AAS 21-725

DYNAMIC PROGRAMMING APPLIED TO RESONANT FLYBY DESIGN

Giulio Campiti,* Alessandro Masat,† and Camilla Colombo‡

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 and is analytically extended in this work to account for their actual eccentricity. The classical model and the extended one 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. A dynamic programming approach to the design and optimization of unperturbed resonant trajectories is proposed, using as discrete decision variables a set of resonance ratios 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 reduce the distance at perihelion.

INTRODUCTION

The implementation of resonant gravity assist maneuvers (GAMs) allows to achieve significant propellant consumption reduction and often makes the difference between feasible and unfeasible trajectories. For instance, a certain number of real-life missions involves substantial variations of the orbital inclination that, if actively performed by the spacecraft engines, would require an excessively large amount of Δv. Two European Space Agency's (ESA) missions can be taken as reference, Solar Orbiter,[1] aimed at exploring the inner heliosphere from moderate out-of-ecliptic latitudes, and JUpiter ICy moons Explorer (JUICE),[2] which requires a high inclination orbit phase to investigate the non-equatorial regions of Jupiter's environment. These missions were only made possible by implementing sequences of resonant GAMs.

A powerful tool used in the preliminary design of resonant trajectories is the b-plane, a target plane through which a body passes on a given encounter with a planet or moon. In 1976, Öpik[3] defined a convenient coordinate system based on this 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. In Valsecchi et al.[4] the theory was extended to the case in which the Minimum Orbital Intersection Distance (MOID) between

*M.Sc. Student in Space Engineering, DAER-Department of Aerospace Science and Technology, Politecnico di Milano, Via G. La Masa 26, 20156, Milano, Italy.

†PhD Candidate, DAER-Department of Aerospace Science and Technology, Politecnico di Milano, Via G. La Masa 26, 20156, Milano, Italy.

‡Associate Professor, DAER-Department of Aerospace Science and Technology, Politecnico di Milano, Via G. La Masa 26, 20156, Milano, Italy.

the orbits of the small body and the flyby body has a finite value. In another paper, the same authors used Öpik's frame to show 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.[5] 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. However, a common limitation to Öpik's theory and the b-plane circle equation is that they were 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.[6]

The proposed work extends the validity of the link between orbits with given semi-major axis and the b-plane circle equation, allowing for the flyby body around the primary to be elliptical. The aim is to improve the accuracy of the classical model through a purely analytical extension, such 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 study trajectories with fair accuracy while requiring very low computational efforts.

In mission analysis, the design of unperturbed resonant trajectories can be formulated as an optimization problem, where the resonance ratio between the orbital period of the spacecraft and that of the planet represents a discrete decision variable at each flyby. Hence, the search for an optimal sequence of resonant encounters features a mixed combinatorial-continuous nature. The dimensionality of the problem is also affected by the total number of flybys, which is unknown a priori. Masat et al.[7] formulated and implemented a multi-level optimization scheme to solve the problem of reaching a target orbit in a prescribed number N of resonant encounters. At the highest level of their strategy, a direct search algorithm finds an optimal splitting of the total Δv target into N sub-Δv. The proposed work expands on those results, including the number of flybys as optimization variable and attempts to define a systematic procedure that does not rely on heuristic methods, which not necessarily discover the optimal solution. Given the presence of discrete elements in the design process, the problem is approached with the technique of dynamic programming, which provides a systematic method to make a sequence of interrelated decisions.[8]

The paper is structured as follows: the first section presents a brief overview of classical Öpik's theory and the circle equation in the b-plane. Then the model is extended to elliptical planetary orbits and the dynamic programming approach is proposed. Finally, both the model extension and the developed algorithm are tested on some mission design applications, including Solar Orbiter's resonant phase with Venus.

CLASSICAL B-PLANE RESONANCE THEORY

Öpik's theory is based on a two-body patched-conics approximation and was derived using Jacobi normalized units, the same commonly employed in the non-dimensionalization of the circular restricted three-body problem. The planet has an orbital period of 2π 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 be the non-dimensional semi-major axis, eccentricity and inclination of a massless particle in orbit around the Sun, where i is reckoned with respect

to the planet's orbital plane. 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, the Y-axis is aligned with the planet's direction of motion and the and Z-axis completes the right-handed triad. In this reference frame, the components of the planetocentric velocity of the particle are:[9]

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

The encounter geometry is defined in terms of two angles, θ and ϕ. The former is the angle between $\mathbf{U}$ and the Y-axis and the latter is the angle formed by the $Y - Z$ plane and the plane containing $\mathbf{U}$ and the X-axis.[9] Thus, the vector $\mathbf{U}$ can be alternatively expressed as:

$$\begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} U\sin\theta\sin\phi \\ U\cos\theta \\ U\sin\theta\cos\phi \end{bmatrix} \tag{2}$$

with U given by

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

Using the previous expressions, the angle θ can be simply expressed as function of a and U:[3]

$$\cos\theta = \frac{1 - 1/a - U^2}{2\,U} \tag{4}$$

B-plane Reference Frame

The b-plane is defined as the plane orthogonal to $\mathbf{U}$ and containing the centre of the planet. Based on this definition, a second planetocentric reference frame (ξ, η, ζ) is introduced such that the (ξ, ζ)-axes lie on the b-plane and $\hat{\boldsymbol{\eta}}$ is perpendicular to it. The ζ-axis is parallel to the projection of the planet's velocity on the b-plane but with opposite direction and $\hat{\boldsymbol{\xi}}$ completes a right-handed reference system:

$$\hat{\boldsymbol{\eta}} = \frac{\mathbf{U}}{\|\mathbf{U}\|}; \quad \hat{\boldsymbol{\xi}} = \frac{\mathbf{U}\times\mathbf{v}_{pl}}{\|\mathbf{U}\times\mathbf{v}_{pl}\|}; \quad \hat{\boldsymbol{\zeta}} = \hat{\boldsymbol{\xi}}\times\hat{\boldsymbol{\eta}} \tag{5}$$

When the particle crosses the b-plane, $\eta = 0$ and the impact parameter b identifies the closest distance between the object and the planet:

$$b^2 = \xi^2 + \zeta^2 \tag{6}$$

According to the patched-conics approximation, the flyby is modeled as an instantaneous rotation of the incoming vector $\mathbf{U}$ into the outgoing velocity $\mathbf{U}'$, with the superscript $'$ identifying post-encounter conditions. $\mathbf{U}$ is rotated by an angle γ, called deflection angle, in the direction given by ψ:[9]

$$\tan\frac{\gamma}{2} = \frac{m}{bU^2} = \frac{c}{b} \tag{7}$$

$$\begin{bmatrix} b\sin\psi \\ b\cos\psi \end{bmatrix} = \begin{bmatrix} \xi \\ \zeta \end{bmatrix} \tag{8}$$

where m is the planet's mass in units of solar mass and c is a characteristic length.

As shown by Valsecchi et al.,[5] the locus of points in the b-plane leading to a post-encounter orbit with prescribed semi-major axis is identified by a unique circle of equation:

$$\xi^2 + \zeta^2 - \frac{2c\sin\theta}{\cos\theta' - \cos\theta}\zeta + \frac{c^2\left(\cos\theta' + \cos\theta\right)}{\cos\theta' - \cos\theta} = 0 \tag{9}$$

A interesting limit condition is obtained when $a' = a$, i.e. the encounter changes the orbital eccentricity and inclination of the particle but keeping constant its orbital period. In this case, the circle degenerates into a straight horizontal line of equation:

$$\zeta = \frac{c\cos\theta}{\sin\theta} \tag{10}$$

Resonance definition

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. This resonance condition can be expressed as:

$$hT' = kT_{pl} \tag{11}$$

Using Jacobi normalized units, $T_{pl} = 2\pi$ and $T' = 2\pi a'^{3/2}$, and the post-encounter semi-major axis can be directly linked to the k/h ratio:

$$a' = \left(\frac{k}{h}\right)^{2/3} \tag{12}$$

Note that, given a generic semi-major axis, the above equation can be inverted to find a k/h ratio also in the non-resonant case, with the only difference that the latter will be a generic real number not obtainable as the ratio between two integers.

EXTENSION OF THE THEORY TO ECCENTRIC PLANETARY ORBITS

The block diagram of Figure 1 presents the main steps required to design a generic flyby in the b-plane. The dashed blue boxes evidence 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 sphere of influence (SOI) is a degree of freedom of the problem. Theoretically, any point on the b-plane could be chosen to analyze 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 generalize 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 $\mathbf{U}$.

Generalization of the Formulae

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, here denoted with l_{ref} and v_{ref}. The former is taken as the planet's distance to the Sun, which is now generally

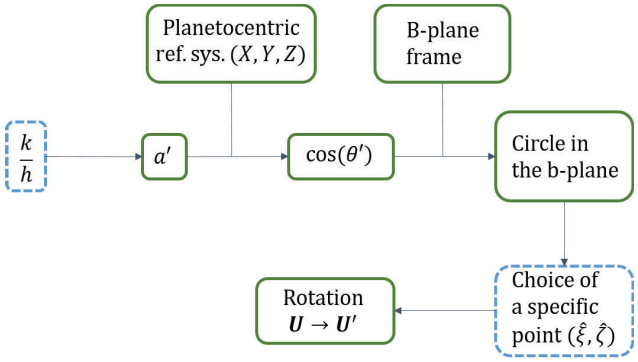

Figure 1. Steps required to design a flyby in the b-plane

different from the planet's semi-major axis a_{pl}. The latter is taken as the velocity of a fictitious planet moving on a circular orbit at a distance l_{ref} from the Sun, and can then be computed as:

$$v_{ref} = \sqrt{\frac{\mu_\odot}{l_{ref}}} \tag{13}$$

where $\mu_\odot$ is the Sun's gravitational parameter. The characteristic velocity is now different from the true planet's velocity $\mathbf{v}_{pl}$. A parameter χ is introduced:

$$\chi = \frac{l_{ref}}{a_{pl}} \tag{14}$$

which is equal to 1 only in case of circular orbit of the planet. Each non-dimensional quantity referred to the above described characteristic units will be denoted with the overline notation (e.g. $\bar{a}, \bar{U}$, etc.). All the other variables are to be considered dimensional quantities.

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 (11), which is independent of the planet's orbital eccentricity. The orbital periods of the planet's and that of the particle are:

$$T_{pl} = 2\pi\sqrt{\frac{a_{pl}^3}{\mu_\odot}}, \qquad T' = 2\pi\sqrt{\frac{a'^3}{\mu_\odot}}; \tag{15}$$

Plugging these expressions into Equation (11) yields:

$$\left(\frac{a'}{a_{pl}}\right)^{3/2} = \frac{k}{h} \quad \longrightarrow \quad \frac{a'}{a_{pl}} = \left(\frac{k}{h}\right)^{2/3} \tag{16}$$

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

$$\frac{1}{l_{ref}} \cdot \frac{a'}{a_{pl}} = \frac{1}{l_{ref}} \cdot \left(\frac{k}{h}\right)^{2/3} \quad \longrightarrow \quad \bar{a}' = \frac{1}{\chi}\left(\frac{k}{h}\right)^{2/3} \tag{17}$$

If the planet was in circular orbit, Equation (4) 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 characterized 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. The new frame can be obtained by performing a clockwise rotation of the X, Y-axes around the Z direction, as clear from Figure 2. The rotation angle is given by the planet's flight path angle. Note that generally the $\hat{Y}$ direction is not perpendicular to the radial direction. The two frames only coincide when the planet is at the apsides or when it is in circular orbit. The encounter geometry is then characterized by introducing again two angles θ and ϕ, with definitions analogous to those of Equation (2) but referred now with respect to the $(\hat{X}, \hat{Y}, \hat{Z})$ reference frame.

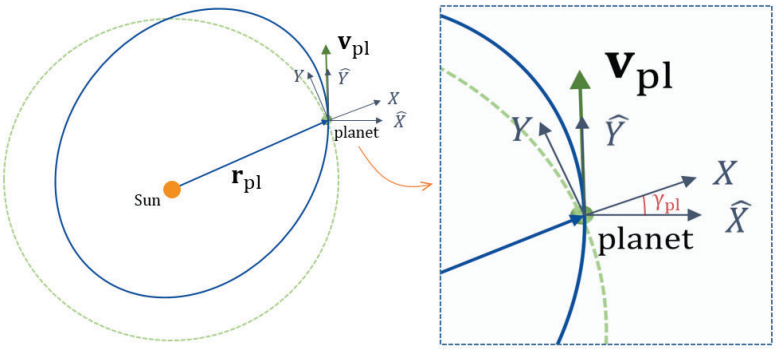

Figure 2. $(\hat{\mathbf{X}}, \hat{\mathbf{Y}}, \hat{\mathbf{Z}})$ reference frame. $\gamma_{\mathbf{pl}}$ identifies the flight path angle of the planet at the encounter position

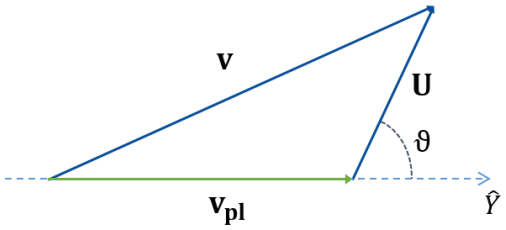

Figure 3. Heliocentric velocity $\mathbf{v}$ as vector sum of $\mathbf{v}_{\mathbf{pl}}$ and $\mathbf{U}$

Referring to Figure 3, the particle's heliocentric velocity can be obtained by applying the cosine law to the triangle formed by the three vectors:

$$v^2 = v_{pl}^2 + U^2 + 2\,U\,v_{pl}\cos\theta \tag{18}$$

Equation (18) can be expressed in non-dimensional units by dividing both sides by the square of the characteristic velocity:

$$\bar{v}^2 = \frac{v_{pl}^2}{v_{ref}^2} + \bar{U}^2 + 2\,\bar{U}\frac{v_{pl}}{v_{ref}}\cos(\theta) \tag{19}$$

For a planet in circular orbit, the ratio v_{pl}/v_{ref} is equal to 1 and the original expression is obtained (Equation (4)), whereas for the elliptical cases, v_{pl} can be expressed as function of v_{ref}. From the principle of conservation of energy:

$$v_{pl} = \sqrt{\mu_\odot\left(\frac{2}{r_{pl}} - \frac{1}{a_{pl}}\right)} \tag{20}$$

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

$$v_{pl} = \sqrt{\frac{\mu_\odot}{r_{pl}}\left(2 - \frac{r_{pl}}{a_{pl}}\right)} = \sqrt{\frac{\mu_\odot}{r_{pl}}\left(2-\chi\right)} \tag{21}$$

Equation (13) can be used to obtain:

$$v_{pl} = v_{ref}\sqrt{2-\chi} \quad \Rightarrow \quad \frac{v_{pl}}{v_{ref}} = \sqrt{2-\chi} \tag{22}$$

The last identity is used to rewrite Equation (19) as function of χ:

$$\bar{v}^2 = 2 - \chi + \bar{U}^2 + 2\bar{U}\sqrt{2-\chi}\cos(\theta) \tag{23}$$

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

$$\begin{gathered} v^2 = \mu_\odot\left(\frac{2}{r} - \frac{1}{a}\right) = \frac{\mu_\odot}{r}\left(2 - \frac{r}{a}\right) = v_{ref}^2\left(2 - \frac{1}{\bar{a}}\right) \\ \Rightarrow \quad \bar{v}^2 = 2 - \frac{1}{\bar{a}} \end{gathered} \tag{24}$$

The two expressions of $\bar{v}^2$ of Equations (23) and (24) can be used to solve for θ, yielding:

$$\cos(\theta) = \frac{\chi - 1/\bar{a} - \bar{U}^2}{2\bar{U}\sqrt{2-\chi}} \tag{25}$$

Note that Equation (25) is a generalization of Equation (4), since $\chi = 1$ provides exactly the original expression from Öpik.

The particle's heliocentric velocity at the encounter can be expressed in the (X, Y, Z) frame as:

$$\bar{\mathbf{v}} = \begin{bmatrix} \pm\sqrt{2 - \frac{1}{\bar{a}} - \bar{a}\left(1-e^2\right)} \\ \sqrt{\bar{a}\left(1-e^2\right)}\cos i \\ \pm\sqrt{\bar{a}\left(1-e^2\right)}\sin i \end{bmatrix} \tag{26}$$

Equation (26) was obtained following the same steps of Öpik,[3] with the only difference being the characteristic units used in the non-dimensionalization. Knowing $\bar{\mathbf{v}}$, the planetocentric velocity $\bar{\mathbf{U}}$ is obtained as the vector difference between $\bar{\mathbf{v}}$ and the planet's velocity $\bar{\mathbf{v}}_{pl}$, always expressed in the frame (X, Y, Z). Referring to Figure 2, $\bar{\mathbf{v}}_{pl}$ can be computed as:

$$\mathbf{v}_{pl} = \|\mathbf{v}_{pl}\|\begin{bmatrix} \sin\gamma_{pl} \\ \cos\gamma_{pl} \\ 0 \end{bmatrix} \quad \rightarrow \quad \bar{\mathbf{v}}_{pl} = \sqrt{2-\chi}\begin{bmatrix} \sin\gamma_{pl} \\ \cos\gamma_{pl} \\ 0 \end{bmatrix} \tag{27}$$

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

$$\bar{\mathbf{U}} = \bar{\mathbf{v}} - \bar{\mathbf{v}}_{pl} = \begin{bmatrix} \pm\sqrt{2 - \frac{1}{\bar{a}} - \bar{a}(1-e^2)} - \sqrt{2-\chi}\sin\gamma_{pl} \\ \sqrt{\bar{a}(1-e^2)}\cos i - \sqrt{2-\chi}\cos\gamma_{pl} \\ \pm\sqrt{\bar{a}\left(1-e^2\right)}\sin i \end{bmatrix} \tag{28}$$

They can be obtained from the orbital elements, so you can take them as known.

The remaining blocks of Figure 1 to be generalized are those associated with the b-plane reference frame. Since the eccentricity of a planet's orbit does not enter the definition of b-plane, Equation (5) can be used to compute the (ξ, η, ζ)-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 in astrodynamics to model flybys in the general case of planets in elliptical orbits. The only necessary changes to Equations (7) and (8) 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-dimensionalization step might lead to different expressions. However, starting from the equations written in dimensional form and using the characteristic units introduced in this section yields the following identities:

$$\tan\frac{\gamma}{2} = \frac{\bar{m}}{\bar{b}\bar{U}^2} \tag{29}$$

$$\begin{bmatrix} \bar{b}\sin\psi \\ \bar{b}\cos\psi \end{bmatrix} = \begin{bmatrix} \bar{\xi} \\ \bar{\zeta} \end{bmatrix} \tag{30}$$

which look the same of Equations (7) and (8). This was not a foregone conclusion, but is due to the particular choice made regarding the definition of l_{ref} and v_{ref}. Despite the similarity, the above equations and those of the classical theory do not lead to the same results. In Equations (29) and (30), 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. Finally, the derivation of b-plane circles is solely based on the deflection equations,[3] thus Equation (9) also remains unchanged in the extended model. However, for the same pre- and post-encounter semi-major axes (a, a'), the two models lead to different values of $\cos\theta$, $\cos\theta'$ and hence to different circles. In Figure 4, several circles are computed with the two models. Each pair of close circles is associated with a same k/h ratio, specified next to each of them.

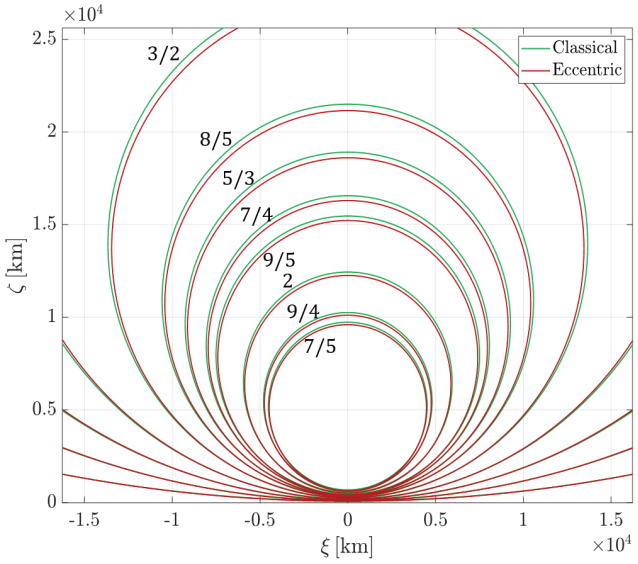

Figure 4. B-plane circles computed through the two models

Lastly, Table 1 provides a summary of the main differences between the classical model based on Öpik's theory and the extended one presented in this section.

Table 1. Main differences between the classical theory and the extended model

	Classical model	Eccentric extension
Characteristic 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 $a \leftrightarrow \cos\theta$	$\cos\theta = \frac{1-1/a-U^2}{2U}$	$\cos\theta = \frac{\chi-1/\bar{a}-\bar{U}^2}{2\bar{U}\sqrt{2-\chi}}$
Velocity $\bar{\mathbf{U}}$ (X, Y, Z-frame)	$\begin{bmatrix} \pm\sqrt{2-1/a-a\,(1-e^2)} \\ \sqrt{a\,(1-e^2)}\cos i - 1 \\ \pm\sqrt{a\,(1-e^2)}\sin i \end{bmatrix}$	$\begin{bmatrix} \pm\sqrt{2-\frac{1}{\bar{a}}-\bar{a}\,(1-e^2)} - \sqrt{2-\chi}\sin\gamma_{pl} \\ \sqrt{\bar{a}\,(1-e^2)}\cos i - \sqrt{2-\chi}\cos\gamma_{pl} \\ \pm\sqrt{\bar{a}\,(1-e^2)}\sin i \end{bmatrix}$

DYNAMIC PROGRAMMING APPLIED TO RESONANT FLYBY DESIGN

A dynamic programming approach is proposed for the efficient preliminary design of multiple resonant flyby trajectories, adopting the above mentioned theory extension.

Design Problem

The design problem addressed in this section 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. The main assumptions used in the design problem are:

- patched-conics approximation
- resonant flybys, except for the last one
- unperturbed trajectory
- unpowered flybys

The initial and final orbits must join the same position in space of an attracting body to allow the execution of resonant flybys. Since neither perturbations nor maneuvers 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.[7]

Another consequence of neglecting perturbations is that the trajectory between two consecutive flybys corresponds to a Keplerian orbit, which could be characterized through a position vector $\mathbf{r}$ and a velocity vector $\mathbf{v}$. However, the resonance condition implies that each close approach takes place at the same position in space. Considering also that the magnitude of the velocity $\mathbf{U}$ is conserved, the problem features four constants of motion that allow to describe the state of the system using only two independent variables.

As previously mentioned, the patched-conics approximation leaves the entry point of the spacecraft into a planet's SOI as a degree of freedom to be exploited in the design phase. 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. However, instead of using the coordinates (ξ, ζ) as free variables to define the point of injection, a more convenient choice 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 α identifying a specific point on the circle. Figure 5 provides more details about the definition of α, taken as the counterclockwise angle between the ξ-axis and the line joining the centre of the circle and a point belonging to it. 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:

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

Clearly, the intermediate semi-major axes must be in resonance with the planet and so 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 (12). This allows to easily compute the set of admissible resonances for the intermediate flybys, as will be made clear later.

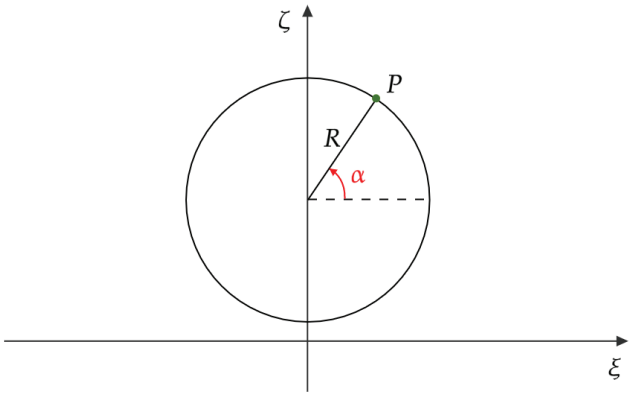

Figure 5. Definition of α

The selection of sequences of resonant semi-major axes introduces a discrete variable into the optimization problem, which combines both continuous and combinatorial optimization. In addition, as the number of total GAMs is allowed to vary, the problem complexity increases due to a search space of varying dimensionality. Masat et al.[7] addressed the same design problem as described above, with the exception that the total number of flybys N was given as input. Their strategy

features a modular structure, where the highest level consists of a stochastic search of an optimal partition of the total Δv target into N contributions, whereas a brute-force approach is used at the lowest level to find the series of pairs $(\bar{a}_i, \alpha_i)$ that most closely lead to the desired sub-Δv targets. The algorithm proposed in this paper attempts to improve on their optimization strategy by avoiding the use of heuristic methods, which do not guarantee to find the optimal solution, and by including the total number of flybys as one of the optimization parameters. The physics of the problem is exploited to formulate a principle that allows to reduce the joint continuous-combinatorial problem to a purely combinatorial one, which can then be tackled with techniques of deterministic dynamic programming.

Dynamic Programming

Dynamic programming[10] provides an efficient computational method to find the optimal solution in sequential decision problems, where the decisions made at one time influence the later available choices. The problem is decomposed into simpler and independent sub-problems, which are tackled one by one starting from the smallest and using the optimal solutions to the smaller problems to gradually solve the larger ones, until the original problem is completely solved. The procedure requires to formulate the problem as a multi-stage decision process,[11] where the *stages* identify the points in which a *policy decision* is required. At each stage, the system might be in different possible conditions called *states*, and the policy decision has the effect to transform the current state into a different one associated with the next stage. According to the Markovian property,[12] each state must retain all the necessary information to determine the optimal policy henceforth. Each decision produces some positive or negative contribution to an overall objective function to be minimized (or maximized). The process of making optimal decisions is based on Bellman's principle of optimality:[13] 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 optimization 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. 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 described in the previous section. Furthermore, since no probability distributions are involved in such problem, it falls into the category of deterministic dynamic programming problems,[14] where the state at stage $n+1$ is uniquely determined by the state and the policy decision at stage n.

Let denote with s_n the state at stage n and with x_n the policy decision selected at that point from a set of admissible decisions S_n. Given s_n, let $f_n(s_n, x_n)$ be the total cost resulting from following the best overall policy at stages $1, 2, \ldots, n-1$ plus the immediate contribution produced by the decision x_n. The optimal decision x_n^* is the one that minimizes the total cost, whose value will be denoted by $f_n^*(s_n)$. Thus,

$$f_n^*(s_n) = \min_{x_n \in S_n} f_n\left(s_n, x_n\right) = f_n\left(s_n, x_n^*\right) \tag{31}$$

Solving the sub-problem associated to stage n means finding the best policy for all possible states. The 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[15] and implies that the stages are numbered in terms of the number of stages completed. For deterministic problems, the optimal policy at each stage can be expressed by the following recursive

relationship:

$$f_n^*(s_n) = \min_{x_n \in S_n} \left\{ c_{s_n x_n} + f_{n-1}^*(x_n) \right\} \tag{32}$$

where $c_{s_n x_n}$ identifies the immediate contribution due to the decision x_n.

Reduction of the Combinatorial Complexity

For the design of multiple-flyby trajectories, the stages are naturally represented by the series of consecutive flybys, and the point of injection into a planet's SOI is the decision variable that determines the evolution of the spacecraft trajectory at each stage. As mentioned before, the presence of four constants of motion allows the state of the system to be described by only two independent variables. These two are chosen as the non-dimensional semi-major axis $\bar{a}$ and the orbital inclination i. The former is a convenient choice because, if the set of admissible resonances is limited, the semi-major axis also assumes only discrete values. On the other hand, for each value of $\bar{a}$, each point on the corresponding b-plane circle is associated to a different possible value of the inclination. This means that this latter varies in continuous intervals, implying the existence of an infinite number of possible states at each stage. Since it is clearly unfeasible to consider them all, a trivial way to overcome this issue would be to discretize the continuous interval, as suggested by Bellman.[13] However, besides the suboptimality introduced by a discretization, even a coarse grid easily would lead to combinatorial explosion issues when the problem involves several stages.

The approach adopted here is to identify a single optimal value of inclination for each b-plane circle, rather than considering the whole continuous range. Identifying with i_f the target inclination of the flybys sequence, the optimal value is simply taken as the closest inclination to i_f attainable on the circle considered. It can be shown that this strategy achieves the target state in as few flybys as possible, which is why it is considered optimal. Unfortunately, as there is no analytical formula linking a point on a circle to the corresponding post-encounter inclination, the optimal value must be sought numerically. Moreover, for realistic GAMs, one must take into account the minimum admissible flyby altitude above the planet.[16] 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. The direct consequence is that all the points on a circle that are below b_{min} are not viable and must be discarded. Therefore, the first step in the search for the optimal inclination is to identify the feasible domain D_α in which the angle α can vary on a circle. Referring to Figure 6, 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[17] (large b), it is appropriate to set a limit on the maximum impact parameter as well.

Let $f_\alpha(\alpha)$ identify the function that, for any $\alpha \in D_\alpha$, returns the value of the final inclination. $f_\alpha(\alpha)$ may be seen as a black-box function involving all the steps in Figure 1 plus a conversion step where the post-encounter velocity and flyby position are used to compute the final inclination. The optimal inclination for any circle is then computed by always minimizing the same function:

$$i' = \min_{\alpha \in D_\alpha} \left(f_\alpha(\alpha) - i_f \right)^2 \tag{33}$$

The minimization is carried out by means of Brent's algorithm for functions of one variable.[18]

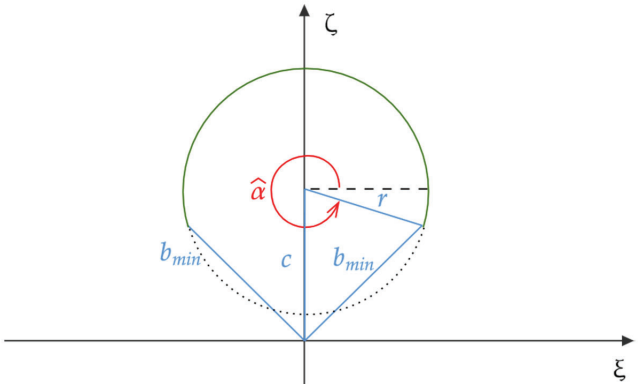

Figure 6. The green arc identifies the feasible domain of α due to b_{min}

Solution Procedure

Let $\{\bar{a}_0, e_0, i_0, \Omega_0, \omega_0\}$ be the orbital elements of a given initial orbit and let $\{\bar{a}_f, e_f, i_f, \Omega_f, \omega_f\}$ be those of a target orbit. Provided that the orbits feature the same Tisserand parameter and cross a planet's orbit in the same position in space, the initial and target states of the spacecraft are $s_0 = (\bar{a}_0, i_0)$ and $s_f = (\bar{a}_f, i_f)$, respectively.

The possible states through which the satellite can move are determined starting from 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 orbits by means of Equation (17). 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 (17) is then used to convert the k/h_i ratios and obtain a set $S_{\bar{a}}$ of admissible semi-major axes. 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.

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:

$$f_n(s_n, x_n) = \sum_{i=1}^{3} \left| \bar{U}_i - \bar{U}_{f,i} \right| \tag{34}$$

where $\bar{U}_i$ are the components of the velocity associated to the current state s_n, and $\bar{\boldsymbol{U}}_f$ 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:

$$f_n^*(s_n) = \min_{x_n \in S_{n-1}} f_n(s_n, x_n) \qquad n = 1, 2, \ldots \tag{35}$$

where S_n is defined as the set of all the possible states in which the system could be at stage 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 $S_{\bar{a}}$. For instance, the elements of S_1 are obtained by starting from state s_0 and computing for each semi-major axis in $S_{\bar{a}}$ the optimal value of inclination using Equation (33). Each pair of semi-major axis−inclination obtained is then saved as an element of S_1. Sometimes, when computing the optimal inclination, D_α may result in an empty set, meaning that the circle is all inside a forbidden region of the b-plane and no feasible state can be computed for that circle.

In practice, at each stage and for each state Equation (35) answers the question: "what is the best pre-encounter state to come from, if the system is currently in state s_n?".

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 ε. 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 ε, and the number of total flybys N is retrieved.

SIMULATIONS AND RESULTS

Comparison Between Classical and Extended Theory

The classical and the extended theory of b-plane circles are tested and compared on two mission design applications. Consider the trajectory design of a spacecraft that, after departing Earth, is required to perform a sequence of five resonant flybys at Venus with the following conditions:

$$\text{Resonance ratios}: \left\{\frac{4}{3}, \frac{7}{5}, \frac{3}{2}, \frac{8}{5}, \frac{7}{4}\right\} \qquad \text{Inclinations}: \{1.5^\circ, 3^\circ, 4.5^\circ, 6^\circ, 7.5^\circ\}$$

The two models are asked to find the sequence of b-plane points leading to these flybys. Each k/h ratio identifies a circle in the b-plane and then Equation (33) is used to find a specific point on each circle. For the elliptical model, the position of Earth at launch and that of Venus at the encounter are selected arbitrarily from their orbits. Then, assuming a realistic transfer time, the initial conditions of the first flyby are obtained by computing the Lambert arc related to the transfer orbit between the two planets. For the classical model, the same position of Earth is used, while Venus is assumed to be at the same heliocentric longitude as before, but in circular orbit.

The sequences of b-plane points computed through the two models are shown in Figure 7. 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. 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 χ 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

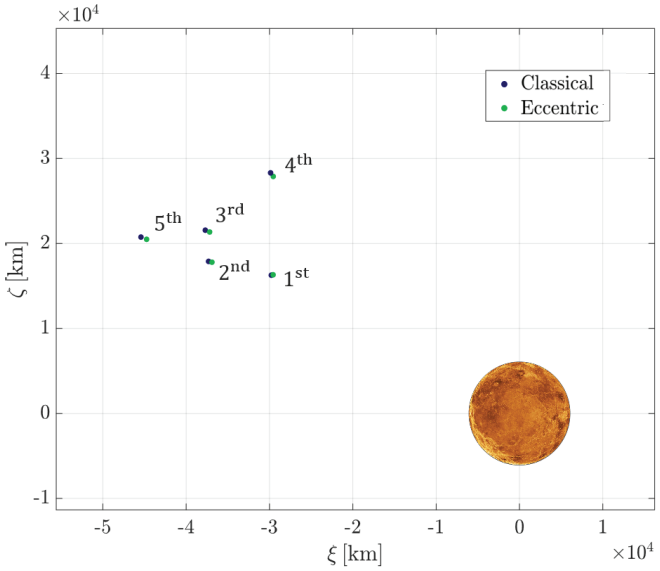

Figure 7. B-plane points for the flybys with Venus. The numbers refer to the flyby number

circular one. For this reason, it is interesting to test the models in a second scenario in which three resonant flybys are performed at Mars near its perihelion and with the following specifications:

$$\text{Resonance ratios}: \left\{\frac{4}{5}, \frac{6}{7}, \frac{9}{10}\right\} \qquad \text{Inclinations}: \{1.5^\circ, 3^\circ, 4.5^\circ\}$$

This time the points predicted by the two models are considerably different, as shown in Figure 8. To gain a better insight into the differences, consider the blue points in the figure, obtained using the classical model. If Mars was actually in circular orbit, those points would accurately lead to the prescribed resonant orbits. Instead, since Mars has a certain orbital 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 predicted ones. Table 2 shows how significant this deviation is. The first and third columns report the errors in the semi-major axes and inclinations of the orbits predicted by the classical model. The former errors inevitably translate into errors in the orbital periods and, therefore, in the k/h ratios predicted. Consequently, 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.

Table 2. Deviations between the orbits obtained with the classical model and the prescribed ones

	$\Delta a_{\%}$ [%]	$\Delta k/h$ [%]	Δi [deg]
1st flyby	1.923	2.87	0.37
2nd flyby	3.134	3.73	0.71
3rd flyby	4.062	5.77	0.99

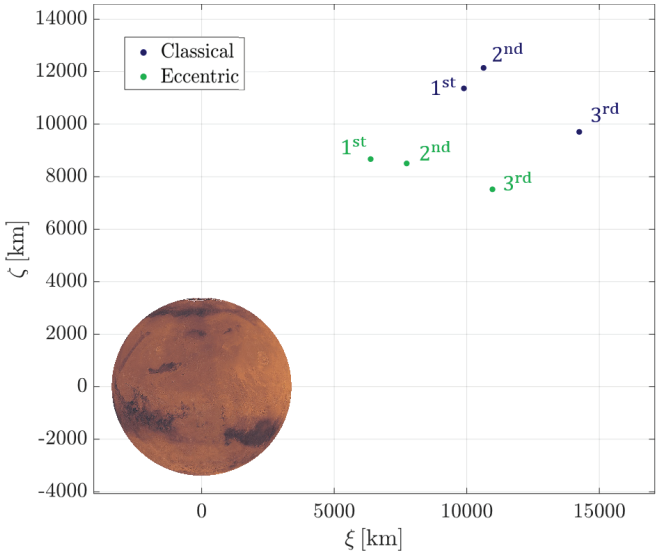

Figure 8. B-plane points for the flybys with Mars

Solar Orbiter Resonant Phase with Venus

The dynamic programming 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 reduce the distance at perihelion. The baseline trajectory is the mission profile with launch in January 2017 proposed in.[1] The algorithm is only given the initial orbit before the resonant sequence and the last resonant orbit as inputs, as reported in Table 3. The total number of necessary resonant flybys is left as a free variable. In this regard, the target orbit is considered achieved when the error as defined in Equation (34) drops below a value of 10^{-4} km/s. The admissible resonance ratios are generated by considering a limit of 5 on both k and h. Given the harsh atmosphere of Venus, a minimum flyby altitude of 300 km is assumed.

Table 3. Initial and final orbit

Orbit	Aphelion [AU]	Perihelion [AU]	Ecl. Incl. [deg]
Initial	0.998	0.311	1.72
Final	0.738	0.320	27.25

The overall algorithm has been implemented in MATLAB® on a machine with an Intel® Core™ i7-6700 CPU @2.60 GHz. 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. In each simulation, the algorithm converged to the final state in four stages, the same number of flybys of the real mission. As can be seen from Table 4, the sequence of resonances predicted with the extended model is identical to that of the baseline trajectory, giving proof of the proposed design strategy. The inclinations also are comparable, with the differences being mainly due to having disregarded the perturbations. Figure 9 provides a visual comparison of the baseline trajectory and that obtained by using dynamic programming and the extended theory. The plot of the baseline trajectory also includes the non-resonant phases of the mission.

Each simulation required a surprisingly low computational time of about 0.06 s. When evaluating the computational efficiency of a dynamic programming approach, the standard benchmark used for comparison is the brute-force search, where all the feasible solutions are systematically evaluated in order to find the optimal one. The two approaches have been tested on the same problem as before but setting a limit of 15 for both k and h, so as to simulate a scenario involving a relatively high number of possible trajectories. Both converged to the same solution but, as clear from Table 5, dynamic programming converged with considerable less effort than the brute-force approach. Figure 10 illustrates all the solutions evaluated by the two algorithms. Each gray or red point represents the final state associated to a trajectory computed with either of the two algorithms.

Table 4. Comparison between the baseline trajectory and the computed ones

	Resonance ratios k/h			**Inclinations** [deg]		
	SOLO baseline	Dyn. prog. (classical)	Dyn. prog. (extended)	SOLO baseline	Dyn. prog. (classical)	Dyn. prog. (extended)
E2-V2	-	-	-	1.72	1.72	1.72
V2-V3	3/4	4/5	3/4	9.93	9.608	9.152
V3-V4	3/4	3/4	3/4	18.11	18.546	18.344
V4-V5	2/3	2/3	2/3	23.83	24.647	24.586
V5-V6	3/5	3/5	3/5	27.25	27.25	27.25

Table 5. Brute-force vs dynamic programming performances

	Brute-force	Dyn. prog.
Time	3.55 min	1.12 s
Number of evaluations	∼168 k	1220

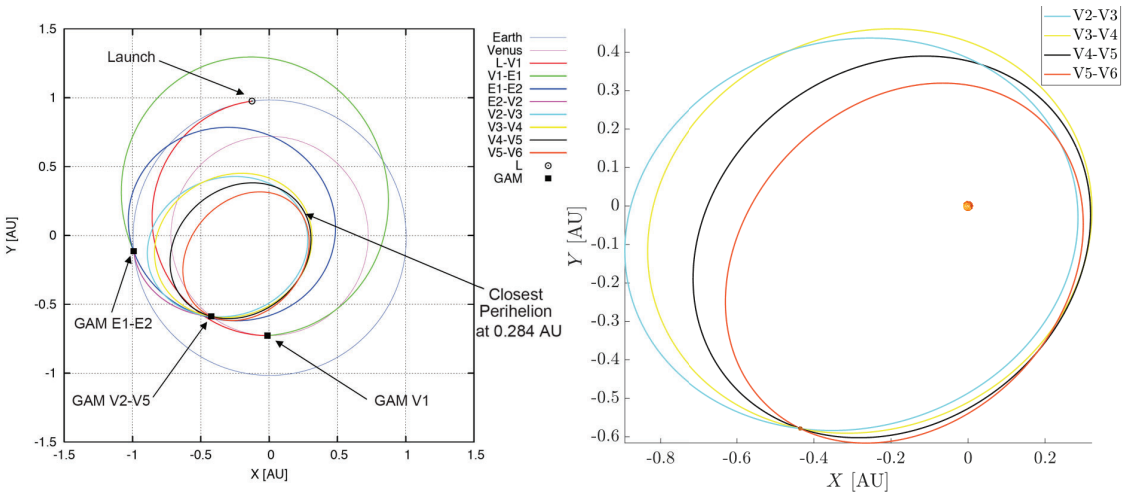

Figure 9. Baseline trajectory (*left*) and dynamic programming solution with extended theory (*right*)

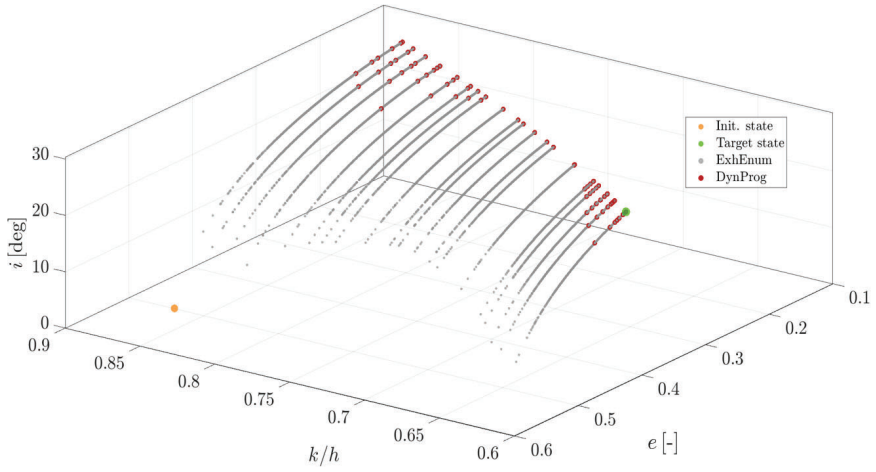

Figure 10. Brute-force (BF) vs dynamic programming (DP)

CONCLUSION

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 two models were tested 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.

In the second part of the work, a dynamic programming approach was proposed to address the unperturbed design of multiple resonant flyby trajectories, where the number of flybys is not known a priori. The intrinsic continuous-combinatorial nature of the problem was handled by identifying a principle that allows reducing the problem to a purely combinatorial one. The developed algorithm was tested against the design of Solar Orbiter's resonant phase with Venus, and the resulting solution featured the same flybys number and the same resonance ratios as the baseline mission trajectory. The computational efficiency of the dynamic programming approach was assessed by comparing its performances with those of a traditional brute-force method. In conclusion, the proposed algorithm is intended to provide 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.

ACKNOWLEDGEMENT

The research leading to these results received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 679086 - COMPASS).

REFERENCES

[1] E. S. Agency, "Solar Orbiter Definition Study Report (Red Book)," tech. rep., 2001.

[2] E. S. Agency, "JUpiter ICy moons Explorer Exploring the emergence of habitable worlds around gas giants (Red Book)," tech. rep., 2014.

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

[4] G. B. Valsecchi, A. Milani, G. F. Gronchi, and S. R. Chesley, "Resonant Returns to Close Approaches: Analytical Theory," *Astronomy & Astrophysics*, Vol. 408, 2003.

[5] G. B. Valsecchi, A. Milani, G. F. Gronchi, and S. R. Chesley, "The Distribution of Energy Perturbations at Planetary Close Encounters," *Celestial Mechanics and Dynamical Astronomy*, Vol. 78, 2000.

[6] Y. Langevin, "Chemical and solar electric propulsion options for a Cornerstone Mission to Mercury," *Acta Astronautica*, Vol. 47, No. 2-9, 2000, pp. 443–452.

[7] A. Masat, "B-plane Orbital Resonance Analysis and Applications," Master's thesis, Politecnico di Milano, 2019.

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

[9] "Planetary Close Encounters: Geometry of Approach and Post-Encounter Orbital Parameters," *Celestial Mechanics and Dynamical Astronomy*, Vol. 49, 1990.

[10] M. Sniedovich, *Dynamic programming: Foundations and Principles*. Boca Raton: Chapman & Hall/CRC Press, 2nd ed., 2010.

[11] M. Carter, C. C. Price, and G. Rabadi, *Operations Research: a Practical Introduction*. Chapman and Hall/CRC, 2nd ed., 2018.

[12] D. W. Stroock, *An Introduction to Markov Processes*. Springer, Berlin, Heidelberg, 2nd ed., 2014.

[13] R. Bellman, *Dynamic Programming*. Dover Publications, 2003.

[14] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, Massachusetts, 3rd ed., 2005.

[15] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, Inc., 2nd ed., 2007.

[16] Y. F. Golubev, A. V. Grushevskii, V. V. Koryanov, A. G. Tuchin, , and D. A. Tuchin, "Formation of High Inclined Orbits to the Ecliptic by Multiple Gravity Assist Maneuvers," *Journal of Computer and System Sciences International*, Vol. 56, No. 2, 2017, pp. 275–299.

[17] R. Greenberg, A. Carusi, and G. B. Valsecchi, "Outcome of Planetary Close Encounters: a Systematic Comparison of Methodologies," *Icarus*, Vol. 75, 1988.

[18] R.P.Brent, *Algorithms for Minimization Without Derivatives*. Prentice-Hall, 1973.