POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

MSc in Space Engineering

Authors:

10723712	Marcello Pareschi	(BSc Aerospace Engineering - Politecnico di Milano)
10836125	Daniele Paternoster	(BSc Aerospace Engineering - Politecnico di Milano)
10711624	Alex Cristian Turcu	(BSc Aerospace Engineering - Politecnico di Milano)
10884250	Tamim Harun Or	(BSc Aerospace Engineering - International Islamic University Malaysia)

Professor: Camilla Colombo

Academic year: 2023-2024

Contents

1. Interplanetary mission

1.1. Symbols

1.1.1 Analisi della missione

A_e	$[m^2]$	area di efflusso totale
ϕ	$[rad]$	angolo di traiettoria del razzo

1.1.2 Analisi della missione 2

A_e	$[m^2]$	area di efflusso totale
ϕ	$[rad]$	angolo di traiettoria del razzo

1.2. Introduction

1.2.1 Description of the problem

The first part of the assignment aims at designing an interplanetary transfer from Mars to asteroid 1036 Ganymed exploiting a powered gravity assist on Earth. The problem is analyzed through the patched conics method, without considering the injection and arrival hyperbolae. The initial and final velocity vector of the satellite are assumed to be the same of the respective celestial body. The two heliocentric legs are calculated through the Lambert problem. The trajectory is selected with the only criteria of minimizing the cost of the mission, assessed through the total ΔV. The latter is computed by summing different contributions:

$$\Delta V_{tot} = \Delta V_1 + \Delta V_2 + \Delta V_3 \tag{1}$$

Where the three terms are defined as:

- ΔV_1 related to the injection in the first heliocentric leg;
- ΔV_2 related to the exit from the second heliocentric leg;
- ΔV_3 related to the impulse given by the engine at pericentre of hyperbola fly-by;

All the manoeuvres are assumed to be impulsive, i.e. they change only the velocity vector of the spacecraft, mantaining invariated the position vector. Note that ΔV_1 and ΔV_2 are related to heliocentric velocities, while ΔV_3 is calculated through relative geocentric velocities.
For each manoeuvre the cost is computed as:

$$\Delta V_i = \|\boldsymbol{V_{+,i}} - \boldsymbol{V_{-,i}}\| \tag{2}$$

Where $\boldsymbol{V_{-,i}}$ and $\boldsymbol{V_{+,i}}$ are the velocity vectors before and after the i-th manoeuvre respectively.

1.2.2 Assigned data and constraints

A few constraints were considered:

- earliest departure date $t_{min,dep} = [01/01/2028\ \ 00:00:00]$ in Gregorian calendar
- latest arrival date $t_{max,arr} = [01/01/2058\ \ 00:00:00]$ in Gregorian calendar
- time of flights of the two heliocentric arcs must be greater than the associated parabolic time
- minimum pericentre radius of the fly-by hyperbola $r_p = r_E + 500\ km$
- single-revolution Lambert problem was considered
- reasonable total time of the mission

Note that the constraint on the pericentre radius of the fly-by is considered both for avoid impact on Earth and to prevent undesired atmoshperic drag effects.

1.3. Algorithms description

The targeting problem previously defined can be seen as an optimization problem with three degrees of freedom (DOFs). Indeed, once the departing date and the two times of flight of the heliocentric legs are chosen, both the Lambert's arcs and the fly-by hyperbola are fully defined. Regarding the formulation of the Lambert's problem, it requires the knowledge of the initial and final position and also the imposed time of flight between them. For two Lambert's arcs it would be needed a total of six DOFs, but this quantities are dependant one to each other:

- the final position vector of the first arc corresponds to the initial position of the second one;
- the initial date for the departing on the second arc corresponds the arrival date on the first arc;
- fixing the first Lambert's arc, the final position and arrival date for the second Lambert's arc are related through the analytical ephemerides.

Once the two heliocentric legs are determined, the powered gravity assist follows as the geocentric velocity vectors are known.
Two methods were implemented to solve the optimization problem: **brute force algorithm** and the **gradient descent algorithm**.

1.3.1 Brute force algorithm

1.3.2 Gradient descent algorithm

1.4. Reduction of the time windows

1.4.1 Resonance period analysis

1.4.2 Cost-plot analysis

1.4.3 Final time window selection

1.5. Conclusion and results

Bibliography