POLITECNICO DI MILANO

POLITECNICO
MILANO 1863

# Resonant flybys in the b-plane: extension of the theory to elliptical planetary orbits and dynamic programming application

*Author:*
Giulio CAMPITI

*Supervisor:*
Camilla COLOMBO

*Co-supervisor:*
Alessandro MASAT

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
Department of Aerospace Science and Technology
*Master of Science in Space Engineering*

**A.Y. 2020/21**

When referring to this work, full bibliographic details must be given:
*Giulio Campiti, "Resonant flybys in the b-plane: extension of the theory to elliptical planetary orbits and dynamic programming application", 2021, Politecnico di Milano, Faculty of Industrial Engineering, Department of Aerospace Science and Technologies, Master in Space Engineering, Supervisor: Camilla Colombo, Co-supervisor: Alessandro Masat*

# *Abstract*

The implementation of resonant gravity assist maneuvers is an essential prerequisite for interplanetary missions requiring complex trajectory solutions. A convenient formalism to design resonant trajectories is the b-plane, as post-encounter orbits with prescribed semi-major axis can be easily mapped on this plane and thus targeted a priori. This result was originally derived in the approximation of pure circular orbits of the flyby bodies. While this may seem like a good approximation for bodies with a low orbital eccentricity, it becomes questionable for those with a markedly elliptical orbit, such as Mars or Mercury. Moreover, if a flyby takes place in the vicinity of the apsidal points, the model precision inevitably worsens. These cases are of interest for actual interplanetary missions, such as ESA's BepiColombo. This thesis presents a generalisation of the classical flyby model in the b-plane allowing for the flyby body around the primary to be elliptical. The classical and the extended models are tested and compared on two mission design applications, showing non-negligible differences when the flyby body has a marked orbital eccentricity and/or the flyby takes place at one of the apsidal points.

The second part of the thesis focuses on the preliminary design and optimisation of unperturbed resonant trajectories. The resonance condition at each flyby is a discrete decision variable that can be advantageously leveraged by techniques of combinatorial optimisation. A dynamic programming approach to the problem is proposed, using as discrete decision variables a set of resonance conditions and the total number of flybys. The developed algorithm is tested by reproducing the design of Solar Orbiter's resonant phase with Venus, used in the actual mission to gradually raise the ecliptic inclination and decrease the perihelion distance. The computational efficiency of the approach is evaluated by comparing it with the more standard brute-force approach.

# *Sommario*

L'esecuzione di sequenze di flyby risonanti è una tecnica utilizzata in diverse missioni interplanetarie che richiedono soluzioni di traiettorie complesse. Un formalismo conveniente per progettare traiettorie risonanti è il b-plane, in quanto permette di mappare facilmente orbite post-flyby a semi-asse maggiore assegnato. Tale risultato fu derivato originariamente nell'ipotesi di orbite planetarie puramente circolari. Sebbene questa possa essere una buona approssimazione per pianeti con una bassa eccentricità orbitale, diventa discutibile per corpi con un'orbita marcatamente ellittica, come Marte o Mercurio. Inoltre, per flyby in prossimità dei punti apsidali la precisione del modello peggiora inevitabilmente. Questi casi sono di interesse per reali missioni interplanetarie, come BepiColombo. In questa tesi si propone dunque una generalizzazione del classico modello di flyby nel b-plane a orbite planetarie eccentriche. Il modello classico e quello esteso verranno testati e confrontati su due applicazioni di progetto di missione, rivelando differenze non trascurabili nei casi in cui il pianeta ha una marcata eccentricità orbitale e/o il flyby avviene in uno dei punti apsidali.

La seconda parte della tesi si concentra sulla progettazione di traiettorie risonanti ottimali. La condizione di risonanza ad ogni flyby è una variabile decisionale discreta che può essere sfruttata vantaggiosamente tramite tecniche di ottimizzazione combinatoria. Si propone un approccio di programmazione dinamica al problema, utilizzando come variabili decisionali discrete un insieme di condizioni di risonanza e il numero totale di flyby. L'algoritmo sviluppato viene testato nella progettazione della fase risonante di Solar Orbiter con Venere, la quale viene utilizzata nella missione reale per aumentare gradualmente l'inclinazione e diminuire la distanza dal Sole al perielio. L'efficienza computazionale dell'approccio è valutata tramite un confronto con il metodo più standard di enumerazione esaustiva.

# *Acknowledgements*

This thesis is part of the COMPASS project. COMPASS has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 679-086 - COMPASS), www.compass.polimi.it.

Some of the contents of this thesis have been presented during the "2021 AAS/AIAA Astrodynamics Specialist Conference, Big Sky, Virtual, August 9-11", and have been submitted for publication in the conference proceedings.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **AU** | **A**stronomical **U**nit |
| **CR3BP** | **C**ircular **R**estricted **3**-**B**ody **P**roblem |
| **ESA** | **E**uropean **S**pace **A**gency |
| **GAM** | **G**ravity **A**ssist **M**anoeuvre |
| **JUICE** | **JU**piter **IC**y moons **E**xplorer |
| **MGA** | **M**ulti **G**ravity **A**ssist |
| **MJD** | **M**odified **J**ulian **D**ays |
| **MOID** | **M**inimum **O**rbital **I**ntersection **D**istance |
| **NASA** | **N**ational **A**eronautics and **S**pace **A**dministration |
| **RAAN** | **R**ight **A**scension of the **A**scending **N**ode |
| **SOI** | **S**phere **o**f **I**nfluence |
| **SOLO** | **SOL**ar **O**rbiter |
| **SOTA** | **S**tate **o**f **t**he **A**rt |
| **SPP** | **S**hortest **P**ath **P**roblem |

# Chapter 1

# Introduction to the thesis work

Since the 60's interplanetary spaceflight has provided a means to gather some scientifically valuable data about the solar system that could not be gained through observations from Earth's surface or from orbit around the Earth. Missions such as Pioneer and Voyager made possible stunning discoveries that deepened our knowledge about the various planets, moons and small bodies of the solar system. Nonetheless, the large velocity changes required to travel between different bodies in the solar system have always posed tough challenges to space exploration. The current launch vehicles and propulsion technologies are still inadequate to reach target orbits with much higher or much lower energy with respect to the Earth, or even to achieve substantial changes in the heliocentric orbital inclination.

To overcome the technological limitations, Gravity Assist Manoeuvres (GAM) have been widely exploited over the past 50 years, allowing to accomplish otherwise unfeasible missions such as Cassini-Huygens, Bepicolombo, and many others. However, in several cases the implementation of single GAMs is not sufficient to achieve the mission objectives and it is necessary to synthesise a sequence of "resonant" GAMs (or resonant flybys), such that each time the orbital period of the spacecraft and that of the small body are commensurate, providing a new encounter at the same position in space [1]. A typical case is when the mission requires large changes in the heliocentric orbital inclination, given that the maximum inclination variation achievable in a single GAM features an upper bound. An analytical estimate of such value was derived by Golubev et al. [2] based on Labunsky's approximation [3], and is computed in Table 1.1 for several planets and moons of the solar system. As it can be seen, when large inclination changes are desired for a spacecraft orbiting the inner solar system or around a giant planet, a single flyby may not be enough. Various real-life missions successfully implemented sequences of resonant flybys, such as Solar Orbiter (SOLO), an European Space Agency's (ESA) mission that exploits four resonant encounters at Venus to gradually increase the solar inclination to 34° and decrease the perihelion distance [4].

The resonance concept is also beneficial for detailed planetary/lunar observations, since repeated close encounters can provide diverse observation angles and thus a better data collection. In the JUpiter ICy moons Explorer

(JUICE) mission, resonant flybys at Callisto are used to raise and then bring back to zero the equatorial inclination of the probe with respect to Jupiter, allowing to investigate the non-equatorial regions of the planet's environment [5].

| **Celestial body** | **Central body** | $\Delta \mathbf{i}_{max}$ [deg] |
|---|---|---|
| Mercury | Sun | 3.37 |
| Venus | " | 10.70 |
| Earth | " | 13.82 |
| Mars | " | 7.58 |
| Jupiter, Saturn, Uranus, Neptune | " | >90 |
| Pluto | " | 13.35 |
| Io, Europa | Jupiter | 5.15 |
| Ganymede | " | 8.78 |
| Callisto | Saturn | 10.87 |
| Titan | " | 17.20 |

TABLE 1.1: Maximum orbital inclination variation achievable in a single GAM around some large bodies of the solar system

For missions including resonant flyby sequences, the preliminary design of the resonant phase is often crucial to the overall trajectory design. Consider, for example, a mission with a highly inclined orbital phase. The maximum inclination attainable with any sequence of resonant flybys is proportional to the magnitude of the planetocentric velocity U of the spacecraft, which is approximately a constant of motion if no manoeuvres are involved. However, U is also inversely proportional to the magnitude of the change in inclination at each flyby in the sequence. Therefore, the choice of U is the outcome of a compromise between the final inclination to be achieved and the rate of change in inclination [6]. Clearly, the value chosen for U notably affects the design of the trajectory prior to the resonant phase. In Solar Orbiter, the required relative arrival velocity at Venus before the first resonant flyby was $\sim 18\,\mathrm{km/s}$ [4], which could not be achieved with a direct injection into a trajectory from Earth to Venus. For this reason, it was necessary to implement an initial sequence of three flybys (the first at Venus and the others at Earth) allowing the spacecraft to arrive at Venus with the required velocity, and then begin the resonant phase.

The design of resonant flyby trajectories involves several challenging aspects. Firstly, the solution space for this kind of problems is very large and with a variable dimensionality due to the unknown number of total flybys required.

Moreover, as the resonance condition at each flyby is a discrete decision variable, the optimisation problem features a mixed-continuous combinatorial nature. For these reasons, efficient methods based on simplified dynamics models are required to quickly explore such complex solution spaces in the early stages of the design, thus allowing to reduce the number of promising trajectories. A fairly accurate analytic model to design resonant flybys was first devised by Öpik [7] and later improved by Valsecchi et al. [8] based on the b-plane, a target plane orthogonal to the incoming hyperbolic velocity vector of a small body on an encounter with a planet or moon.

In the first part of the thesis, the model is further improved by removing one of its main simplifying assumptions, namely that the flyby body moves on a purely circular orbit. In this way, the designed solutions will be more realistic, thus facilitating the transition process from a preliminary trajectory to one based on a full ephemeris model. In the second part of the thesis, the developed model is used in conjunction with a dynamic programming approach to address the design of resonant flyby sequences. Dynamic programming provides a very efficient optimisation method for problems involving combinatorial elements, and allows to find the global optimal solution by investigating only a reduced portion of the solution space. Overall, the objective of the thesis is to provide a fast algorithm for computing fairly accurate preliminary solutions to resonant flyby design problems, whose results could serve as reasonable starting point for numerical methods to faster converge to more complete trajectory solutions.

# Part I

# B-plane resonance theory for elliptical planetary orbits

# Chapter 2

# Introduction

In 1976, Öpik [7] defined a convenient coordinate system based on the b-plane and laid the foundations for the analytical modeling of close approaches. Since then, several authors have expanded on its work, removing some simplifying assumptions and obtaining different kinds of results. Farnocchia et al. [9] proposed a method to map the uncertainties in the estimated trajectory onto the b-plane. Greenberg et al. [10] proved that Öpik's formulation is valid over a wide range of conditions, but for very slow encounters it fails to predict the results of a precise three-body numerical integration. In Valsecchi et al. [11] the theory was extended to the case in which the Minimum Orbital Intersection Distance (MOID) between the orbits of the small body and the flyby body has a finite value. A remarkable result was obtained by the same authors in another paper, showing that the locus of points leading to a post-encounter orbit with a prescribed semi-major axis is identified by a unique circle in the b-plane [8]. As the resonance concept is directly linked to that of semi-major axis, this allows resonant orbits to be easily mapped on the b-plane and thus targeted a priori. Several other authors extended the original Öpik's theory, such as Carusi et al. [12] and Milani et al. [13].

A common limitation to Öpik's theory and the b-plane circle equation is that they were both derived in the approximation of pure circular orbit of the flyby body. While this may seem like a good approximation for bodies with a low orbital eccentricity, it becomes questionable for those with a markedly elliptical orbit, such as Mars or Mercury. Moreover, if a flyby takes place in the vicinity of the apsidal points, the model precision inevitably worsens. These cases are of interest for actual interplanetary missions, an example of which is ESA's BepiColombo, whose baseline trajectory includes a first flyby close to Mercury's perihelion and another one 180° later [14]. Despite Valsecchi et al. [11] has already suggested the possibility of extending the theory for elliptical orbits, to the best of the author's knowledge this problem has not yet been addressed and will be the focus of the first part of the thesis, which is outlined as follows:

- **Chapter 3** provides an overview of the b-plane formalism for close encounters and the b-plane circle equation.
- **Chapter 4** generalises the classical model allowing for the flyby body's around the primary to be elliptical. Simulations are carried out to validate the developed model and compare it with the classical one. A

brief discussion is given on the validity of one hypothesis behind the extended model, namely that the magnitude of the planetocentric velocity is conserved across a GAM.

# Chapter 3

# Close encounters and orbital resonances in the b-plane

This chapter presents an overview of Öpik's theory of planetary close encounters, together with the classical derivation of the circle equation in the b-plane. These concepts provide the necessary theoretical basis for understanding the model extension developed in Chapter 4. More detailed information can be found in the works of Öpik [7], Carusi et al. [12] and Valsecchi et al. [11].

## 3.1 Öpik's theory of planetary close encounters

Öpik's theory was derived based on a two-body linked-conics approximation and by including Tisserand's criterion [15]. In this framework, a massless particle is assumed to move on a Keplerian orbit relative to the Sun, until it enters the sphere of influence (SOI) of a perturbing planet moving on a circular orbit. From a heliocentric perspective, the SOI is assumed to have an infinitesimal radius, implying that the orbits of the two objects are actually touching, i.e. the MOID is zero. In this approximation, the point in which the spacecraft enters the SOI is not defined, leaving a degree of freedom in the flyby dynamics [16]. The encounter is then modeled in the b-plane an instantaneous rotation of the incoming planetocentric velocity vector, without changing its magnitude. The flyby deflection has the effect of changing the particle's heliocentric orbit from before the encounter to immediately after it. Despite these simplifying assumptions, the true nature of planetary encounters is approximated quite well by this model and provides accurate initial estimates of the actual trajectory [17]. On the other hand, the model fails for very slow encounters or very low velocity of the small body [10]. As a general rule of thumb, it is inapplicable for a Tisserand parameter exceeding the value of 3 [11].

### 3.1.1 Planetocentric reference frame

Öpik's theory was formulated by using the Jacobi normalised units, the same commonly employed in the non-dimensionalisation of the Circular Restricted Three-Body Problem (CR3BP). The period of the planet is $2\pi$ and its distance

to the Sun is 1. It is also assumed that both the mass of the Sun and the gravitational constant are equal to 1, such that the planet's velocity is 1 as well. Let $a, e, i, \Omega, \omega$ be the orbital elements of the particle's heliocentric orbit, where i is reckoned with respect to the orbital plane of the planet. It is introduced a planetocentric reference frame $(X, Y, Z)$ centred in the planet's center of mass such that the X-axis is directed from the Sun to the planet's position, the Y-axis is aligned with its direction of motion and the and Z-axis completes the right-handed triad. In this reference frame, the components of the planetocentric velocity vector of the particle are [12]:

$$\mathbf{U} = \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} \pm\sqrt{2 - 1/a - a\left(1-e^2\right)} \\ \sqrt{a\left(1-e^2\right)}\cos i - 1 \\ \pm\sqrt{a\left(1-e^2\right)}\sin i \end{bmatrix} \tag{3.1}$$

Calling with T the Tisserand parameter of the particle's orbit, it can be shown a relation between T and the magnitude of the planetocentric velocity U:

$$U = 3 - T = \sqrt{3 - \frac{1}{a} - 2\sqrt{a\left(1-e^2\right)}\cos i} \tag{3.2}$$

Since T is an invariant in the CR3BP, Equation (3.2) allows Öpik's theory to include a sort of three-body information, stating that the magnitude of the planetocentric velocity before and after a planetary encounter is approximately conserved. The encounter geometry can be defined by two angles $\vartheta$ and $\varphi$, as showed in Figure 3.1.

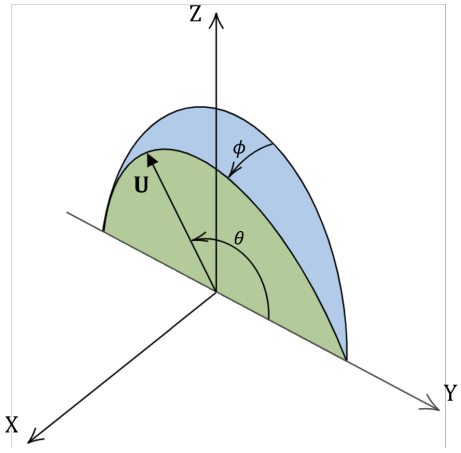


FIGURE 3.1: Planetary encounter geometry

Hence, the planetocentric velocity vector can be alternatively expressed as:

$$\begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} U \sin\vartheta \sin\varphi \\ U \cos\vartheta \\ U \sin\vartheta \cos\varphi \end{bmatrix} \tag{3.3}$$

such that:

$$\begin{bmatrix} \cos\vartheta \\ \tan\varphi \end{bmatrix} = \begin{bmatrix} U_y/U \\ U_\chi/U_z \end{bmatrix} \tag{3.4}$$

Substituting now Equations (3.1) and (3.2) inside Equation (3.4) yields

$$\cos\vartheta = \frac{\sqrt{a\left(1-e^2\right)}\cos i - 1}{\sqrt{3-\frac{1}{a}-2\sqrt{a\left(1-e^2\right)}\cos i}} \tag{3.5}$$

Taking the square of Equation (3.2), the following relation is obtained

$$\sqrt{a\left(1-e^2\right)}\cos i = \frac{3-1/a-U^2}{2} \tag{3.6}$$

which can be used inside Equation (3.5) to finally get

$$\begin{aligned} \cos\vartheta &= \frac{\frac{3-1/a-U^2}{2}-1}{\sqrt{3-\frac{1}{a}-\left(3-1/a-U^2\right)}} \\ &= \frac{3/2-1/2a-U^2/2-1}{U} \\ &= \frac{3-1/a-U^2-2}{2U} \\ &= \frac{1-1/a-U^2}{2U} \end{aligned} \tag{3.7}$$

Equation (3.5) provides an interesting insight into the geometrical meaning of $\vartheta$. In a close encounter, the potential energy of the particle is fixed by the distance of the planet relative to the Sun. Instead, the kinetic energy is linked to the magnitude of the vectorial sum of the heliocentric velocities of the particle and the planet. For fixed U, this magnitude does only depends on $\vartheta$. Therefore, the total energy (potential plus kinetic), and thus the semi-major axis, are only function of $\vartheta$, as evidenced in Equation (3.5).

An expression of $\varphi$ as function of the particle's orbital elements can be derived as well. This, however, does not only depend on the semi-major axis

and moreover suffers from sign ambiguity:

$$\tan\varphi = \pm\sqrt{\frac{2a-1}{a^2\left(1-e^2\right)}-1}\frac{1}{\sin i} \tag{3.8}$$

If $0 < \varphi < \pi/2$ the encounter is from inside planet's orbit, and vice versa if $-\pi/2 < \varphi < 0$.

### 3.1.2 B-plane frame

The b-plane is defined as the plane orthogonal to **U** and containing the centre of the planet. In this context it is introduced another planetocentric reference frame $(\xi, \eta, \zeta)$ such that the $(\xi, \zeta)$-axes lie on the b-plane and $\hat{\eta}$ is perpendicular to it. In particular, $\hat{\zeta}$ is parallel to the projection of the planet's velocity $\mathbf{v}_{pl}$ on the b-plane but with opposite direction and $\hat{\xi}$ completes a right-handed reference system:

$$\hat{\eta} = \frac{\mathbf{U}}{\|\mathbf{U}\|}; \quad \hat{\xi} = \frac{\mathbf{U}\times\mathbf{v}_{pl}}{\|\mathbf{U}\times\mathbf{v}_{pl}\|}; \quad \hat{\zeta} = \hat{\xi}\times\hat{\eta} \tag{3.9}$$

This set of axes is particularly convenient to characterise a close approach between two objects, as it decouples the two key quantities of an encounter. The $\zeta$ coordinate is indeed associated to the shift in the time of arrival of the particle with respect to the planet, whereas $\hat{\xi}$ is related to the minimum distance between the two orbits. When the particle crosses the b-plane, $\eta = 0$ and a quantity called impact parameter identifies the closest distance between the object and the planet:

$$b^2 = \xi^2 + \zeta^2 \tag{3.10}$$

A vector in the $(X, Y, Z)$ frame can be expressed in $(\xi, \eta, \zeta)$ coordinates by means of the direction cosine matrix relating the two systems:

$$\mathbf{x}|_{b-plane} = \left[\hat{\xi}\ \hat{\eta}\ \hat{\zeta}\right]^T \cdot \mathbf{x}|_{XYZ} \tag{3.11}$$

### 3.1.3 B-plane deflection model

As stated before, the magnitude of the velocity vector **U** remains unchanged from before to after the encounter. The flyby is modelled as an instantaneous rotation of such vector by a given amount that depends on its magnitude, on the impact parameter and on the planet's mass m in units of the Sun's mass. Introducing the characteristic length $c = m/U^2$, the deflection angle $\gamma$ can be computed as:

$$\tan\frac{\gamma}{2} = \frac{m}{bU^2} = \frac{c}{b} \tag{3.12}$$

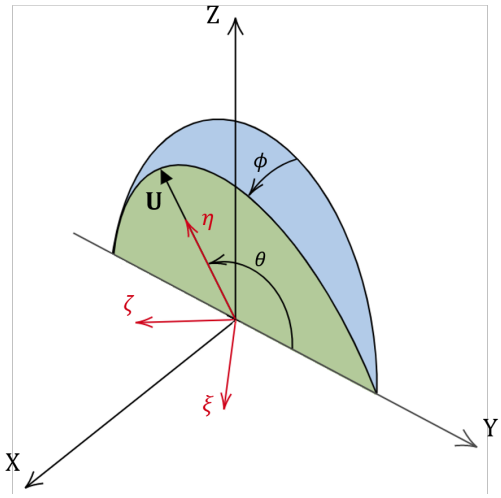


FIGURE 3.2: The two planetocentric reference systems (X, Y, Z) and $(\xi, \eta, \zeta)$

or, equivalently

$$\begin{aligned} \cos\gamma &= \frac{b^2 - c^2}{b^2 + c^2} \\ \sin\gamma &= \frac{2bc}{b^2 + c^2} \end{aligned} \tag{3.13}$$

After the encounter, the vector $\mathbf{U}$ is rotated by $\gamma$ in the direction given by the direction angle $\psi$, defined as:

$$\begin{bmatrix} b\sin\psi \\ b\cos\psi \end{bmatrix} = \begin{bmatrix} \xi \\ \zeta \end{bmatrix} \tag{3.14}$$

A simple way to compute the outgoing planetocentric velocity $\mathbf{U}'$ is to view the rotation as an eigen-axis one. The magnitude of the angle is $\gamma$ and the direction of the eigen-axis $\hat{\lambda}$ is identified by the cross product $\mathbf{U} \times \mathbf{U}'$, perpendicular to the plane of rotation. Since $\mathbf{U}$ by definition is aligned with $\hat{\eta}$ and orthogonal to $\mathbf{U} \times \mathbf{U}'$, the eigen-axis will lie on the b-plane and it can thus be readily obtained in this frame with the aid of geometrical considerations:

$$\hat{\lambda} = \begin{bmatrix} -\cos\psi \\ 0 \\ \sin\psi \end{bmatrix} \tag{3.15}$$

The eigen-axis can then be expressed in the (X, Y, Z) frame by means of Equation (3.11) and the rotation matrix from $\mathbf{U}$ to $\mathbf{U}'$ can be finally computed through the Rodrigues' formula [18].
The direction of $\mathbf{U}'$, aligned with the outgoing asymptote, can be identified

through two post-encounter angles $\vartheta'$ and $\varphi'$ (Figure 3.3), where the superscript $'$ stands for post-encounter conditions. An expression of $\vartheta'$ can be obtained in terms of $\vartheta$, $\gamma$ and $\psi$:

$$\cos\vartheta' = \cos\vartheta\cos\gamma + \sin\vartheta\sin\gamma\cos\psi \tag{3.16}$$

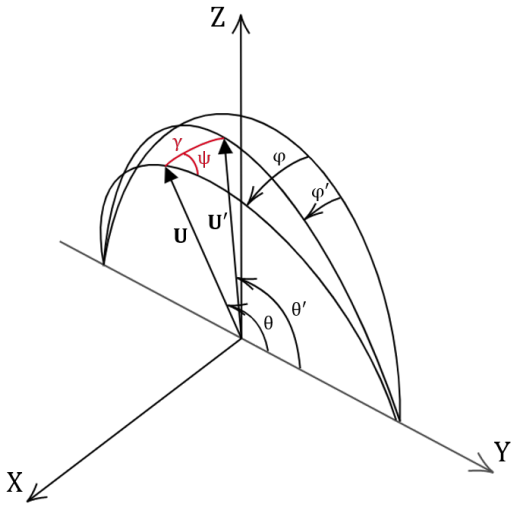


FIGURE 3.3: Deflection of the incoming velocity vector **U** by means of the angles $\gamma$ and $\psi$

## 3.2 Resonant returns and b-plane circles

The here reported summary follows what presented in Valsecchi et al. [8]and later refined by the same authors in [11].

When a close approach alters the particle's orbit in such a way that the orbital period of the two objects becomes commensurable, then a further encounter takes place after h periods of the particle and k of the planet have passed, with h, k integers. Identifying with T the particle's orbital period, the resonance condition can be expressed as follows:

$$hT'_{obj} = kT_{pl} \tag{3.17}$$

Using Jacobi normalised units, $T_{pl} = 2\pi$ and $T'_{obj} = 2\pi(a')^{3/2}$, and the resonant post-encounter semi-major axis can be directly linked to the resonance ratio k/h

$$a' = \left(\frac{k}{h}\right)^{2/3} \tag{3.18}$$

Recalling Equation (3.7), a given resonance ratio can be directly linked to a geometrical condition in terms of the angle $\vartheta$.

### 3.2.1 B-plane circles

Considering Equations (3.13) and (3.14), Equation (3.16) can be rearranged as

$$\cos\vartheta' = \frac{b^2 - c^2}{b^2 + c^2}\cos\vartheta + \frac{2c\zeta}{b^2 + c^2}\sin\vartheta \tag{3.19}$$

which can be solved for $\zeta$

$$\zeta = \frac{\left(b^2 + c^2\right)\cos\vartheta'}{2c\sin\vartheta} - \frac{\left(b^2 - c^2\right)\cos\vartheta}{2c\sin\vartheta} \tag{3.20}$$

The impact parameter can be replaced by means of Equation (3.10) and, rearranging the terms it is obtained

$$\xi^2 + \zeta^2 - \frac{2c\sin\vartheta}{\cos\vartheta' - \cos\vartheta}\zeta + \frac{c^2\left(\cos\vartheta' + \cos\vartheta\right)}{\cos\vartheta' - \cos\vartheta} = 0 \tag{3.21}$$

that is the equation of a circle in the b-plane centred on the $\zeta$-axis. Calling with C and R respectively the centre and the radius of such circle, Equation (3.21) can be written in a more compact way:

$$\xi^2 + \zeta^2 - 2C\zeta + C^2 = R^2 \tag{3.22}$$

where

$$C = \frac{c\sin\vartheta}{\cos\vartheta' - \cos\vartheta} \quad R = \left|\frac{c\sin\vartheta'}{\cos\vartheta' - \cos\vartheta}\right| \tag{3.23}$$

From the last equation one notices that $C > 0$ when $\cos\vartheta' > \cos\vartheta$, which, recalling Equation (3.7), occurs when $a' > a$. As a consequence, all the circles centred on the positive $\zeta$-axis lead to a larger post-encounter semi-major axis than the initial one, and vice versa. A interesting limit condition is when $a' = a$, i.e. the encounter modifies the orbital eccentricity and inclination while keeping constant the orbital period, a condition known as "orbit cranking" [19]. In this case, the circle degenerates into a straight horizontal line in the b-plane of equation:

$$\zeta = \frac{c\cos\vartheta}{\sin\vartheta}. \tag{3.24}$$

This kind of manoeuvre was implemented for example in the "cranking down" phase of Cassini's orbit, where a sequence of pure cranking flybys with Titan was used to quickly reduce the orbital inclination of the spacecraft, making the orbit increasingly close to the ring plane of Saturn. Considering Equation (3.23) and recalling the link between semi-major axis and angle $\vartheta$, it can be

noted that the smaller a circle, the larger the variation of semi-major axis between the pre- and post-encounter orbit.

In Figure 3.4 several b-plane circles are computed for an encounter with Venus and for different resonance conditions, assuming a planetocentric velocity of U = 0.5. The color bar reflects the number of revolutions of the planet before a new close approach, whereas the black circle represents the image of Venus augmented due to the gravitational focusing, which is approximately 1.2 times the radius of the planet.

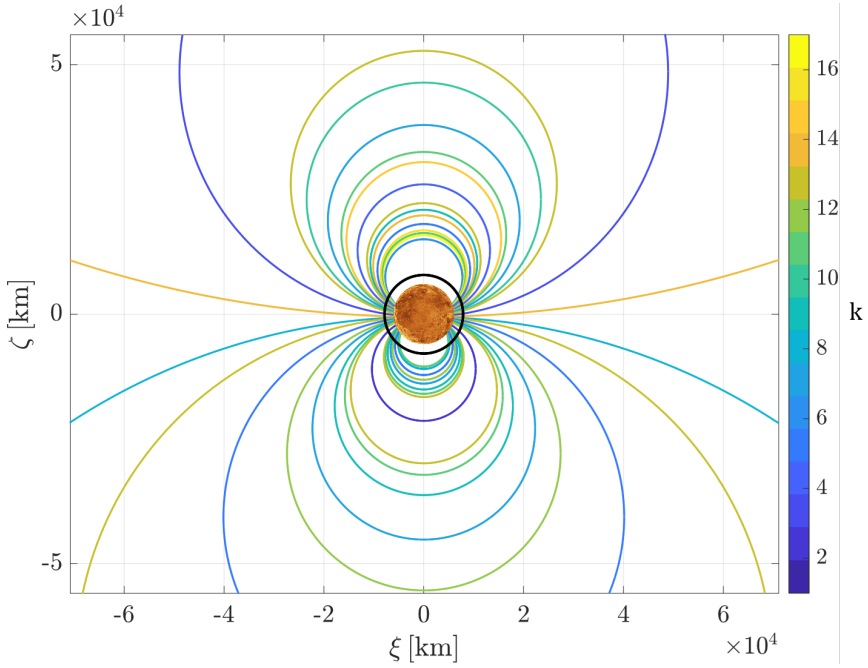


FIGURE 3.4: B-plane circles for a resonant encounter with Venus

For deeper insight into the b-plane circles, Figure 3.5 gives an example of post-encounter orbits computed by choosing different points on a same circle. Each point and its associated orbit are marked with the same colour. Since all the points belong to the same circle, what changes between the various solutions is the eccentricity and inclination, while the semi-major axis is the same. It can be noted that the closer the point is to the planet, the larger the variation in inclination with respect to the initial orbit. This provides a good rule of thumb and follows a rather intuitive logic, given that a stronger gravitational force has a greater ability to modify the orbital plane of an object.

For a visual comparison, Figure 3.6 shows what happens by choosing different circles and the same polar coordinate on each circle. This time, the semi-major axis also changes, as well as the inclination and eccentricity. Having considered the circles in the positive part of the plane, in any case the post-encounter orbit features a larger semi-major axis than the initial one. It can be seen that the largest orbits are associated with the points on the smallest circles, as further evidence of what stated before.

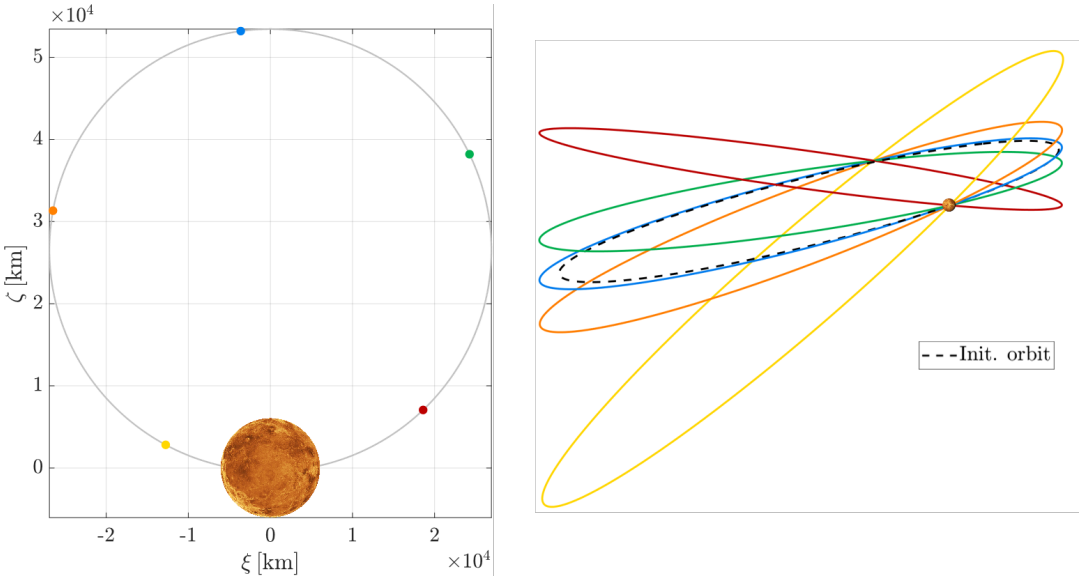


FIGURE 3.5: Post-encounter orbits for different points on a same b-plane circle

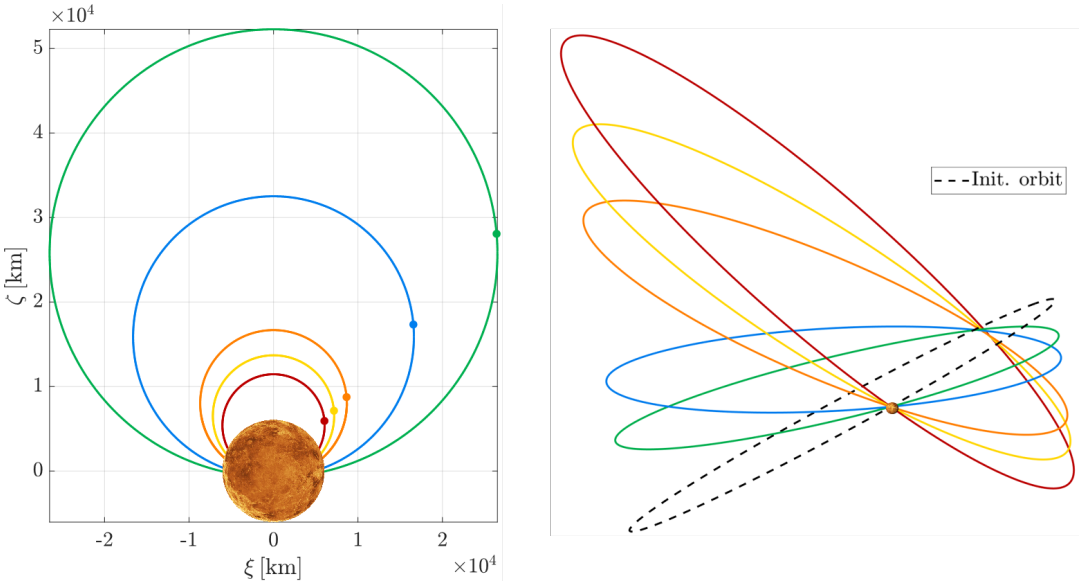


FIGURE 3.6: Post-encounter orbits for different b-plane circles

# Chapter 4

# Extention of the theory to elliptical planetary orbits

This chapter presents a generalisation of the formulae described in Chapter 3 for elliptical planetary orbits. The aim is to improve the accuracy of the classical model through a purely analytical extension, so that its implementation does not increase the computational costs compared to those required by the classical theory. This is a key aspect since one of the main advantages of modelling flybys in the b-plane is the possibility to analyse trajectories with fair accuracy while requiring very low computational efforts.

## 4.1 Generalisation of the formulae

The block diagram of Figure 4.1 summarises the main steps required to design a generic flyby using the the b-plane. The dashed blue boxes highlight

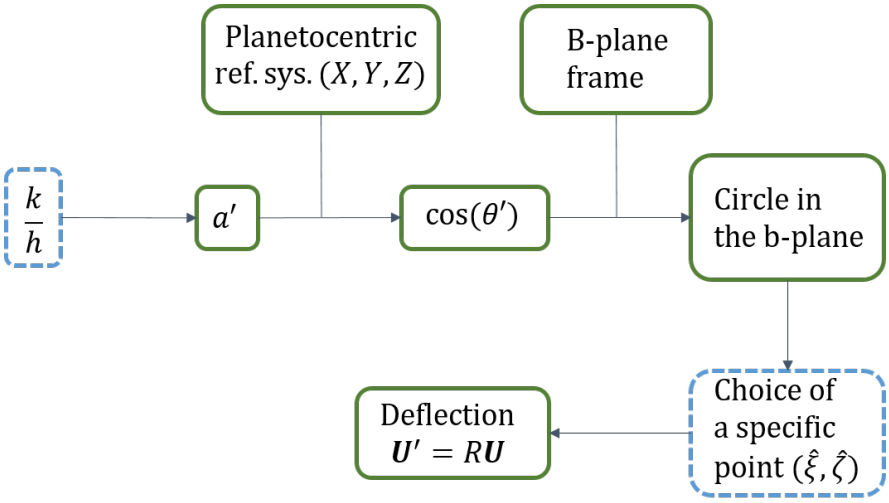


FIGURE 4.1: Sequence of steps required to design a resonant flyby in the b-plane

the free design variables, i.e. the k/h ratio and the point on the b-plane circle. Due to the patched-conics approximation, the entry point of a spacecraft into a planet's SOI is a degree of freedom of the problem. Theoretically, any point

on the b-plane could be chosen to analyse the evolution of the spacecraft's trajectory. However, if the target is an orbit with a specific post-encounter semi-major axis, the locus of points available drastically reduces to a simple circle. To generalise the model to elliptical planetary orbits, it is necessary to revise all the intermediate steps shown in the figure, leading from an input k/h ratio to the deflection of the incoming velocity **U**.

### 4.1.1 Notation and non-dimensional quantities

The notation used to derive the formulae is introduced, together with the characteristic units used to formulate the problem in non-dimensional coordinates. The notation is as follows:

- $a_{pl}, e_{pl}, i_{pl}, \Omega_{pl}, \omega_{pl}, \vartheta_{pl}$ : orbital elements of the planet referred to its orbital plane. This means that $i_{pl} = 0$. If the planet's orbital elements are available in another reference system, say the ecliptic, they first have to be converted in cartesian coordinates, which can then be expressed in the new frame through a rotation matrix;
- $a, e, i, \Omega, \omega, \vartheta$ : orbital elements of the particle's heliocentric orbit referred to the planet's orbital plane. In such reference frame, the particle can only cross a planet's orbit at the ascending or descending node, whereby a close encounter can take place at one of those points only. Both e and i can be in general non-zero, as per Öpik's classical theory;
- **v**: heliocentric velocity vectors;
- **U**: particle's velocity vector relative to the planet;
- $\bar{(\cdot)}$: non-dimensional variables. For instance, $\bar{\mathbf{U}}$ is the non-dimensional planetocentric velocity vector of the particle;
- $(\cdot)'$: post-encounter quantities.

Consider a close approach between a massless particle in orbit around the Sun and a planet with an elliptical orbit. The problem is once again formulated in non-dimensional form by using the same reference quantities as before, with the exception of the units of length and velocity that are now defined:

- $l_{ref}$: planet's distance from the Sun. Referring to Figure 4.2, $l_{ref} = \|\mathbf{r}_{pl}\|$. Note that this is now generally different from the planet's semi-major axis;
- $v_{ref}$: velocity of a fictitious planet moving on a circular orbit at a distance $l_{ref}$ from the Sun. It can then be computed as:

$$v_{ref} = \sqrt{\frac{\mu_\odot}{l_{ref}}} \tag{4.1}$$

where $\mu_\odot$ is the Sun's gravitational parameter. Clearly, the reference velocity is now different from the magnitude of the true planet's velocity $\|\mathbf{v}_{pl}\|$.

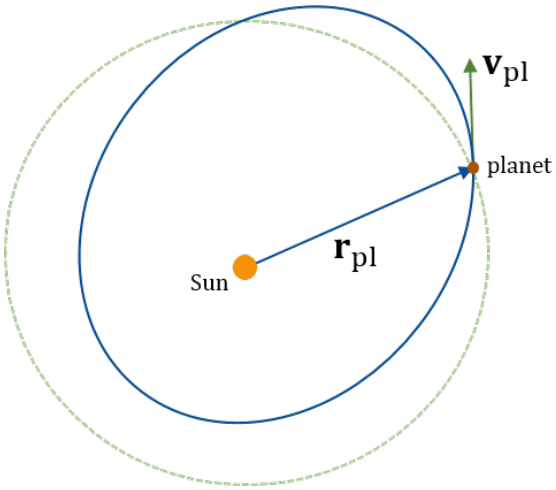


FIGURE 4.2: Blue solid line: planet's orbit. Dashed green line: circular orbit of a fictitious planet located at a distance $l_{ref}$ from the Sun

A parameter $\chi$ is introduced:

$$\chi = \frac{l_{ref}}{a_{pl}} \tag{4.2}$$

In case of circular orbit of the secondary body, $\chi = 1$. For particularly elliptical planets such as Mercury, $\chi$ reaches a minimum of 0.83.

### 4.1.2 Relationship between a given resonance ratio and the post-encounter semi-major axis

The relationship between a given k/h ratio and the particle's post-encounter semi-major axis can be extended for non-circular planetary orbits starting from Equation (3.17), which is independent of the planet's orbital eccentricity. The orbital periods of planet and particle respectively are:

$$T_{pl} = 2\pi\sqrt{\frac{a_{pl}^3}{\mu_\odot}}; \qquad T' = 2\pi\sqrt{\frac{a'^3}{\mu_\odot}}; \tag{4.3}$$

Plugging these expressions into Equation (3.17) yields:

$$\left(\frac{a'}{a_{pl}}\right)^{3/2} = \frac{k}{h} \quad \longrightarrow \quad \frac{a'}{a_{pl}} = \left(\frac{k}{h}\right)^{2/3} \tag{4.4}$$

and dividing both sides of the equation by $l_{ref}$ leads to:

$$\frac{1}{l_{ref}} \cdot \frac{a'}{a_{pl}} = \frac{1}{l_{ref}} \cdot \left(\frac{k}{h}\right)^{2/3} \quad \longrightarrow \quad \boxed{\bar{a}' = \frac{1}{\chi}\left(\frac{k}{h}\right)^{2/3}} \tag{4.5}$$

The only difference with Equation (3.18) is the presence of $\chi$ in the denominator.

### 4.1.3 Planetocentric reference frame

If the planet was in circular orbit, Equation (3.7) would provide a geometrical definition associated with the semi-major axis. For elliptical orbits, an analogue of that relation can be conveniently obtained if the encounter geometry is characterised with respect to a new planetocentric reference frame $(\hat{X}, \hat{Y}, \hat{Z})$, where the $\hat{Y}$-axis is aligned with the planet's velocity vector, the $\hat{Z}$-axis coincides with the Z-axis and the $\hat{X}$-axis completes a right handed reference system. Note that generally the $\hat{Y}$ direction is not perpendicular to the radial direction. As clear from Figure 4.3, the new frame can be obtained by

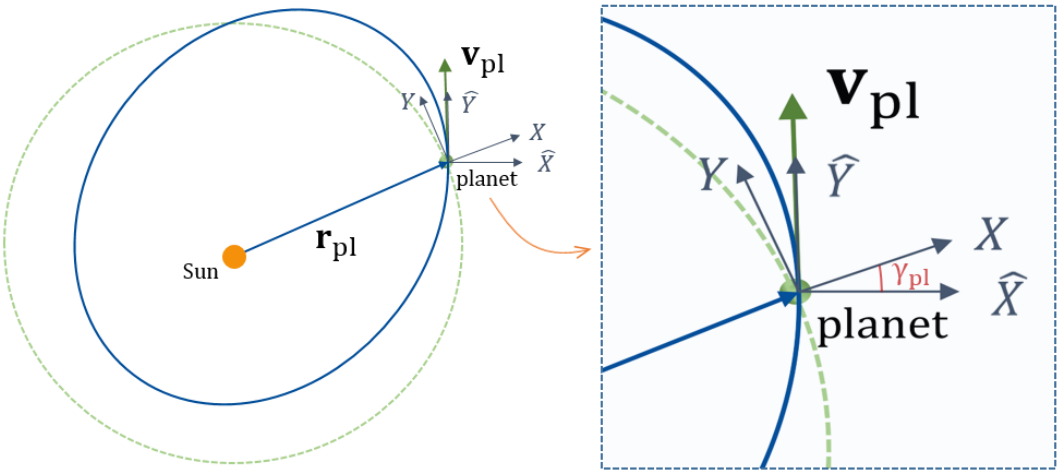


FIGURE 4.3: Comparison between the reference systems (X, Y, Z) and $(\hat{X}, \hat{Y}, \hat{Z})$.

performing a clockwise rotation of the X, Y-axes around the Z direction. The rotation angle is given by the planet's flight path angle $\gamma_{pl}$, defined as [20]:

$$\gamma_{pl} = \tan^{-1}\left(\frac{e_{pl}\sin(\vartheta_{pl})}{1 + e_{pl}\cos(\vartheta_{pl})}\right) \tag{4.6}$$

Then the following coordinate transformations can be used to rotate a vector quantity from one frame to the other:

$$\mathbf{x}|_{\hat{X},\hat{Y},\hat{Z}} = \begin{bmatrix} \cos(\gamma_{pl}) & -\sin(\gamma_{pl}) & 0 \\ \sin(\gamma_{pl}) & \cos(\gamma_{pl}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}|_{X,Y,Z} = R_Z(-\gamma_{pl})\mathbf{x}|_{X,Y,Z} \tag{4.7}$$

$$\mathbf{x}|_{X,Y,Z} = R_Z^T(-\gamma_{pl})\mathbf{x}|_{\hat{X},\hat{Y},\hat{Z}}$$

Note that the two frames only coincide when the planet is at the apsides, i.e. where $\gamma_{pl} = 0^\circ$, or when the planet is in circular orbit.

### 4.1.4 Link between a semi-major axis and the angle θ

The encounter geometry is then characterised by introducing again two angles $\vartheta$ and $\varphi$, with definitions analogous to those of Equation (3.4) but referred now with respect to the $(\hat{X}, \hat{Y}, \hat{Z})$ reference frame. Thus, $\vartheta$ identifies the angle formed by **U** with the $\hat{Y}$-axis, whereas $\varphi$ is the angle between the $\hat{X}\hat{Y}$ plane and the plane containing **U** and the $\hat{Y}$-axis.
Although it is possible to find an expression of $\varphi$ as a function of the particle's orbital elements, the focus here is on the link between the semi-major axis and $\vartheta$, as this is what is needed to generalise the classical theory (Figure 4.1).

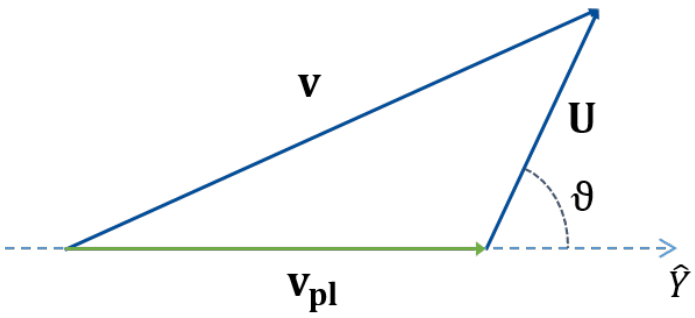


FIGURE 4.4: Angle $\vartheta$

Referring to Figure 4.4, the particle's heliocentric velocity can be obtained by applying the cosine law to the triangle formed by the three vectors:

$$\begin{aligned} v^2 &= v_{pl}^2 + U^2 - 2\,U\,v_{pl}\cos(\pi - \vartheta) \\ &= v_{pl}^2 + U^2 + 2\,U\,v_{pl}\cos\vartheta \end{aligned} \tag{4.8}$$

Equation (4.8) can be expressed in non-dimensional units by dividing both sides by the square of the characteristic velocity:

$$\begin{aligned} \frac{v^2}{v_{ref}^2} &= \frac{v_{pl}^2}{v_{ref}^2} + \frac{U^2}{v_{ref}^2} + 2\,\frac{U}{v_{ref}}\,\frac{v_{pl}}{v_{ref}}\,\cos(\vartheta); \\ \rightarrow \bar{v}^2 &= \frac{v_{pl}^2}{v_{ref}^2} + \bar{U}^2 + 2\,\bar{U}\,\frac{v_{pl}}{v_{ref}}\,\cos(\vartheta) \end{aligned} \tag{4.9}$$

For a planet in circular orbit, the ratio $v_{pl}/v_{ref}$ is equal to 1 and the original expression is obtained (Equation (3.7)), whereas for the elliptical cases, $v_{pl}$ can be expressed as function of $v_{ref}$. From the principle of conservation of energy:

$$v_{pl} = \sqrt{\mu_\odot \left(\frac{2}{r_{pl}} - \frac{1}{a_{pl}}\right)} \tag{4.10}$$

Collecting the term $1/r_{pl}$ under the square root and recalling the definition of $\chi$:

$$v_{pl} = \sqrt{\frac{\mu_\odot}{r_{pl}}\left(2 - \frac{r_{pl}}{a_{pl}}\right)} = \sqrt{\frac{\mu_\odot}{r_{pl}}\left(2-\chi\right)} \tag{4.11}$$

Equation (4.1) can be used to obtain:

$$v_{pl} = v_{ref}\sqrt{2-\chi} \quad \Rightarrow \quad \frac{v_{pl}}{v_{ref}} = \sqrt{2-\chi} \tag{4.12}$$

The last identity is used to rewrite Equation (4.9) as function of $\chi$:

$$\bar{v}^2 = 2 - \chi + \bar{U}^2 + 2\bar{U}\sqrt{2-\chi}\cos(\vartheta) \tag{4.13}$$

Exploiting again the principle of conservation of energy, the particle's heliocentric velocity is computed as:

$$\begin{aligned} v^2 = \mu_\odot\left(\frac{2}{r} - \frac{1}{a}\right) = \frac{\mu_\odot}{r}\left(2 - \frac{r}{a}\right) = v_{ref}^2\left(2 - \frac{1}{\bar{a}}\right) \\ \Rightarrow \quad \bar{v}^2 = 2 - \frac{1}{\bar{a}} \end{aligned} \tag{4.14}$$

The two expressions of $\bar{v}^2$ of Equations (4.13) and (4.14) can be matched:

$$2 - \chi + \bar{U}^2 + 2\bar{U}\sqrt{2-\chi}\cos(\vartheta) = 2 - \frac{1}{\bar{a}} \tag{4.15}$$

and solving for $\vartheta$ yields:

$$\boxed{\cos(\vartheta) = \frac{\chi - 1/\bar{a} - \bar{U}^2}{2\bar{U}\sqrt{2-\chi}}} \tag{4.16}$$

Note that Equation (4.16) is a generalisation of Equation (3.7), since $\chi = 1$ provides exactly the original expression from Öpik. It is stressed again that this result could only be achieved by referring $\vartheta$ to the planetocentric reference system introduced in the previous section.

### 4.1.5 Components of the planetocentric velocity U

When a particle encounters a planet moving on a circular orbit around the Sun, its relative velocity can be computed as function of the semi-major axis, eccentricity and inclination only. This result can be easily extended to elliptical planetary orbits, provided that the orbital position of the planet at the encounter is known.

Following Öpik [7], as a first step the particle's heliocentric velocity vector is derived in the (X, Y, Z) reference system. Figure 4.5 shows why this frame is convenient in this case. Regardless of the eccentricity of the planet's orbit, the X-axis is always aligned with the radial direction of the particle's orbit, whereas the transverse direction forms an angle i with the X, Y plane. Therefore, the vector **U** is easily found in X, Y, Z-coordinates. The angular

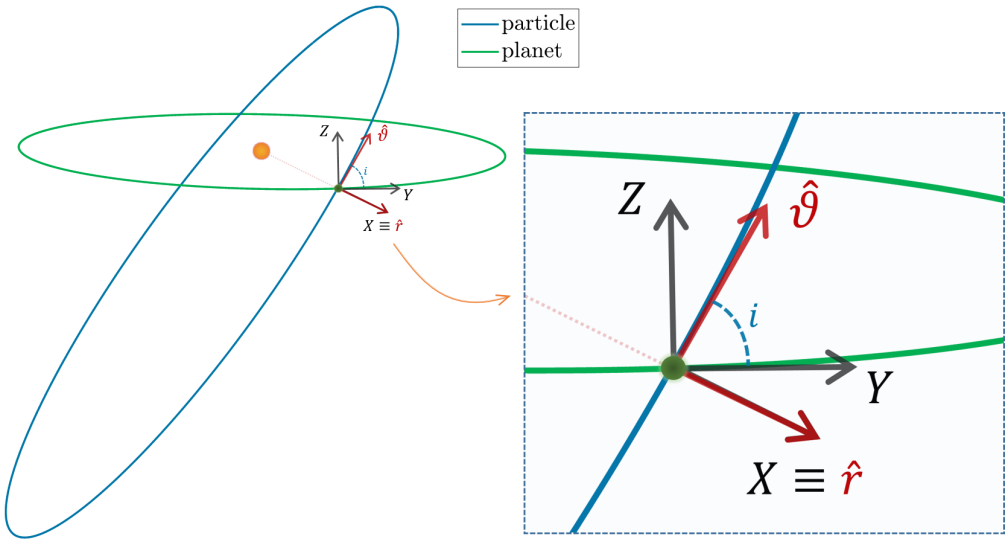


FIGURE 4.5: Radial and transverse direction of the particle's orbit at the encounter position

momentum per unit mass of a Keplerian orbit is given by:

$$h = \sqrt{\mu_\odot \, a \, (1 - e^2)} \tag{4.17}$$

or, alternatively:

$$h = v_t\, r \tag{4.18}$$

where $v_t$ is the transverse component of the velocity. The two expressions can be used to obtain $v_t$:

$$v_t = \frac{\sqrt{\mu_\odot\, a\,(1-e^2)}}{r} \tag{4.19}$$

which can be expressed in non-dimensional units as:

$$\begin{aligned} v_t = \sqrt{\frac{\mu_\odot\, a\,(1-e^2)}{r^2}} = \sqrt{\frac{\mu_\odot}{r}}\sqrt{\frac{a}{r}\,(1-e^2)} = v_{ref}\sqrt{\bar{a}\,(1-e^2)} \\ \Rightarrow \quad \bar{v}_t = \sqrt{\bar{a}\,(1-e^2)} \end{aligned} \tag{4.20}$$

Since the radial direction is always aligned with the X-axis, it results:

$$\bar{v}_r^2 \equiv \bar{v}_X^2 = \bar{v}^2 - \bar{v}_t^2 \tag{4.21}$$

and by substituting Equation (4.14) and (4.20) into Equation (4.21) leads to:

$$\bar{v}_X^2 = 2 - \frac{1}{\bar{a}} - \bar{a}\,(1-e^2) \quad \Rightarrow \quad \bar{v}_X = \pm\sqrt{2 - \frac{1}{\bar{a}} - \bar{a}\,(1-e^2)} \tag{4.22}$$

where $\pm$ distinguishes encounters from inside and outside planet's orbit. As the radial and transverse directions are orthogonal, $v_t$ is decomposed in the Y, Z plane as:

$$\begin{cases} \bar{v}_Y = \bar{v}_t\, \cos i = \sqrt{\bar{a}\,(1-e^2)}\, \cos i \\ \bar{v}_Z = \pm\bar{v}_t\, \sin i = \pm\sqrt{\bar{a}\,(1-e^2)}\, \cos i \end{cases} \tag{4.23}$$

where $\pm$ identifies ascending or descending node.
Given the absolute velocity $\bar{\mathbf{v}}$, the relative velocity $\bar{\mathbf{U}}$ is obtained as the vector difference between $\bar{\mathbf{v}}$ and the non-dimensional planet's velocity $\bar{\mathbf{v}}_{pl}$, always expressed in the frame (X, Y, Z). Referring to Figure 4.3, $\mathbf{v}_{pl}$ can be written as:

$$\mathbf{v}_{pl} = \|\mathbf{v}_{pl}\| \begin{bmatrix} \sin\gamma_{pl} \\ \cos\gamma_{pl} \\ 0 \end{bmatrix} \tag{4.24}$$

and Equation (4.12) can be used to switch to non-dimensional coordinates:

$$\bar{\mathbf{v}}_{pl} = \begin{bmatrix} \sqrt{2-\chi}\,\sin\gamma_{pl} \\ \sqrt{2-\chi}\,\cos\gamma_{pl} \\ 0 \end{bmatrix} \tag{4.25}$$

and $\bar{\mathbf{U}}$ is finally given by:

$$\bar{\mathbf{U}} = \bar{\mathbf{v}} - \bar{\mathbf{v}}_{pl} = \begin{bmatrix} \pm\sqrt{2-\frac{1}{\bar{a}}-\bar{a}\,(1-e^2)} - \sqrt{2-\chi}\,\sin\gamma_{pl} \\ \sqrt{\bar{a}\,(1-e^2)}\,\cos i - \sqrt{2-\chi}\,\cos\gamma_{pl} \\ \pm\sqrt{\bar{a}\,(1-e^2)}\,\sin i \end{bmatrix} \tag{4.26}$$

which can be expressed in the $(\hat{X}, \hat{Y}, \hat{Z})$ frame through the rotation matrix of Equation (4.7). Compared to the case of circular orbit, the elements $a, e$ and $i$ are no longer sufficient to compute $\bar{\mathbf{U}}$, but the knowledge of $\chi$ and $\gamma_{pl}$ is also required. Note that, once $\bar{\mathbf{U}}$ is known, Equation (3.4) can be used to recover the angle $\varphi$.

### 4.1.6 B-plane circles and deflection model

The remaining blocks of Figure 4.1 to be generalised are those associated with the b-plane reference system. Since the eccentricity of a planet's orbit does not enter the definition of b-plane, Equation (3.9) can be used to compute the $(\xi, \eta, \zeta)$-axes also in this case.

The deflection model does not require major modifications either, as it was not specifically developed to account only for circular planetary orbits. Indeed, prior to Öpik's work on close encounters, the concepts of impact parameter and deflection angle had already been used for some time in astrodynamics to model flybys in the general case of planets in elliptical orbits [21]. The only necessary changes to the deflection equations are due to the use of characteristic units different from those used by Öpik. In other words, even though the deflection equations were derived basing on the general model of two-body hyperbolic trajectory, where the planet is not necessarily in a circular orbit, the use of $l_{ref}$ and $v_{ref}$ in the non-dimensionalisation step might lead to different expressions. However, starting from the equations written in dimensional form and using the characteristic units introduced in this chapter yields the following identities:

$$\cos\gamma = \frac{\bar{b}^2 - \bar{c}^2}{\bar{b}^2 + \bar{c}^2}; \quad \sin\gamma = \frac{2\bar{b}\bar{c}}{\bar{b}^2 + \bar{c}^2} \tag{4.27}$$

$$\tan\frac{\gamma}{2} = \frac{\bar{m}}{\bar{b}\bar{U}^2} \tag{4.28}$$

$$\begin{bmatrix} \bar{b}\sin\psi \\ \bar{b}\cos\psi \end{bmatrix} = \begin{bmatrix} \bar{\bar{\xi}} \\ \bar{\bar{\zeta}} \end{bmatrix} \tag{4.29}$$

which look the same of Equations (3.13) and (3.14). This was not a foregone conclusion, but is due to the particular choice made regarding the definition of $l_{ref}$ and $v_{ref}$. This statement can be verified by trying to use different characteristic units to reformulate the problem described in this chapter, such as $l_{ref} = \|\mathbf{r}_{pl}\|$ and $v_{ref} = \|\mathbf{v}_{pl}\|$. Despite the similarity, the above equations and those of the classical theory do not lead to the same results. In Equations (4.27), (4.28) and (4.29), the eccentricity of the planet's orbit is implicitly accounted for in the definitions of $\bar{U}$ and $\bar{b}$, and this in turn affects the final value of the deflection angle.

The last equation to be generalised for the deflection model is given by Equation (3.4), which provides a fundamental link between the deflection angle $\gamma$ and the post-encounter angle $\vartheta'$. As long as $\vartheta$ is referred to the planet's velocity, Equation (3.4) holds true. Therefore, since $\vartheta$ was defined with respect to the $\hat{Y}$-axis, Equation (3.4) can be used in the elliptical case with no modifications required.

In summary, because of the way the problem has been formulated, the non-dimensional equations associated with the deflection model are the same for both the cases of circular and elliptical orbit of the planet. Moreover, as can be seen in Section 3.2.1, the derivation of b-plane circles is solely based on those equations, thus implying that Equation (3.22) also remains unchanged. However, the classical and extended model do not lead to the same b-plane circles. Indeed, for the same pre- and post-encounter semi-major axes $a, a'$, the two methods yield different values of $\cos\vartheta$, $\cos\vartheta'$ and hence to different b-plane circles. As evidence of this, Figures 4.6 and 4.7 show several resonance circles computed using the two models. Each pair of close circles is associated with a same $k/h$ ratio, specified next to each of them. The flyby planet is Venus and the pre-encounter conditions are the same for all circles. It can be seen that the effect of the planet's orbital eccentricity is a slight modification of the radius and the centre of the circles.

Lastly, Table 4.1 provides a summary of the main differences between the classical model based on Öpik's theory and the one presented in this chapter. The deflection equations, the circle equations and some of the reference quantities are not reported as they are not different in the two cases. Since for circular orbits $\chi = 1$ and $\gamma_{pl} = 0°$, one can easily verify that in this case the two models yield the same results. This is to emphasise again that the eccentric extension is a generalisation of the classical model.

| | Classical | Eccentric extension |
|---|---|---|
| **Reference length and velocity** | $a_{pl}, v_{pl}$ | $r_{pl}, \sqrt{\frac{\mu_\odot}{r_{pl}}}$ |
| **Planetocentric ref. frame** | $(X, Y, Z)$ | $(\hat{X}, \hat{Y}, \hat{Z})$ |
| **Link** $k/h \leftrightarrow a'$ | $a' = \left(\frac{k}{h}\right)^{2/3}$ | $\bar{a}' = \frac{1}{\chi}\left(\frac{k}{h}\right)^{2/3}$ |
| **Link** $a \leftrightarrow \cos\vartheta$ | $\cos\vartheta = \frac{1-1/a-U^2}{2U}$ | $\cos\vartheta = \frac{\chi-1/\bar{a}-\bar{U}^2}{2\bar{U}\sqrt{2-\chi}}$ |
| **Vector $\bar{U}$** (X, Y, Z-frame) | $\begin{bmatrix} \pm\sqrt{2-1/a-a\left(1-e^2\right)} \\ \sqrt{a\left(1-e^2\right)}\cos i - 1 \\ \pm\sqrt{a\left(1-e^2\right)}\sin i \end{bmatrix}$ | $\begin{bmatrix} \pm\sqrt{2-\frac{1}{\bar{a}}-\bar{a}\left(1-e^2\right)}-\sqrt{2-\chi}\sin\gamma_{pl} \\ \sqrt{\bar{a}\left(1-e^2\right)}\cos i-\sqrt{2-\chi}\cos\gamma_{pl} \\ \pm\sqrt{\bar{a}\left(1-e^2\right)}\sin i \end{bmatrix}$ |

TABLE 4.1: Classical theory and eccentric extension compared

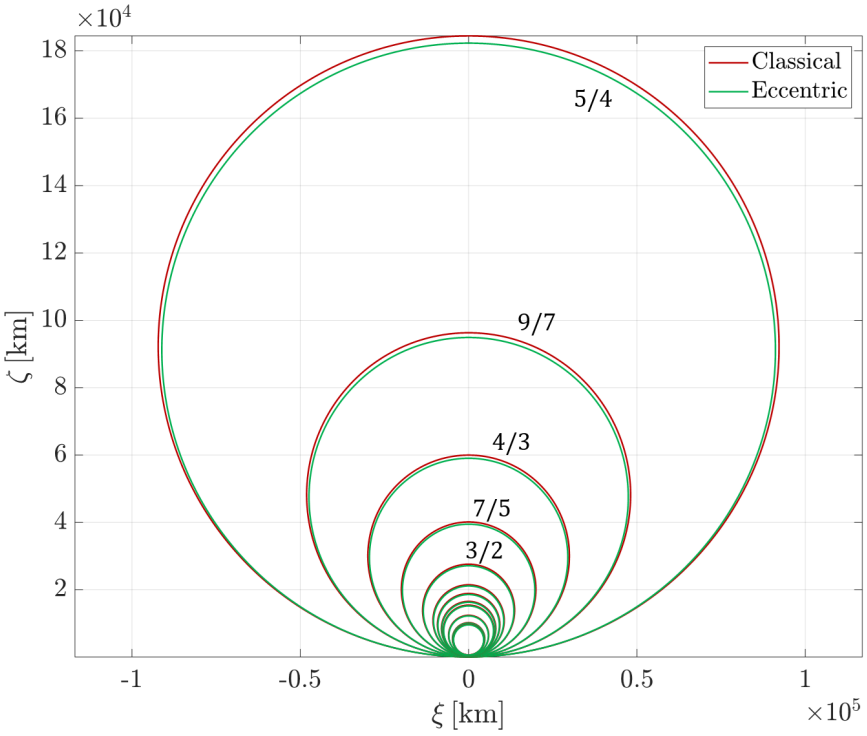


FIGURE 4.6: Resonance circles computed through the two methods

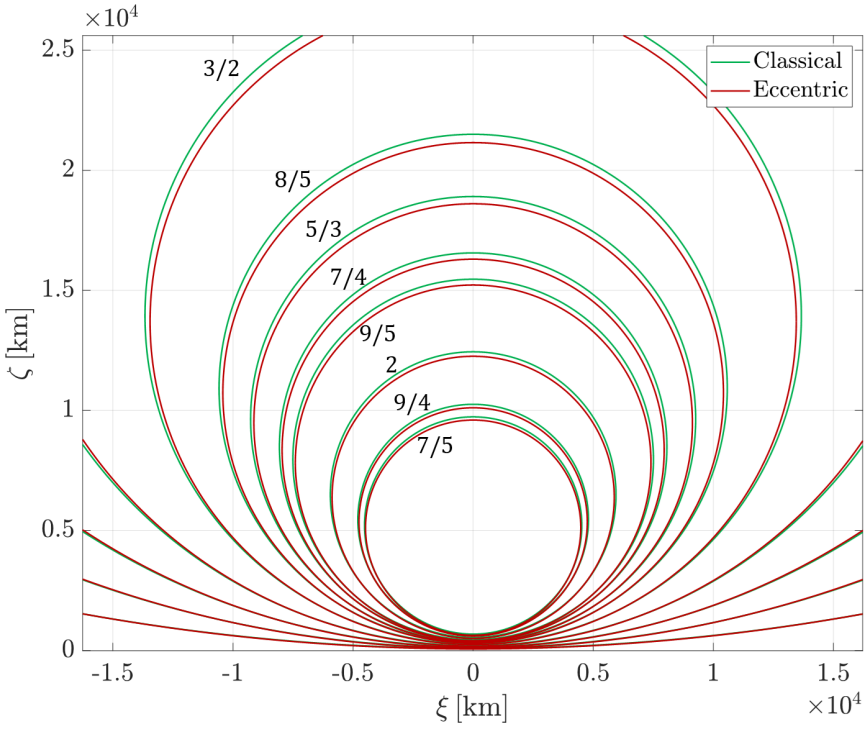


FIGURE 4.7: Enlargement of Figure 4.6. Different resonances are emphasised

## 4.2 Validation and comparison

The purpose of this section is twofold: validating the model and comparing it with the classical one on some mission design application. The former goal can be achieved with a simple test: given a close approach between a spacecraft and a planet in elliptical orbit, some desired resonance ratio k/h is chosen for the post-encounter conditions. Then, a generic point on the resulting b-plane circle is selected and the outgoing velocity is computed. The extended model can be considered correct if the final orbit of the spacecraft and that of the planet meet the resonance condition specified at the beginning, regardless of the point chosen on the circle. On the other hand, the comparison test aims at understanding how different the results produced by the two models are in mission design applications.

The two tasks, validation and comparison, can be carried out all at once through the following single test. Consider the trajectory design of a spacecraft that is launched from Earth on a date $t_0$ and arrives at Venus after a time of flight $\Delta t$. Once at the planet, it is required to perform a sequence of five resonant flybys characterised by the post-encounter conditions specified in Tab 4.2. The two models are asked to find the sequence of b-plane points leading to the orbits reported in the table. Note that each pair of (k/h, inclination) is enough to define a unique post-encounter orbit. Indeed, each resonance ratio corresponds to a b-plane circle and the inclination values identify a single point on each circle. The procedure to identify the point corresponding to a given inclination value will be deeply detailed in Chapter 6.

| Flyby | k/h | Inclination |
|---|---|---|
| $1^{st}$ | 4/3 | 1.5° |
| $2^{nd}$ | 7/5 | 3° |
| $3^{rd}$ | 3/2 | 4.5° |
| $4^{th}$ | 8/5 | 6° |
| $5^{th}$ | 7/4 | 7.5° |

TABLE 4.2: Post-encounter orbits

The test was performed assuming $t_0$ = 6756 Modified Julian Days (MJD) 2000 and $\Delta t$ = 150 days. For the elliptical model, the position of Earth at launch and that of Venus at the encounter are taken from the ephemeris, and then the initial conditions at the first flyby are obtained by computing the Lambert arc related to the transfer orbit between the two planets. For the classical model, the only difference is in the position of Venus, which is assumed to be at the same heliocentric longitude as before but in circular orbit.

For the validation side of the test, the points predicted by the elliptical model correctly lead to the prescribed resonance conditions, giving proof of the accuracy of the equations described in this chapter. Figure 4.8 shows the points predicted through the two models. The number next to each couple of close

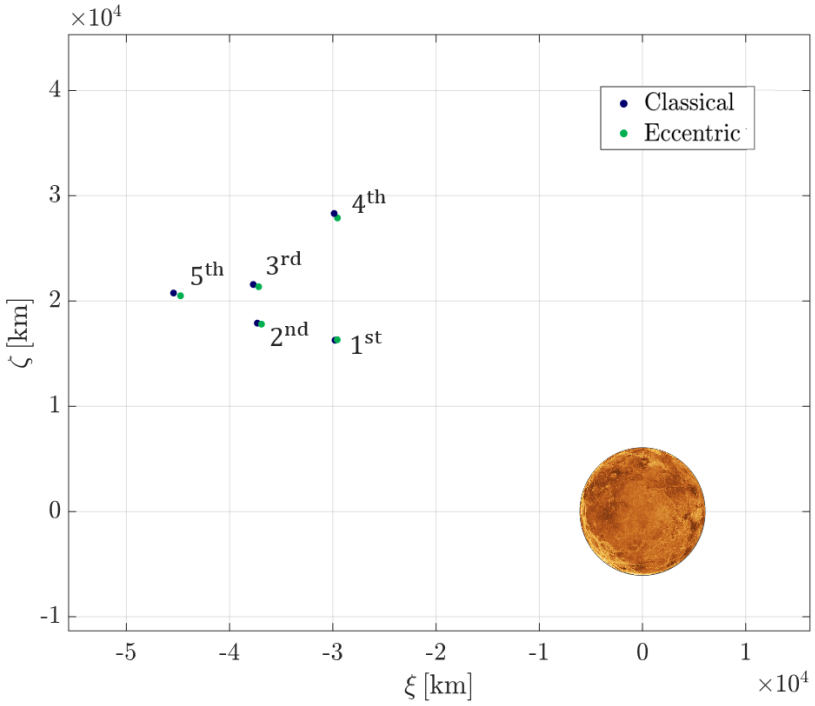


FIGURE 4.8: B-plane points for the flybys with Venus.

points refers to the flyby number. For the sake of comparison, all the points

are plotted in the same graph even though the orientation of the b-plane in space changes at each flyby. As can be seen, the two models yield very similar results.

To gain a better insight into the differences, consider the blue points in Figure 4.8, obtained using the classical model. If Venus was actually in circular orbit, those points would accurately lead to the orbits reported in Table 4.3. Instead, since Venus actually has a small but non-zero eccentricity, its real position and velocity will be somehow different from those used to calculate the blue points. Therefore, if these latter are used to compute the spacecraft trajectory, the resulting orbits are likely to slightly deviate from the desired ones. Table 4.3 shows how significant this deviation is. The second and third columns report the errors in terms of semi-major axis and inclination of the orbits. The first error inevitably translates into an error on the orbital period and, consequently, on the resonance ratio of the orbits. Therefore, on each subsequent return, the spacecraft will not exactly match the position of the planet, but will be either a little early or a little late, thus introducing another source of error. Anyway, Table 4.3 again highlights small differences in the outcomes of the two models. This might have been expected considering that Venus has the lowest orbital eccentricity among the planets of the solar system. Indeed, for Venus' orbit the parameter $\chi$ is comprised between $0.9932 \leq \chi \leq 1.0068$, which is why the differences between the two models are limited. However, as mentioned in the introduction, some missions involve series of flybys with planets with non-negligible orbital eccentricities and located near the apsides, where the orbit most deviates from the circular one. For this reason, it is interesting to see how the two models perform in such a scenario.

| **Flyby** | $\Delta$k/h [%] error | $\Delta a_{\%}$ [%] | $\Delta$i [deg] |
|---|---|---|---|
| **1st** | 0.075 | 0.067 | 0.021 |
| **2nd** | 0.214 | 0.122 | 0.041 |
| **3rd** | 0.266 | 0.199 | 0.053 |
| **4th** | 0.438 | 0.267 | 0.059 |
| **5th** | 0.629 | 0.354 | 0.048 |

TABLE 4.3: Deviations of the solutions obtained by the classical model with respect to the specified orbits

In the second test the spacecraft is required to perform three resonant flybys at Mars near its perihelion and with the following specifications:
The points computed through the elliptical model once again correctly satisfy the resonance conditions given as input. However, as can be noticed from Figure 4.9, the solutions produced by the two models are considerably

| Flyby | k/h | Inclination |
|---|---|---|
| **1st** | 4/5 | 1.5° |
| **2nd** | 6/7 | 3° |
| **3rd** | 9/10 | 4.5° |

TABLE 4.4: Post-encounter orbits (Mars test)

different this time.

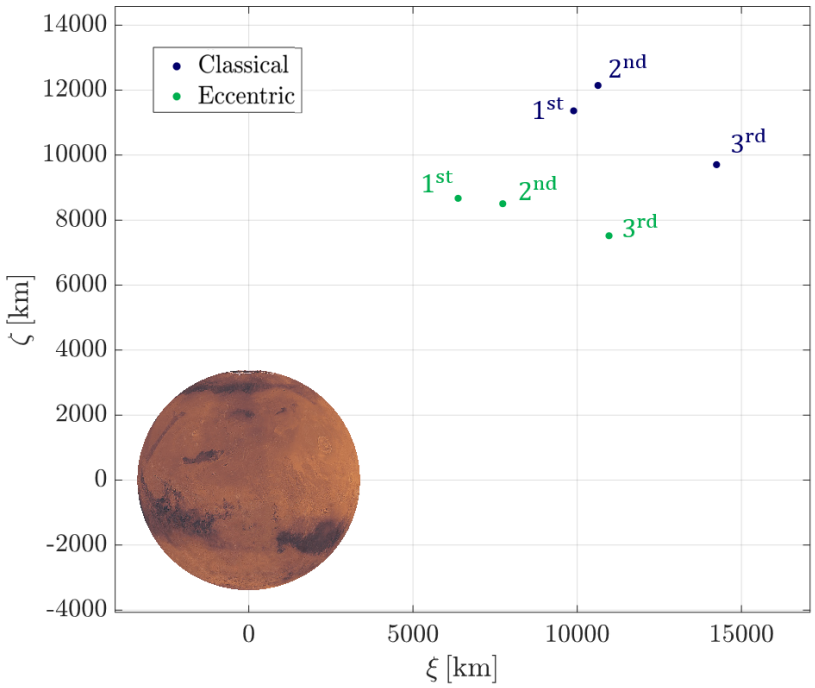


FIGURE 4.9: B-plane points for the flybys with Mars

When dropping the assumption of circular orbit of the planet, the outcomes of the classical model lead to significant errors, as reported in Table 4.5. The magnitude of the errors on the resonance condition is of particular concern, as it implies large time delays between the arrival of the spacecraft and the planet at the encounter position.

In summary, the equations derived in this chapter resulted correct, allowing to overcome the assumption of circular planetary orbits made by the classical model. When designing flybys at planets with a marked orbital eccentricity, adopting the classical model results in a poor approximation of the estimated trajectory, and therefore the use of the extended model is more advisable. In the other cases, the two models produce similar outcomes, suggesting that both are suitable in a preliminary design phase. Nevertheless, as the model extension was simply obtained by generalising the equations of the classical one, the two models require the same computational time. Therefore, one would never be wrong in always implementing the extended model.

| **Flyby** | $\Delta$k/h [%] error | $\Delta a_{\%}$ [%] | $\Delta$i [deg] |
|---|---|---|---|
| **1st** | 2.87 | 1.923 | 0.37 |
| **2nd** | 3.73 | 3.134 | 0.71 |
| **3rd** | 5.77 | 4.062 | 0.99 |

TABLE 4.5: Deviations of the solutions obtained by the classical model with respect to the specified orbits (Mars test)

## 4.3 On the conservation of U

Several authors have investigated the accuracy of Öpik's model in predicting the outcomes of close approaches (Greenberg et al. [10], Wetherill and Cox [22]). The general result was that the two-body behavior can approximate quite well the true nature of planetary encounters. Although the extended model has been obtained as a generalisation of the classical one, care must be taken before drawing similar conclusions about such model. In particular, some concern arises around one assumption on which the extended model is based, namely that the magnitude of the relative velocity is conserved across a close encounter. Before giving more details, a brief introduction to Tisserand's criterion is necessary.

During planetary close encounters, the orbital parameters of the small body change drastically. In this context, Tisserand's criterion is a particularly powerful concept, stating that a function of the semi-major axis, eccentricity and inclination stays approximately constant before and after the approach, thus restricting the possible changes in orbital parameters. The conserved quantity is the Tisserand parameter already introduced in Equation (3.2) and was originally derived from the Jacobi integral of the CR3BP in the particular case in which the primary body is much more massive than the secondary one, e.g. the Sun and a planet. For circular planetary orbits and in the approximation of zero-SOI, the incoming and outgoing velocities of a small body approaching a planet are a function of the Tisserand parameter only. Since the latter does not change during a flyby, the relative velocity is also conserved. In other words, Tisserand's criterion somehow supports the hypothesis of constant relative velocity underlying the patched conics model. However, the Tisserand parameter is not conserved for elliptical orbits [23], suggesting that U may be more subject to variations as well. Since the extended model however relies on the hypothesis of constant U, it is interesting to see to what extent the latter holds true for orbits that are far from circular. To this end, numerical tests are carried out in the following section to study the variation of the particle's velocity for different eccentricities of the planet's orbit.

### 4.3.1 Simulations setup

In each test, a close encounter between a massless particle and a planet moving in elliptical orbit around the Sun is simulated by using a full three-body dynamics model. The eccentricity of the planet's orbit is varied at each simulation, allowing to study the variation of the particle's velocity under different conditions.

#### Initial conditions for the integration

Firstly, a planet is chosen and, starting from its true orbit, the eccentricity is changed by selecting a value. Then, a position vector $\mathbf{r}_{pl}$ is randomly chosen on the orbit to identify the planet's position at the time of the encounter. The particle's position $\mathbf{r}$ at encounter is defined by choosing a point close to $\mathbf{r}_{pl}$ and that is inside the planet's SOI. More specifically, referring to Figure 4.10, $\mathbf{r}$ is obtained by taking the vectorial sum of $\mathbf{r}_{pl}$ and a planet-centred position vector $\Delta\mathbf{r}$ that has random orientation and a random magnitude ranging in between a minimum value $r_{min}$ and the radius of the SOI. Once $\mathbf{r}$ is known, a Keplerian orbit is generated for the particle such that it joins this position and has random values of semi-major axis, eccentricity and inclination, but still within reasonable limits. In practise, the orbits of the two objects feature a finite but small MOID.

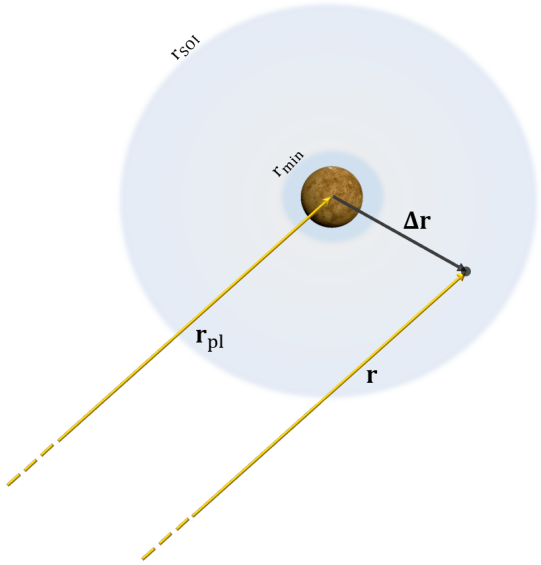


FIGURE 4.10: Particle's and planet's position at time of closest approach

$\mathbf{r}_{pl}$ and $\mathbf{r}$ are then propagated backward in time through a two-body model until the particle is well outside the planet's SOI, obtaining two initial position vectors $\mathbf{r}_{pl,0}$ and $\mathbf{r}_0$, as shown in Figure 4.11 on the left.

#### Numerical integration

The motion of the objects is propagated forward in time using $\mathbf{r}_{pl,0}$ and $\mathbf{r}_0$ as initial conditions and this time employing a full three-body dynamics model

for the particle:

$$\begin{cases} \dfrac{d\mathbf{r}(t)}{dt} = \mathbf{v}(t) \\ \dfrac{d\mathbf{v}(t)}{dt} = -\mu_{\odot} \dfrac{\mathbf{r}(t)}{\|\mathbf{r}(t)\|^3} - \mu_{pl} \dfrac{\mathbf{r}(t) - \mathbf{r}_{pl}(t)}{\|\mathbf{r}(t) - \mathbf{r}_{pl}(t)\|^3} \end{cases} \tag{4.30}$$

where $\mathbf{v}(t)$ is the particle's velocity and $\mu_{pl}$ is the planet's gravitational parameter. The integration is performed using Matlab®'s `ode113` and setting respectively $10^{-13}$ and $10^{-14}$ as relative and absolute tolerances. The stopping criterion for the integration is given by the SOI exit, as shown in Figure 4.11 on the right.

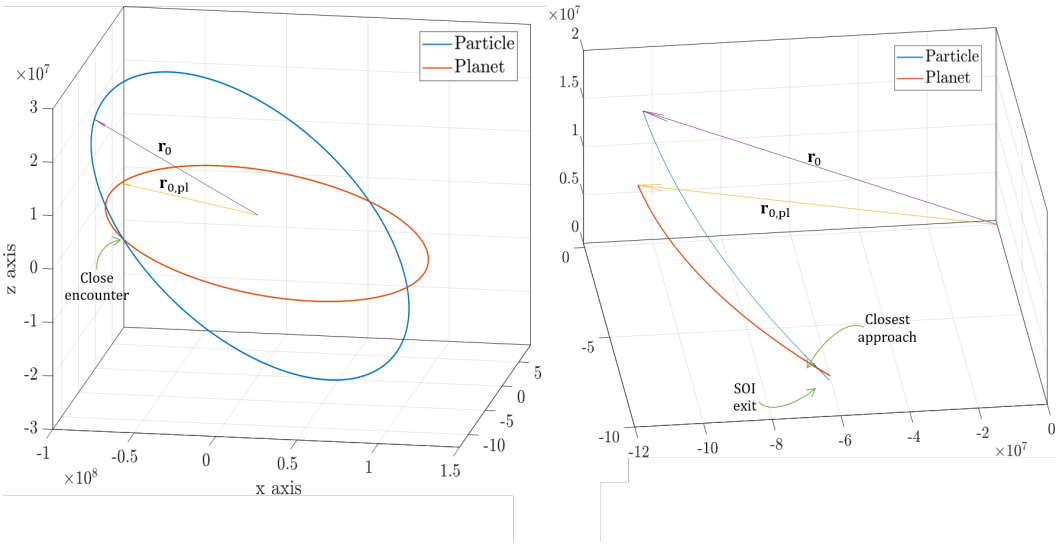


FIGURE 4.11: *Left:* vectors $\mathbf{r}_0$ and $\mathbf{r}_{pl,0}$ obtained by backward propagation of $\mathbf{r}$ and $\mathbf{r}_{pl}$. *Right:* forward propagation of $\mathbf{r}_0$ and $\mathbf{r}_{pl,0}$ using three-body model

Identifying with $t_{in}$ and $t_{out}$ the time instants at which the particle respectively enters and leaves the SOI, the particle's velocities $\mathbf{v}(t_{in})$ and $\mathbf{v}(t_{out})$ are retrieved from the integration of the trajectory. Then, the relative velocities $\mathbf{U}_{in}$ and $\mathbf{U}_{out}$ are obtained by subtracting the planet's velocity from $\mathbf{v}(t_{in})$ and $\mathbf{v}(t_{out})$. Finally, the following parameter of merit is computed to evaluate the magnitude variation between incoming and outgoing velocity:

$$\Delta U_{\%} = \frac{\|\mathbf{U}_{in}\| - \|\mathbf{U}_{out}\|}{\|\mathbf{U}_{in}\|} \cdot 100 \tag{4.31}$$

### 4.3.2 Simulations and results

The test involved the simulation of $2 \cdot 10^4$ close encounters between a particle and Jupiter. At each run, the orbital eccentricity of the planet is modified to a different value starting from 0 and eventually reaching 0.9. Each black dot in Figure 4.12 represents the result of one simulation, whereas the red

line is computed with a moving average method. As a first consideration, note that $\Delta U_{\%}$ is different than zero also for $e_{pl} = 0$. This is because, as previously mentioned, although the conservation of U holds true for circular orbits, it was strictly demonstrated under the hypothesis of zero-SOI, where the planet's velocity is considered constant during the relatively short time interval of the encounter. In reality, the change in planet's velocity between $t_{in}$ and $t_{out}$ directly affects the computation of $\mathbf{U}_{in}$ and $\mathbf{U}_{out}$. However, $\Delta U_{\%}$ seems approximately similar in all test cases, except for very large values of eccentricity, where the red curve shows a slight upward inflection. The causes of this effect can be understood by considering Fig 4.13, where $\Delta U_{\%}$ is plotted as a function of the true anomaly of the planet at the time of closest approach.

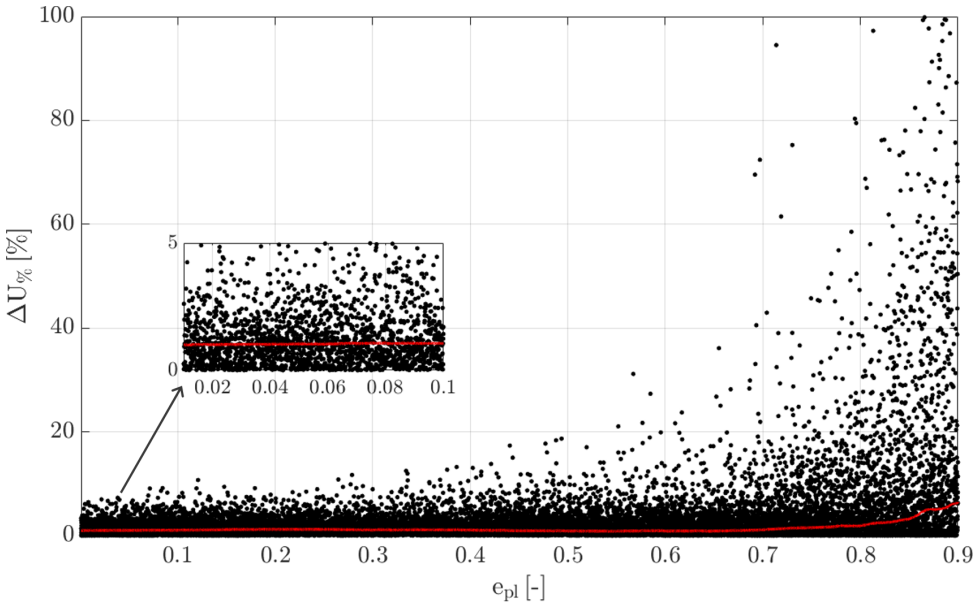


FIGURE 4.12: Variations between relative incoming and outgoing velocity as function of the orbital eccentricity of Jupiter

As can be seen, the largest velocity variations all occur when Jupiter is at perihelion, whereas the opposite happens at aphelion. This can be explained considering that near perihelion:

(i) the planet's velocity is larger, causing the relative velocity **U** to be smaller. Therefore, the encounter is slower and the particle spends more time inside the planet's SOI;

(ii) the Sun's gravitational force is on its greatest, and so also the planet's acceleration reaches its maximum value. Consequently, the planet's velocity will change more during the duration of the flyby.

The combination of these two factors results in larger variations of the particle's relative velocity.

Overall, the more elliptical an orbit is, the larger $\Delta U_{\%}$ will be for encounters near perihelion. However, as clear from Figure 4.12, this characteristic

becomes appreciable only for particularly large eccentricities. As none of the largest bodies in the solar system has a higher eccentricity than 0.25, the assumption of constant U underlying the extended model causes no problem in any practical application.

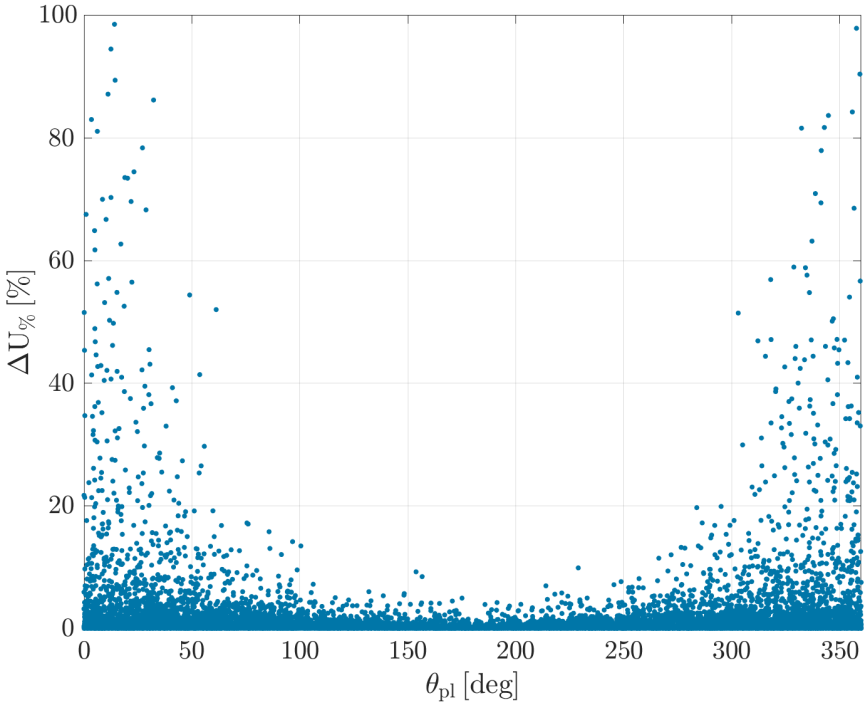


FIGURE 4.13: $\Delta U_{\%}$ vs planet's true anomaly

# Part II

# Dynamic programming approach to resonant flyby trajectory design

# Chapter 5

# Introduction

In mission design, resonant orbits have been extensively studied for several purposes and using different frameworks. In Anderson [17], heteroclinic connections between distinct resonant orbits have been chained together to obtain nearly ballistic trajectories in the CR3BP. Within this framework, Vaquero et al. [24] approached the preliminary design of a transfer to a Saturn's moon by exploiting the invariant manifolds associated with periodic resonant orbits. A strict relation was shown between the resonance transition mechanism and the concept of weak capture by Topputo et al. [25], who also studied the motion of resonant orbits for the Earth-Moon-Sun system within a four-body dynamics. More restricted to the patched conics model, many studies are available that developed and compared a variety of optimisation techniques for the design of multi-GAM trajectories including sequences of resonant orbits, such as Ceriotti [16] and Vasile and Locatelli [26]. Federici et al. [27] proposed a method based on the combined use of the Tisserand graph and differential evolution algorithms to design sequences of resonant orbits for a transfer between Jupiter and Europa. Vasile et al. [28] also investigated optimal transfers to Europa involving resonant flybys and including low-thrust manoeuvres.

The solution space of this kind of problems is very large, and efficient strategies are needed to quickly explore it during the early stages of the design. Few works can be found in literature that exploit the advantages of the b-plane for the design of resonant flybys. In this case, a convenient decision variable of the problem is the resonance condition between the spacecraft and the planet at each flyby, as it can easily be mapped on the b-plane. In this framework, Masat [29] implemented a multi-level optimisation scheme to solve the problem of reaching a target orbit in a fixed number N of resonant flybys. At the highest level of its strategy, a direct search algorithm finds an optimal splitting of the total $\Delta$v between the initial and target orbit into N contributions. At the lowest level, a brute-force approach is used at each flyby to select the resonant orbit that gets the closest to each sub-$\Delta$v. The proposed work expands on those results, including the number of flybys as optimisation variable and attempts to define a systematic procedure that does not rely on heuristic methods, which not necessarily discover the optimal solution. The choice of the resonance condition at each flyby introduces a discrete element in the design process that can be advantageously leveraged by methods of combinatorial optimisation. For this reason, the problem is

approached with the technique of dynamic programming, which provides a systematic method to make a sequence of interrelated decisions[30]. The developed algorithm is tested by reproducing the design of Solar Orbiter's resonant phase with Venus, used in the actual mission to gradually raise the ecliptic inclination and decrease the perihelion distance. The computational efficiency of the approach is evaluated by comparing it with an exhaustive enumeration approach.

This second part is outlined as follows:

- **Chapter 6** provides an introduction to dynamic programming and discusses an illustrative example, the shortest path problem.
- **Chapter 7** introduces the problem of unperturbed resonant flyby trajectories in the b-plane and proposes a dynamic programming approach.
- **Chapter 8** presents the application of the proposed algorithm to the preliminary design of Solar Orbiter and compares the computational efficiency of the method with the more standard brute-force approach.

# Chapter 6

# Dynamic programming

Dynamic programming provides a computationally efficient method for finding optimal solutions to problems that can be formulated as multi-stage decision processes, where the decisions made at one time influence the later available choices. It was first introduced by Richard Bellman in the 1950s and has since found application in a variety of disciplines, from engineering to economics. Examples of applications include landing aircraft, managing blood inventories, scheduling fleets of vehicles, and many more. In space mission design, several works have demonstrated the effectiveness of this technique for different optimisation problems. In Lin [31], a dynamic programming method is used to optimise the total propellant consumption required to control of the orbital altitude of a space station. The optimisation of low-thrust trajectories is a typical problem approached with dynamic programming, as it allows to reduce the high dimensional problem into a succession of low dimensional sub-problems. In this direction, Colombo et al. [32] proposed an efficient algorithm based on differential dynamic programming that computes an optimal feedback control law by discretising the dynamics in correspondence of a fixed number of decision times. Alternative strategies based on this approach can be found in Nugnes and Colombo [33] and Lantoine and Russel [34].

Unlike other mathematical programming techniques such as linear or quadratic programming, dynamic programming is not concerned with a specific class of problems, but can be approached by a variety of angles depending on the problem at hand. Indeed, it often takes a lot of creativity as well as experience to formulate a problem as a dynamic programming one. As there is no general definition of a "dynamic programming problem", most textbooks introduce the method by directly showing its application to some illustrative examples. In the following section, one of the most popular dynamic programming problems is presented so as to introduce the reader to the general notation and terminology, and explain the logic behind the approach. Despite the renowned performance of dynamic programming for stochastic problems, this chapter will focus on deterministic decision processes as they reflect the structure of the problem presented in the following chapter. For further exposure to stochastic control, refer to Bellman [35], Bertsekas [36], and Puterman [37].

## 6.1 Shortest path problem

Consider a driver who has to travel from an initial point P to a terminal point Q but can choose numerous alternative routes in between. Each path in the road network in Figure 6.1 is one possible strategy, and the objective is to find the shortest path possible. The nodes represent the possible *states* in which the system (i.e. the driver) may be at a certain time. In this particular example, the driver has a total of four decisions to make, called *stages* of the problem. The numbers in the graph indicate the distances between the nodes, and the total cost of a path is equal to the sum of all distances between the nodes crossed by that path.

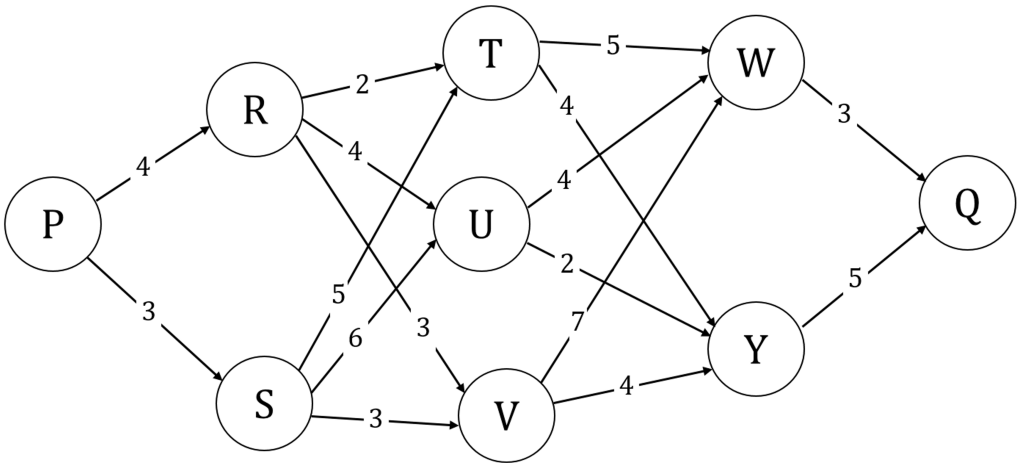


FIGURE 6.1: Road network

Tackling the problem with a naïve greedy approach would lead to the solution P $\rightarrow$ S $\rightarrow$ V $\rightarrow$ Y $\rightarrow$ Q, with a total distance of 15. However, this is not the best solution, and a different method is required that considers the overall cost of a sequence of nodes, rather than blindly making the locally optimal choice at each stage. A trivial approach would be to calculate the total distance for each possible path and then select the shortest. Although effective, this method features a high time complexity and rapidly becomes inefficient for large problems.

Dynamic programming provides a clever method that avoids calculating every possible solution but still ensures that the optimal one is found. The general idea behind the approach is to decompose the problem into simpler and independent sub-problems, which are tackled one by one starting from the smallest. The optimal solution to the smaller problems is then used to gradually solve the larger ones, until the original problem is completely solved. For the shortest path problem (SPP), the procedure starts by considering the last stage, where the driver is left with only one last decision to make. Whatever the current state at this stage, the driver has no choice but to go to the terminal node and so the optimal solution to this first sub-problem is trivial. Then, by moving back one stage at a time (from right to left), each subsequent sub-problem answers the question: "for each possible state at the current stage,

what is the shortest path to the final destination?". The advantage of this approach is that, given the solution to the first j sub-problems, it takes relatively little effort to find the answer to the (j + 1)-th. The original problem is automatically solved once the optimal solution to the sub-problem associated with the first stage has been found.

The same notation as Hillier and Lieberman [30] will be used to formalise the problem. Let denote with $s_n$ the state of the system at stage n and with $x_n$ the policy decision made at that point, namely what state to go next. Being in state $s_n$ and selecting $x_n$ at the next stage will add a net contribution $c_{s_n x_n}$ to the overall path. For example, if $s_n = R$ and $x_n = U$, then $c_{s_n x_n} = 4$. For a system that is in $s_n$ at stage n, let $f_n(s_n, x_n)$ be the total distance resulting from choosing $x_n$ as next destination and then following the best overall policy from stage n + 1 up to the last one. Of the various choices $x_n$ available for each state $s_n$, let $x_n^*$ denote the one that minimises $f_n(s_n, x_n)$, and let $f_n^*(s_n)$ be the corresponding value of the function. Thus,

$$f_n^*(s_n) = f_n\left(s_n, x_n^*\right) = \min_{x_n} f_n\left(s_n, x_n\right) \tag{6.1}$$

In practise, $f_n^*(s_n)$ represents the minimum distance required to reach Q from a node $s_n$. Starting from the last stage (n = 4) and moving backwards, the solution procedure then consists in iteratively finding $f_4^*(s_4)$, $f_3^*(s_3)$, ... for each possible state, until computing $f_1^*(s_1 = P)$, which provides the optimal solution to the original problem. Having solved the sub-problem associated with stage n + 1, it is easy to solve the one associated with stage n as the objective function can be computed by definition as:

$$f_n(s_n, x_n) = c_{s_n x_n} + f_{n+1}^*(x_n) \tag{6.2}$$

for each $s_n$ and each $x_n$. Storing the results of one iteration is then necessary to perform the next one. In the analysed example, the solution to the first sub-problem is straightforward to compute:

$$\begin{array}{llll} n = 4 & s_4 = W & f_4(W, Q) = 3 & \rightarrow \quad f_4^*(W) = 3 \\ & s_4 = Y & f_4(Y, Q) = 5 & \rightarrow \quad f_4^*(Y) = 5 \end{array}$$

The second iteration involves some calculations, as the driver could be in either T, U or V and may choose to go either W or Y. Hence:

$$\begin{array}{llll} n = 3 & s_3 = T & \begin{array}{l} f_3(T, W) = c_{T,W} + f_4^*(W) = 5 + 3 = 8 \\ f_3(T, Y) = c_{T,Y} + f_4^*(Y) = 4 + 5 = 9 \end{array} & \rightarrow \quad f_3^*(T) = 8 \\ & s_3 = U & \begin{array}{l} f_3(U, W) = c_{U,W} + f_4^*(W) = 4 + 3 = 7 \\ f_3(U, Y) = c_{U,Y} + f_4^*(Y) = 2 + 5 = 7 \end{array} & \rightarrow \quad f_3^*(U) = 7 \\ & s_3 = V & \begin{array}{l} f_3(V, W) = c_{V,W} + f_4^*(W) = 7 + 3 = 10 \\ f_3(V, Y) = c_{V,Y} + f_4^*(Y) = 4 + 5 = 9 \end{array} & \rightarrow \quad f_3^*(V) = 9 \end{array}$$

Note that the optimal policy need not be unique, as is the case for a system in state $s_3 = U$. By repeating the same steps, the $2^{nd}$-stage sub-problem is solved in a similar fashion:

$$
\begin{array}{llll}
n = 2 & s_2 = R & f_2(R,T) = c_{R,T} + f_3^*(T) = 2 + 8 = 10 & \\
 & & f_2(R,U) = c_{R,U} + f_3^*(U) = 4 + 7 = 11 & \rightarrow \quad f_2^*(R) = 10 \\
 & & f_2(R,V) = c_{R,V} + f_3^*(V) = 3 + 9 = 12 & \\
 & s_2 = S & f_2(S,T) = c_{S,T} + f_3^*(T) = 5 + 8 = 13 & \\
 & & f_2(S,U) = c_{S,U} + f_3^*(U) = 6 + 7 = 13 & \rightarrow \quad f_2^*(S) = 12 \\
 & & f_2(S,V) = c_{S,V} + f_3^*(V) = 3 + 9 = 12 &
\end{array}
$$

And, finally

$$
\begin{array}{llll}
n = 1 & s_1 = P & f_1(P,R) = c_{P,R} + f_2^*(R) = 4 + 10 = 14 & \rightarrow \quad f_1^*(P) = 14 \\
 & & f_1(P,S) = c_{P,S} + f_2^*(S) = 3 + 12 = 15 &
\end{array}
$$

The latter result provides the solution to the original problem, stating that the shortest path from P to Q features a distance of 14. By keeping track of the optimal decisions that minimise the objective function at each stage, the optimal route results in P $\rightarrow$ R $\rightarrow$ T $\rightarrow$ W $\rightarrow$ Q. This is also illustrated in Figure 6.2, where the blue arrows indicate the optimal decisions at each node. The figure also highlights one important property of the dynamic programming approach: besides providing the optimal policy for the overall problem, an optimal decision policy is also available for any state in the network, regardless of this state being ever reached. The prescription of an optimal policy for each state and at each stage is a typical feature of dynamic programming that can be useful in several ways, including sensitivity analysis [30].

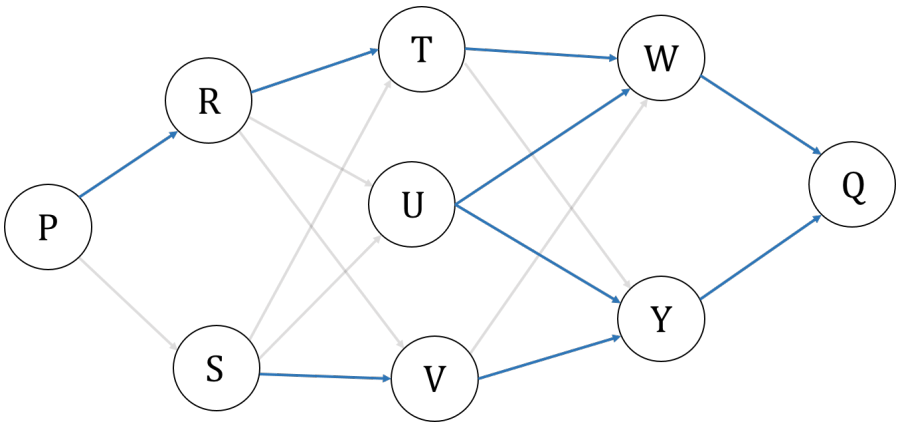


FIGURE 6.2: Optimal decision policies

## 6.2 Formalisation of the dynamic programming approach

The SPP presented in the previous section illustrates the working principle of dynamic programming. Although this method can be applied to a variety of different classes of problems, there are three essential features that never lack in any dynamic programming problem: states, stages and a recursive relation.

As a first step, the procedure requires to structure the problem as a multi-stage decision process, where the *stages* identify the points in which a *policy decision* is required. The problem is approached starting from the first or last stage and proceeding one stage at a time, such that the solution to a one-stage problem is necessary to solve the next one. Any problem lacking this property cannot be dealt with dynamic programming. In some problems, the stages corresponds to discrete points in time, whereas in others they do not have time implications. In the latter case, it often takes more imagination and intuition to formulate the problem as a dynamic programming one. At each stage, the system might be in different possible conditions called *states*, and a policy decision has the effect to transform the current state into a different one associated with the next stage. The definition of the states is usually the most crucial design parameter of the model, as there are often several available options and the effectiveness of the method may change drastically depending upon which one is made. The only rule to follow in making such choice is that each state must retain all the necessary information to determine the optimal policy henceforth, which is known as Markovian property [38]. In addition, as a practical guideline, the number of state variables should be contained, since the computational efficiency of dynamic programming rapidly decreases as the dimensionality of the state space increases. This property is referred to as "curse of dimensionality" [35] and significantly limits the applicability of the method in practise.

The decision variables are associated to the degrees of freedom of the problem. As for the SPP, each policy decision produces some positive or negative contribution to an overall objective function to be minimised (or maximised). Problems formulated with discrete states and the decision variables falls under the category of "Markov decision processes". Conversely, problems where continuous states and decisions are involved are often addressed under the umbrella of "control theory" [39]. In any case, the process of making optimal decisions is based on Bellman's principle of optimality [35]: whatever the current state, the remaining decisions must constitute an optimal policy for all successive stages, regardless of the history of decisions made to arrive at that point. The optimisation is carried out in practice by formulating a recursive relationship that provides the optimal policy to any sub-problem, given the solution to all the smaller sub-problems. Solving sub-problem associated to stage n means finding the best policy for all possible states. For the SPP, the problems are solved starting from the last stage and then moving backward one stage at a time, until the original problem is solved completely.

This kind of logic is referred to as backward induction [39]. Vice versa, some problems are more conveniently approached starting from the last stage, so that the recursive relation is based on a forward induction process. In this case, the stages are numbered in terms of the number of stages completed.

All these features define the underlying structure that any problem must have to be addressed with dynamic programming. As will be shown, this is the case for the design problem addressed in the following chapter. Furthermore, since no probability distributions are involved in such problem, it falls into the category of deterministic dynamic programming problems [36], where the state at stage $n+1$ is uniquely determined by the state and the policy decision at stage n. For this kind of problems, the recursive relationship can be expressed in mathematical terms as:

$$f_n^*(s_n) = \min_{x_n \in S_n} \left\{ c_{s_n x_n} + f_{n-1}^*(x_n) \right\} \tag{6.3}$$

where $S_n$ identifies the set of all admissible decisions, given that the system is in state $s_n$ at stage n.

# Chapter 7

# Unperturbed ballistic resonant flyby design

Resonant flybys are increasingly included in today's missions, allowing to achieve unique scientific objectives. However, their design involves several challenging aspects. The problem can be framed within the broader framework of Multi Gravity Assist (MGA) trajectory design and has several features in common with it, such as the intrinsic mixed-continuous combinatorial nature. The discrete variables of such hybrid domains result in the non-smoothness of the solution space, whose dimensionality grows exponentially with the number of alternative options available. For resonant flyby trajectories, the discrete degrees of freedom are the resonance condition at each flyby and the total number of encounters. If the latter is allowed to vary, each additional GAM adds another dimension to the solution space, whereby even small sequences of GAMs can result in combinatorial explosion issues. For this reason, directly using high-fidelity models to optimise such trajectories is short-sighted and leads to prohibitive computational times. The standard approach consists of a first phase in which the solution space of these problems is explored through low-fidelity techniques to identify a limited number of promising solutions. Each of these alternatives can serve as a reasonable starting point for higher-fidelity models to converge to fully-integrated trajectories. Since the convergence of such models and the quality of the optimised solution strongly depend on the initial guess, the preliminary design phase is of great importance for the final solution.

Over time, mission analysts have proposed a variety of methods to compute preliminary solutions in ballistic tour design problems. An intuitive approach is to directly use global methods such as differential evolution or genetic algorithm, as they can be immediately applied to hybrid domains. However, previous studies evidenced the risk of such techniques to ignore most of the continuous gradient information within the search space [40]. A more systematic approach is to divide the problem into its continuous and discrete parts, and then handle their dynamics on two levels. In practise, a high-level loop scans possible solutions in terms of categorical variables (e.g. resonance conditions), while at the low-level a local method optimises the remaining continuous part of the trajectory. While such approach effectively exploits the dual nature of the problem, the evaluation of all the possible

combinations of the discrete variables should be avoided in order to limit the computational effort required. To this end, many analytical tools are available to reduce the number of potentially optimal solutions. The most simple of them is the Tisserand graph, a two-dimensional map used to quickly assess the feasibility of sequences of GAMs. In Federici et al. [27], this tool was used to identify possible sequences of resonant flybys for a transfer between Jupiter and Europa. A common problem with the Tisserand graph is that it completely neglects the phasing of the planets, as well as the inclination and eccentricity of their orbits, making it more complex to find the optimal final solution. Numerous more advanced methods exist that are typically used in conjunction with a patched-conics orbital model, such as the branch and bound algorithm, grid searches or heuristic-based pruning techniques.

In this context, dynamic programming has been hardly incorporated into typical MGA design problems, whose structure is not particularly suited to the implementation of the method. Indeed, the need to solve a Lambert's problem at each flyby and the huge number of possibilities associated with it can easily cause curse of dimensionality problems. On the other hand, resonant GAMs always involve encounters at the same position in space, meaning that there is no need to compute Lambert arcs. As will be shown later, the problem features a similar structure to that of the shortest path problem, thus allowing for a convenient implementation of dynamic programming.

## 7.1 Problem statement

The design problem addressed in this work can be stated as follows: given a spacecraft on an initial orbit, the goal is to find the minimum sequence of resonant flybys that gradually lead to a target orbit, which would not be achievable in a single flyby.

### 7.1.1 Assumptions and considerations

The main assumptions underlying the model are:

- zero-SOI patched-conics model
- planets in elliptical and inclined orbits
- resonant flybys, except for the last one
- unperturbed trajectory
- unpowered flybys

The use of the patched-conics model for a preliminary trajectory design is a consolidated practise. Despite its simplifying assumptions, the application of the model to real-world missions such as Voyagers 1 and 2 revealed to work rather well [17]. Once a patched-conics solution is computed, there are

several techniques to convert it to a trajectory governed by n-body dynamics (e.g. differential correction) [41] [42]. As previously mentioned, the zero-SOI approximation leaves the entry point of the spacecraft into the planet's SOI as a degree of freedom to be exploited in the design phase. The model also implies that each heliocentric orbit of the small body must intersect exactly that of the planet at the encounter position. This is not a major limitation and still provides a good approximation for resonant encounters, since relaxing this assumption only results in a slight distortion of the circles in the b-plane, as shown in Valsecchi et al. [11]. However, although those hypothesis are reasonable for various interplanetary missions, there are notable cases in which they result in a bad approximation of the trajectory. A typical example is the Earth-Moon system, as the Moon is well inside the SOI of the Earth.

The b-plane model extension developed in Part I allows to consider the real orbits of the planets. Taking their eccentricity and inclination into account in the preliminary design allows an easier conversion of the trajectory to higher-fidelity models [27].

Disregarding perturbations can be seen as a main limitation of the model, since the planetocentric dynamics can be altered in a non-negligible way by disturbing effects. Although a method to include such effects in the computation of the b-plane circles has already been developed by Masat [29], the focus of this work is to demonstrate the effectiveness of a dynamic programming approach to the problem. Future work could address the inclusion of perturbations in the model in order to refine the dynamics within the SOI.

Since neither perturbations nor manoeuvres are considered, the effect of each GAM is a pure rotation of the incoming velocity $\mathbf{U}$, without changing its magnitude. For this reason, the target orbit must be ballistically reachable, otherwise the design would not be effective. This condition is equivalent to saying that the initial and final orbits must feature the same Tisserand parameter[29]. Clearly, the two orbits must also join the same position in space of an attracting body, otherwise the execution of resonant flybys would not be possible. Another consequence of those hypothesis is that the trajectory between two consecutive flybys corresponds to a Keplerian orbit, which can be characterised through a position vector $\mathbf{r}$ and a velocity vector $\mathbf{v}$. However, as the resonance condition implies that each flyby occurs at the same position in space, this position provides in practise three constants of motion. In addition, since the magnitude of the velocity $\mathbf{U}$ is also conserved, the problem features in total four constants that allow to describe the state of the system using only two independent variables.

The evolution of the state, and therefore of the overall trajectory, depends entirely on the degree of freedom left by the entry point of the spacecraft into the SOI. At each flyby, this point is chosen on the b-plane and then the deflection model is used to rotate the incoming velocity, leading to a fully defined post-encounter orbit. The natural choice to specify such injection point would be to directly use the b-plane coordinates $(\xi, \zeta)$ as free variables.

Nevertheless, a more convenient option is to use the post-encounter semi-major axis $\bar{a}$, which is associated to a circle in the b-plane, and a polar coordinate $\alpha$ identifying a specific point on the circle. Fig 7.1 provides more details about the definition of $\alpha$, taken as the counterclockwise angle between the $\xi$-axis and the line joining the centre of the circle and a point belonging to it. Therefore, the b-plane point corresponding to a given couple $(\bar{a}, \alpha)$ can be computed as:

$$\begin{cases} \xi = R\cos\alpha \\ \zeta = C + R\sin\alpha \end{cases} \tag{7.1}$$

where C and R are the centre and radius of the circle associated to $\bar{a}$.

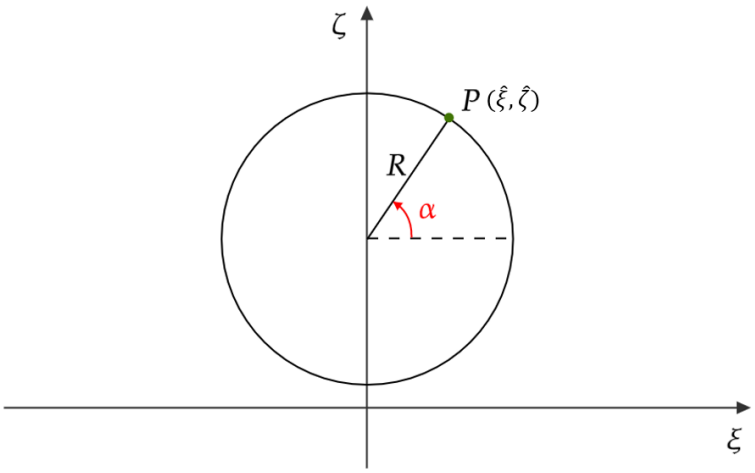


FIGURE 7.1: Definition of $\alpha$

The benefits of this description is twofold. First, one of the decision variables is directly related to an orbital feature of the post-encounter orbit, namely the semi-major axis, which in turn can be linked to a specific resonance condition through Equation (4.5). In addition, the discrete nature of the resonance condition translates into a limited set of admissible semi-major axes. This introduces a discrete element that can be conveniently exploited through a dynamic programming approach. A second discrete variable is the number N of total encounters, which increases the problem complexity due to the variable dimensionality of the solution space. Furthermore, since $\alpha$ varies in a continuous domain ($[0; 2\pi]$), the optimisation problem features a mixed-continuous combinatorial nature.

Finding the optimal sequence of resonant GAMs thus translates into the search for the optimal series of pairs $(\bar{a}_i, \alpha_i)$ that leads from the initial to the target orbit in the least possible number of close encounters:

$$\text{Initial orbit} \rightarrow (\bar{a}_1, \alpha_1) \rightarrow (\bar{a}_2, \alpha_2) \rightarrow \ldots \rightarrow \underset{(\bar{a}_N, \alpha_N)}{\text{Target orbit}}$$

As the intermediate semi-major axes must be in resonance with the planet, each of them will be associated to a k/h ratio, with k and h integers. However, although the initial and target orbits do not necessarily have to be resonant, they can still be associated to a "dummy" k/h ratio by means of Equation (4.5). This allows to easily compute the set of admissible resonances for the intermediate flybys, as will be clear later.

Masat [29] addressed the same design problem as described above, with the exception that the total number of flybys was given as input. Besides proposing a solution to the unperturbed problem, the author also implemented a strategy to account for perturbing effects inside the SOI by means of some corrective coefficients. Although these could have been included in the present work as well, the original scope of the thesis was to improve the performances and generality of their unperturbed solution strategy, with the possibility of including the perturbations in a second moment. The approach proposed by the author features a modular structure with three optimisation layers, as shown in Figure 7.2. At the highest level, a direct-search algorithm finds an optimal splitting of the total velocity variation between the initial and target orbits into N contributions. At the lowest level, a nested *for* loop is used to analyse all possible combinations of resonance conditions and angle $\alpha$ to find the one that best matches the desired sub-$\Delta$v at each flyby.

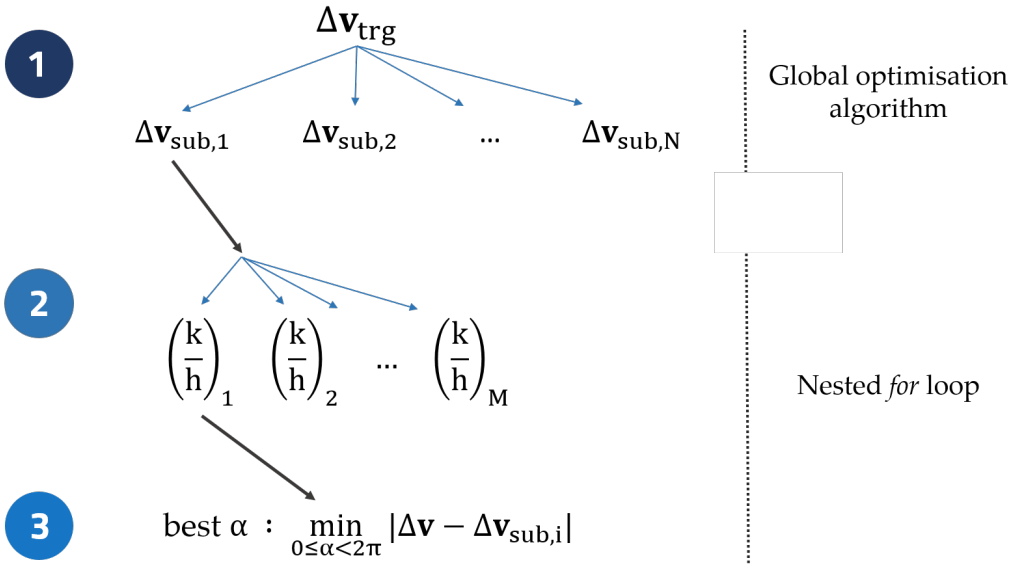


FIGURE 7.2: Unperturbed solution strategy by Masat

This thesis extends on their work and attempts to improve on the optimisation strategy by avoiding both the use of heuristic methods, which do not guarantee finding the optimal solution, as well as brute-force searches, which are computationally expensive. This is accomplished by proposing a deterministic dynamic programming approach that, on the one hand, always converges to the globally optimal solution of the problem and, on the other hand,

allows to find such solution without systematically evaluating all the possible combinations of flyby sequences. Moreover, a further improvement is given by the inclusion of the total number of flybys as one of the optimisation parameters. This information is valuable as it could be used to quickly reduce the solution space to be explored when moving to more advanced dynamics models.

## 7.2 Dynamic programming approach

The structure of the considered problem naturally reflects that of a multi-stage decision-making process, where the stages are the series of consecutive flybys. The state of the system only requires two variables to be described, and they are chosen as the non-dimensional semi-major axis $\bar{a}$ and the orbital inclination i. The former is convenient because it can only assume discrete values and moreover can be immediately associated with a locus of points in the b-plane, namely a circle. As a second state variable, also the eccentricity would have worked, but the inclination is more convenient since it is easier to define the range of intermediate values it can assume for a certain problem. However, given a generic semi-major axis $\bar{a}$, each value of $\alpha$ on the corresponding circle leads to a different value of post-encounter inclination. Thus, i varies in continuous intervals and there are an infinite number of states at each stage.

### 7.2.1 Reduction of the combinatorial complexity

Since it is clearly impossible to consider all possible states, a trivial way out of the problem is to discretise the continuous interval and interpolate the function values not associated with the grid points, as suggested by Bellman [35]. The type of discretisation scheme depends on the classical trade-off between accuracy and computational runtime. However, besides the suboptimality introduced by the discretisation, even a coarse grid easily leads to curse of dimensionality when the problem involves several stages.

The approach adopted here is to select a single optimal value of inclination for each b-plane circle, rather than considering the whole continuous range. Denoting with $i_f$ the target inclination of the overall flybys sequence, the optimal inclination on any given circle is taken as the one closest to $i_f$. Making such choice at each flyby allows to reach the final inclination as quickly as possible. Note that the choice of the points on the circles only affects the evolution of the inclination through the sequence of flybys, and not that of the semi-major axis. Indeed, the possible values of $\bar{a}$ at stage j only depend on the value of $\bar{a}$ at stage j-1, regardless of the inclination. This observation is based on the following expression of the post-encounter angle $\vartheta'$:

$$\cos\vartheta' = \cos\vartheta\cos\gamma + \sin\vartheta\sin\gamma\cos\psi \tag{7.2}$$

Recalling the relationship between $\bar{a}$ and $\vartheta$ (Equation (3.5)), Equation (7.2) implies that the post-encounter semi-major axis is only a function of its pre-encounter value and the angles $\gamma$ and $\varphi$, which do not depend on the pre-encounter inclination. This simple argument explains why an optimal policy for the evolution of inclination can be established regardless of the major semi-axes.

In this way, the originally hybrid optimisation problem is reduced to fully combinatorial, thereby facilitating the implementation of dynamic programming. Unfortunately, there is no analytical formula linking a point on a circle to the corresponding inclination, and so the optimal value must be sought numerically. Moreover, for realistic GAMs, one must take into account the minimum admissible flyby altitude above the planet. In the b-plane, this limitation translates into a constraint on the minimum impact parameter $b_{min}$, which is generally taken as the planet's radius augmented by the gravitational focusing [43]:

$$b_{min} = R_{pl}\sqrt{1 + \frac{2c}{R_{pl}}} \tag{7.3}$$

where c is the speed of light and $R_{pl}$ is the physical radius of the planet. The consequence is that all the points on a circle that are below $b_{min}$ are not feasible and must be discarded. For this reason, the search for the optimal value of inclination requires a first step to identify the feasible domain $D_\alpha$ in which the angle $\alpha$ can vary. Referring to Figure 7.3, $b_{min}$ imposes two bounds on this domain, one of which is denoted by $\hat{\alpha}$. This angle is obtained by applying the cosine law to the triangle $rcb_{min}$ and then making some considerations about the quadrant in which $\hat{\alpha}$ is located. Since Öpik's theory is less accurate for shallow encounters (large b) [10], it is appropriate to set a limit on the maximum impact parameter as well.

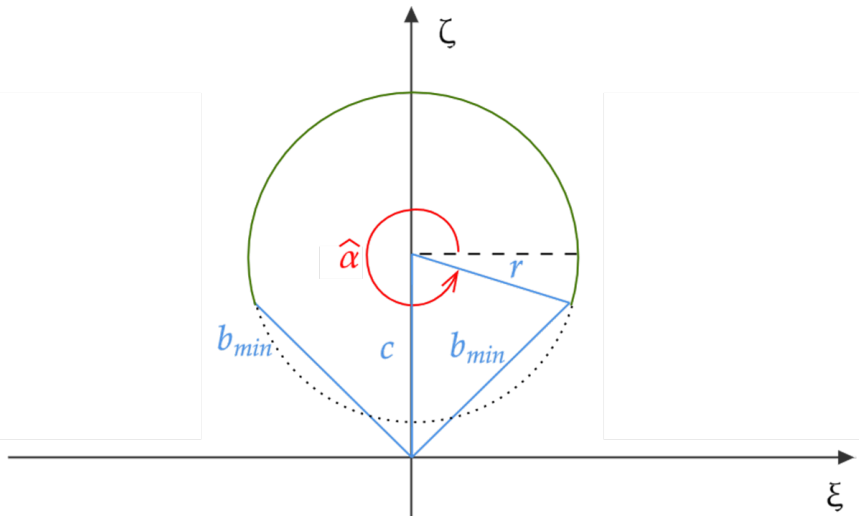


FIGURE 7.3: The green arc identifies the feasible domain of $\alpha$ due to $b_{min}$

Once $D_\alpha$ is computed, let $f_\alpha(\alpha)$ identify the function that, for any $\alpha \in D_\alpha$, returns the value of the corresponding inclination. $f_\alpha(\alpha)$ may be seen as a black-box function involving all the steps in box diagram of Figure 4.1 plus a conversion step where the post-encounter velocity and flyby position are used to compute the post-encounter inclination. The optimal inclination for any circle is then computed by always minimising this function:

$$i_{opt} = \min_{\alpha \in D_\alpha} (f_\alpha(\alpha) - i_f)^2 \tag{7.4}$$

Although $f_\alpha(\alpha)$ is continuous on $D_\alpha$, this last can be discontinuous due to the limitations imposed by the impact parameter, as shown in the example of Figure 7.4. This occurs when the desired jump of the semi-major axis is relatively small, resulting in a b-plane circle so large that it occupies both the unfeasible regions due to $b_{min}$ and $b_{max}$. In these cases, the minimisation is carried out once for each continuous sub-part of the domain and the best function value is retained.

Figure 7.4 also highlights a typical feature of GAMs already mentioned in Chapter 3, namely that the closer a small body flies to the planet, the greater the variation in inclination. Indeed, the plot shows a case in which the post-encounter semi-major axis is smaller than the initial one, whereby the circle is located below the $\zeta$-axis in the b-plane and so the point at $\alpha = 90°$ is the closest point to the planet. However, this is only a general rule, but it is not always the case, so numerical minimisation is also necessary when searching for the largest inclination change.

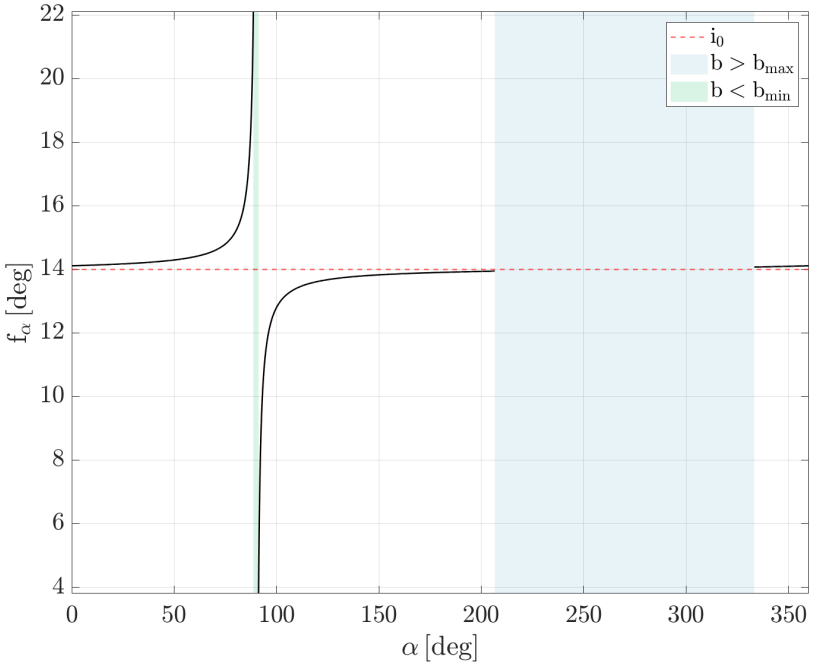


FIGURE 7.4: Example of how $f_\alpha(\alpha)$ varies on its domain. $i_0$ denotes the pre-encounter inclination

### 7.2.2 Solution procedure

Let $\{\bar{a}_0, e_0, i_0, \Omega_0, \omega_0\}$ be the orbital elements of the initial orbit and $\{\bar{a}_f, e_f, i_f, \Omega_f, \omega_f\}$ those of the target orbit. Provided that the orbits feature the same Tisserand parameter and they cross a planet's orbit in the same position in space, the initial and target states of the system are $s_0 = (\bar{a}_0, i_0)$ and $s_f = (\bar{a}_f, i_f)$, respectively. Having reduced the problem to fully combinatorial, the set of possible solutions is finite and so the problem presents a similar structure to the SPP. However, unlike this latter, the possible states through which the system can move are not known in advance. For this reason, the solution strategy consists in practice of two steps, the first of which is aimed at generating the possible intermediate states.

**States generation step.** The procedure starts by identifying a set of admissible resonances. First, two "dummy" k/h ratios, $k/h_0$ and $k/h_f$, are computed from the initial and final semi-major axes by means of Equation (4.5). A set of resonance ratios $k/h_1, k/h_2, \ldots, k/h_r$ is then computed such that $k/h_0 \leq k/h_i \leq k/h_f$ and both k and h are not larger than a certain value, so as not to consider orbits with too large orbital periods. Equation (4.5) is then used to convert the $k/h_i$ ratios and obtain a finite set $S_{\bar{a}}$ of admissible semi-major axes. This, however, does not represent the range of alternative $\bar{a}$ available *at each stage*. Indeed, depending on the current state, often only a subset of $S_{\bar{a}}$ is feasible for the evolution of the system to the next stage. This is because the jump in semi-axis major cannot be arbitrarily large due to the constraint on $b_{min}$. For too large $\Delta\bar{a}$, the b-plane circle results all inside the unfeasible region and the considered semi-major axis cannot be reached. The maximum $\Delta\bar{a}$ for a given pre-encounter condition can be found by writing:

$$\left| C + R \right| = \left| c \frac{\sin\vartheta + \sin\vartheta'}{\cos\vartheta' - \cos\vartheta} \right| = b_{min} \tag{7.5}$$

Solving for $\vartheta'$ and then using Equation (4.16) one obtains the maximum and minimum semi-major axes achievable by single GAM. This information allows to limit the number of possibilities available, given the current state. Therefore, Equation (7.5) is computed at the beginning once for each element of $S_{\bar{a}}$, so as to determine the possible state transitions in terms of semi-major axis. On the other hand, the possible intermediate inclinations are not computed at this stage. This is because, by using dynamic programming, not all possible sequences of pre- and post-encounter semi-major axes are evaluated. Therefore, it is unnecessary to compute in advance the optimal inclinations associated with each possible circle, and so they are only calculated whenever needed.

**Dynamic programming.** The problem is then formulated with dynamic programming as follows:

- Number of stages: N = total number of flybys (unknown in advance)
- Stages numbering: n = number of flybys completed

- States: $s = (\bar{a}, i)$
- Policy decision at stage n: $x_n$ = state at stage $n-1$
- Objective function being in state $s_n$ at stage n and making the decision $x_n$:

$$f_n(s_n, x_n) = \sum_{i=1}^{3} \left| \bar{U}_i - \bar{U}_{f,i} \right| \tag{7.6}$$

  where $\bar{U}_i$ are the components of the velocity associated with the current state $s_n$ and $\bar{\mathbf{U}}_f$ is the target velocity, computed from the final state $s_f$. The function represents the error between the current velocity and the desired one.

- Recursive formula:

$$f_n^*(s_n) = \min_{x_n \in S_{n-1}} f_n(s_n, x_n) \qquad n = 1, 2, \ldots \tag{7.7}$$

  where $S_n$ is defined as the set of all possible states in which the system could be at stage n. In practice, at each stage and for each state Equation (7.7) answers the question: "what is the best pre-encounter state to come from, if the system is currently in state $s_n$?".

  Clearly, $S_0$ only includes $s_0$. Then $S_1, S_2$, etc. are computed one by one as the algorithm progresses from one stage to the next. The generic $S_j$ is computed starting from all the states in $S_{j-1}$ and considering that, at stage j, the semi-major axis could be any of those in the set $S_{\bar{a}}$, except those not reachable due to $b_{min}$. For instance, the elements of $S_1$ are obtained by starting from state $s_0$ and computing for each feasible semi-major axis in $S_{\bar{a}}$ the optimal value of inclination using Equation (7.4). Each pair of semi-major axis–inclination obtained is then saved as an element of $S_1$. Note that, in this way, even when the system has reached a generic $\bar{a}_r \in S_{\bar{a}}$ at some stage, the semi-major axes that are further from $\bar{a}_f$ are still considered as valid options for the next stage. The reasons for this are shown in the example of Figure 7.5. Given the current semi-major axis $\bar{a}_r$ and given that the target one is larger, the plot shows the maximum inclination achievable by choosing different semi-major axes at the next stage. As it can be seen, in this case by moving backwards with respect to $\bar{a}_f$ it is possible to obtain a greater gain in inclination. Therefore, not necessarily each flyby has to get a step closer to $\bar{a}_f$, since some solutions could benefit from a trade-off between losing in semi-axis and gaining in inclination.

Unlike for the SPP, here the sub-problems are solved starting from stage one and then moving forward one stage at a time, until the original problem is solved completely. This kind of logic is referred to as forward induction [39]. Recalling that the objective function provides a measure of how close is the current state to the target one, the final orbit is considered achieved when the error drops below a certain threshold $\varepsilon$, defined depending on the problem. By keeping track of the best function value obtained at the end of each

stage, the algorithm is stopped when such value becomes lower than $\varepsilon$, and the number of total flybys N is retrieved. If more than one solution has an error smaller than $\varepsilon$ for the same number of flybys, the one with the smallest error is chosen.

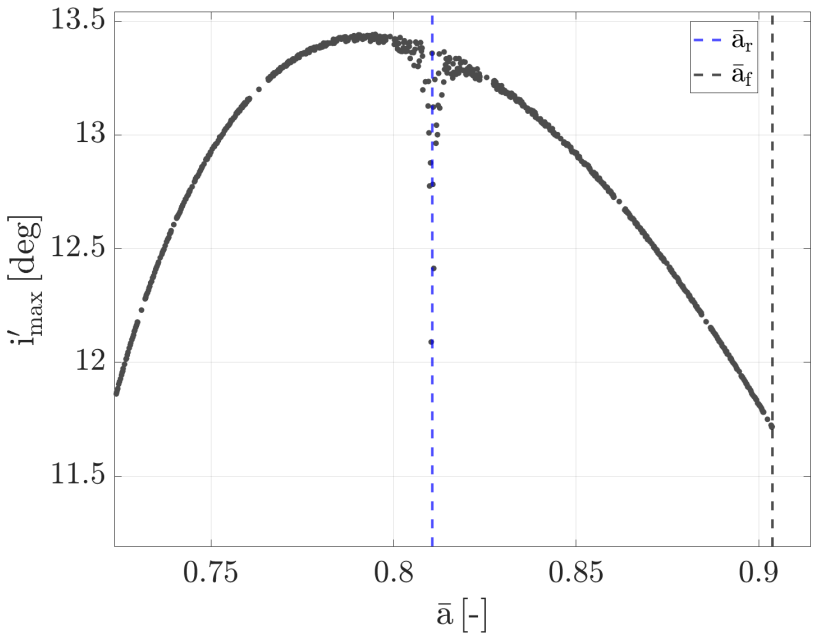


FIGURE 7.5: Example of maximum inclinations achievable by choosing different semi-major axes at the next stage. The initial inclination is $i_r = 3°$

Note that, since the state at stage $j+1$ is completely determined by the state and policy decision at stage j, the problem can be classified under the category of deterministic problems.

## 7.3 Dynamic programming compared to exhaustive enumeration

In order to highlight the computational efficiency of the dynamic programming approach, a comparison is made with the more standard technique of exhaustive enumeration, which iterates through all possible solutions to a given problem and then determines which is the best one. Consider a simple case in which, starting from an initial state, the set of admissible resonant semi-major axes is composed of only three elements, denoted with $a_1, a_2, a_3$. Obviously this is a quite reductive example, but is helpful to easily understand the advantages of using dynamic programming. Figure 7.6 shows the logic behind the two approaches. The state of the system is represented only by the semi-major axes, since for each node a single inclination value is calculated using Equation (7.4). For each n-th stage, "ev." indicates the number of evaluations performed by each of the two methods. Each line between two nodes corresponds to one evaluation, and the bold ones identify the optimal paths. Keep in mind that, at each stage and for each stage, dynamic

programming finds the best state to come from. As the initial state is fixed, each alternative is considered optimal at stage 1.

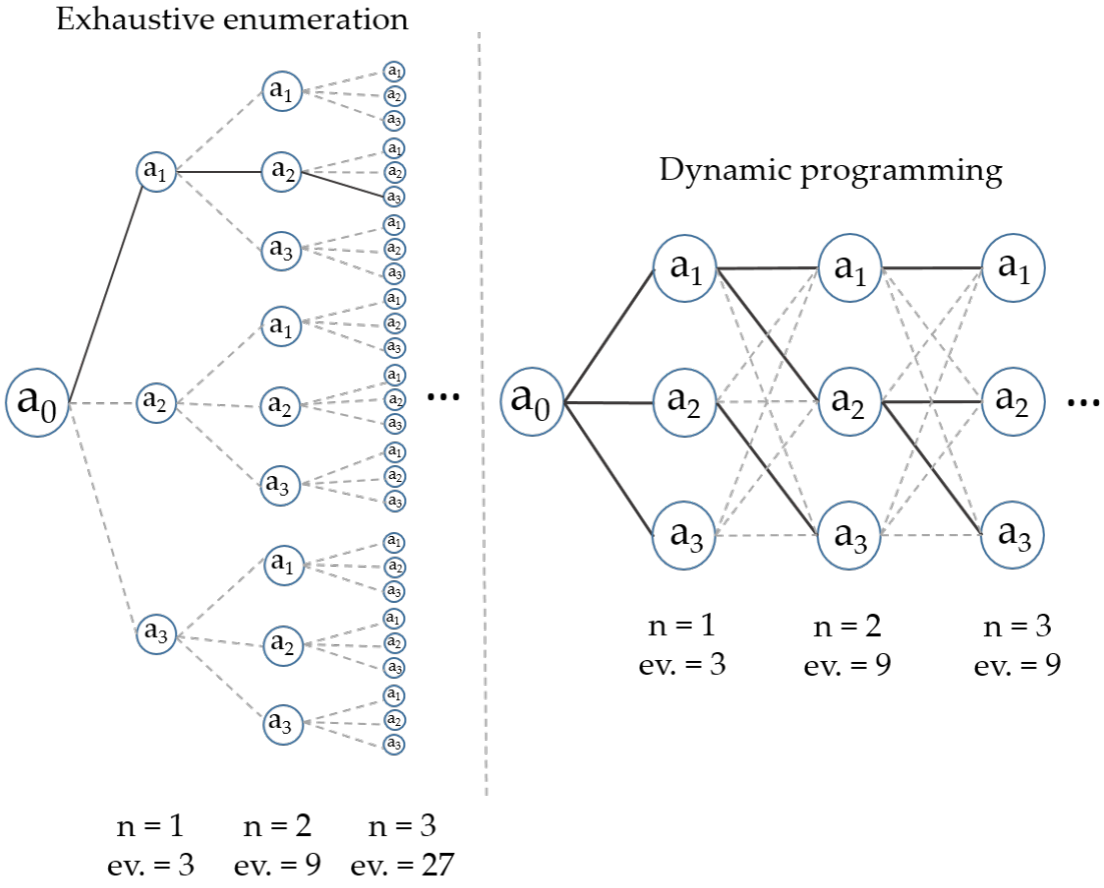


FIGURE 7.6: Logic behind the two methods

First of all, note that dynamic programming provides an optimal solution to get from the initial stage to any of the intermediate stages and in any number of flybys. Exhaustive enumeration, on the other hand, only identifies the optimal solution to the original problem. The first two stages require the same computational effort from both methods, whereas from stage 3 onwards the number of evaluations performed by exhaustive enumeration grows exponentially, while that of dynamic programming remains unchanged. This is because of the way such approach splits the problem in smaller pieces and exploits the solution to the smaller sub-problems to solve the larger ones, without considering many unnecessary paths. For instance, consider the path $a_0 \rightarrow a_2 \rightarrow a_2 \rightarrow a_2$. Although it leads to sub-optimal solutions, at stage 4 exhaustive enumeration will still evaluate three continuations of that path, one for each admissible semi-major axis. This in turn will result in another 9 evaluations at stage 5, 27 at stage 6, and so on. On the other hand, this does not happen with dynamic programming that, once it has established that the optimal path to be in $a_2$ at stage 3 is $a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow a_2$, it considers no other alternative to find the solution at stage 5.

Note that the computational savings are significantly higher for larger versions of the considered problem. For a set of k admissible semi-major axes, the number of evaluation performed after n stages are:

- $\sum_{i=1}^{n} k^i$ with exhaustive enumeration,
- $k + n \cdot k^2$ with dynamic programming.

For instance, for $k = 10$ and $n = 4$, the former method performs about 10 thousands evaluations more than the other. Consider also that, if in future works the flyby model was to be improved by including, for example, perturbations, the discrepancy between the two approaches would weigh considerably on the overall computational time. The number of evaluations is the most significant parameter, as it does not depend on the performance of the machine used.

# Chapter 8

# Test cases and results

Several tests have been carried out to assess the performance of the proposed algorithm. This chapter reports three of them, selected to highlight different aspects of the method. For all reported performances in terms of computational time, please note that all tests have been implemented in Matlab® on a machine with an Intel® Core™ i7-6700 CPU @2.60 GHz.

## 8.1 Solar Orbiter

Solar Orbiter is an exemplary case of a mission whose objectives could only be achieved by implementing resonant flybys. Its orbital requirements were to (i) achieve a minimum perihelion between [0.28; 0.30] AU and (ii) raise the solar inclination angle to a minimum of 25° and with a goal of 35°. These objectives were accomplished by exploiting a series of four resonant flybys at Venus. In the first test, the dynamic programming algorithm is asked to reproduce the design of such resonant phase of the mission. The baseline trajectory is the mission profile with launch in January 2017 proposed in [4], which is shown in Figure 8.1. The resonant orbits are labelled with the notation $V_i - V_j$, with $V_i$ denoting the i-th flyby at Venus. The designed resonances are, respectively, (3/4, 3/4, 2/3, 3/5), and more details about the orbits are given in Table 8.1. Note that the gravitational assist obtained from the second flyby consists of a pure cranking, i.e. the period remains unchanged.

| **Orbit** | Aphelion [AU] | Perihelion [AU] | Ecliptic Inclination [°] |
|---|---|---|---|
| **E2-V2** | 0.998 | 0.311 | 1.72 |
| **V2-V3** | 0.910 | 0.284 | 9.93 |
| **V3-V4** | 0.870 | 0.315 | 18.11 |
| **V4-V5** | 0.789 | 0.315 | 23.83 |
| **V5-V6** | 0.740 | 0.290 | 27.25 |

TABLE 8.1: Resonant orbits of SOLO for the mission profile with launch in January 2017 (source: [4])

Since the trajectory involves no mid-course manoeuvres other than navigation ones [44], it offers an ideal test case for the proposed design strategy.

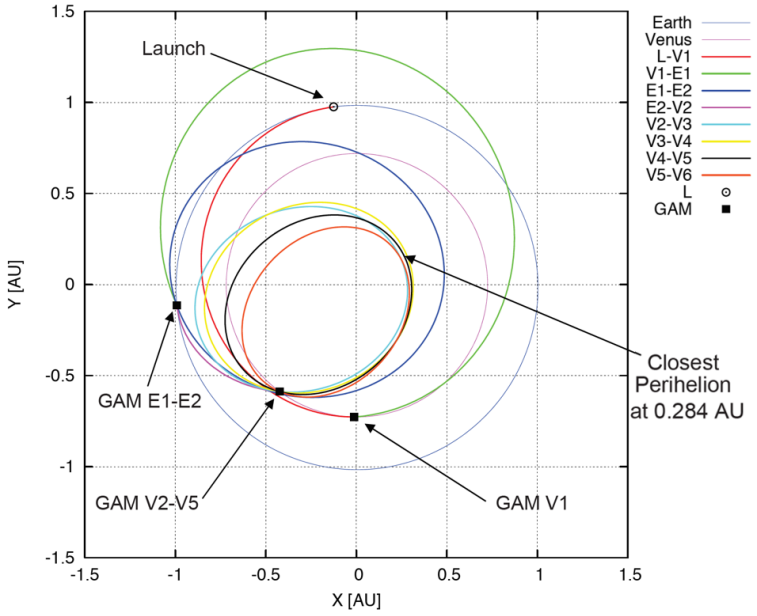


FIGURE 8.1: SOLO baseline trajectory with launch in January 2017 (source: [4])

The algorithm is only given the initial and last orbits of Table 8.1 as inputs, whereas the total number of necessary resonant flybys is left as a free variable. The planet's position at encounters is taken from the ephemeris, considering that for the selected mission profile the first flyby takes place on 22/05/2020. Thus, for each orbit, three orbital elements plus the three components of the flyby position are known. Interestingly, these six parameters do not provide a unique orbital representation, but rather are associated with four possible orbits. The differences in the orbits are due to whether the flyby takes place at the ascending or descending node, and from inside or outside the planet's orbit. For the sake of conciseness, the proof of this is left in Appendix A, where a method for identifying the four alternative orbits is proposed. In this test, all the encounters are assumed to occur at the ascending node and from outside Venus' orbit, as per the baseline trajectory.

The set of admissible resonance ratios is generated by considering a limit of 5 on both k and h. Since the algorithm is designed to reach the target orbit in as few encounters as possible, the solution is likely to involve low-altitude flybys. Therefore, the outcomes are affected by the choice of minimum admissible flyby altitude, here selected to be 300 km, as suggested by Janin [44] and Jehn et al. [45]. The target orbit is considered achieved when the error as defined in Equation (7.6) drops below a value of $10^{-4}$ km/s. The test has been repeated twice, once using the classical model and once using the extended one, thus providing a further opportunity to compare the differences in their outcomes. However, these are not expected to differ much given the low eccentricity of Venus.

### 8.1.1 Classic model solution

The algorithm converged to the final state in four stages, the same number of flybys of the real mission. Table 8.2 compares the solution trajectory with the baseline one in terms of k/h ratios and inclinations. Note that, except for the first orbit, the predicted resonance conditions are equal. The inclinations also are comparable, with the differences being mainly due to having disregarded the perturbations. A visual representation of the obtained solution is given in Figure 8.2. For the sake of comparison, the colours used in the figure are the same as those used for the resonant orbits in Figure 8.1.

The simulation required a surprisingly low computational time of about 0.06 s, highlighting the efficiency of the method.

| | **k/h ratios** | | **Inclinations** [deg] | |
|---|---|---|---|---|
| | SOLO | DP | SOLO | DP |
| **E2-V2** | - | - | 1.72 | 1.72 |
| **V2-V3** | 3/4 | 4/5 | 9.93 | 9.608 |
| **V3-V4** | 3/4 | 3/4 | 18.11 | 18.546 |
| **V4-V5** | 2/3 | 2/3 | 23.83 | 24.647 |
| **V5-V6** | 3/5 | 3/5 | 27.25 | 27.25 |

TABLE 8.2: Comparison between SOLO's baseline trajectory and that obtained through DP, considering a circular orbit of Venus

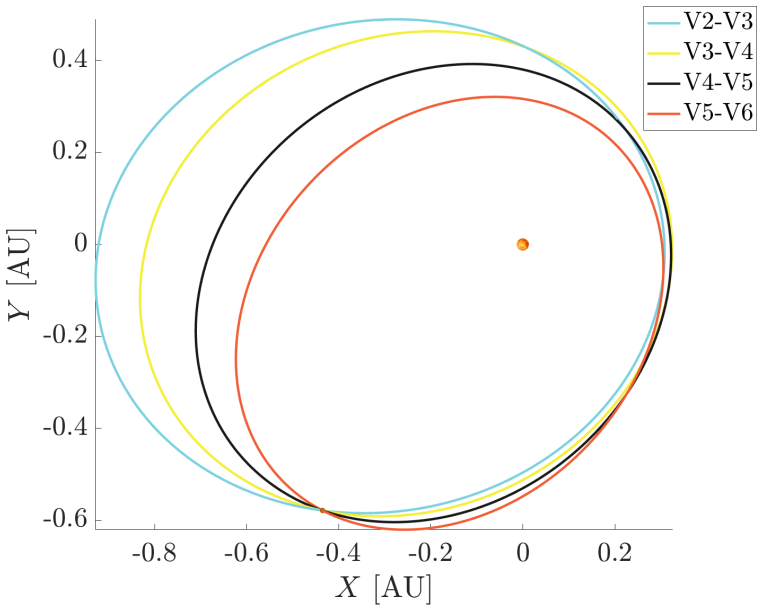


FIGURE 8.2: Solution trajectory using the classical model

### 8.1.2 Extended model solution

Also using the extended model the algorithm designed four orbits, whose details are reported in Table 8.3. However, in this case the predicted resonances are identical to those of the baseline solution, giving proof of the proposed design strategy. Also from a graphical point of view, there is no appreciable difference between the obtained trajectory, shown in Figure 8.3, and the baseline one. On the other hand, no particular improvement is observed in the predicted inclinations.

| | **k/h ratios** | | **Inclinations** [deg] | |
|---|---|---|---|---|
| | SOLO | DP | SOLO | DP |
| **E2-V2** | - | - | 1.72 | 1.72 |
| **V2-V3** | 3/4 | 3/4 | 9.93 | 9.152 |
| **V3-V4** | 3/4 | 3/4 | 18.11 | 18.344 |
| **V4-V5** | 2/3 | 2/3 | 23.83 | 24.586 |
| **V5-V6** | 3/5 | 3/5 | 27.25 | 27.25 |

TABLE 8.3: Comparison between SOLO's baseline trajectory and that obtained through DP, considering the real orbit of Venus

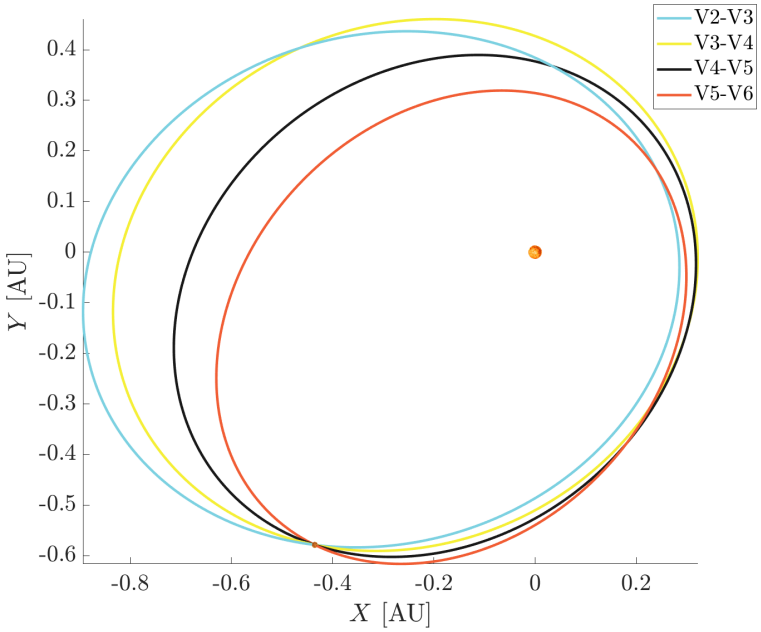


FIGURE 8.3: Solution trajectory using the extended model

The algorithm converged in about 0.06 s, as with the classical model, emphasising again that the two models require the same computational time.

For comparison, by performing the same test using the solution strategy proposed by Masat [29], it takes about 1.3 s only to run one nested loop at the low level. In addition, as pointed out by the same author, the heaviest part of their optimisation strategy is related the direct-search implemented at the high level, which results in rather high computational times when the solution space is meticulously analysed.

## 8.2 Dynamic programming vs exhaustive enumeration

When evaluating the computational efficiency of a dynamic programming approach, the standard benchmark used for comparison is the exhaustive search. In Section 7.3, a simple theoretical example was illustrated to highlight the differences between the two methods. To compare them on a real application, a second experiment is performed here by testing both on the same design problem seen in the previous section. The only difference in this case is that a limit of 15 is set for both k and h, so as to simulate a scenario involving a relatively high number of possible trajectories.

As it could be expected, the two algorithms converged to the same final solution. However, dynamic programming required considerable less effort than the other approach. Their performances are reported in Table 8.4 in terms of computational runtime and number of evaluations. The latter is the most meaningful parameter, as it does not depend on the performance of the machine used.

| | Exhaustive enumeration | Dynamic programming |
|---|---|---|
| **Runtime** | 3.55 min | 1.12 s |
| **Number of evaluations** | ~168 k | 1220 |

TABLE 8.4: Computational runtime and number of evaluations required by the two methods

Figure 8.4 provides a graphical representation of the number of solutions evaluated by both approaches. Each gray or red point identifies the final state associated to a trajectory computed with either of the two algorithms. The distinctive feature of the solutions evaluated by dynamic programming is that they are all relatively close to the target state in terms of eccentricity and inclination.

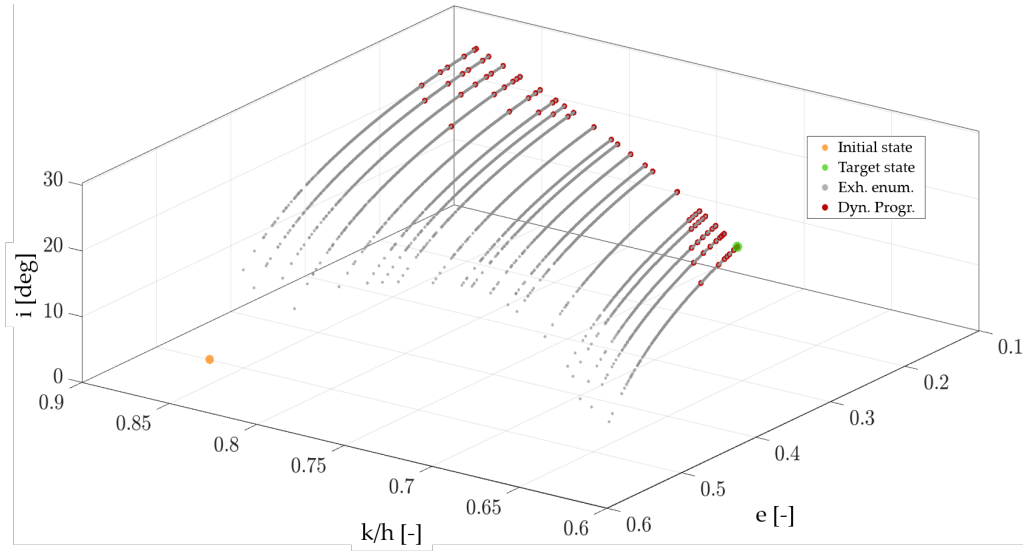


FIGURE 8.4: Solutions evaluated by the two methods

## 8.3 Resonant flybys at Callisto with target inclinations

The solution strategy proposed in Chapter 7 applies to scenarios in which large orbital changes are desired, and is designed to reach the target state in as few stages as possible. This usually results in trajectories with series of low-altitude flybys, which maximise the gravitational interaction. Several real-life missions have been designed based on such approach. This was the case, for example, for the lowering-inclination phase of the Cassini mission, where a series of resonant flybys with Titan were designed by keeping the flyby altitudes at minimum in order to quickly reduce the inclination down to near Saturn's equator [19].

However, as mentioned in the introduction, resonant orbits are also typically used for detailed observations of planets and moons, providing different observation geometries and thus better data collection. The high-inclination phase of the JUICE mission is a case on point. As shown in Table 8.5, a series of five resonant flybys at Callisto were specifically designed to provide repeated observations of the same latitudes of the inner Jovian system [5]. For missions of this kind, the proposed algorithm can still provide useful information. The way it was described, the inclination value at each flyby is chosen as close as possible to the inclination of the final state. Instead, by specifying a different target inclination for each stage, Equation 7.4 can be used to target specific latitudes at each flyby. The major axes can then vary as before within a range of permissible values, allowing different mission options to be examined, or they too can be specified with target values at each flyby, perhaps according to requirements on the frequency of the observations. In the latter case, the algorithm simply computes the b-plane coordinates that leads to the prescribed orbits.

A simple test is performed where the solver is asked to compute such coordinates for the orbits reported in Table 8.5. The same test was carried out by Valsecchi et al. [43] using a different method, thus providing a useful feedback for the results.

| | Perijove [$R_C$] | Apojove [$R_C$] | Inclination [deg] |
|---|---|---|---|
| $C_N$ | 13.5 | 39.1 | 7 |
| $C_{N+1}$ | 14.7 | 37.9 | 15 |
| $C_{N+2}$ | 14.9 | 33.3 | 22 |
| $C_{N+3}$ | 14.7 | 37.9 | 15 |
| $C_{N+4}$ | 13.5 | 39.1 | 7 |

TABLE 8.5: Resonant orbits of the high-inclination phase. The perijove and apojove are expressed in units of the radius of Callisto ($R_C$). Source: Definition Study Report of JUICE [5]

The position of Callisto has been taken from the ephemeris, considering that the first flyby is scheduled on 13/10/2030 [5]. All encounters are assumed to take place at the ascending node and in the pre-perijove branch of the orbit of the spacecraft, as in Valsecchi. Table 8.6 reports the results of the simulation. The b-plane coordinates computed by the algorithm are very similar to those computed by Valsecchi. Little differences exist, perhaps due to factors such as the value of Jupiter's radius or the position of Callisto used in the calculations.

| | **Valsecchi** | | | **Developed algorithm** | | |
|---|---|---|---|---|---|---|
| | $\xi$ | $\zeta$ | b | $\xi$ | $\zeta$ | b |
| $C_{N+1}$ | -1.25 | -0.04 | 1.25 | -1.26 | -0.06 | 1.26 |
| $C_{N+2}$ | -1.11 | -0.46 | 1.20 | -1.12 | -0.44 | 1.20 |
| $C_{N+3}$ | 1.14 | 0.37 | 1.20 | 1.14 | 0.35 | 1.20 |
| $C_{N+4}$ | 1.25 | -0.04 | 1.25 | 1.26 | -0.04 | 1.26 |

TABLE 8.6: B-plane coordinates computed by Valsecchi et al. [43] and by the developed algorithm

Although Valsecchi's method was found to be slightly faster, the algorithm presented in this thesis is more versatile for applications of resonant flyby design. In fact, if a given target inclination is too high (or too low), the algorithm computes what is the fastest way to get there using multiple flybys

and taking into account the limits on the impact parameter. On the other hand, using Valsecchi's method one simply obtains the b-plane coordinates to achieve the target inclination in one flyby, but the point will be within the infeasible zone due to the impact parameter and so will be unusable.

# Chapter 9

# Conclusion and outlook

The classical flyby model in the b-plane and the b-plane circle equation were analytically extended to the case of non-circular orbit of the flyby body. The extended model was first validated and then compared with the classical one against two mission scenarios, highlighting non-negligible differences when the flyby body has a marked orbital eccentricity and/or the flyby takes place at one of the apsidal points. In these cases, the classical model results inadequate to predict the b-plane point leading to a prescribed post-encounter semi-major axis and so the use of the extended model is suggested. Moreover, as the extension is purely analytical, the two models are no different in terms of computational effort, and hence one makes no mistake in always implementing the extended model. The way the model currently stands, its main limitation is that it was developed under the approximation of zero MOID between the orbits of the small body and the flyby body, as per Öpik's classical theory. Since Valsecchi et al. [11] already extended the classical theory to finite MOID close encounters, a future work could do the same for the model developed in this thesis. Another step forward could be to include perturbations within the planet's SOI in the model. One way to do so would be to adjust the b-plane circle equation through some correction coefficients, as already done by Masat [29] for the classical model.

In the second part of the work, a dynamic programming approach based on the extended b-plane model was proposed to address the unperturbed design of multiple resonant flybys trajectories, including the number of flybys as one of the optimisation variables. The complexity of the problem lies in the size of the solution space and its mixed continuous-combinatorial nature, owing to the presence of both discrete and continuous decision variables at each flyby, respectively the semi-major axis and the inclination. The continuous part of the problem was reduced to purely combinatorial by identyfing a principle for the choice of the inclination that allows to achieve a target orbital state in the minimum number of flybys as possible. The developed algorithm was tested against the design of Solar Orbiter's resonant phase with Venus. The resulting solution featured the same flybys number and the same resonance ratios as the baseline mission trajectory, giving strong proof of the proposed design method. The efficiency of the dynamic programming approach was assessed by comparing its performances with those of a conventional brute-force method. The computational savings grow exponentially with the number of consecutive flybys and the size of the set of admissible

resonance conditions. Several directions could be followed to increase the exploitability of the algorithm. For instance, although achieving a target orbit in the fewest number of flybys reflects a typical goal in various mission instances, it would be useful to base the choice of inclination also on other kinds of constraints, perhaps related to coverage requirements or communication time in safe mode. Another certainly valuable direction for future work is to include perturbing effects in the model, in order to increase the accuracy of the solution and facilitate the transition to a fully-integrated trajectory. In conclusion, the proposed algorithm provides a fast and reliable tool for the preliminary design of multiple resonant flyby trajectories, which could serve as reasonable starting point for numerical methods to faster converge to more complete trajectory solutions. Indeed, even the information on the total number of required resonant GAMs alone allows to drastically limit the dimensionality of the solution space to probe.

# Appendix A

# Orbit determination from a, e, i and a position vector

Suppose that the semi-major axis a, eccentricity e, inclination i and a position vector $\mathbf{r}$ of a Keplerian orbit are known. The goal is to find the remaining osculating elements of the orbit, i.e. the right ascension of the ascending node (RAAN) $\Omega$, the argument of periapsis $\omega$ and the true anomaly $\vartheta$.

The solution starts by determining the angular momentum versor $\hat{\mathbf{h}}$, where the hat notation is used for unit vectors. Identifying with $\hat{\mathbf{k}}$ the direction of the z-axis, the inclination is the angle between $\hat{\mathbf{h}}$ and $\hat{\mathbf{k}}$. Moreover, $\hat{\mathbf{h}}$ is always perpendicular to the plane of the orbital motion. Hence,

$$\begin{cases} \hat{\mathbf{h}} \cdot \hat{\mathbf{k}} = \cos i \\ \hat{\mathbf{h}} \cdot \mathbf{r} = 0 \\ \|\hat{\mathbf{h}}\| = 1 \end{cases} \tag{A.1}$$

Denoting with $h_1, h_2, h_3$ the components of $\hat{\mathbf{h}}$ and with $r_1, r_2, r_3$ those of $\mathbf{r}$, Equation (A.1) can be written as:

$$\begin{cases} h_3 = \cos(i) \\ r_1 h_1 + r_2 h_2 + r_3 h_3 = 0 \\ \sqrt{h_1^2 + h_2^2 + h_3^2} = 1 \end{cases} \tag{A.2}$$

Collecting the components of $\hat{\mathbf{h}}$ on the left-hand-sides of the three identities leads to:

$$\begin{cases} h_3 = \cos(i) \\ h_2 = -\frac{1}{r_2} [r_3 \, h_3 + r_1 \, h_1] \\ h_1 = \pm\sqrt{1 - h_2^2 - h_3^2} = \pm\sqrt{1 - \frac{1}{r_2^2} [r_3 \cos(i) + r_1 \, h_1]^2 - \cos(i)^2} \end{cases} \tag{A.3}$$

The $\pm$ in Equation (A.3) highlights the existence of two versors $\hat{\mathbf{h}}_\mathbf{1}$, $\hat{\mathbf{h}}_\mathbf{2}$ and hence two possible orbits associated with a position vector and a value of inclination. The third identity in Equation (A.3) can be solved by minimisation

or a root-finding algorithm. In any case, the interval of possible values for $h_1$ is determined by the argument of the square root, which must be positive. The endpoints $x_1$ and $x_2$ of such interval are:

$$x_{1/2} = \frac{-B \pm \sqrt{B^2 - 4\,AC}}{2A} \tag{A.4}$$

where

$$\begin{cases} A = r_1^2 \\ B = 2 r_3 r_1 h_3 \\ C = r_3^2 h_3^2 - r_2^2 (1 - h_3^2) \end{cases} \tag{A.5}$$

The directions of the two lines of nodes $\hat{\mathbf{n}}_\mathbf{1}$, $\hat{\mathbf{n}}_\mathbf{2}$ associated with $\hat{\mathbf{h}}_\mathbf{1}$, $\hat{\mathbf{h}}_\mathbf{2}$ are:

$$\begin{aligned} \hat{\mathbf{n}}_1 &= \hat{\mathbf{h}}_1 \times \hat{\mathbf{k}} \\ \hat{\mathbf{n}}_2 &= \hat{\mathbf{h}}_2 \times \hat{\mathbf{k}} \end{aligned} \tag{A.6}$$

Denoting with $n_{ij}$ the j-th component of the i-th vector $\hat{\mathbf{n}}$, $\hat{\mathbf{n}}_\mathbf{1}$, $\hat{\mathbf{n}}_\mathbf{2}$ are used to compute the two values of RAAN

$$\begin{aligned} \Omega_1 &= \operatorname{atan2}(n_{12}, n_{11}) \\ \Omega_2 &= \operatorname{atan2}(n_{22}, n_{21}) \end{aligned} \tag{A.7}$$

and those of the argument of latitude

$$\begin{aligned} u_1 &= \angle(\hat{\mathbf{n}}_1, \mathbf{r}) \\ u_2 &= \angle(\hat{\mathbf{n}}_2, \mathbf{r}) \end{aligned} \tag{A.8}$$

Note that $u_1$ and $u_2$ are oriented angles, whereby Equation (A.8) must be computed considering the direction of rotation. The orbit equation in polar coordinates is:

$$r = \frac{a\left(1 - e^2\right)}{1 + e \cdot \cos(\vartheta)} \tag{A.9}$$

thus

$$\vartheta = \cos^{-1}\left(\left(\frac{p}{r} - 1\right) \cdot \frac{1}{e}\right) \tag{A.10}$$

where p is the semi-latus rectum ($p = a(1 - e^2)$). Because of the ambiguity of the inverse cosine function, Equation (A.10) leads to two possible values $\vartheta_1$, $\vartheta_2$ of true anomaly. Both can be associated to each of $u_1$, $u_2$, leading in turn to four possible orbits, whose arguments of periapsis can be computed as:

$$\begin{aligned} \omega_1 &= u_1 - \vartheta_1 \\ \omega_2 &= u_1 - \vartheta_2 \\ \omega_3 &= u_2 - \vartheta_1 \\ \omega_4 &= u_2 - \vartheta_2 \end{aligned} \tag{A.11}$$

In conclusion, there are four orbits joining **r** and featuring the same values of a, e, i, and their orbital elements are:

$$\begin{aligned} \text{kep}_1 &= \{\, a\, e\, i\, \Omega_1\, \omega_1\, \vartheta_1 \,\} \\ \text{kep}_2 &= \{\, a\, e\, i\, \Omega_1\, \omega_2\, \vartheta_2 \,\} \\ \text{kep}_3 &= \{\, a\, e\, i\, \Omega_2\, \omega_3\, \vartheta_1 \,\} \\ \text{kep}_4 &= \{\, a\, e\, i\, \Omega_2\, \omega_4\, \vartheta_2 \,\} \end{aligned} \tag{A.12}$$

The procedure has been applied to the initial orbit of the resonant phase of Solar Orbiter, which is reported in Table 8.1. The four resulting orbits are shown in Figure A.1. It can be noted that the differences relate to whether the encounter occurs at the ascending or descending node, and from inside or outside the planet's orbit.

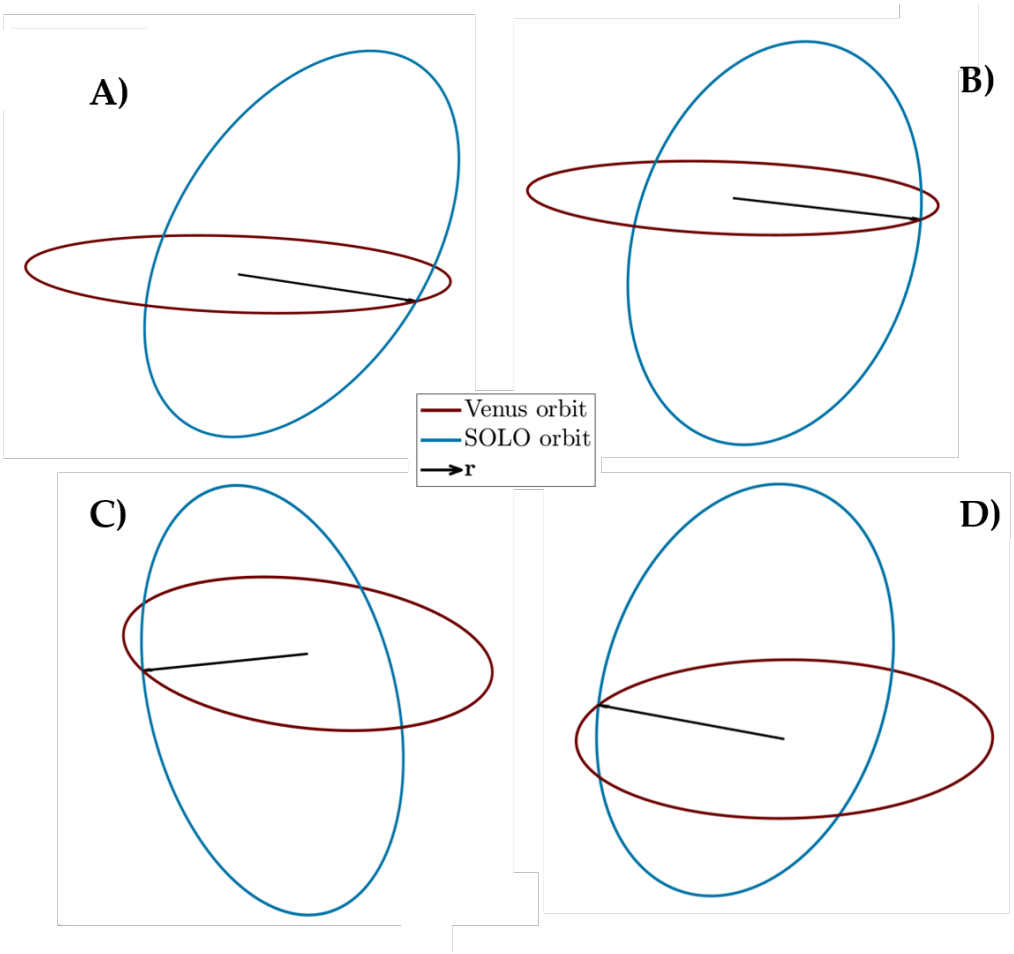


FIGURE A.1: Encounters at A) ascending node, from inside planet's orbit, B) ascending node, from outside planet's orbit, C) descending node, from inside planet's orbit, D) descending node, from outside planet's orbit

# Bibliography


[1] Yu. F. Golubev et al. «Gravity Assist Maneuvers Near Venus for Exit to Non-Ecliptic Positions: Resonance Asymptotic Velocity». In: *Solar System Research* 53.4 (2019), pp. 245–253. DOI: 10.20948/prepr-2018-222.

[2] Yu. F. Golubev et al. «Formation of High Inclined Orbits to the Ecliptic by Multiple Gravity Assist Maneuvers». In: *Journal of Computer and System Sciences International* 56.2 (2017), pp. 275–299. DOI: 10.1134/S1064230717020083.

[3] A. V. Labunsky, O. V. Papkov, and K. G. Sukhanov. *Multiple Gravity Assist Interplanetary Trajectories*. CRC Press: Earth Space Institute Book Series, 1998. ISBN: 9780203746288. DOI: 10.1201/9780203746288.

[4] European Space Agency. *Solar Orbiter Definition Study Report (Red Book)*. Tech. rep. 2001.

[5] European Space Agency. *JUpiter ICy moons Explorer Exploring the emergence of habitable worlds around gas giants (Red Book)*. Tech. rep. 2014.

[6] A. Grushevskii et al. «Optimal Spacecraft Asymptotic Velocity for the High Inclined Orbits Formation Using Gravity Assists in the Planetary Systems». In: *Journal of Physics: Conference Series* 1730.1 (2021). DOI: 10.1088/1742-6596/1730/1/012061.

[7] E. J. Öpik. *Interplanetary Encounters: Close-Range Gravitational Interactions*. Elsevier Scientific Publishing Co., 1976.

[8] G. B. Valsecchi et al. «The Distribution of Energy Perturbations at Planetary Close Encounters». In: *Celestial Mechanics and Dynamical Astronomy* 78 (2000). DOI: 10.1023/A:1011151925074.

[9] D. Farnocchia et al. «Planetary Encounter Analysis on the B-plane: a Comprehensive Formulation». In: *Celestial Mechanics and Dynamical Astronomy* 131.36 (2019). DOI: 10.1007/s10569-019-9914-4.

[10] R. Greenberg, A. Carusi, and G. B. Valsecchi. «Outcome of Planetary Close Encounters: a Systematic Comparison of Methodologies». In: *Icarus* 75 (1988). ISSN: 0019-1035. DOI: 10.1016/0019-1035(88)90125-X.

[11] G. B. Valsecchi et al. «Resonant Returns to Close Approaches: Analytical theory». In: *Astronomy & Astrophysics* 408 (2003). DOI: 10.1051/0004-6361:20031039.

[12] A. Carusi, G. B. Valsecchi, and R. Greenberg. «Planetary Close Encounters: Geometry of Approach and Post-Encounter Orbital Parameters». In: *Celestial Mechanics and Dynamical Astronomy* 49 (1990). DOI: 10.1007/bf00050709.

[13] A. Milani, G. B. Valsecchi, and S. R. Chesley. «Close Approaches of Asteroid 1999 AN10: Resonant and Non-Resonant Returns». In: *Astronomy & Astrophysics* 346 (1999).

[14] Y. Langevin. «Chemical and Solar Electric Propulsion Options for a Cornerstone Mission to Mercury». In: *Acta Astronautica* 47.2-9 (2000), pp. 443–452. ISSN: 0094-5765. DOI: 10.1016/S0094-5765(00)00084-9.

[15] F. F. Tisserand. *Traité de Mécanique Céleste*. Vol. 4. Gauthier-Villars, Paris, 1896, p. 203. ISBN: 026639955X.

[16] M. Ceriotti. «Global Optimisation of Multiple Gravity Assist Trajectories». PhD thesis. University of Glasgow, 2010.

[17] R. L. Anderson. «Resonant Flyby and Tour Design Using Heteroclinic Connections». In: *American Astronomical Society, 43th annual DDA meeting* 53.4 (2012), pp. 245–253.

[18] C. W. Cryer. *Rodrigues' Formula and the Classical Orthogonal Polynomials*. Tech. rep. 1968.

[19] A. A. Wolf. «Touring the Saturnian System». In: *Space Science Reviews* 104 (2002), pp. 101–128. DOI: 10.1023/A:1023692724823.

[20] H. D. Curtis. *Orbital Mechanics for Engineering Students*. Third ed. Amsterdam: Elsevier Butterworth Heinemann, 2010. ISBN: 978-0-08-097747-8. DOI: 10.1016/C2011-0-69685-1.

[21] L. W. Bandermann and R. D. Wolstencroft. «Energy Changes in Close Planetary Encounters». In: *Monthly Notices of the Royal Astronomical Society* 152.4 (July 1971), pp. 377–382. ISSN: 0035-8711. DOI: 10.1093/mnras/152.4.377.

[22] G. W. Wetherill and L. P. Cox. «Range of Validity of the Two-Body Approximation in Models of Terrestrial Planet Accumulation. I - Gravitational Perturbations». In: *Icarus* 60 (Oct. 1984). DOI: 10.1016/0019-1035(84)90137-4.

[23] A. Carusi, L. Kresák, and G. B. Valsecchi. «Conservation of the Tisserand Parameter at Close Encounters of Interplanetary Objects with Jupiter». In: *Earth, Moon, and Planets* 68 (1995), pp. 71–94. ISSN: 1573-0794. DOI: 10.1007/BF00671499.

[24] M. Vaquero and K. C. Howell. «Poincaré Maps and Resonant Orbits in the Circular Restricted Three-Body Problem». In: *AAS/AIAA, Astrodynamic Specialist Conference* (2011).

[25] F. Topputo, E. Belbruno, and M. Gidea. «Resonant Motion, Ballistic Escape, and their Applications in Astrodynamics». In: *Advances in Space Research* (2008). ISSN: 0273-1177. DOI: 10.1016/j.asr.2008.01.017.

[26] M. Vasile and M. Locatelli. «A Hybrid Multiagent Approach for Global Trajectory Optimization». In: *Journal of Global Optimization* 44 (2009), pp. 461–479. DOI: 10.1007/s10898-008-9329-3.

[27] L. Federici, A. Zavoli, and G. Colasurdo. «Preliminary Capture Trajectory Design for Europa Tomography Probe». In: *International Journal of Aerospace Engineering* (2018). ISSN: 1687-5966. DOI: 10.1155/2018/6890173.

[28] M. Vasile and S. Campagnola. «Design of Low-Thrust Multi-Gravity Assist Trajectories to Europa». In: *Journal of the British Interplanetary Society* 62.1 (2009), pp. 15–31.

[29] A. Masat. «B-plane Orbital Resonance Analysis and Applications: Perturbed Semi-Analytical Model for Planetary Protection and Defence Applied to Ballistic Resonant Flyby Design». MA thesis. Scuola di Ingegneria Industriale e dell'Informazione, DAER - Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano. Supervisor: Camilla Colombo. Co-supervisor: Matteo Romano., 2019.

[30] F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research*. 9th ed. McGraw-Hill, New York (NY), United States, 2010.

[31] K. P. Lin, Y. Z. Luo, and G. J. Tang. «Space Station Orbit Design Using Dynamic Programming». In: *Acta Astronautica* 89 (2013), pp. 195–204. ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2013.04.011.

[32] C. Colombo, M. Vasile, and G. Radice. «Optimal Low-Thrust Trajectories to Asteroids through an Algorithm Based on Differential Dynamic Programming». In: *Celestial Mechanics and Dynamical Astronomy* 105 (2009), pp. 75–112. DOI: 10.1007/s10569-009-9224-3.

[33] M. Nugnes and C. Colombo. «Low-Thrust Trajectory Optimisation through Differential Dynamic Programming Method based on Keplerian Orbital Elements». In: 70th International Astronautical Congress: IAC Proceedings (2019). URL: http://hdl.handle.net/11311/1117583.

[34] G. Lantoine and R. Russell. «A Hybrid Differential Dynamic Programming Algorithm for Robust Low-Thrust Optimization». In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. DOI: 10.2514/6.2008-6615.

[35] R. Bellman. *Dynamic Programming*. Dover Publications, 2003. ISBN: 978-0-486-42809-3.

[36] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. 3rd ed. Athena Scientific, Belmont, Massachusetts, 2005. ISBN: 1886529094.

[37] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994. ISBN: 0471619779.

[38] Daniel W. Stroock. *An Introduction to Markov Processes*. 2nd ed. Springer, Berlin, Heidelberg, 2014. ISBN: 978-3-540-26990-8.

[39] W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. 2nd ed. John Wiley & Sons, Inc., 2007. ISBN: 978-0-470-17155-4.

[40] D. Myatt et al. «Advanced Global Optimisation for Mission Analysis and Design». In: 2004. URL: http://www.esa.int/gsp/ACT/publications/index.htm.

[41] N. Bradley and R. Russell. «A Continuation Method for Converting Trajectories from Patched Conics to Full Gravity Models». In: *Journal of the Astronautical Sciences* 152 (Oct. 2014). DOI: 10.1007/s40295-014-0017-x.

[42] D. Ellison and J. Englander. «High-Fidelity Multiple-Flyby Trajectory Optimization Using Multiple-Shooting». In: *AAS/AIAA Astrodynamics Specialist Conference*. Aug. 2019.

[43] G.B.Valsecchi, E.M.Alessi, and A.Rossi. «An Analytical Solution for the Swing-By Problem». In: *Celestial Mechanics & Dynamical Astronomy* 123 (2016), pp. 151–166. ISSN: 1572-9478. DOI: 10.1007/s10569-015-9631-6.

[44] G. Janin. «Trajectory Design for the Solar Orbiter Mission». In: *Monografías de la Real Academia de Ciencias de Zaragoza* 25 (2004), pp. 177–218.

[45] R. Jehn et al. «BepiColombo - A Mission to Mercury». In: 21st International Symposium on Space Flight Dynamics (ISSFD). 2009.