim of decreasing required computational time and, at the same time, avoiding substantial accuracy losses. Those canonical units are illustrated in Table 3.2.

Canonical Unit	Symbol	Value
Distance	DU	1 Astronomical Unit (AU)
Time	TU	1 Earth period (365.25 days)
Velocity	VU	DU/TU
Standard gravitational parameter	μ_U	DU^3/TU^2

Table 3.2: Canonical Units implemented in SKSA and NKSA.

As may be seen, a core part in this algorithm is orbit propagation (the Kepler's problem) and delta-V calculation subroutines. Kepler's problem subroutine was attained from a JPL code, whose author was Stefano Campagnola. Such code is a validated implementation of the "Algorithm 8" present in [54], where an universal variable χ is used in order to avoid accuracy losses when orbits are nearly parabolic. Besides, it also uses canonical units to simplify some expressions involving the standard gravitational parameter and, hence, speeding up the process, something essential in this subroutine as it is constantly called in SKSA.

Delta-V calculation using linear approximations is given in the following subsection.

3.2.3. Delta-V calculation

Subsection 3.2.1 deals with how reachable KBOs were filtered in SKSA, subsection 3.2.2 explained SKSA framework, this section addresses how maneuvers were calculated.

In accordance with Figure 3.3, there are three families of epochs to apply maneuvers (FM1, SM1, and FM2) whose delta-V values must be found out. This problem is equal to solving three Lambert's problem families FM1-SKBO1, SM1-nominal KBO, and FM2-SKBO2. Nonetheless, as said before, solving Lambert's problem is computationally expensive when applied millions of times. Instead, a trajectory linearization around the nominal trajectory has been used to overcome this handicap at the expense of sacrificing accuracy.

Nonetheless, this should not suppose a very problematic assumption a priori for two reasons. Firstly, only reachable (and thus near) bodies are to be considered and, secondly, this calculation is only intended to be a first approximation with the aim of finding KBO encounter sequences for a later pathsolving phase in EMTG. Therefore, potential secondary KBO flybys are assumed to be linear perturbations of their pertinent nominal trajectories. Such linear perturbations may be calculated using Equation (1.6), whose matrix components are given in Equations (1.7)(1.8)(1.9)(1.10).

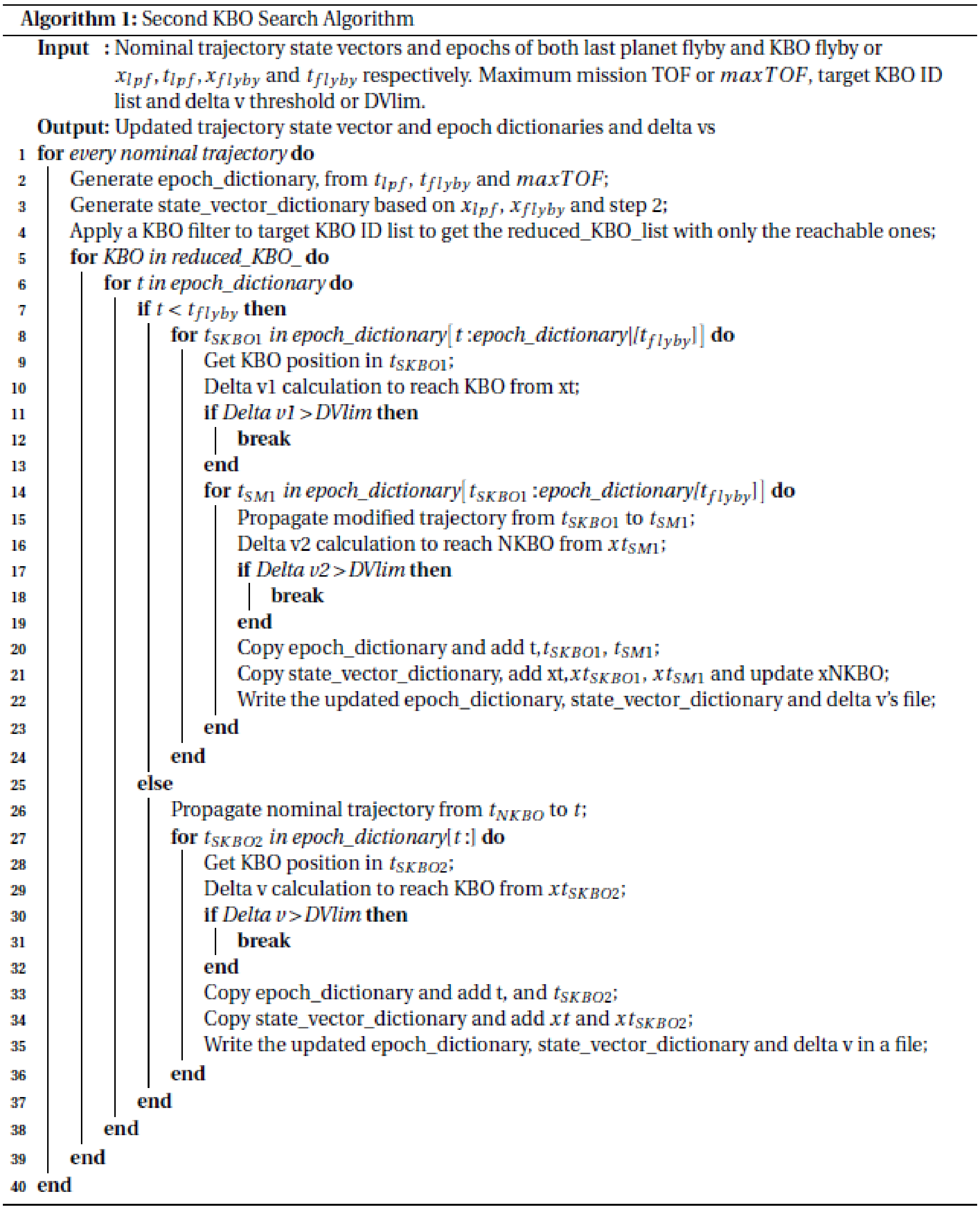
Consequently, the equation system to solve is (3.1) wherein the necessary delta-V to be applied in order to reach a certain target at epoch t is equal to the $\delta\mathbf{v_0}$ array. Looking at equation (3.1) one could reckon a priori

[5] i.e., the second KBO encounter epoch

that there are more unknowns than equations, twelve versus six. Nevertheless, it is important to notice $\boldsymbol{\delta r_0}$ is a 3x1 zero-array since impulses are assumed as instantaneous and, hence, occurring in NT at epoch t_0.

In addition, $\delta\mathbf{r}$ (the difference between the reference and the new nominal trajectory at epoch t) is known since both target and NT positions in t are known as well. Therefore the only unknown 3x1 vectors are $\boldsymbol{\delta v_0}$ and $\delta\mathbf{v}$, making system (3.1) a determined linear system and, thus, solvable.

$$\begin{pmatrix}\delta\mathbf{r}\\ \delta\mathbf{v}\end{pmatrix} = \Phi(t,t_0)\begin{pmatrix}\delta\mathbf{r_0}\\ \delta\mathbf{v_0}\end{pmatrix} \Rightarrow \begin{pmatrix}\delta\mathbf{r}\\ \delta\mathbf{v}\end{pmatrix} = \begin{pmatrix}\tilde{\mathbf{R}} & \mathbf{R}\\ \tilde{\mathbf{V}} & \mathbf{V}\end{pmatrix}\begin{pmatrix}\emptyset\\ \delta\mathbf{v_0}\end{pmatrix} \Rightarrow \delta\mathbf{v_0} = \mathbf{R}^{-1}\delta\mathbf{r} \Rightarrow \delta\mathbf{v} = \mathbf{V}\delta\mathbf{v_0} \tag{3.1}$$



Algorithm 1: Second KBO Search Algorithm

```
   Input  : Nominal trajectory state vectors and epochs of both last planet flyby and KBO flyby or
            $x_{lpf}, t_{lpf}, x_{flyby}$ and $t_{flyby}$ respectively. Maximum mission TOF or $maxTOF$, target KBO ID
            list and delta v threshold or DVlim.
   Output: Updated trajectory state vector and epoch dictionaries and delta vs
1  for every nominal trajectory do
2      Generate epoch_dictionary, from $t_{lpf}$, $t_{flyby}$ and $maxTOF$;
3      Generate state_vector_dictionary based on $x_{lpf}$, $x_{flyby}$ and step 2;
4      Apply a KBO filter to target KBO ID list to get the reduced_KBO_list with only the reachable ones;
5      for KBO in reduced_KBO_ do
6          for t in epoch_dictionary do
7              if $t < t_{flyby}$ then
8                  for $t_{SKBO1}$ in epoch_dictionary[ t :epoch_dictionary|[$t_{flyby}$]] do
9                      Get KBO position in $t_{SKBO1}$;
10                     Delta v1 calculation to reach KBO from xt;
11                     if Delta v1 > DVlim then
12                         break
13                     end
14                     for $t_{SM1}$ in epoch_dictionary[ $t_{SKBO1}$ :epoch_dictionary[$t_{flyby}$]] do
15                         Propagate modified trajectory from $t_{SKBO1}$ to $t_{SM1}$;
16                         Delta v2 calculation to reach NKBO from $xt_{SM1}$;
17                         if Delta v2 > DVlim then
18                             break
19                         end
20                         Copy epoch_dictionary and add t,$t_{SKBO1}$, $t_{SM1}$;
21                         Copy state_vector_dictionary, add xt,$xt_{SKBO1}$, $xt_{SM1}$ and update xNKBO;
22                         Write the updated epoch_dictionary, state_vector_dictionary and delta v's file;
23                     end
24                 end
25             else
26                 Propagate nominal trajectory from $t_{NKBO}$ to $t$;
27                 for $t_{SKBO2}$ in epoch_dictionary[ t :] do
28                     Get KBO position in $t_{SKBO2}$;
29                     Delta v calculation to reach KBO from $xt_{SKBO2}$;
30                     if Delta v > DVlim then
31                         break
32                     end
33                     Copy epoch_dictionary and add t, and $t_{SKBO2}$;
34                     Copy state_vector_dictionary and add $xt$ and $xt_{SKBO2}$;
35                     Write the updated epoch_dictionary, state_vector_dictionary and delta v in a file;
36                 end
37             end
38         end
39     end
40 end
```

Figure 3.4: Secondary KBO Search Algorithm pseudocode.

3.3. Trajectory selection for further searches

Before running SKSA, a TOF and launch date filter was applied to the NELLS results. This filter was applied due to a large amount of similar trajectories yielded by NELLS, which sharply increased computational time. Besides, It was also noticed that similar trajectories led to similar results which are indistinguishable regarding the standpoint of serving as first guesses for EMTG.

The filter works as follows.

For a given set of already filtered trajectories, another trajectory ("candidate trajectory") can be added if any of these two conditions is met.

1. The launch date difference between the candidate trajectory and any member of the set shall be equal or larger than 10 days.
2. In case condition 1 is not fulfilled, the TOF difference between the candidate trajectory and the one(s) in the set whose launch date difference with respect to the candidate trajectory is less than 10 days shall be equal or larger than 1 year.

Obviously the first trajectory to iterate will go directly to that set as it will be void at that moment.

The results after performing a second KBO search for all the trajectories attained from NELLS using SKSA are displayed in Figure 3.5 and Figure 3.6.

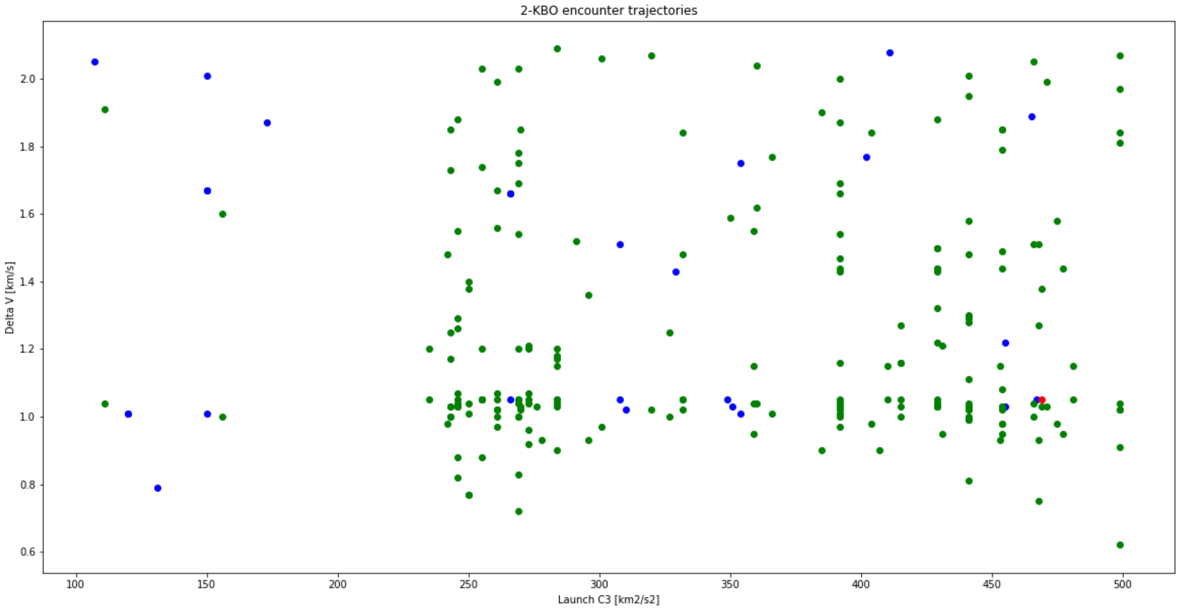


Figure 3.5: Launch C3 vs total delta-V for all trajectories which encounter two KBOs. Colors refer to the last planet flyby, corresponding red to Jupiter, green to Uranus and blue to Neptune respectively.

The number of trajectories generated is around a hundred, much lower than the previous phase which consisted of thousands of orbits. All of them share as a common feature a total delta-V necessary to reach the second KBO below 2.2 km/s. Nonetheless, there exists a broad range of options when it comes to Earth launch/last flyby of the pump-up phase[6] energy.

Due to time constraints, only a subset of Figure 3.5 trajectories will be chosen to apply a further search for a third KBO. That subset consists of the orbits whose last Earth departure C3 is below 200 km^2/s^2 as they are the most promising candidates to generate[7] trajectories where no pump-up phase is needed. A rundown of these selected trajectories are shown in Table 3.3.

[6] The early gravity-assist sequence prior to these trajectories discussed before in this thesis. See subsection 1.2.2 for further details.

[7] After a later optimization stage.

Nonetheless, trajectories with higher Earth departure C_3 and short TOFs should also be evaluated in future work to check whether a pump-up maneuver may be applied beforehand like (for instance) a trajectory whose TOF is 10.53 years keeping at the same time a low delta-V value, compared to the rest of solutions. A rundown of this trajectory is provided in Table 3.4.

It is important to note that the delta-V values shown in Tables 3.3 and 3.4 may represent a poor estimation of the actual values due to the linear approximation of the STM. The exception is for low delta-V values where the linear approximation will be very close to the real solution. Therefore, delta-V values should be considered to be more of a metric of reachability to the KBO sequence, rather than a literal DV value.

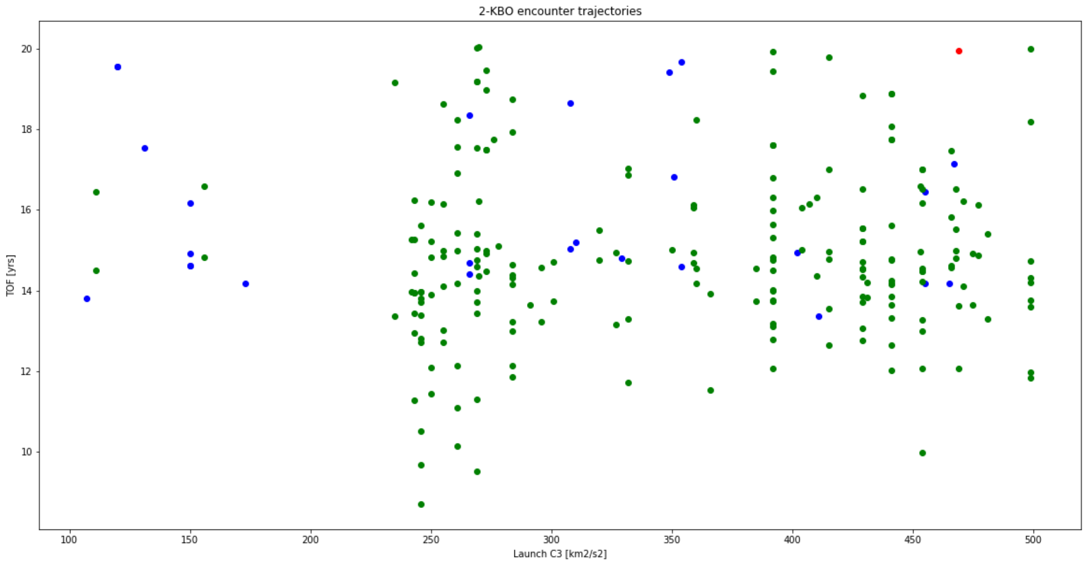


Figure 3.6: Launch C_3 vs total Time of Flight in years for all trajectories which encounter two KBOs. Colors refer to the last planet flyby, corresponding red to Jupiter, green to Uranus and blue to Neptune respectively.

Launch Date	Launch C3	Total Δv	1^{st} KBO	2^{nd} KBO	Trajectory ID	FMi	SM1
2463321.5	106.97	1.99	JN	2385446	2138537	2466832.3	2467912.3
2463351.5	172.71	1.87	JN	3160774	2160256	2466781.2	2467861.2
2463346.5	149.9	1.97	JN	2427581	2385446	2466976.6	2468236.6
2463346.5	149.9	1.67	JN	3046117	3031899	2466976.6	2468236.6
2463346.5	149.9	1.67	JN	3046117	3031899	2466976.6	2468236.6
2464156.5	156.36	1.6	JU	2015788	3027724	2466619.9	2467999.9
2464141.5	111.21	1.91	JU	2456826	3272185	2466728.3	2468768.3
2463346.5	149.9	1.01	JN	2385446	2066652	2468836.6	N/A
2463336.5	120.38	1.01	JN	3031899	2120178	2468681.4	N/A
2463336.5	120.38	1.01	JN	3031899	2120178	2468681.4	N/A
2463301.5	130.74	0.79	JN	3755650	2138537	2468809.4	N/A
2464156.5	156.36	1.0	JU	3027724	2019255	2469619.9	N/A
2464141.5	111.21	1.04	JU	3272185	2143991	2469488.3	N/A

Table 3.3: Outline of the chosen trajectories for a third KBO search. Units: JD, km2/s2, km/s.

Launch Date	Launch C3	Total Δv	Planet Sequence	1^{st} KBO	2^{nd} KBO	FM2
2465341.5	246.07	0.88	JU	2119473	2015820	2468886.6

Table 3.4: Example of trajectory that could be studied in future work. Units: JD, km2/s2, km/s.

3.4. n^{th} KBO search

The SKSA is only able to perform a grid search for a second KBO. Therefore, it should be generalized in order to be able to seek n KBO encounters from the NELLS outputs. As anticipated at the beginning of the chapter, this new algorithm is called the n^{th} KBO Search Algorithm or just NKSA, and its workflow is illustrated in Figure 3.8. The NKSA updates with respect to the SKSA are explained in the following paragraphs.

The NKSA seeks a potential n+1 KBO flyby from a trajectory already containing n KBO encounters which (analogously to SKSA) constitutes the NT in this case. That is, the NELLS output files parsed as inputs in SKSA are generalized as files containing n KBO encounters[8].

Besides, as may be seen in Figure 3.4, the SKSA uses an important event (hereafter referred to as **milestone**) to split the second KBO search in two different NT deviations, the green and red NT deviations of Figure 3.3. Such event is the nominal KBO flyby epoch, since for $t > t_{flyby}$ there is obviously no need to revisit any point of the nominal trajectory once a delta-V has been applied. These two phases are reflected in the two parts of the **if** statement, present in line 7 of Figure 3.4.

There will be more than one of these milestones in a more complex NT, like multiple maneuver points and KBO encounters, as Figure 3.3 shows. Consequently, the NKSA used for a nominal trajectory containing n milestones consists of applying the if-statement's[9] first block in SKSA a n number of times instead of one, as was done in SKSA the. Nonetheless, just as the SKSA, the if-statement's second block has only to be run once.

Another important difference with respect to the SKSA is the addition of a third maneuver in the if-statement's first block. A second maneuver is calculated to reach the next milestone both in SKSA and NKSA. In Figure 3.3 such an event is the nominal KBO flyby. Nonetheless, in the NKSA, a third maneuver is determined right when reaching the next important event to steer the spacecraft back in the nominal path, in order to make sure it encounters the remaining KBOs of the NT. This additional maneuver is TM1 in Figure 3.7, wherein the NKSA concept is summarized.

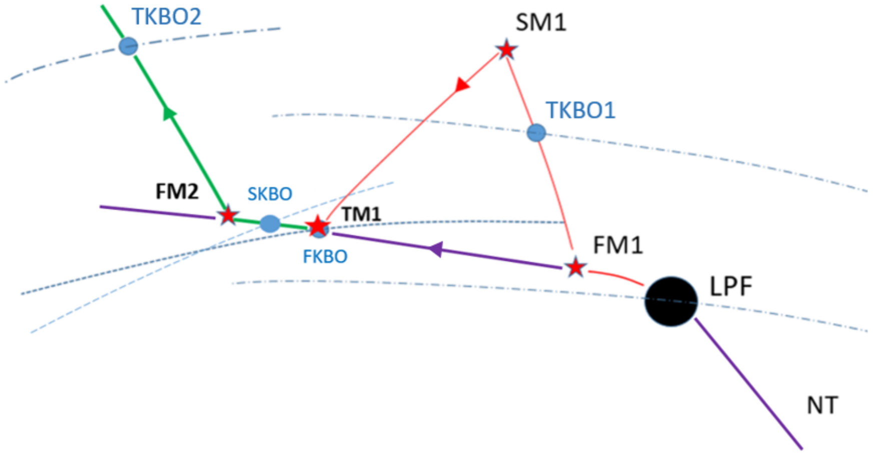


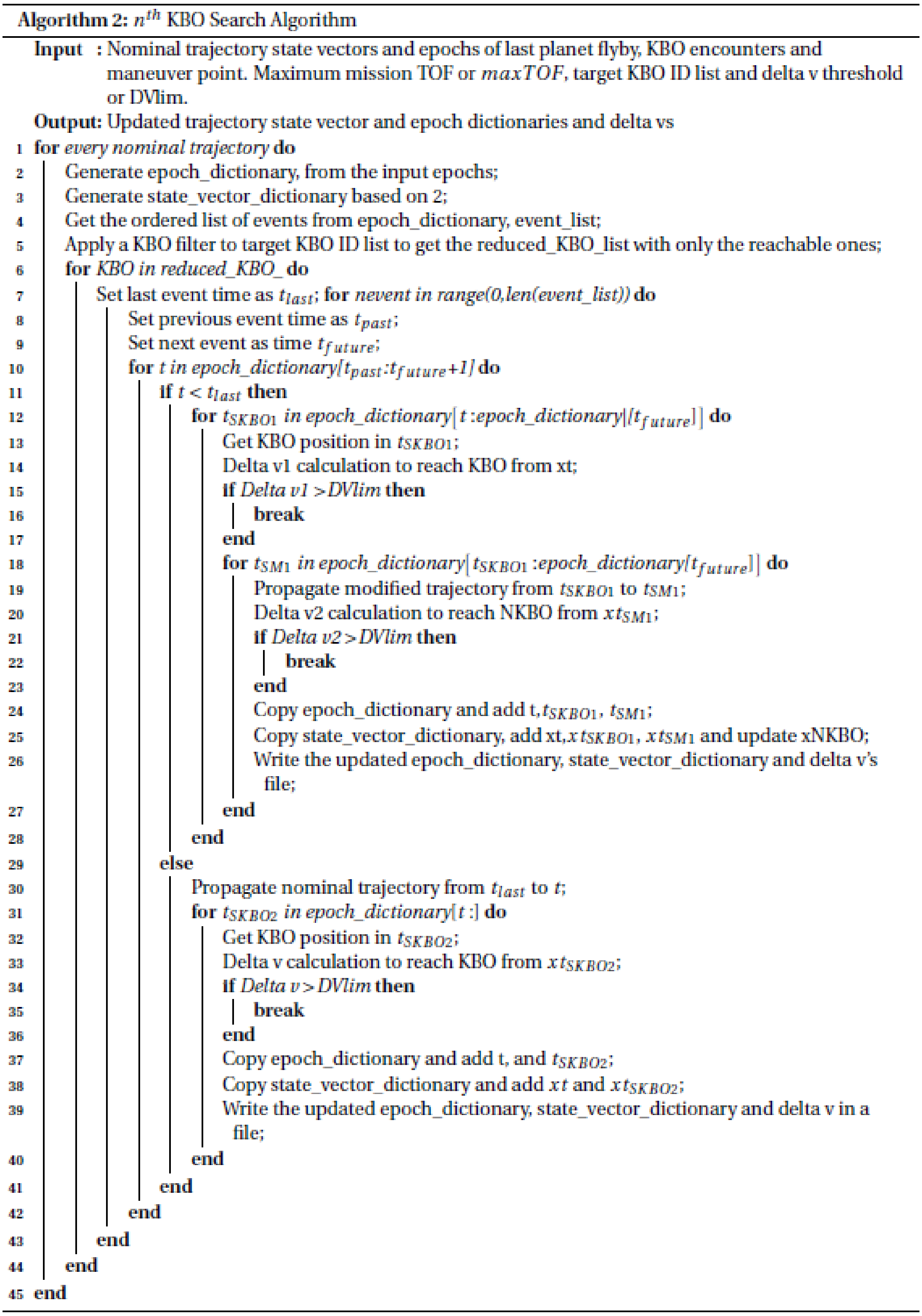
Figure 3.7: n^{th} KBO grid search concept. Purple trajectory stands for the NT encountering n-1 KBOs. There are two possible deviations of the NT (red the first deviation and green the second one) to encounter a n^{th} KBO, either TKBO1 in the first deviation or TKBO2 in the second deviation. In this case n equals 3.

Where the numbers in the events refer to the two possible deviations, 1 for red and 2 for green. The FKBO and SKBO are the two KBOs that the NT encounters. The arrows and the stars indicate the motion direction and maneuver points respectively. Noteworthy, the NT and overlaps with the green deviation between TM1 and FM2 in the second deviation. FMi and SMi stand for the first maneuver and second maneuver in the i^{th}[10] NT deviation. TM1 represents a third maneuver in the first deviation used to steer spacecraft back in the NT. the TM1 position coincides with FKBO.

[8] The SKSA output files contain two KBOs, therefore n=2 in this case

[9] The if statement present in line 7 of Figure 3.4

[10] Notation, it represents both deviation types.



```
Algorithm 2: $n^{th}$ KBO Search Algorithm
   Input  : Nominal trajectory state vectors and epochs of last planet flyby, KBO encounters and
            maneuver point. Maximum mission TOF or $maxTOF$, target KBO ID list and delta v threshold
            or DVlim.
   Output: Updated trajectory state vector and epoch dictionaries and delta vs
 1 for every nominal trajectory do
 2     Generate epoch_dictionary, from the input epochs;
 3     Generate state_vector_dictionary based on 2;
 4     Get the ordered list of events from epoch_dictionary, event_list;
 5     Apply a KBO filter to target KBO ID list to get the reduced_KBO_list with only the reachable ones;
 6     for KBO in reduced_KBO_ do
 7         Set last event time as $t_{last}$; for nevent in range(0,len(event_list)) do
 8             Set previous event time as $t_{past}$;
 9             Set next event as time $t_{future}$;
10             for t in epoch_dictionary[$t_{past}$:$t_{future}$+1] do
11                 if $t < t_{last}$ then
12                     for $t_{SKBO1}$ in epoch_dictionary[ t :epoch_dictionary[[$t_{future}$]] do
13                         Get KBO position in $t_{SKBO1}$;
14                         Delta v1 calculation to reach KBO from xt;
15                         if Delta v1 >DVlim then
16                             break
17                         end
18                         for $t_{SM1}$ in epoch_dictionary[ $t_{SKBO1}$ :epoch_dictionary[$t_{future}$]] do
19                             Propagate modified trajectory from $t_{SKBO1}$ to $t_{SM1}$;
20                             Delta v2 calculation to reach NKBO from $xt_{SM1}$;
21                             if Delta v2 > DVlim then
22                                 break
23                             end
24                             Copy epoch_dictionary and add t,$t_{SKBO1}$, $t_{SM1}$;
25                             Copy state_vector_dictionary, add xt,$xt_{SKBO1}$, $xt_{SM1}$ and update xNKBO;
26                             Write the updated epoch_dictionary, state_vector_dictionary and delta v's
                                file;
27                         end
28                     end
29                 else
30                     Propagate nominal trajectory from $t_{last}$ to $t$;
31                     for $t_{SKBO2}$ in epoch_dictionary[t :] do
32                         Get KBO position in $t_{SKBO2}$;
33                         Delta v calculation to reach KBO from $xt_{SKBO2}$;
34                         if Delta v > DVlim then
35                             break
36                         end
37                         Copy epoch_dictionary and add t, and $t_{SKBO2}$;
38                         Copy state_vector_dictionary and add $xt$ and $xt_{SKBO2}$;
39                         Write the updated epoch_dictionary, state_vector_dictionary and delta v in a
                            file;
40                     end
41                 end
42             end
43         end
44     end
45 end
```

Figure 3.8: n^{th} KBO Search Algorithm pseudocode.

3.5. Third KBO Search Results

The results after running NKSA, using SKSA trajectories (section 3.3) as input, are shown in Figure 3.9 and 3.10.

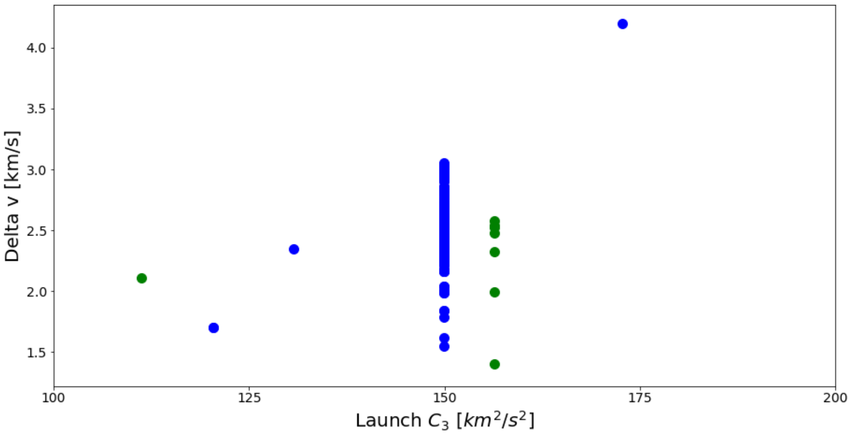


Figure 3.9: Launch C3 vs total delta-V for all trajectories which encounter three KBOs. Colors refer to the last planet flyby, corresponding green to Uranus and blue to Neptune respectively.

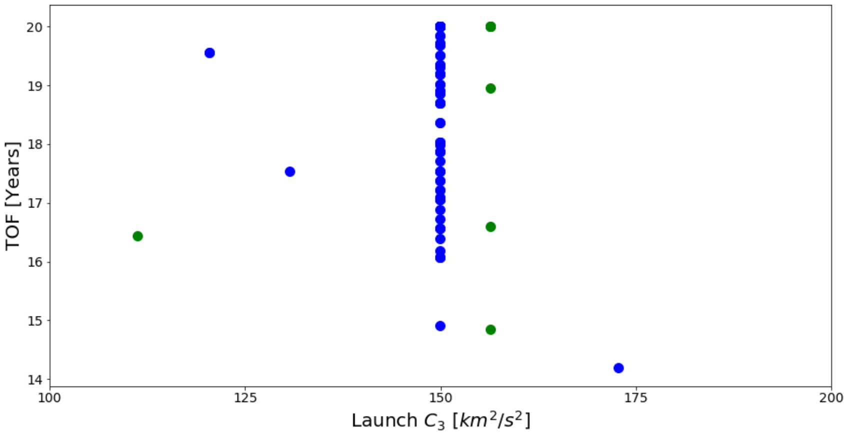


Figure 3.10: Launch C_3 vs TOF for all trajectories encountering three KBOs. Colors refer to the last planet flyby, corresponding green to Uranus and blue to Neptune respectively.

Again, the TOF and launch date filter explained in section 3.3 is applied here right before conducting the third KBO search.

The inclusion of a third KBO encounter significantly increases total delta-V with respect to the attained two-KBO trajectories, shown in Figure 3.5, since more maneuvers have to be applied. It is also worth noting that the launch C_3 of the majority of the attained trajectories is very close to New Horizon's 158 km^2/s^2 [59].

A fourth KBO was conducted exceeding its results the 2 km/s threshold for delta-V imposed in the mission constraints, in the section 1.2.2.

Consequently, the trajectory with the minimum delta-V was selected to perform a proof of concept with EMTG, see Chapter 4. The trajectory details are displayed in Table 3.5 and 3.6.

Launch Date	Launch C3	Total Δv	Jupiter Flyby	Uranus Flyby	1^{st} Maneuver	3755735 Flyby	2^{nd} Maneuver	3^{rd} Maneuver	3027724 Flyby	4^{th} Maneuver	2019255 Flyby
2464156.5	156.36	1.40	2464720.90	2466619.99	2467879.98	2468839.98	2468899.98	2469576.76	2469576.76	2469619.98	2470219.98

Table 3.5: Chosen three-KBO trajectory to run in EMTG. Units: JD, km2/s2, km/s.

Maneuver number	X	Y	Z
1	-2762720645	2618574361	1424368596
2	-3527389871	3038108348	1901295415
3	-4167919628	3307363487	2058777002
4	-4202698556	3324058576	2074429347

Table 3.6: Maneuver positions in ICRF of the chosen three-KBO trajectory to run in EMTG. Units: km.

If a first guess is provided, EMTG requires for instance the approximate values of the trajectory's main epochs[11], spacecraft wet mass, and spacecraft dry mass [55]. In case only wet mass approximate value is known, dry mass can automatically be derived from the well-know Rocket Equation, assuming the delta-V value of Table 3.5 and assigning a characteristic value to Isp. Regarding wet mass approximation, it can be assumed as the maximum mass that can be carried by the chosen launcher.

Due to the high launch C_3 required, a Falcon Heavy (Expendable) with STAR 48-BV kick motor was used for the proof of concept. The performance curve for this vehicle is provided in Figure 3.11 and 3.12. As may be seen in Figure 3.12, it is essential to include the kick-motor in Falcon Heavy in order to be able to launch in high C_3 values.

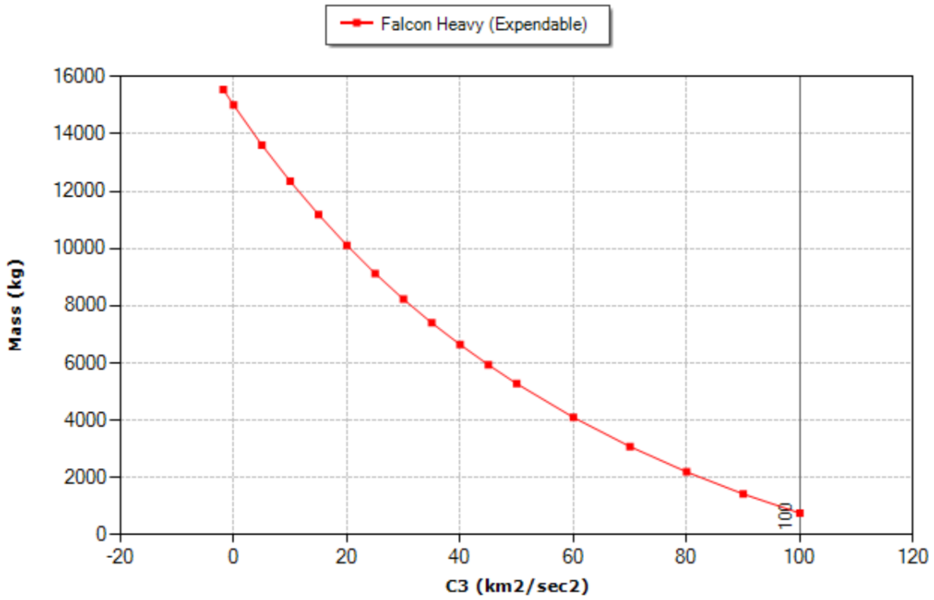


Figure 3.11: Falcon Heavy (Expendable) performance [1].

Consequently, the launcher (maximum) final capability (or, equivalently in this first guess, the spacecraft wet mass) is calculated by combining the STAR 48BV capabilities[12] with the Falcon Heavy (Expendable) nominal performance. The result is a modified curve based on the Falcon Heavy Expendable's present in Figure 3.11, to be evaluated in the launch C_3 value of 156.36 km^2/s^2.

In order to produce such a modified curve of launcher capability versus launch C_3 the following steps were done.

1. The launcher nominal capability (i.e., the curve of Figure 3.11) is interpolated with a fifth-order polynomial in order to compute more point and to expand the curve beyond the launch C_3 threshold of 100

[11] The epochs needed to reconstruct the trajectory

[12] All the parameters related to this kick motor required in the thesis may be found in [3]

km^2/s^2. Once this is done, the subsequent steps are done for every generated point of the resulting interpolated curve.

2. Approximate the nominal launcher periapse velocity (V_p) with Equation (3.2), being ${v_\infty}^2$ the points of the interpolated curve of step 1 and V_{esc} the escape velocity at the STAR 48BV burn altitude of 6500 km.

3. Subtract the adapter mass from the launch vehicle delivered mass, defined by the generated points of step 1, the result is the 'mass after separation' or m_{sep}, see Equation (3.3).

4. Calculate the delta-V given by STAR 48BV (ΔV_{KM}) using the well-known Rocket Equation [54], Equation (3.4), using the STAR 48BV information given in [3]. The initial mass is m_{sep} whereas the final mass is m_{sep} minus the upper stage propellant mass or simply 'mass after burnout' (m_{bout}).

5. Calculate the updated launch vehicle delivered mass (m_{LV1}) by subtracting the upper stage (STAR 48BV) dry mass ($m_{drymass}$) to m_{bout}, see Equation (3.5).

6. Update the new periapse velocity (V_{p1}) by adding ΔV_{KM} to V_p.

7. Calculate the updated launch C_3 (${v_\infty}^2$) at the STAR 48BV burn altitude using Equation (3.2).

$${V_p}^2 = {V_{esc}}^2 + {v_\infty}^2 \tag{3.2}$$

$$m_{sep} = m_{LV} - m_{adapter} \tag{3.3}$$

$$\Delta V_{KM} = g_0 I_{sp} LN \frac{m_{sep}}{m_{bout}} \tag{3.4}$$

$$m_{LV1} = m_{bout} - m_{drymass} \tag{3.5}$$

Therefore, the spacecraft wet mass (which is assumed to equal m_{LV1}) is attained by subtracting both the adapter mass, the upper stage propellant mass, and the upper stage dry mass to launcher nominal capability. Besides, the velocity increments with respect to the nominal delivered mass-C_3 curve are due to the upper stage propellant mass and may be found applying the Rocket Equation, taking into account ${v_\infty}^2 = C_3$ and assuming $I_{sp} = 292.1s$ [8]. The modified Falcon Heavy expendable- C_3 curve along with is interpolated formula is present in Figure 3.12.

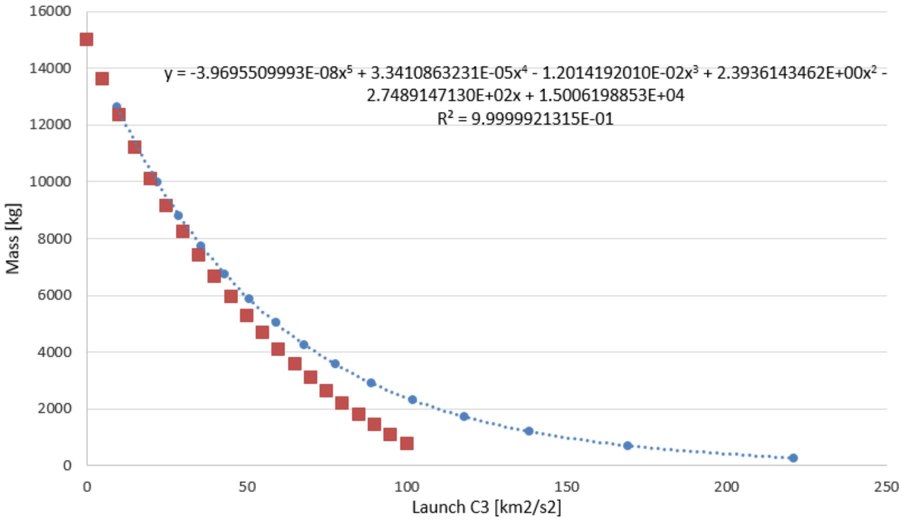


Figure 3.12: Falcon Heavy expendable original (red) and modified (blue) curves. Polynomial formula refers to the blue curve.

The attained spacecraft wet and dry mass values for a launch C_3 value of 156.36 km^2/s^2 were 878 and 442 kilograms respectively. Such dry mass turned out to be pretty close to New Horizons' 401 kg. Consequently, the necessary inputs for the proof of concept are provided in Tables 3.5, 3.6 and 3.7.

Parameter	Value
Spacecraft Wet Mass [kg]	878
Spacecraft Dry Mass [kg]	442
Spacecraft Propellant Isp [s]	228

Table 3.7: Spacecraft parameters for the chosen three-KBO trajectory to run in EMTG.

4

End-to-end Trajectory Example

As said at the beginning of the thesis, the SA purpose is to generate candidate KBO paths (i.e., pathfinding) for a higher fidelity tool (EMTG in this case) to be used as the pathsolving subproblem solver. This chapter discusses the EMTG results attained from the SA trajectory, illustrated in Tables 3.5, 3.6, and 3.7.

The SA trajectory was used as first guess in EMTG yielding the trajectory depicted in Figure 4.1 and summarized in Table 4.1[1]. As may be seen, despite the KBO alignment, the resulting total delta-V from EMTG turned out to be one order of magnitude larger than the trajectory yielded by SA. This means the SA trajectory turned out to be a poor delta-V estimator in this case.

As explained in section 3.4, the NKSA relies on successive linear approximations to calculate delta-V. Therefore, the SA is only able to provide reliable[2] delta-V approximations for KBOs close enough to the attained NELLS output trajectories, being delta-V a measure of the reliability of a given SA output trajectory. Consequently, two conclusions may be drawn for sufficiently low delta-V values in a SA output trajectory, high reliability and its vicinity with respect to the NT.

High attained delta-V values indicate that the influence in the neglected high order terms may actually be relevant and, hence, the computed delta-V values can not be trusted. The study on determining which delta-V values represent accurate approximations with respect to EMTG outputs is definitely an important research topic, although beyond the scope of this thesis.

This SA weakness is aggravated by the fact NKSA relies on successive linear approximations to seek trajectories encountering n KBOs. Consequently, the more KBOs are added in the search the less accurate the total delta-V computation *tend* to be. The reason is, as said in the previous chapter, n-KBO trajectories are based on trajectories which encounter n-1 KBOs and, hence, the error tends to accumulate.

Delta-V error *tends* to increase but not *must* increase with the number of encountered KBOs in a certain trajectory. The reason is, for a trajectory with n+1 KBO flybys, errors associated with having excluded higher order terms in the n SA iterations could (in theory) cancel among themselves to some extent. In any case, the takeaway is the reliability of SA results decreases with the number of KBOs encountered by the trajectory and high delta-V values.

In consequence, two simulations were performed having the NT[3] of the chosen SA trajectory as reference this time, being its delta-V of 1 km/s. That is, another simulation in EMTG was conducted for a 2-KBO trajectory. The results are shown in Figures 4.2 and 4.3 and Tables 4.2 and 4.3.

As expected, the delta-V error is much lower than the 3-KBO case, being the same order of magnitude of the computed value this time. This is an evidence of NELLS delta-Vs may be deemed as a KBO alignment metric,

[1] Simulations in EMTG were conducted by the author's supervisor at NASA GSFC, Dr. Kyle M. Hughes.

[2] Reliability shall be understood as similarity in the attained total delta-V between a SA trajectory with respect to its corresponding EMTG trajectory.

[3] Whose chain of events are exactly the same as the previously chosen trajectory until the second KBO flyby.

further discussed in Chapter 5. More accurate predictions are expected to happen if trajectories with lower approximate delta-V values are found.

One possible cause that the NKSA did not found any solutions with lower delta-V could be that the chosen 60-day time step to discretize each KBO trajectories was not fine enough. Therefore a more exhaustive grid could be an important future research work to seek trajectories in the vicinity of the output trajectories provided by NELLS. Another factor that could have contributed to this lack of low delta-V trajectories is the limited KBO number present in the JPL SSD compared to the estimated amount of bodies that are expected to be in the Kuiper Belt region [57].

As a final remark, despite still having high delta-V values, the last 2-KBO trajectory attained by EMTG could be realistic for flight. For instance this trajectory has a dry mass of 489 kg, which is higher than New Horizons'. Nonetheless, there are key differences between both missions like delta-V budget; New Horizons has a budget of 397 m/s about six times lower than the attained EMTG output trajectory. Therefore, future work with high-fidelity tools could further investigate this trajectory for a potential mission.

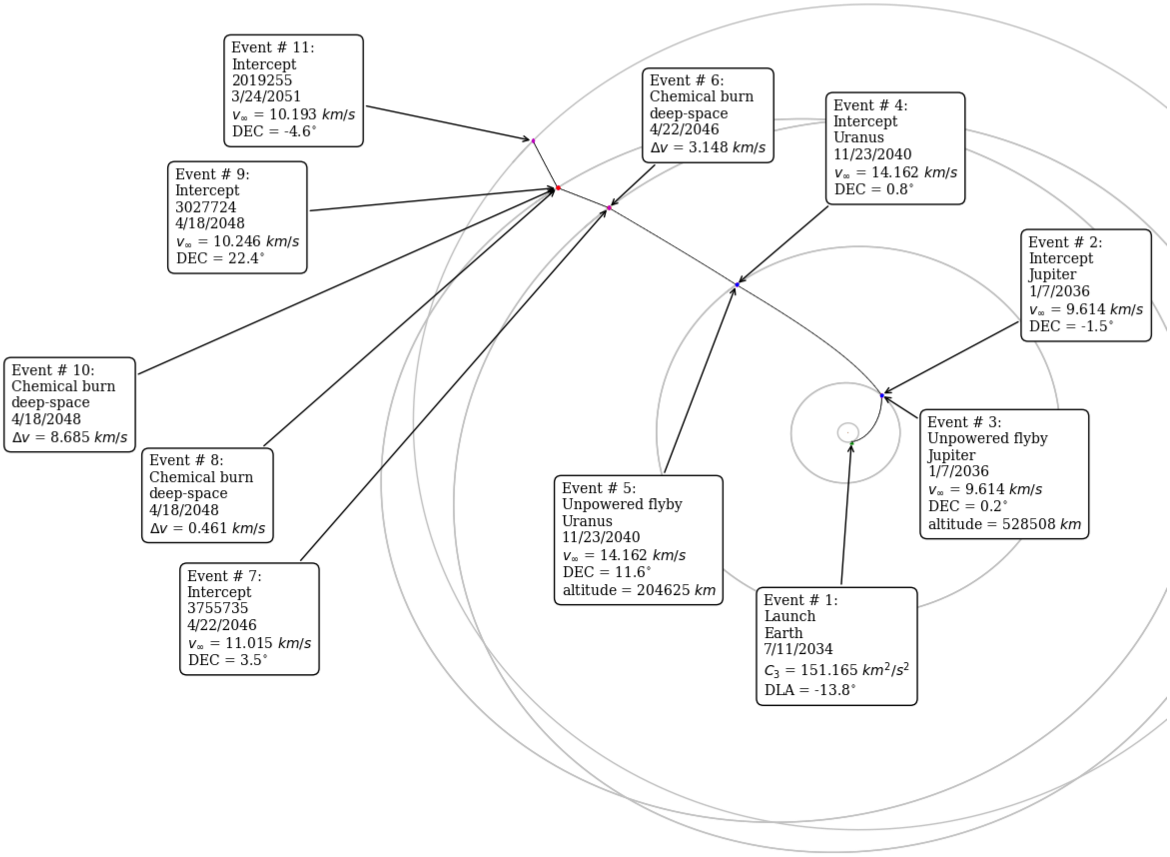


Figure 4.1: EMTG result using SA candidate trajectory as first guess. Total delta-V is 12.29 km/s. Chosen objective function used 'minimize delta-V'.

Launch Date	Launch C3	Total Δv	Jupiter Flyby	Uranus Flyby	1^{st} Maneuver	3755735 Flyby	2^{nd} Maneuver	3^{rd} Maneuver	3027724 Flyby	4^{th} Maneuver	2019255 Flyby
2464156.5	156.36	1.40	2464720.90	2466619.99	2467879.98	2468839.98	2468899.98	2469576.76	2469576.76	2469619.98	2470219.98
Launch Date	Launch C3	Total Δv	Jupiter Flyby	Uranus Flyby	1^{st} Maneuver	3755735 Flyby	2^{nd} Maneuver	3^{rd} Maneuver	3027724 Flyby	2019255 Flyby	
2464154.5	151.17	12.29	2464699.5	2466481.5	2468457.5	2468457.5	2469184.5	2469184.5	2469184.5	2470254.5	

Table 4.1: Comparison between the chosen SA trajectory used as first guess and the EMTG output trajectory, Figure 4.1. SA trajectory and EMTG trajectory are first and second row respectively. Units: JD, km2/s2, km/s.

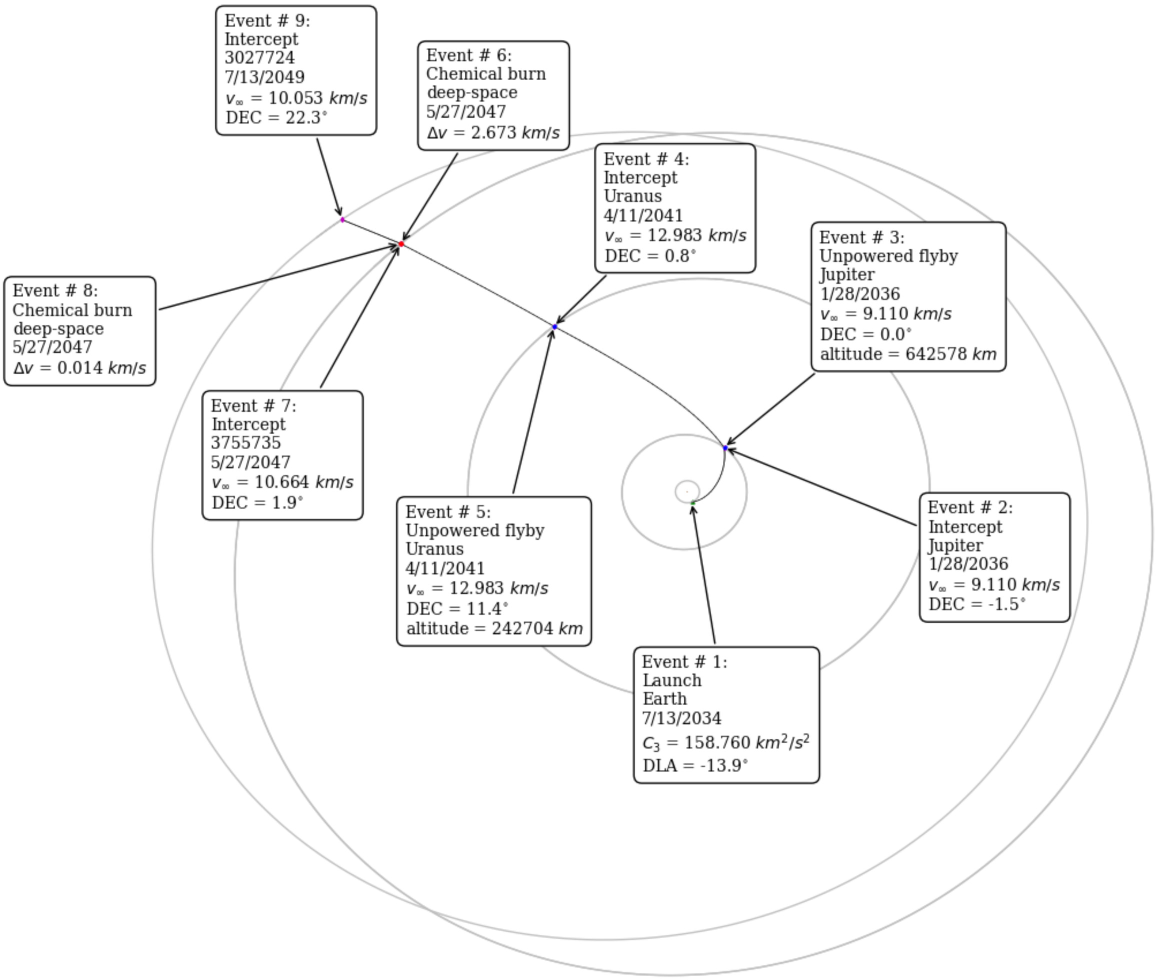


Figure 4.2: EMTG result using the predecessor of the chosen SA trajectory as first guess. Total delta-V is 2.68 km/s. Chosen objective function used 'minimize delta-V'.

Launch Date	Launch C3	Total Δv	Jupiter Flyby	Uranus Flyby	1^{st} Maneuver	3755735 Flyby	3027724 Flyby
2464156.5	156.36	1.00	2464720.90	2466619.99	2467879.98	2468839.98	2469576.76
Launch Date	Launch C3	Total Δv	Jupiter Flyby	Uranus Flyby	1^{st} Maneuver	3755735 Flyby	3027724 Flyby
2464156.5	158.76	2.687	2464720.5	2466620.5	2468857.5	2468857.5	2469635.5

Table 4.2: Comparison between the predecessor of the chosen SA trajectory used as first guess and its corresponding EMTG output trajectory when the aim is to minimize the total mission delta-V, Figure 4.2. The predecessor of the chosen SA trajectory and the EMTG trajectory are given in the first and second row respectively. Units: JD, km2/s2, km/s.

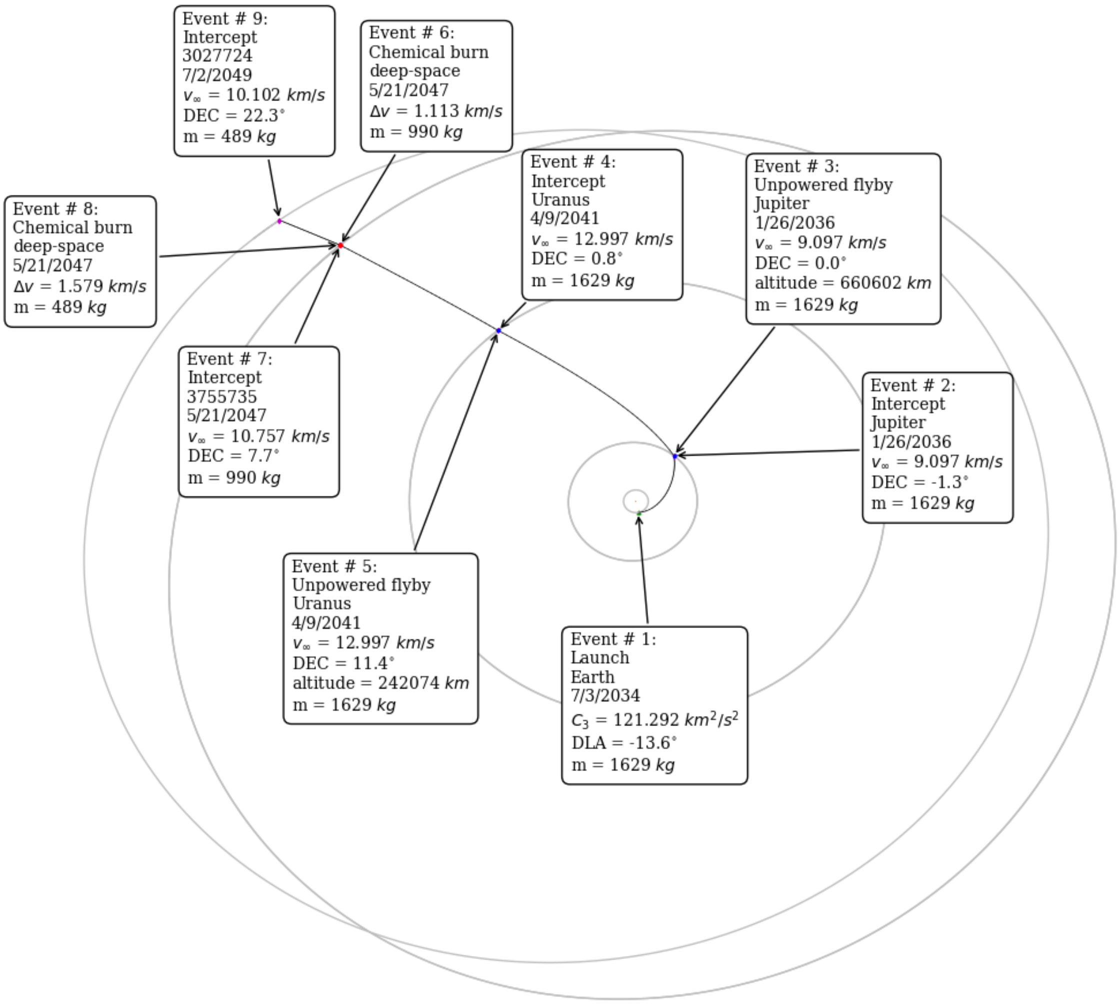


Figure 4.3: EMTG result using the predecessor of the chosen SA trajectory as first guess. Total delta-V is 2.69 km/s. Chosen objective function used 'maximize log10(final mass)'

Launch Date	Launch C3	Total Δv	Jupiter Flyby	Uranus Flyby	1^{st} Maneuver	3755735 Flyby	3027724 Flyby
2464156.5	156.36	1.00	2464720.90	2466619.99	2467879.98	2468839.98	2469576.76
Launch Date	Launch C3	Total Δv	Jupiter Flyby	Uranus Flyby	1^{st} Maneuver	3755735 Flyby	3027724 Flyby
2464146.5	121.29	2.692	2464718.5	2462965.5	2468851.5	2468851.5	2469624.5

Table 4.3: Comparison between the predecessor of the chosen SA trajectory used as first guess and its corresponding EMTG output trajectory when the aim is to maximize the final spacecraft mass, Figure 4.3. The predecessor of the chosen SA trajectory and the EMTG trajectory are given in the first and second row respectively. Units: JD, km2/s2, km/s.

5 Result Validation

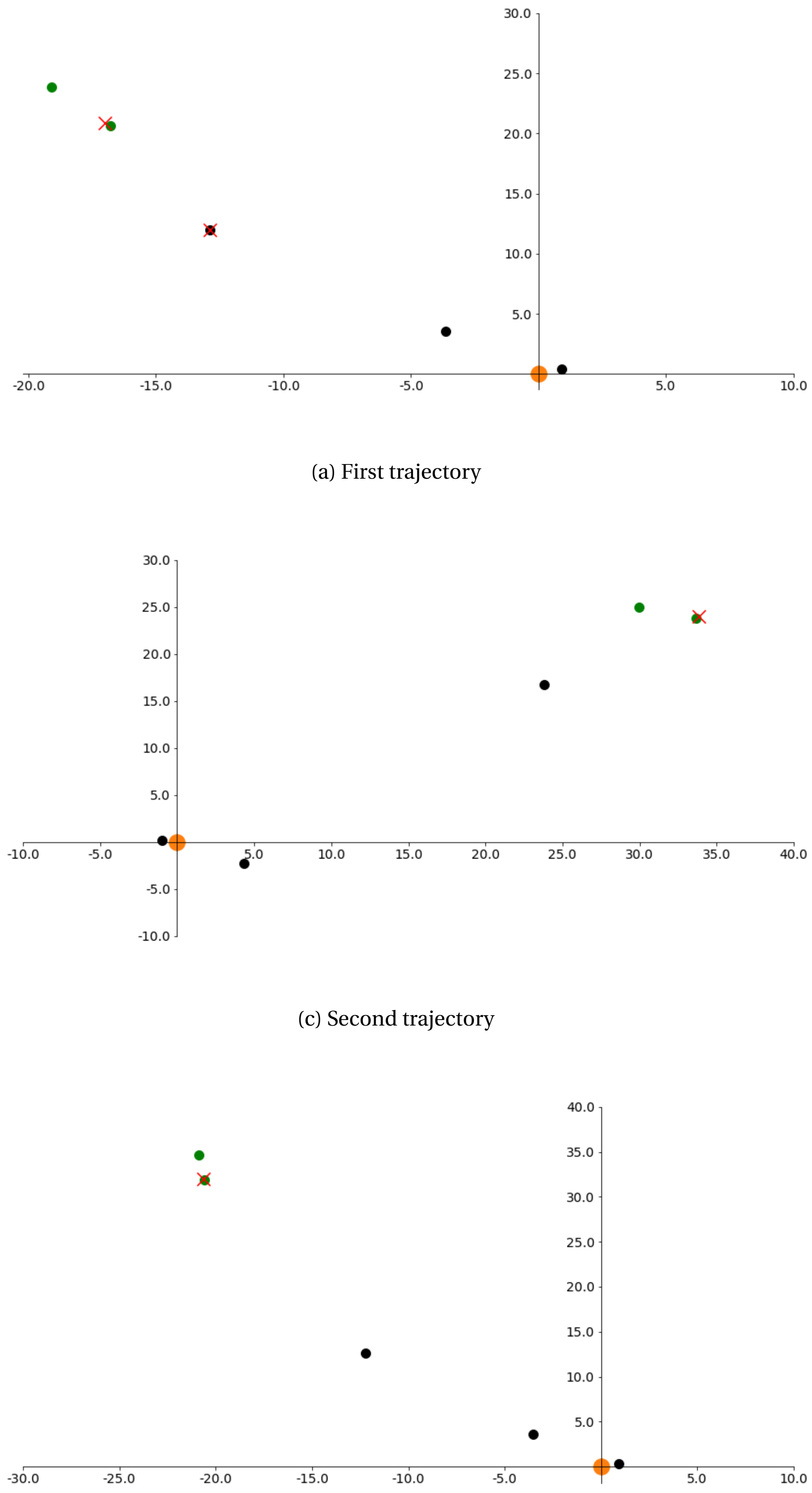
In the previous chapters, the two-phase algorithm SA[1] was implemented and used for a rapid low-fidelity search for potential KBO encounter sequences. This chapter deals with the discussion on the reliability of SA as a pathfinding technique. The validation strategy consisted of plotting several low-delta-V trajectories to visually check how aligned the KBOs in those trajectories are.

It was concluded in the previous chapter that the proposed SA trajectory encountering three KBOs had too high delta-V values to represent an accurate approximation with respect to the amount of propellant needed in EMTG. The delta-V threshold which a candidate SA trajectory can be deemed to have a reliable delta-V calculation is unknown. Taking into account there is still research to be done in this algorithm, several plots are given down below of the trajectories with the lowest delta-V values encountering both two and three KBOs.

The goal is to visually evaluate in every trajectory how KBOs are aligned among themselves and with respect to the last planet flyby. A high degree of alignment would give an extra support to the argument about SA reliability in particular with trajectories having low delta-V values. The plots are given in Figures 5.1, 5.2, 5.3, 5.4, 5.5, and 5.6. As may be seen, KBOs tend to be either aligned like Figure 5.3 (g) or closely grouped together like Figure 5.4 (a).

Therefore, based on the results of these sections, it may be concluded that SA seems to be able to track KBO combinations and it is expected to provided accurate approximations of real trajectories provided that delta-V does not exceed a certain threshold, which could be investigated in future work.

[1] The combination of a grid search performed by NELLS and NKSA.

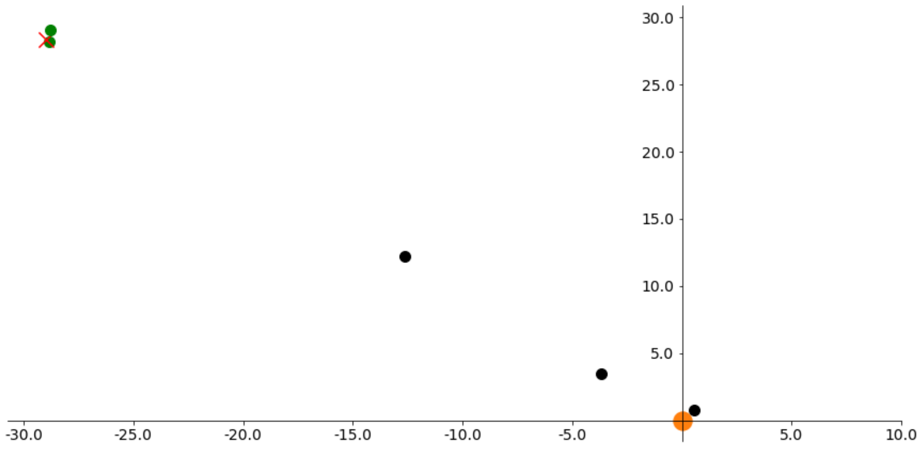


(a) First trajectory

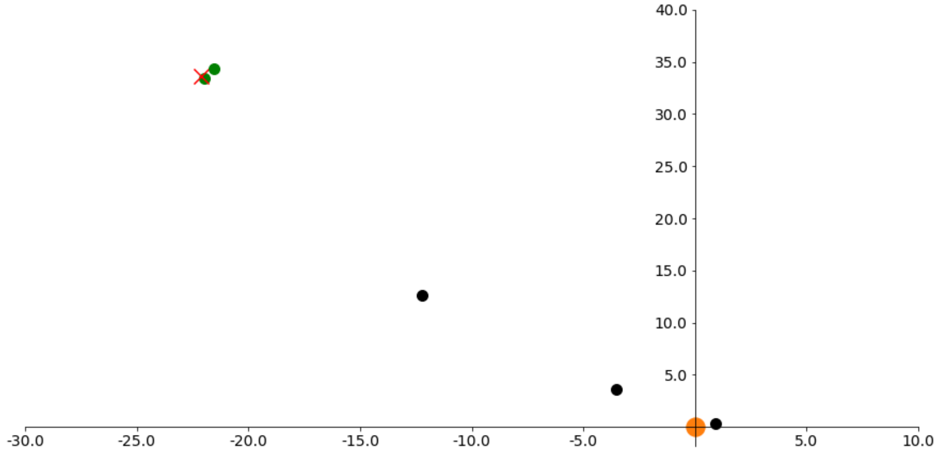
(c) Second trajectory

(c) Third trajectory

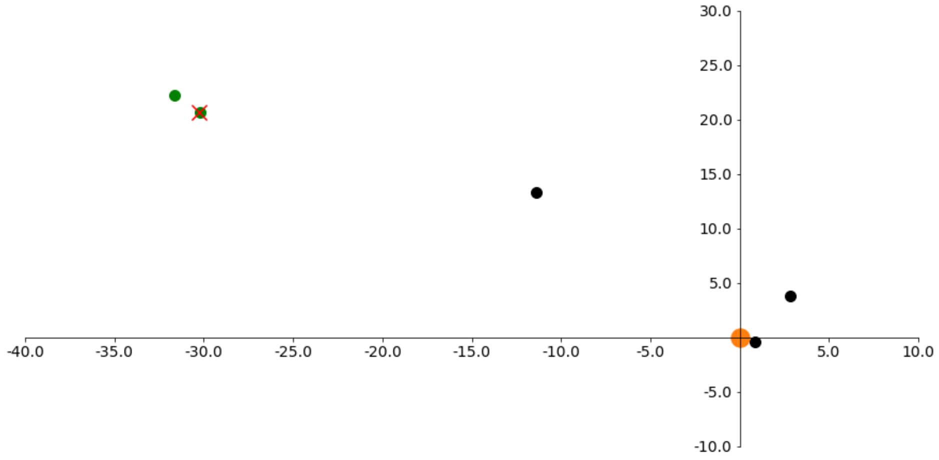
Figure 5.1: Plots for 2-KBO trajectories 1(3). X and Y axes correspond to the X and Y axes in ICRF. Green points stand for KBOs, black points stand for planets, red crosses stand for maneuvers, orange point represents the Sun.



(d) Forth trajectory

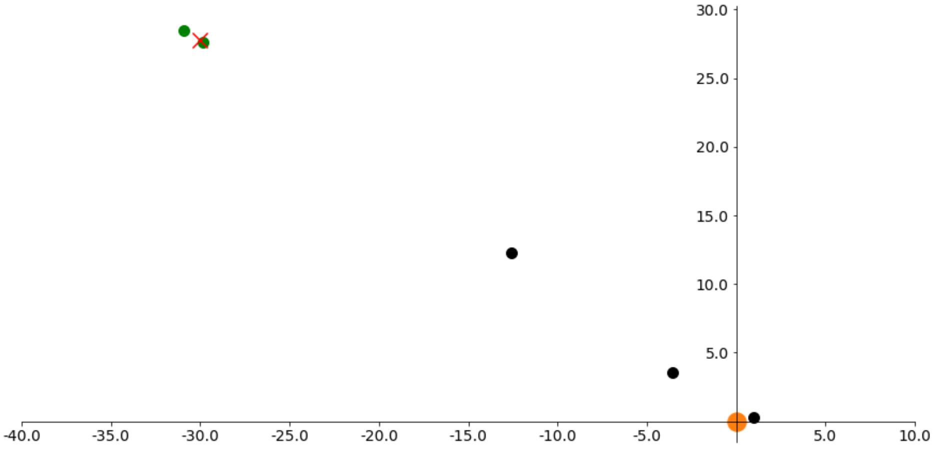


(e) Fifth trajectory

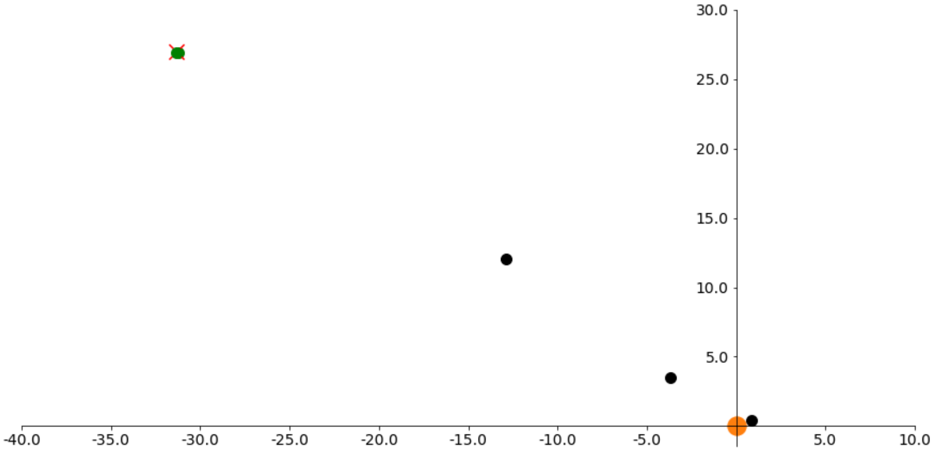


(f) Sixth trajectory

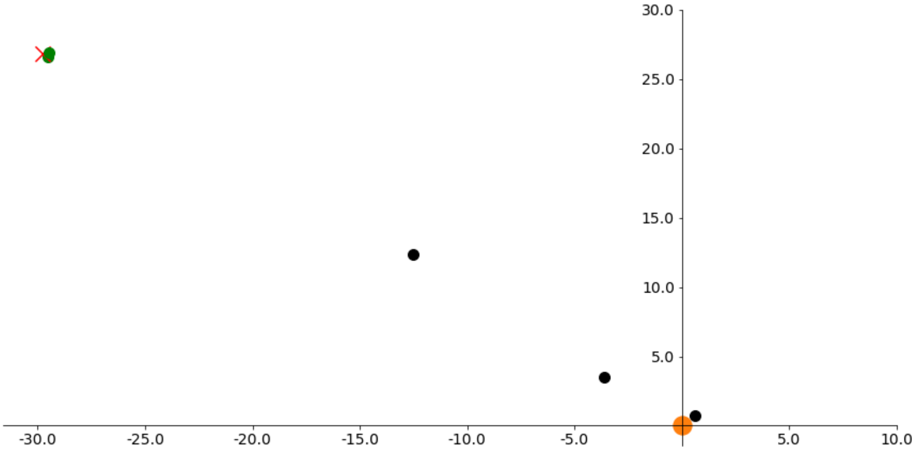
Figure 5.2: Plots for 2-KBO trajectories 2(3). X and Y axes correspond to the X and Y axes in ICRF. Green points stand for KBOs, black points stand for planets, red crosses stand for maneuvers, orange point represents the Sun.



(g) Seventh trajectory

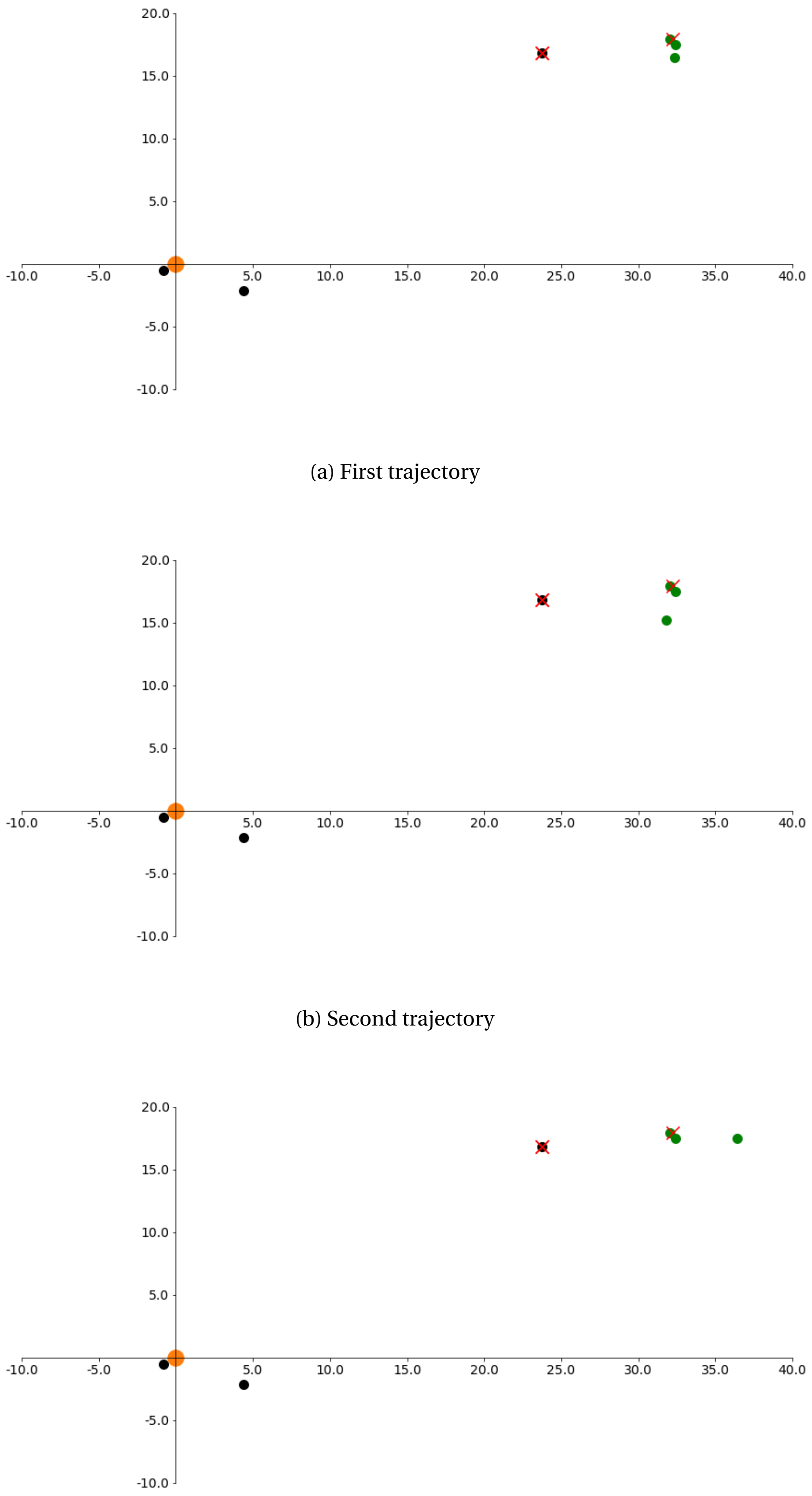


(h) Eighth trajectory



(i) Ninth trajectory

Figure 5.3: Plots for 2-KBO trajectories 3(3). X and Y axes correspond to the X and Y axes in ICRF. Green points stand for KBOs, black points stand for planets, red crosses stand for maneuvers, orange point represents the Sun.

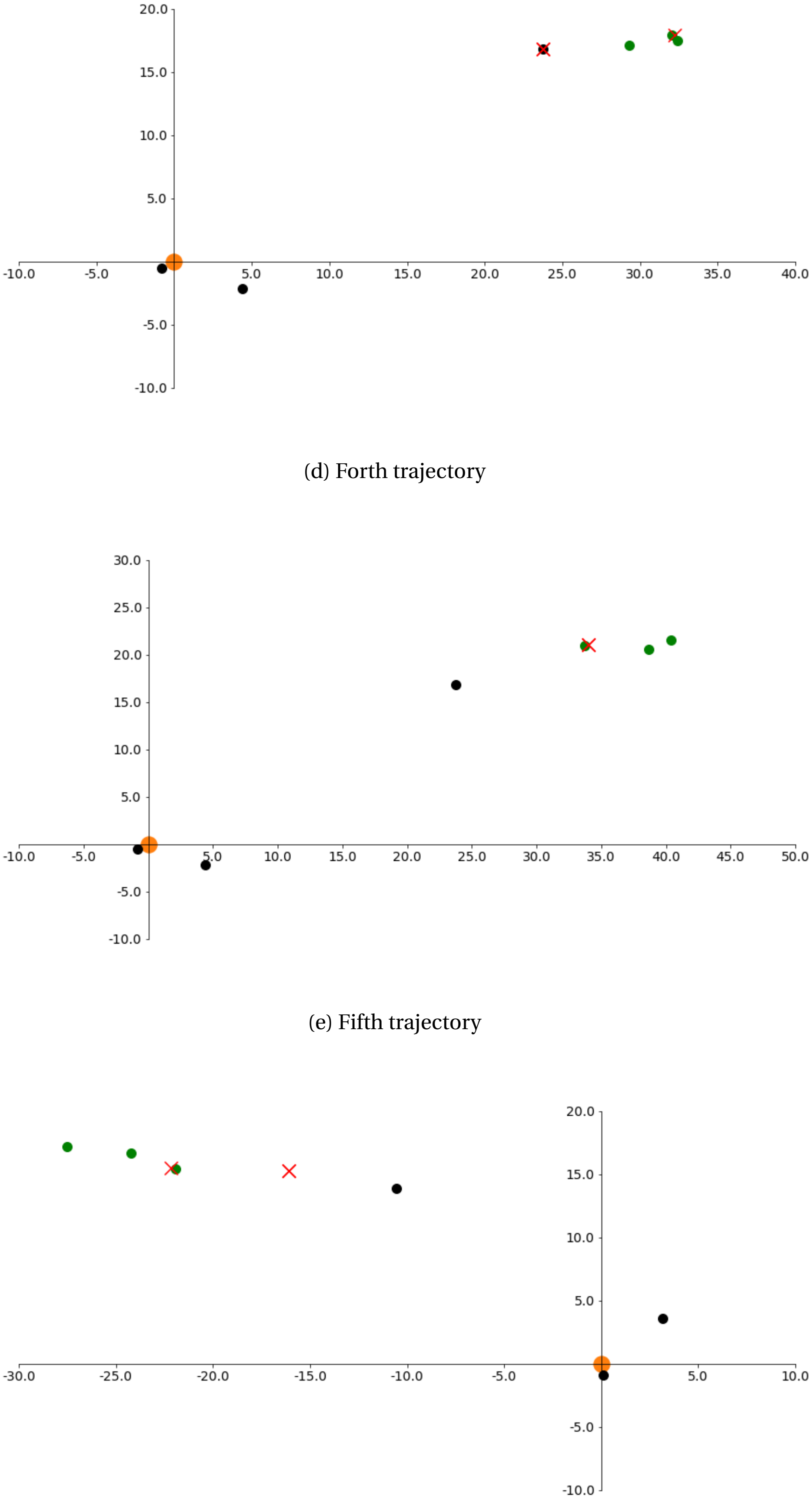


(a) First trajectory

(b) Second trajectory

(c) Third trajectory

Figure 5.4: Plots for 3-KBO trajectories 1(3). X and Y axes correspond to the X and Y axes in ICRF. Green points stand for KBOs, black points stand for planets, red crosses stand for maneuvers, orange point represents the Sun.

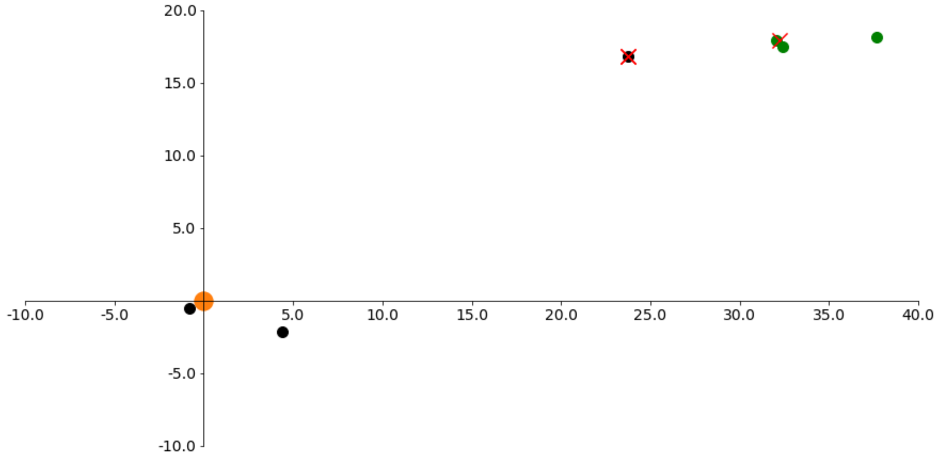


(d) Forth trajectory

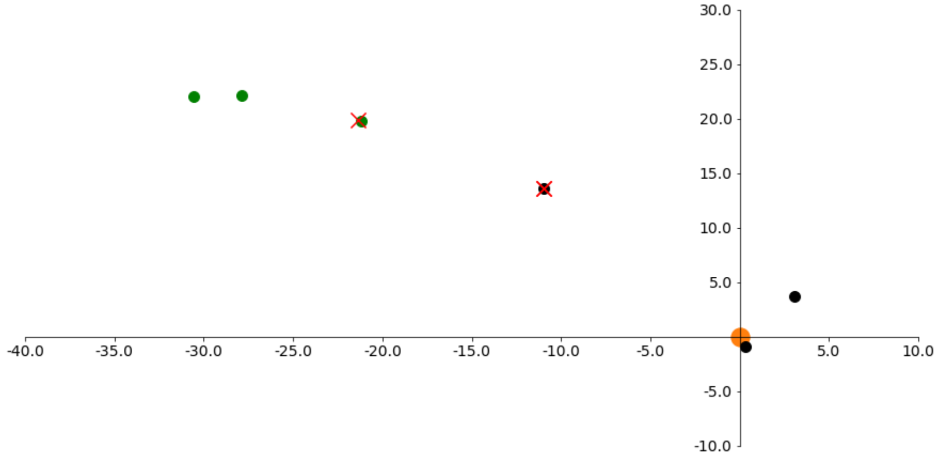
(e) Fifth trajectory

(f) Sixth trajectory

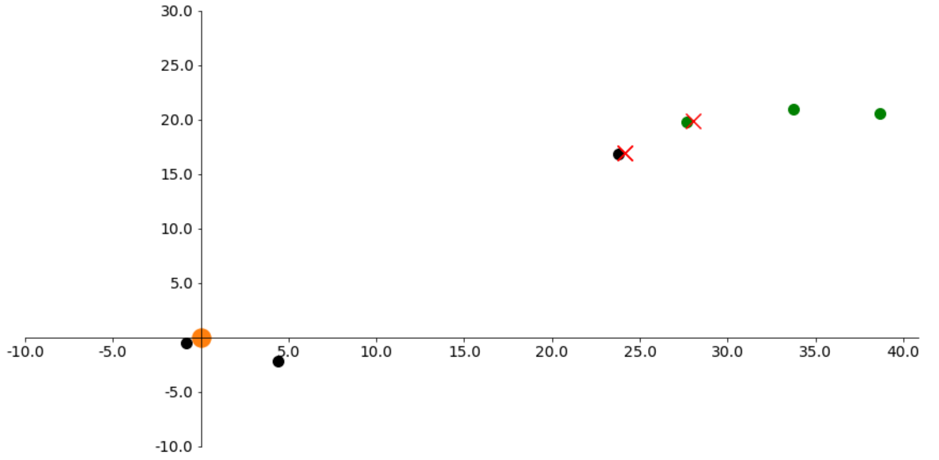
Figure 5.5: Plots for 3-KBO trajectories 2(3). X and Y axes correspond to the X and Y axes in ICRF. Green points stand for KBOs, black points stand for planets, red crosses stand for maneuvers, orange point represents the Sun.



(g) Seventh trajectory



(h) Eighth trajectory



(i) Ninth trajectory

Figure 5.6: Plots for 3-KBO trajectories 3(3). X and Y axes correspond to the X and Y axes in ICRF. Green points stand for KBOs, black points stand for planets, red crosses stand for maneuvers, orange point represents the Sun.

6 Conclusions

A rapid low-fidelity technique (SA) has been designed to tackle the problem of finding trajectories encountering at least two KBOs present in the JPL SSD. Due to the combinatiorial complexity of this problem, it was necessary to develop a technique that could map search space under a tractable computational time, before using higher fidelity tools like EMTG.

The SA is composed by two different grid search types sequentially run along with an intermediate phase of trajectory filtering. That filter was intended to prune out similar solutions to avoid a sharply increase of computational time.

Not only that trajectory filtering was applied between those two phases of the SA, but also in every iteration of the second grid search type, when evaluating the addition of an extra KBO encounter.

The first grid search type was performed with a NASA in-house tool called NELLS; wherein a new, systematic approach to identify the step sizes in target-body arrival date for the Lambert-based grid search was developed. This first search had the goal of finding trajectories encountering one KBO.

The second grid search type was done from NELLS results and by a new algorithm developed in this thesis and consists of a time grid wherein delta-Vs were calculated using a linear propagation of STM. Its general form is called NKSA in the thesis. This second grid search type was meant to seek trajectories in the vicinity of the ones found in the first grid search, encountering an extra KBO at least.

A fine planetary orbit discretization has been attained for the outer Solar System planets, which led to an empirical law for KBO orbit discretization. This discretization can be directly used in other missions/tools relying on Enumerative Searches, saving the time required by the parameter tuning phase.

A filter to prune out KBOs was designed to avoid a sharp increase of computational time, which could be applicable for future Kuiper Belt exploration mission designs.

The evaluation of a SA trajectory with EMTG evidenced that SA trajectory delta-Vs are accurate only with low values as well as addition of more KBO encounters tend to decrease delta-V accuracy. Nonetheless, the SA delta-V can be understood as a KBO alignment measure.

A wide range of ballistic trajectories encountering one KBO have been found, some of the also visiting relatively unknown planets like Uranus and Neptune. Consequently, these trajectories have a high scientific potential as well, taking also into account that (likewise it was done in New Horizons) more KBOs may be detected in the vicinity of those trajectories in future. In addition, an example of a flyable 2-KBO trajectory was attained from the SA results.

Therefore, the SA is an useful pathfinding technique for high-thrust interplanetary trajectories. Nonetheless, there are some features that a future research could tackle which would expand its performance in general and, in particular, to search KBO sequences.

1. Reduce the human-in-the-loop factor by further automating the SA workflow. That is, NELLS input generation, NELLS run, solution filter application, and NKSA run. To date, every phase requires mission designer to be run.

2. Seek a sufficiently fine trajectory discretization for the inner planets especially Mars, Venus, and Earth. This should be done so the SA be able to evaluate a previous pump-up phase with respect to the trajectories found in this thesis.

3. Study possible flyby sequences that a spacecraft could perform before conducting a final Earth flyby for the time period evaluated in the the thesis.

4. Evaluate more thoroughly whether the 60-days discretization in NKSA is actually good for trajectory discretization and maneuver calculation.

5. Study the inclusion of delta-V maneuvers in the first grid search type.

6. Research up to what delta-V value is NKSA's the delta-V calculation accurate.

A

NELLS Input File Template

```
\\ NASA Exhaustive Lambert Lattice Search (NELLS)
\\ Jeremy Knittel
\\ Written Oct 2015 ~ June 2017
\\ Tree search added by Donald Ellison in 2017

\\ NELLS is an iterative patched-conic solver to find 2 body trajectories using
\\ ballistic or powered gravity assist, DSMs, and rendezvous.

missionName = EarthJPluto

\\ Specify the number of threads that you want to use (setting to "max" will use all available)
num_OpenMP_threads = max

\\ Example options file. When run in NELLS, this script finds 16 trajectories that
\\ ballistically go from Earth to Jupiter over a 1000 day launch window.

\\ -------- Route options ---------------------------------------------------------
\\ NELLS in grid mode requires a fixed flyby sequence.
\\ Specify the trajectory sequence. The numbers are indices into
\\ the "bodyNames" vector below (first index 1, not 0). The first
\\ body is the launch body (Earth launch is not assumed, so you must
\\ specify Earth if so desired). The final body is the arrival body.
\\ All intermediary bodies will be visited with a ballistic or
\\ powered flyby or rendezvous. All subsequent options in this section
\\ match the order of the route vector, and should be the same length.
route           = {3,5,9}

\\ Are any flybys powered? The first and final values are ignored,
\\ and the intermediary values should be 1 or 0 to indicate if the
\\ visit to the corresponding body in the route vector is powered
\\ or not
ifPowered       = {0,0,0}

\\ Is there a rendezvous proscribed at any of the intermediary bodies?
\\ The first value is ignored. The final value is not. If rendezvous
\\ are proscribed at intermediary bodies, then minWait and maxWait must
\\ also be set.
ifArrRendezvous = {0,0,0}

\\ If rendezvous was proscribed at an intermediary body, specify
\\ the allowable range of rendezvous time in days. If the rendezvous body
\\ is a planet, then delta-v calculations will be made to put the
\\ spacecraft into a circular orbit at an altitude of "finalOrbitAlt".
\\ finalOrbitAlt should probably be a vector, but as of now it is not.
minWait         = {0,0,0}
maxWait         = {0,0,0}
finalOrbitAlt   = 50000

\\ Are deep space maneuvers (DSMs) allowed on a given trajectory leg?
\\ The first value is ignored. The second value corresponds to the leg
\\ between the second and third bodies in the "route". The third value
\\ corresponds to the leg between the third and fourth bodies in the
\\ "route", etc. The final value is ignored.
ifDSM           = {0,0,0}
\\ ---------------------------------------------------------------------------------

\\ -------- Grid options -----------------------------------------------------------
\\ All trajectories are built up from an initial grid of launch date
\\ and time of flight from the first body in "route" to the second
\\ body in "route".

\\ In Julian Days, specify the launch window open date
startDate = 2463783.5

\\ Specify how long the launch window is open for in days
searchLength = 2414

\\ Specify the minimum and maximum time of flight from the first to
\\ second bodies in "route" in days
maxTOF = 7306.0
minTOF = 0.0

\\ Specify if body to body encounter dates are to be gridded in time (time)
\\ or true anomaly (TA)
gridType = TA

\\ Specify if DSM placement is to be gridded in time (time), true anomaly (TA) or
\\ some custom (custom) placement
dsmGridType = time

\\ If gridding launch dates in time,
\\ specify how to discretize the launch window dates to search and
\\ the time of flight from body 1 to body 2 in "route" in days
dt = 5

\\ If gridding orbits by true anomaly, specify
\\ the number of discretization points to use.
\\ TA spacing will then be 2*pi/numNodes
\\ gridNodesPerOrbit are the number of nodes to discretize each body's orbit into
\\ NOTE: there must be one entry in gridNodesPerOrbit for each body in the Universe file
gridNodesPerOrbit = {50,50,366,50,616,50,50,50,70,50,50,50,50,50,50}

\\ If the first two bodies in the route are the same, and
\\ "ifCheckResonances" is turned on below, then NELLS will also create a
\\ a grid of departure C3 values from "minLaunchC3" and "maxLaunchC3"
\\ both specified below. How should that grid be discretized? [km2/s2]
```

```
deltaC3 = 5

\\ If "ifCheckResonances" is turned on below, or "ifDSM" is turned on
\\ for any leg above, then NELLS will also create a grid of crank angles
\\ for viable flyby departures. Specify the discretization of the
\\ crank angle grid in degrees
deltaAngle = 5

\\ If "ifArrRendezvous" was turned on for any of the bodies above (except
\\ for the final body), then  NELLS will create a grid between "minWait"
\\ and "maxWait", specified above. Further, NELLS will create a grid for
\\ allowable times of flight from the rendezvous body to the next body in
\\ the route. What is the desired discretization of that grid in days?
rendDT = 50

\\ If "ifPowered" was turned on for any of the flybys above, then NELLS
\\ will create a grid of allowable times of flight from the flyby body
\\ to the next body in the grid. What is the desired discretization of
\\ that grid in days? [32 ’nominal’, tried 1 as well]
poweredDT = 1

\\ If "ifDSM" was turned on for any of the legs above, then NELLS will
\\ create a grid to place the dsm. The grid is a percentage of the
\\ overall flight time of the leg.
deltaPct = .5

\\ If dsmGridType is set to TA, then the DSM placement grid will be in
\\ true anomaly instead of percentage of the overall leg flight time (specified in degrees)
deltaDSMangle = 5.0

\\ ---------------------------------------------------------------------------------

\\ -------- Solver options ---------------------------------------------------------
\\ Should NELLS check for near resonant trajectory arcs when a given leg
\\ goes from and to the same body? Using real ephemeris, and 2-body
\\ propagation, ballistic rendezvous solutions arent possible, so
\\ NELLS will insert small delta-v maneuvers to compensate.
ifCheckResonances = 1

\\ If NELLS is going to check resonant trajectory arcs, what is the desired
\\ body:spacecraft orbits? Numbers do not need to be integers. Specify as many
\\ as desired.
resonances = {1:1,3:1,2:1,3:2,1:2}

\\ Should NELLS search for a trajectory backwards? NELLS will still move
\\ forward through the "route" above. The launch and arrival constraints below
\\ still correspond to the first and final bodies in the route respectively,
\\ however their meanings are reversed. "minTOF" and "maxTOF" are still
\\ time of flight between the first to bodies in the route. NELLS will
\\ reverse the direction of time, so that it is effectively searching backwards.
\\ "startDate" and "searchLength" will then be the allowable arrival dates at
\\ the first body in the route.
ifBackwards = 1

\\ What is the maximum number of spacecraft revolutions allowable in a given leg?
maxRevs = 1
```

```
\\ When calculating the parameters of a flyby, how should NELLS patch the
\\ incoming and outgoing trajectories?
\\ c3match_algorithm = 1 indicates that it will create a grid of time
\\ of flight from the flyby body to the next body in the "route". Then, if
\\ NELLS detects a sign change in the value of outgoingC3(TOF) - incomingC3, then
\\ NELLS will iterate to find the exact date. This method is slow but much
\\ more exhaustiv3
\\ c3match_algorithm = 0 uses a novel bracketing algorithm to where dt starts
\\ large and decreases to find sign changes in outgoingC3(TOF) - incomingC3 by
\\ assuming a typical structure to that function. This method is very fast
\\ but can miss solutions
c3match_algorithm = 1

\\ If c3match_algorithm = 1, and you selected a time grid, what discretization should
\\ be used for the grid in days?
c3match_dt = 1

\\ Is the initial departure energy provided by the spacecraft not a launch vehicle?
\\ If so, should NELLS aggregate the delta-v for departure in the trajectory
\\ delta-v?
countLaunchAsDV = 0

\\ NELLS can skip searching for multi-spacecraft revolution legs if the
\\ target body has a period greater than "minMultiRevPer" days. Should NELLS
\\ skip all multi-spacecraft rev trajectories for a given target body period? If so,
\\ also specify that period.
ifSkipMultiRev = 1
minMultiRevPer = 1200

\\ When NELLS is iterating to find the exact time of flight from a flyby body
\\ to the next body in the "route", how close must C3 of the incoming and
\\ outgoing v-infinity vectors match?
maxC3miss = 0.01

\\ --------------------------------------------------------------------------------

\\ -------- Universe options -----------------------------------------------------

centralBodyName = Sun

\\ Should NELLS use SPICE and splineEphem instead of linear interpolation
\\ of HORIZONS data? Linear interpolation is 10~20% faster, but is less
\\ accurate. From experience, if the HORIZONS data is 1-day interval,
\\ then, it is plenty accurate to feed a higher fidelity solver. For example,
\\ a NELLS trajectory converted to an EMTG seed will come back as feasible.
useSplineEphem = 0

\\ Where is the ephemeris data?
universeFolder = C:/Users/miguel/Desktop/NELLS/Universe

\\ SPICE leap second kernel
SPICE_leap_second_kernel = naif0012.tls

\\ SPICE reference frame kernel
SPICE_reference_frame_kernel = pck00010.tpc
```

```
\\ Specify the range of Julian dates that you want to load ephemeris data for
lastEphemDate = 2476885.5
firstEphemDate = 2448119.5

\\ What spacing should be used for ephemeris interpolation points (in days)?
ephemeris_grid_size = 1.0

\\ ---------------------------------------------------------------------------------

\\ -------- Constraint options -----------------------------------------------------
\\ What are the allowable ranges of launch energy in km^2/s^2?
maxLaunchC3 = 500
minLaunchC3 = 0

\\ What are the allowable ranges of arrival energy in km^2/s^2?
maxArrivalC3 = 500
minArrivalC3 = 0

\\ What is the maximum mission delta-v allowable in km/s?
maxDV = 0

\\ What is the latest allowable arrival date in Julian days? If no such
\\ constraint exists, use 1e10
arrDateConstraint = 1e10

\\ What is the maximum allowable mission length in days? Even if no such constraint
\\ exists, make up a reasonable number, because the grids are sized according
\\ to this variable.
maxMissionLength = 7306
\\ ---------------------------------------------------------------------------------

\\ -------- Execution options ------------------------------------------------------
\\ Should NELLS display the number of options found so far as it moves past
\\ each node in the initial TOF vs launch date grid?
ifDispDates = 0

\\ Should NELLS display detailed information about its search? Probably
\\ only useful for debugging
ifDisp = 0

\\ Should NELLS store or only the best trajectories at each node (1) or
\\ store all of them (0)?
skimNode = 0

\\ Would you like to interact with the NELLS user interface after running the
\\ trajectory search?
interact = 1

\\ ---------------------------------------------------------------------------------

\\ -------- Output options ---------------------------------------------------------
\\ How should NELLS sort the trajectories found?
\\ sort = 1: by overall time of flight
\\ sort = 2: by overall delta-v
\\ sort = 3: by launch C3
\\ sort = 4: by score. Score is calculated as the norm of the three metrics above
\\           scaled by maxMissionLength, maxDV, and maxLaunchC3 respectively
sort = 4

\\ How many trajectories should be written to the .gstraj file?
saveTraj = 2000

\\ Directory where bin and traj files will be stored
resultsFolder = NELLS_results

\\ What is the name of the .gsbin file? This file will contain a binary of
\\ all trajectories found. This file can be re-loaded into the NELLS
\\ user interface later so that a user can interact with it.
binFile = JupiterNeptune1d.gsbin

\\ What is the name of the .gstraj file? This file will contain plain text
\\ written descriptions of the best "saveTraj" trajectories
trajFile = JupiterNeptune1d.gstraj

\\ Where on your computer is NELLS? This is needed if plotting is desired.
nellsPath = C:/Users/miguel/Desktop/NELLS

\\ Where is EMTG on your computer, if at all? This is needed if you want to
\\ output an EMTG seed of a given trajectory, because NELLS will create
\\ an EMTG universe that you can use with the EMTG seed.
emtgPath = /Users/jknittel/EMTG/branch/EMTGJ/Universe/
\\ ---------------------------------------------------------------------------------

\\ -------- TREETOP options --------------------------------------------------------

node_pool_size = 1000000

\\ ---------------------------------------------------------------------------------
```

B

Tisserand Graphs

Tisserand graphs are based on the Tisserand's criterion, which comes from a first integral of the well-known *circular restricted three-body problem*[1] . Tisserand criterion states that there is a constant T of motion before and after a (non-powered) swingby about a body [53]. It is possible from Tisserand constant and specific orbital energy (ϵ) expressions (Equations B.1 and B.2) to create a Tisserand contour plot (Figure B.1) for spacecraft flybys about the Solar System bodies [53].

$$T = \frac{r}{a} + 2\left[\frac{a\left(1-e^2\right)}{r_{planet}}\right]^{1/2} \cos(i) \tag{B.1}$$

$$\epsilon = -\frac{\mu}{2a} \tag{B.2}$$

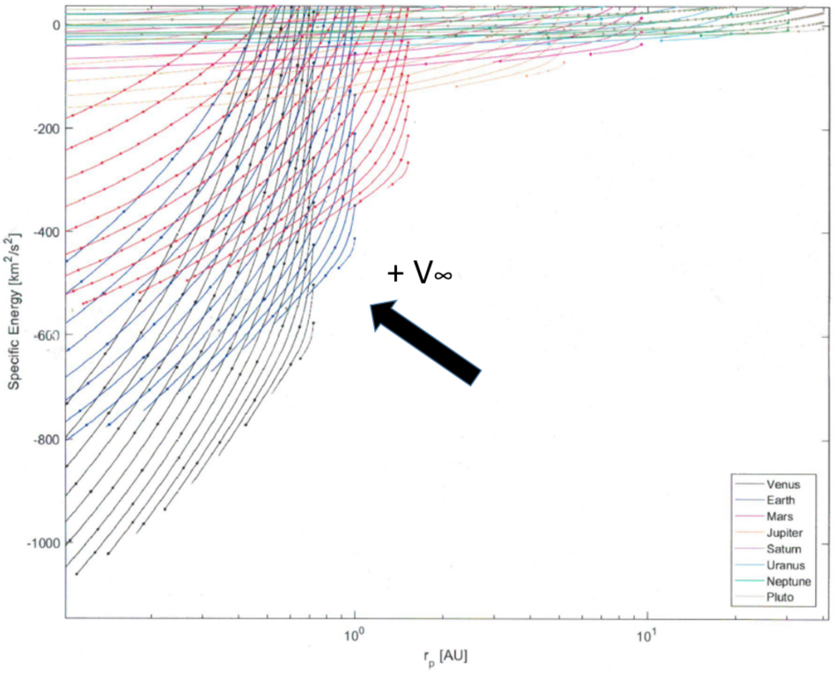


Figure B.1: Solar System Tisserand graph. Perihelion vs Specific Energy. The black arrow indicates the direction wherein spacecraft escape velocity norm with respect to a body increases. Own source based on Kyle Hughes' MATLAB code, the author's NASA supervisor.

[1] The reader is referred to [56] for instance for further details on that mathematical approximation.

Being r_p the planet perihelion, a the semi-major axis, e the eccentricity, i the inclination and μ the standard gravitational parameter respectively.

In Figure B.1, each point of any curve stands for an inbound/outbound (heliocentric) trajectory with respect to a planet. Each color represents a family of curves associated with a planet. Each planet or, in other words, its corresponding family of curves - has its own T value since the parameters which determine T vary in every planet.

Each curve is formed by a family of trajectories which have the same spacecraft escape velocity norm (V_∞) with respect to the planet in question. Within each curve family, V_∞ contours increase by steps of 2 km/s in the direction of the arrow displayed in Figure B.1.

Bearing in mind that, under the patched-conics model, V_∞ remains the same when performing a flyby between the inbound and outbound trajectories, flybys are modelled as displacements across their pertinent contour line. Such displacements in Figure B.1 are limited by the minimum flyby altitude constraint value, 200 km in this thesis, expressed as the points on each curve.

The minimum (conservative) feasible launch C_3 value corresponds to the Earth curve with the highest V_∞ value without intersecting any curve of Jupiter family in Figure B.1. This conclusion is due to Mars and Venus flybys are excluded in the grid search[2] and the Jupiter curves are closer to the Earth curves than the ones belonging to the rest of the outer planets. As may be seen in that Figure, such value is a $V_\infty = 7 km/s$ or (equivalently) a launch C_3 of 49 km^2/s^2.

As a final remark, it is important to note these graphs only assess the availability of transfers between planets from an energy standpoint, but not from a phasing point of view which later is figured out in the pathsolving subproblem.

[2] See section 1.2.3 for further details.

Bibliography

[1] Nasa launch vehicle performance website. URL https://elvperf.ksc.nasa.gov/.

[2] New horizons trajectory. URL https://airandspace.si.edu/multimedia-gallery/new-horizons-trajectory.

[3] Northrop grumman corporation, star 48bv. URL https://www.northropgrumman.com/Capabilities/PropulsionSystems/Documents/NGIS_MotorCatalog.pdf.

[4] Nasa, curiosity's daily launch windows. URL https://mars.nasa.gov/msl/mission/timeline/launch/launchwindow/.

[5] Freeflyer. URL https://ai-solutions.com/freeflyer/.

[6] International earth rotation and reference systems service. URL https://www.iers.org/IERS/EN/DataProducts/ICRF/icrf.html.

[7] Jpl, solar system database,small-body database browser. URL https://ssd.jpl.nasa.gov/sbdb.cgi.

[8] Northrop grumman's star 48bv kick stage motor specifications. URL https://www.northropgrumman.com/Capabilities/PropulsionSystems/Documents/NGIS_MotorCatalog.pdf.

[9] Monte, a mission analysis operations and navigation toolkit environment. URL https://montepy.jpl.nasa.gov/.

[10] Nasa decadal survey, . URL https://science.nasa.gov/about-us/science-strategy/decadal-surveys.

[11] Nasa science, solar system exploration, kuiper belt, . URL https://solarsystem.nasa.gov/solar-system/kuiper-belt/in-depth/.

[12] Spherical coordinate system used in this msc thesis. URL https://www.semanticscholar.org/paper/Enhancing-Stereo-Signals-with-High-Order-Ambisonics-Trevino-Sakamoto/0240c681d68c10797fb3c32edc0d4c6c81b495bb.

[13] Systems tool kit. URL https://www.agi.com/products.

[14] Bernardetta Addis, Andrea Cassioli, Marco Locatelli, and Fabio Schoen. Global optimization for the design of space trajectories. *Optimization On Line*, 11:191, 2008.

[15] Nitin Arora and Ryan P Russell. A fast and robust multiple revolution lambert algorithm using a cosine transformation. *Paper AAS*, 13:728, 2013.

[16] RR Bate, DD Mueller, and JE White. *Fundamentals of Astrodynamics, chap. 7.* Dover Pub. Inc., New York, 1971.

[17] Richard H Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, revised edition.* American Institute of Aeronautics and Astronautics, 1999.

[18] Miguel Benayas Penas. Literature study. *Delft University of Technology Repository*, 2018.

[19] Brian E Blank and Steven George Krantz. *Calculus: Multivariable*, volume 2. Springer Science & Business Media, 2006.

[20] Harold S.M. Coxeter. *Introduction To Geometry, 2 ed.* Wiley Classics Library, 1989.

[21] Louis A D'Amario, Larry E Bright, and Aron A Wolf. Galileo trajectory design. *Space science reviews*, 60 (1-4):23–78, 1992.

[22] Theresa Debban, T McConaghy, and James Longuski. Design and optimization of low-thrust gravity-assist trajectories to selected planets. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, page 4729, 2002.

[23] Dave Doody. *Deep space craft: an overview of interplanetary flight*. Springer Science & Business Media, 2010.

[24] Donald Hamilton Ellison. *Robust preliminary design for multiple gravity assist spacecraft trajectories*. PhD thesis, University of Illinois at Urbana-Champaign, 2018.

[25] NASA Facts. Voyager to the outer planets and into interstellar space. *NASA*, 2013.

[26] J. Balbas S. Ivy E. Merkurjev T. Hall G. Peterson, E. T. Campbell and P. Rodriguez. Relative performance of lambert solvers 1: Zero revolution methods. *Paper AAS*, 136:1495–1510, 2010.

[27] Arthur Guez, David Silver, and Peter Dayan. Scalable and efficient bayes-adaptive reinforcement learning based on monte-carlo tree search. *Journal of Artificial Intelligence Research*, 48:841–883, 2013.

[28] Yanping Guo and Robert W Farquhar. New horizons mission design. *Space science reviews*, 140(1-4): 49–74, 2008.

[29] Daniel Hennes and Dario Izzo. Interplanetary trajectory planning with monte carlo tree search. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[30] Kyle M Hughes. Gravity-assist trajectories to venus, mars, and the ice giants: Mission design with human and robotic applications. *Purdue University*, 2016.

[31] Dario Izzo, Luís F Simões, Marcus Märtens, Guido CHE de Croon, Aurelie Heritier, and Chit Hong Yam. Search for a grand tour of the jupiter galilean moons. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 1301–1308. ACM, 2013.

[32] Moriba Jah, Steven Hughes, Matthew Wilkins, and Tom Kelecy. The general mission analysis tool (gmat): A new resource for supporting debris orbit determination, tracking and analysis. In *Fifth European Conference on Space Debris*, volume 672, 2009.

[33] Drew Ryan Jones, Troy Goodson, Paul Thompson, Powtawche Valerino, and Jessica Williams. Solar probe plus: unique navigation modeling challenges. *JPL Technical Report Server*, 2016.

[34] Morris Kline. *Calculus: an intuitive and physical approach*. Courier Corporation, 1998.

[35] Allan R Klumpp. New developments in astrodynamics algorithms for autonomous rendezvous. *NASA Technical Reports Server*, 1991.

[36] C Kohlhase. Mariner jupiter/saturn 1977 mission design tradeoffs. In *13th Annual Meeting and Technical Display Incorporating the Forum on the Future of Air Transportation*, page 281, 1977.

[37] CE Kohlhase and PA Penzo. Voyager mission description. *Space science reviews*, 21(2):77–101, 1977.

[38] Demyan Vasilyevich Lantukh et al. *Preliminary design of spacecraft trajectories for missions to outer planets and small bodies*. PhD thesis, The University of Texas at Austin, 2015.

[39] Jack J Lissauer and Imke De Pater. *Fundamental planetary science: physics, chemistry and habitability*. Cambridge University Press, 2013.

[40] M Marley, L Dudzinski, T Spilker, R Moeller, C Borden, W Smythe, et al. Planetary science decadal survey jpl rapid mission architecture neptune-triton-kbo study final report. *NASA Mission Concept Study Report*, 2010.

[41] Lucy-Ann McFadden, Torrence Johnson, and Paul Weissman. *Encyclopedia of the solar system*. Elsevier, 2006.

[42] Paul Musegaas. Optimization of space trajectories including multiple gravity assists and deep space maneuvers. *Delft University of Technology Repository*, 2013.

[43] Cesar Ocampo. An architecture for a generalized spacecraft trajectory design and optimization system. 05 2003. doi: 10.1142/9789812704849_0023.

[44] Moonish R Patel. *Automated design of Delta-V gravity-assist trajectories for solar system exploration*. PhD thesis, Purdue University, 1993.

[45] Fernando Peralta and Steve Flanagan. Cassini interplanetary trajectory design. *Control Engineering Practice*, 3(11):1603–1610, 1995.

[46] Fernando Peralta and JC Smith Jr. Cassini trajectory design description. *JPL Technical Report Server*, 1993.

[47] Edward A Rinderle. Galileo user's guide, mission design system, satellite tour analysis and design subsystem. *Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, JPL D-263*, 1986.

[48] Christopher T Russell. *New Horizons: Reconnaissance of the Pluto-Charon System and the Kuiper Belt*. Springer Science & Business Media, 2009.

[49] AB Sergeyevsky and GC Snyder. Interplanetary mission design handbook. volume 1, part 3: Earth to jupiter ballistic mission opportunities, 1985-2005. *NASA Technical Reports Server*, 1982.

[50] AB Sergeyevsky and Nanette H Yin. Interplanetary mission design handbook. volume 1, part 1: Earth to venus ballistic mission opportunities, 1991-2005. *NASA Technical Reports Server*, 1983.

[51] Andrey B Sergeyevsky, Gerald C Snyder, and Ross A Cunniff. Interplanetary mission design handbook. volume 1, part 2: Earth to mars ballistic mission opportunities, 1990-2005. *NASA Technical Reports Server*, 1983.

[52] S Alan Stern. The new horizons pluto kuiper belt mission: an overview with historical context. In *New Horizons*, pages 3–21. Springer, 2009.

[53] Nathan J Strange and James M Longuski. Graphical method for gravity-assist trajectory design. *Journal of Spacecraft and Rockets*, 39(1):9–16, 2002.

[54] David A Vallado. *Fundamentals of astrodynamics and applications*, volume 12. Springer Science & Business Media, 2001.

[55] Matthew A Vavrina, Jacob A Englander, and Donald H Ellison. Global optimization of n-maneuver, high-thrust trajectories using direct multiple shooting. *NASA Technical Reports Server*, 2016.

[56] Karel F. Wakker. *Fundamentals of astrodynamics*. Institutional Repository Library Delft University of Technology, 2015.

[57] John Wilkinson. The new solar system. In *The Solar System in Close-Up*, pages 1–24. Springer, 2016.

[58] Steve Neal Williams. *Automated design of multiple encounter gravity-assist trajectories*. AIAA/AAS Astrodynamics Conference, Portland, OR, Aug. 20-22, 1990, Technical Papers. Part 2 (A90-52957 24-13), 1990.

[59] Amanda M Zangari, Tiffany J Finley, S Alan Stern, and Mark B Tapley. Return to the kuiper belt: launch opportunities from 2025 to 2040. *Journal of Spacecraft and Rockets*, 56(3):919–930, 2018.