N92-24778

NASA J59-12

427

8/35 633562

RAPID DESIGN OF GRAVITY ASSIST TRAJECTORIES*

J. Carrico,[1] H.L. Hooper,[1] L. Roszman,[1] and C. Gramling[2]

[1] Computer Sciences Corporation, Lanham-Seabrook, MD.; [2] NASA/Goddard Space Flight Center

ABSTRACT

Several International Solar-Terrestrial Physics (ISTP) missions require the design of complex gravity-assisted trajectories in order to investigate the interaction of the solar wind with the Earth's magnetic field. These trajectories present a formidable trajectory design and optimization problem. This paper discusses the philosophy and methodology that enable an analyst to design and analyze such trajectories. The authors describe what is called "floating end-point" targeting, which allows the inherently nonlinear multiple body problem to be solved with simple linear techniques. This paper demonstrates how floating end-point targeting combines analytic approximations with a Newton method targeter to achieve trajectory design goals quickly, even for the very sensitive double lunar swingby trajectories used by the ISTP missions. A Multiconic orbit integration scheme allows fast and accurate orbit propagation. This paper also describes a prototype software tool, Swingby, which has been built for trajectory design and launch window analysis.

Keywords: gravity assist, ISTP missions, double-lunar swingby, Multiconic, floating end-point

1. GRAVITY-ASSIST TRAJECTORY DESIGN

Designing spacecraft trajectories that are influenced by several gravitational bodies presents many problems for the mission analyst, such as the difficulties of targeting a moving body, the lack of a general solution to the N-body problem, and the massive amount of numerical data that must be interpreted. Additionally, for missions within the Earth-Moon system, the non-sphericity of the Earth's geopotential field, the Sun's gravitational force, the solar radiation pressure, and the gravitational forces of the larger planets all must be taken into account to achieve the accuracy necessary for mission planning.

Several missions have used trajectories that were heavily influenced by both the Earth and the Moon. In particular, the NASA mission International Cometary Explorer (ICE) (Ref. 1) and ISAS's Hiten (Ref. 2) used paired lunar encounters in what is called a double lunar swingby (DLS) (Ref. 3). In a DLS, the first encounter accelerates the spacecraft from an orbit with a radius at apogee of about 60 Earth radii (R_e) into an orbit with a radius at apogee ranging from 140 to 240 R_e.

Two spacecraft from the ISTP Project will use DLS trajectories: GEOTAIL (to be launched in July of 1992) and WIND (to be launched in December, 1992). This paper will concentrate on designing DLS trajectories, although the techniques will be clearly applicable to other gravity-assisted missions.

Figure 1 shows a DLS trajectory in the Geocentric Inertial (GCI) coordinate frame with its major features noted, including (A) the transfer orbit, (B) the first lunar swingby (acceleration), (C) the outer loop, (D) the second lunar swingby (deceleration), and (E), an inner loop. This inner loop usually has a period such that after one or more revolutions the spacecraft will again encounter the Moon and start another DLS sequence.
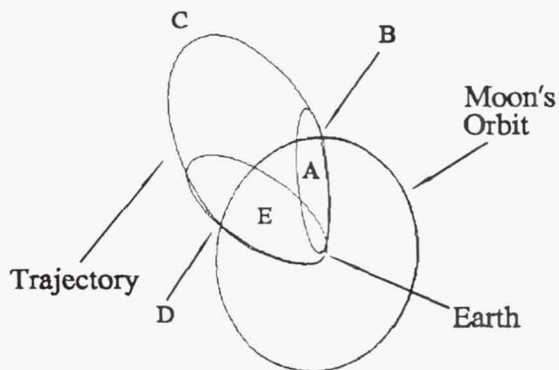

Figure 1. A double lunar swingby.

A major problem in designing gravity-assisted trajectories comes from the fact that there are no analytic solutions to the N-body problem. In the case of DLS trajectories, the gravity of the Earth, Moon, and Sun must be modeled. In addition, experience at NASA's Goddard Space Flight Center (GSFC) Flight Dynamics Facility (FDF) has shown that the models must also include the Earth's oblateness, solar radiation pressure, and sometimes even the effect of the larger planets (Ref. 4). It is therefore necessary to use fully integrated trajectories during the design and pre-mission analysis.

Another problem when designing gravity-assist maneuvers is that the analyst is trying to hit a moving target (the Moon). Because a gravity assist is so sensitive to the arrival conditions, especially for DLSs, even slight deviations from the nominal can have drastic effects on the resultant trajectory. The targeting scheme must be able to handle the sensitivities.

2. IMMEDIATE FEEDBACK AND VISUALIZATION

Our solutions to the problems stated above are to use a quick, yet accurate, orbit integrator for generating trajectories, and to use a simple targeting scheme with immediate graphic information for the mission analyst. The graphic information allows the analyst to observe convergence or divergence and to adjust parameters to force convergence. The details of each of these solutions follow.

TRAJECTORY INTEGRATION

There are many methods of integrating a trajectory while modeling several gravitational bodies with perturbations, but they are usually characterized by intensive calculations that take a great deal of time unless a very powerful computer is used. The mainframe computers at NASA's GSFC are indeed capable of computing such trajectories, but the ever-increasing operational data traffic from new spacecraft has pushed much mission analysis to microcomputers. Therefore, a quick method to integrate orbits accurately is needed to make up for the decrease in computing speed. Fortunately, the Apollo and Galileo missions have already shown that the Multiconic method (also known as the pseudostate method) is a fast and accurate method to compute spacecraft trajectories with multiple body perturbations (Refs. 5, 6).

The Multiconic method for orbit integration bridges the gap between the fast but inaccurate patched-conic methods and the slow but accurate numerical integration techniques. Basically, the Multiconic method is to overlap conics for each gravitational body while accounting for the relative motion of these bodies, and to add averaged accelerations for any other perturbations.

To generate trajectories accurate enough for mission analysis, we have found it necessary to use a variable step size with the Multiconic integrator. Some previous studies of the errors of the Multiconic method (Refs. 7, 8) have shown that the optimum step size, h, that can be taken for a given error tolerance is a function of the distances from the gravitational bodies and their gravitational constants:

$$h = \left(\frac{6 \epsilon R_e^3 R_m^2}{\mu_e \mu_m} \right)^{\frac{1}{3}} \qquad [1]$$

where:

> h = step size
> ϵ = error tolerance (km/s)
> R_e = Distance from the Earth
> R_m = Distance from the Moon
> μ_e = Gravity constant of the Earth
> μ_m = Gravity constant of the Moon

Our implementation of Multiconic currently models the Earth's and Moon's effects as conics and uses averaged solar and oblateness effects. We have compared our Multiconic integrator trajectories calculated with a variable-step twelfth-order Stormer-Cowell orbit integrator (Ref. 9). After more than 68.5 days, the position differences were less than 40 km, and the velocity differences were less than 4 mm/sec. (The Multiconic trajectory used an error tolerance of 1×10^{-9} km/sec.) At the outer loop apogee, the position differences were less than 5 km and the velocity differences less than 3 mm/sec. It took our present implementation of Multiconic about 26 seconds to generate such a trajectory, running on an AST 386/33 Mhz PC with a math coprocessor.

GRAPHICS

Displaying trajectories graphically is a great help to the mission analyst. Although the mechanics of DLS trajectories are not trivial, a graphic display of the trajectory shapes makes clear some rather subtle information that is difficult to

ascertain from numbers alone. For instance, in Figure 1 it is not hard to spot the perigees, apogees, and lines of apsides. The encounters with the Moon also can be clearly seen, along with the resulting effects. Furthermore, the repetitive integration of DLS trajectories during targeting creates massive amounts of data, which are difficult to sift through and interpret. It is much easier to determine the differences between two trajectories when their shapes are overlaid on each other than to read through two spacecraft ephemeris listings. A static display of the trajectory, however, still hides some important information from the analyst: the time history.

It was mentioned earlier that targeting a gravity assist is essentially an attempt to hit a moving target. More accurately, it is a process of properly missing the moving target— hitting the body would severely shorten most missions! To "properly miss" a target, the spacecraft must pass on the proper side of the assisting body, at the right distance. In the case of a DLS, the analyst often poses the question, "Which arrived at the crossing point first, the spacecraft or the Moon?" We have found that when the trajectories are plotted as they are calculated, a sort of animation takes place, and the analyst can clearly see the relative positions of the spacecraft and the Moon as they move. Our implementation of Multiconic draws the spacecraft's position onto the screen at every intermediate step, along with the positions of the Earth and the Moon. To assist the analyst, a trail of the past trajectory is left on the computer screen. Depending on the phase of the mission, the analyst may want to view the trajectory in any of a variety of coordinate frames. In addition to GCI, analysts in GSFC's FDF have found the Sun-Earth Rotating, Moon-Earth Rotating, Selenocentric, Ground Track, and Celestial Sphere coordinates helpful.

The purpose of graphics, as mentioned before, is to supply immediate feedback to changes made by the analyst. In order for feedback to indeed be "immediate," the analyst must be able to quickly and simply alter the initial parameters. We have found that the simplest way to do this is with a "pull-down" and "pop-up" menu system. This allows the analyst to input data easily, check results, and then quickly modify the data and try again.

TARGETING —
DETERMINING A FIRST GUESS

In order to design a DLS, the analyst first uses a trial-and-error method, adjusting the initial conditions until the Moon is encountered after some suitable revolutions in the transfer orbit. The initial conditions normally are an initial date and time, position and velocity vectors or parking orbit coast time and upper-stage burn duration, and some set of orbit maneuvers. This portion of mission planning does not need to be automated. Allowing the analyst to quickly adjust the initial conditions and supplying immediate feedback via appropriate graphics enables the analyst to quickly determine the first guess at the initial conditions. This not only gets the job done, but starts to give the analyst an intuitive feel for controlling the trajectory.

REFINING THE FIRST GUESS

Once the analyst is satisfied with the first guess, he or she must refine the initial conditions to achieve an acceptable lunar swingby to meet the mission goals. In the case of a DLS, the

mission goals usually relate to the duration spent in the outer loop and the orientation of the outer loop orbit plane. Because the Moon takes about a month to go around the Earth, and since a DLS trajectory must pass the Moon a second time, the duration in the outer loop is nearly an integer number of months, ranging between 1 and 7. The first lunar swingby of a DLS must be a gravity acceleration, which means that the spacecraft must pass on the trailing side of the Moon. A proper gravity assist will increase the semimajor axis, altering the period so that when the spacecraft again crosses the Moon's orbit, it will also encounter the Moon. The fact that a DLS requires the spacecraft to go through a second lunar swingby constrains the spacecraft orbit to lie close to the Moon's orbit. Therefore, in order to target a proper first lunar swingby, the analyst must somehow describe the conditions at the first swingby that will give the outer loop the proper semimajor axis and inclination with respect to the Moon's orbit plane.

The B-Plane coordinate system described by Kizner (Ref. 10) has proved itself to be very efficient for describing the arrival conditions at an assisting body. Tucker states (Ref. 11) that the B-Plane parameters ($\vec{B}\cdot\hat{T}$ and $\vec{B}\cdot\hat{R}$) responded more linearly to variations in the initial conditions for the Apollo mission than any other they knew about. The B-Plane is shown in Figure 2. The first axis, $\hat{S}$, is parallel to the
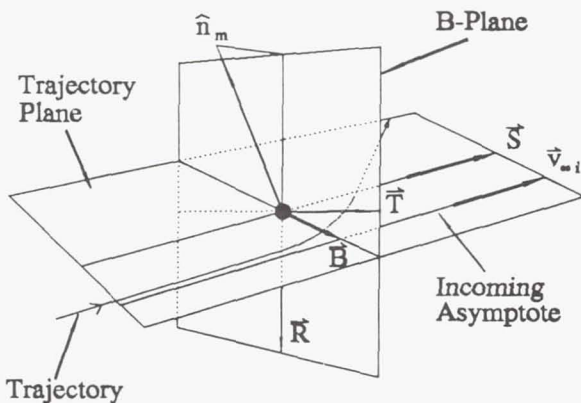

Figure 2. The B-Plane.

incoming asymptote. Crossing $\hat{S}$ with the Moon's orbit normal $\hat{n}_m$ yields the second axis, $\hat{T}$. The last member of the orthogonal triad, $\hat{R}$, is $\hat{S}\times\hat{T}$. The B-Vector ($\vec{B}$) is formed by crossing $\hat{S}$ with the spacecraft's orbit normal:

$$\hat{B} = \hat{S}\times\hat{n}_{sc} \qquad [2]$$

The magnitude of $\vec{B}$ is the semiminor axis of the spacecraft's hyperbolic trajectory. It is convenient to describe $\vec{B}$ in terms of its components along the $\hat{T}$ and $\hat{R}$ axes, $\vec{B}\cdot\hat{T}$ and $\vec{B}\cdot\hat{R}$. During targeting, these components behave much better than $|\vec{B}|$ and an angle from $\hat{T}$, because the changes with respect to the initial conditions (the partial derivatives) for each are of the same order of magnitude, whereas the change in the angle (on the order of 0.01) is usually much less than the change in the $|\vec{B}|$ (on the order of 10 to 1000).

The analyst can refine the first guess by using a Newton method to vary the initial conditions to achieve a specified

$\vec{B}\cdot\hat{T}$ and a $\vec{B}\cdot\hat{R}$. The Newton method comes from truncating the Taylor series of a function. In general, if the independent variables are $(Q_1,Q_2,...Q_n)$, and the goals $(f_1,f_2,...f_n)$ are a function of the variables, then a linear approximation yields the equation:

$$\vec{f}(\vec{Q}) = \vec{f}(\vec{Q}_0) + S\cdot(\vec{Q}-\vec{Q}_0) \qquad [3]$$

where the sensitivity matrix S is:

$$S = \begin{bmatrix} \dfrac{\partial f_1}{\partial Q_1} & \dfrac{\partial f_1}{\partial Q_2} & \cdots & \dfrac{\partial f_1}{\partial Q_n} \\[2mm] \dfrac{\partial f_2}{\partial Q_1} & \dfrac{\partial f_2}{\partial Q_2} & \cdots & \dfrac{\partial f_2}{\partial Q_n} \\[2mm] \vdots & & & \\[2mm] \dfrac{\partial f_n}{\partial Q_1} & \dfrac{\partial f_n}{\partial Q_2} & \cdots & \dfrac{\partial f_n}{\partial Q_n} \end{bmatrix} \qquad [4]$$

Rearranging Equation [3], we get:

$$\Delta\vec{Q} = S^{-1}\cdot\Delta\vec{f} \qquad [5]$$

where:
$$\Delta\vec{Q} = \vec{Q} - \vec{Q}_0$$
$$\Delta\vec{f} = \vec{f}(\vec{Q}) - \vec{f}(\vec{Q}_0)$$

For example, to achieve a swingby with the target values $\vec{B}\cdot\hat{T}_t$ and $\vec{B}\cdot\hat{R}_t$, select two components of a pre-swingby delta-V (the tangential and normal components, V_t and V_n) as the part of the initial conditions to vary. Generate a nominal trajectory with, say, zero delta-V ($\vec{Q}_0 = (V_{t0},V_{n0}) = (0,0)$), and then record the differences from the target values, $\Delta\vec{f}=(\vec{B}\cdot\hat{T}_t-\vec{B}\cdot\hat{T}_0, \vec{B}\cdot\hat{R}_t-\vec{B}\cdot\hat{R}_0)$. Then the sensitivity matrix of partials is formed by varying V_t and V_n separately by small perturbations and dividing the resulting changes in $\vec{B}\cdot\hat{T}$ and $\vec{B}\cdot\hat{R}$ by the perturbations. Equation [5] can be solved numerically by using a Gauss-Jordan technique; and then the next guess as to the delta-V is calculated by multiplying the inverted matrix by $\Delta\vec{f}$ and adding the results to the initial delta-V. This is repeated until the changes in $\vec{B}\cdot\hat{T}$ and $\vec{B}\cdot\hat{R}$ are within a specified tolerance. Our experience has shown that with the proper choice of tolerances and perturbations, the solution converges in about 3 to 5 iterations.

However, the trick is not just to achieve a certain $\vec{B}\cdot\hat{T}$ and $\vec{B}\cdot\hat{R}$, but to choose values for $\vec{B}\cdot\hat{T}$ and $\vec{B}\cdot\hat{R}$ that meet the mission goals (which, as mentioned above, relate to the duration spent in the outer loop and the orientation of the orbit plane). It is possible to derive analytic equations that calculate B-Plane parameters based on the post-swingby orbit; however, these equations are a function of the inbound trajectory, which changes during targeting, causing the B-Plane parameters to change. This then becomes a matter of trying to target on moving numbers. Let us now look at an illustration of the problem, and show how it can be solved using a "floating end-point" scheme.

An analyst wants to design a DLS with a 1-month outer loop in the Moon's orbit plane, as shown in Figure 3. The

semimajor axis of the outer loop, a_o, is about 461,000 km. The goal is then to achieve a lunar swingby that will increase the trajectory's energy to this amount and put the trajectory into the Moon's orbit plane. First, we calculate the magnitude of $\vec{B}$ that will give the desired energy increase, and then we will calculate components along $\hat{T}$ and $\hat{R}$.
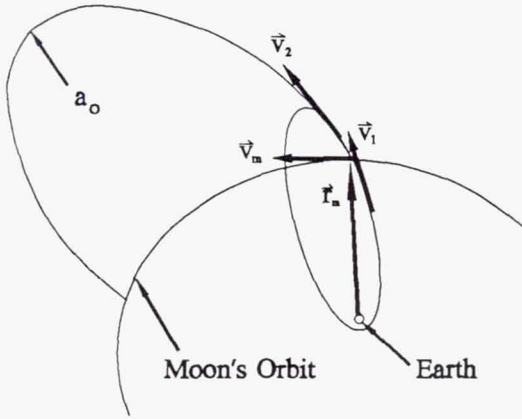

Figure 3. A 1-month outer loop double lunar swingby.

$|\vec{B}|$ is a function of the bend angle, δ. We can calculate δ from the hyperbolic asymptotes, which in turn can be calculated from the velocities. First, calculate the desired velocity after the swingby, v_2, from the energy equation:

$$v_2 = \sqrt{\mu_e\left(\frac{2}{r_m} - \frac{1}{a_o}\right)} \qquad [6]$$

where:

μ_e = Earth's gravity constant
r_m = Moon's orbit radius
a_o = Semimajor axis of the outer loop
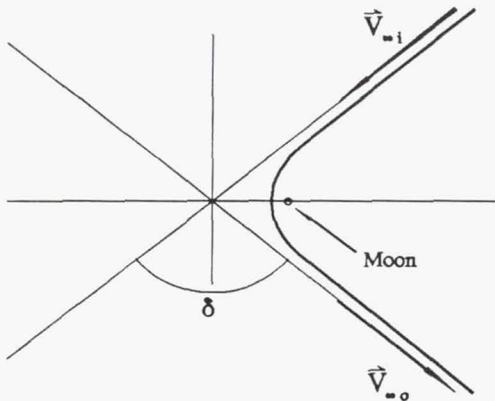v_2 = Spacecraft velocity after swingby

Figure 4. Selenocentric hyperbolic trajectory.

The geometry of the hyperbolic swingby allows us to calculate the asymptotes. Figure 4 shows the lunar swingby in the selenocentric frame. The incoming and outgoing asymptotic

velocities are defined from simple vector addition:

$$\vec{v}_{\infty i} = \vec{v}_1 - \vec{v}_m$$
$$\vec{v}_{\infty o} = \vec{v}_2 - \vec{v}_m \qquad [7]$$

where:

$\vec{v}_m$ = Moon's velocity vector
$\vec{v}_{\infty i}$ = Incoming asymptotic velocity vector
$\vec{v}_{\infty o}$ = Outgoing asymptotic velocity vector

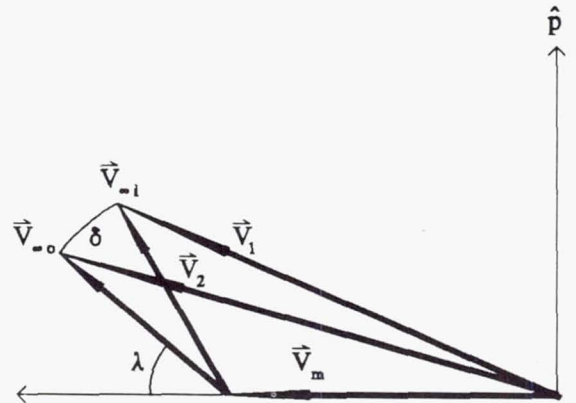Figure 5 shows these vectors, along with some related angles:

Figure 5. Vector diagram of gravity assist.

Assuming the outgoing asymptote to lie in the Moon's orbit plane, the angle between the Moon's velocity vector and the outgoing asymptote, λ, can be calculated from the law of cosines as a function of our desired v_2, and is:

$$\lambda = \pi - \cos^{-1}\left(\frac{v_m^2 + v_{\infty o}^2 - v_2^2}{2 v_m v_{\infty o}}\right) \qquad [8]$$

However, $v_{\infty o}$, which has the same magnitude as $v_{\infty i}$, cannot be evaluated until the incoming trajectory is defined. This is solved by running a first-guess trajectory to the Moon and using the resulting position and velocity vectors at periselenium to calculate $v_{\infty i}$. After converting the state vectors into a selenocentric semimajor axis, a_m, we have from the energy equation of a Moon-centered hyperbola:

$$|\vec{v}_{\infty o}| = |\vec{v}_{\infty i}| = v_\infty = \sqrt{-\frac{\mu_m}{a_m}} \qquad [9]$$

Equation [9] can then be used to calculate the angle λ by substituting into Equation [8]. Now we can describe the desired outgoing asymptote in terms of its vector components:

$$\hat{v}_{\infty o} = \sin(\lambda)\hat{p}_m + \cos(\lambda)\hat{v}_m \qquad [10]$$

where $\hat{p}$ is orthogonal to the velocity vector in the direction of the Moon's position vector, $\hat{r}$:

$$\hat{p}_m = \hat{v}_m \times \hat{n}_m = \hat{v}_m \times (\hat{r}_m \times \hat{v}_m) \qquad [11]$$

The bend angle, δ, is then:

$$\delta = \cos^{-1}(\hat{v}_{\infty o} \cdot \hat{v}_{\infty i})$$ [12]

(Note here that again we need an incoming trajectory for these calculations.) Now we must calculate B magnitude as a
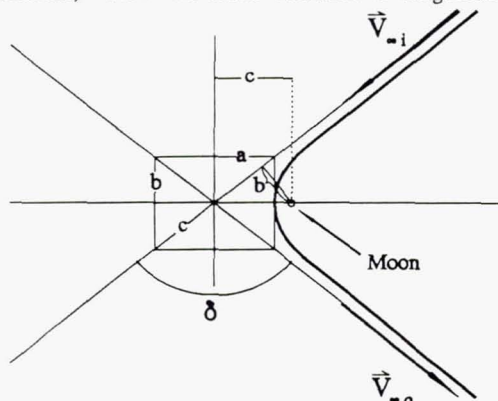

Figure 6. Geometry of the hyperbola.

function of the bend angle, δ. Figure 6 shows the geometry of the hyperbola. From Figure 6 it can be shown that:

$$\sin\left(\frac{\delta}{2}\right) = \frac{a}{c}$$ [13]

and since $c^2 = a^2 + b^2$ we can rearrange Equation [13] and get:

$$b = a \cdot \sqrt{\frac{1}{\sin^2\left(\frac{\delta}{2}\right)} - 1} = a \cdot \cot\left(\frac{\delta}{2}\right)$$ [14]

Solving Equation [9] for a_m as a function of v_∞ and substituting into Equation [14], we find the magnitude of the B vector as:

$$|\vec{B}| = \frac{\mu_m}{v_\infty^2} \cdot \cot\left(\frac{\delta}{2}\right)$$ [15]

Now that we have the vector magnitude, we need to calculate the components along the $\hat{T}$ and $\hat{R}$ axes. To do this, we must realize that the lunar encounter is a three-dimensional problem. The hyperbola's orbit plane can have any orientation in space. As mentioned earlier, we define $\hat{s}$ to be a vector centered at the Moon and parallel to the incoming asymptotes. We also let $\hat{o}$ be the generalized outbound asymptote. Figure 7 shows
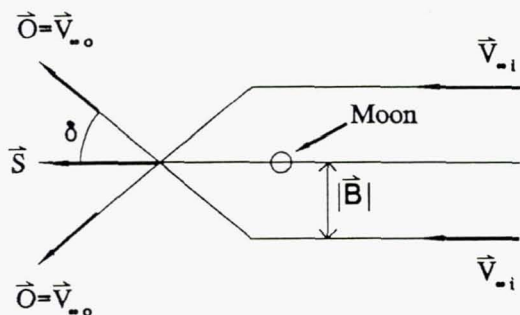

Figure 7. Hyperbolic asymptotes.

that for incoming asymptotes with a constant B magnitude, there is a family of outbound asymptotes that are at a constant angle, δ, from $\hat{s}$. One can then imagine in three dimensions that for a *cylinder* of incoming asymptotes, there is a *cone* of outbound asymptotes. Figure 8 shows this on a Moon-centered unit sphere. The cone becomes a circle centered about $\hat{s}$.
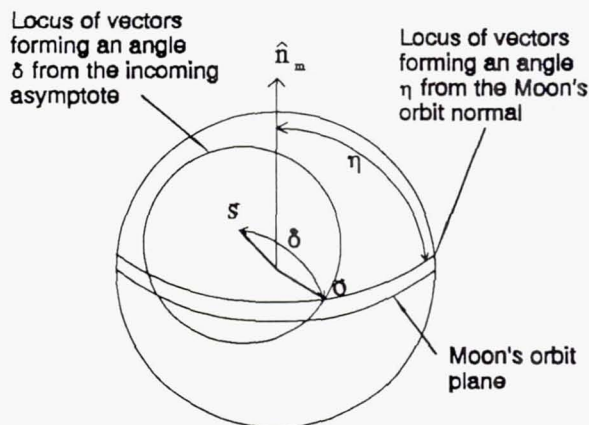

Figure 8. The Moon-centered unit sphere.

In general, $\hat{o}$ will not lie in the Moon's orbit plane, so it is shown slightly above the plane. $\hat{o}$ makes an angle η with the Moon's orbit normal, $\hat{n}_m$. (Ultimately, we will set η to 90°, forcing $\hat{o}$ to lie in the Moon's orbit plane.) The locus of all possible outgoing asymptotes that are at an angle η from $\hat{n}_m$ forms a cone, which is also shown on Figure 8. The problem then is to find the intersection of two cones, one centered about $\hat{s}$, and the other about $\hat{n}_m$. The solution to this is found in Reference 12, and is attributed to Grubin (Ref. 13). The information that we have is:

$$\hat{o} \cdot \hat{n}_m = \cos(\eta)$$
$$\hat{o} \cdot \hat{s} = \cos(\delta)$$ [16]
$$\hat{o} \cdot \hat{o} = 1$$

If we let:

$$x = \frac{\cos\delta - \hat{s} \cdot \hat{n}_m \cos\eta}{1 - (\hat{s} \cdot \hat{n}_m)^2}$$ [17]

$$y = \frac{\cos\eta - \hat{s} \cdot \hat{n}_m \cos\delta}{1 - (\hat{s} \cdot \hat{n}_m)^2}$$ [18]

$$z = \pm\sqrt{\frac{1 - x\cos\delta - y\cos\eta}{1 - (\hat{s} \cdot \hat{n}_m)^2}}$$ [19]

$$\vec{C} = \hat{s} \times \hat{n}_m$$ [20]

then our desired outgoing asymptote is defined as:

$$\hat{o} = x\hat{s} + y\hat{n}_m + z\vec{C} \qquad [21]$$

The sign of z is chosen so that the outgoing trajectory will have the proper orbit normal; an outbound gravity acceleration should be positive. (Note that once again we need an incoming trajectory for these calculations, this time for $\hat{s}$.) The spacecraft's orbit normal is formed from the incoming and outgoing asymptotes:

$$\hat{n}_{sc} = \frac{\hat{s} \times \hat{o}}{|\hat{s} \times \hat{o}|} \qquad [22]$$

And then, as previously defined, the B vector is:

$$\hat{B} = \hat{s} \times \hat{n}_{sc} \qquad [23]$$

Then, using the previous definitions of the T and R vectors, we have:

$$\vec{B} \cdot \hat{T} = |\vec{B}| \hat{B} \cdot \hat{T}$$
$$\vec{B} \cdot \hat{R} = |\vec{B}| \hat{B} \cdot \hat{R} \qquad [24]$$

Targeting on these components will give a trajectory that has the desired post-swingby semimajor axis and lies in the Moon's orbit plane.

FLOATING END-POINT TARGETING

As noted earlier, the major difficulty with the equations above is that to calculate the target $\vec{B}$ we need a trajectory to the Moon. This means we need to make a first guess at the initial conditions, run a trajectory to the Moon, calculate the target $\vec{B}$, and then vary the initial conditions until we achieve the target. Many of the equations presented above, however, were dependent on the incoming trajectory, so that when the trajectory is changed, $\vec{B}$ will change. It would become quite tedious to reenter a new $\vec{B}$ at every iteration, and there is no need to, since we can let the computer calculate it every time it runs a nominal trajectory. As Tucker wrote (Ref. 11), "... the program has a much better estimate of the $[a_m]$ than the user can get without considerable previous work having been done on similar trajectories." The Newton method modified to include the floating end-point is, then, to (1) run a best-guess trajectory to periselenium; (2) calculate the target $\vec{B}$ as a function of the desired outer loop semimajor axis; (3) create the sensitivity matrix by running perturbations and update the best guess; and (4) Repeat (1) to (3) until the target is within tolerance.

Our implementation of these equations has shown that the solution converges very quickly: between three and six iterations for the first swingby of a DLS.

3. AN IMPLEMENTATION: SWINGBY

We have implemented the techniques described above in a program called Swingby. Swingby runs on a IBM 386 PC. It
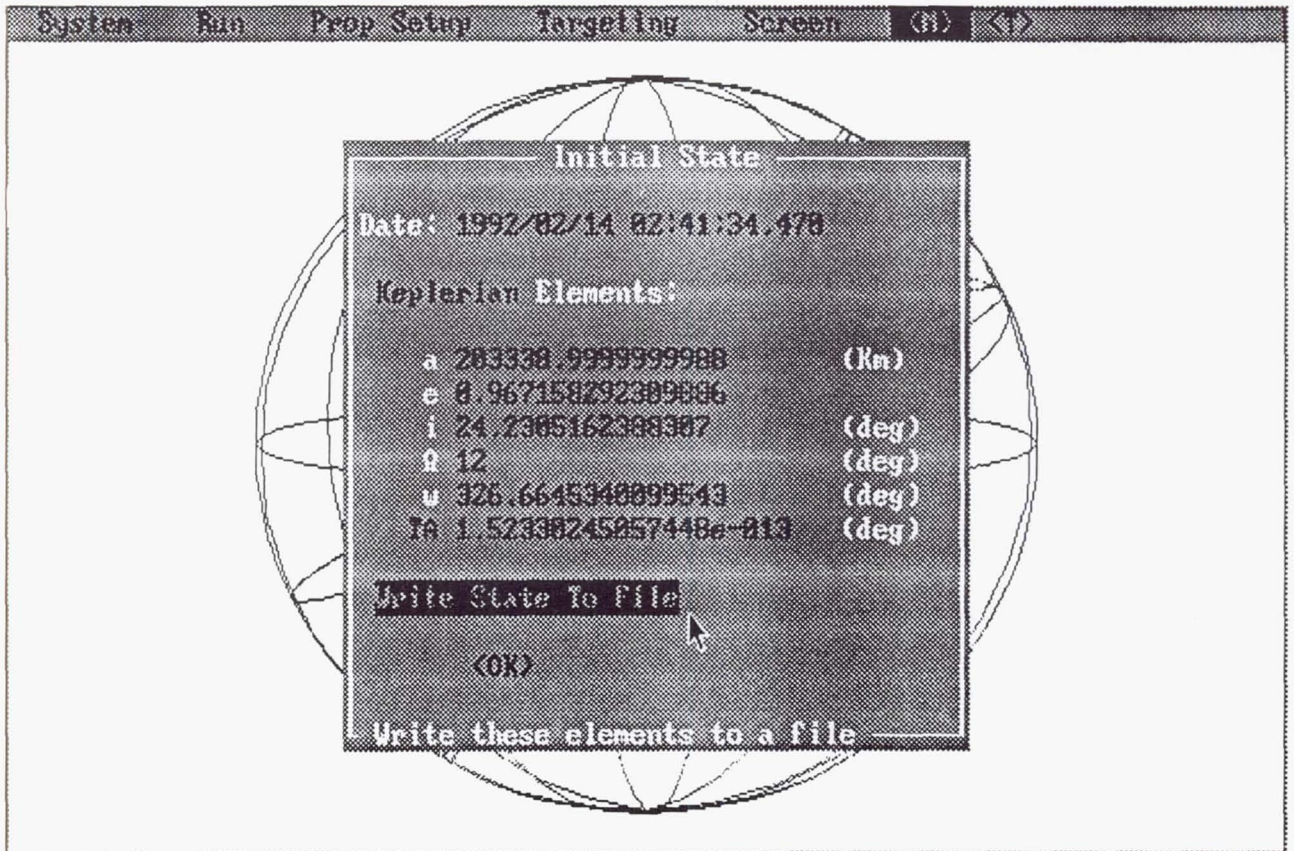
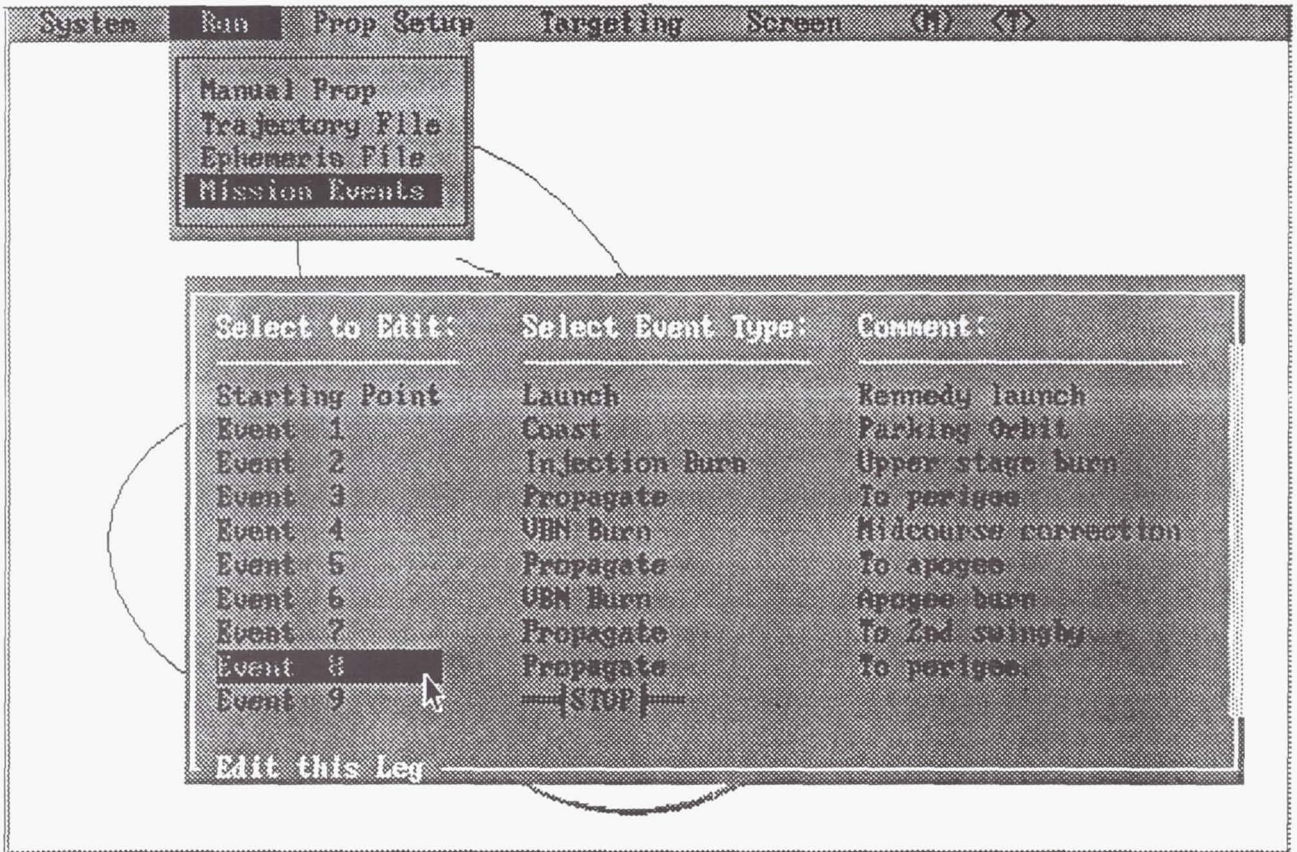*Figure 9. A screen from the program Swingby.*

Figure 10. Mission Event Sequence Screen from the program Swingby.

is designed for both pre-mission analysis and maneuver planning during missions. Figure 9 shows a typical data entry screen from the program. The analyst inputs data through a series of pull-down and pop-up menus. He or she may select one of several different coordinate frames in which to plot the trajectories. (The trajectories shown earlier were created with Swingby). The analyst may also leave a trajectory on the screen, make a change to one of the initial conditions, and then plot the new trajectory (in a different color, if desired). This allows the analyst to gain a intuitive feel for controlling trajectories.

No trajectory, however, can be generated with a simple propagation from an initial state. DLS trajectories usually require several maneuvers. After launch there is usually a maneuver at the first inner loop apogee to raise perigee to a safe level, and a maneuver at the outer loop apogee to control the arrival conditions at the second lunar swingby. Swingby allows the analyst to create such a "mission." Through the menus, the analyst can string together a sequence of "events" that determine the "mission." An example menu is shown in Figure 10.

The first event launches the spacecraft. Swingby reads a file of coefficients that model the launch vehicle by representing the insertion state as a function of launch site, launch azimuth, and time and date of launch. By selecting "Starting Point" on the menu, the analyst may enter the launch time and date, along with the launch azimuth. The second event coasts in the parking orbit. Selecting this menu allows the analyst to enter the duration to spend in the parking orbit, and allows him or her to choose the color in which the orbit will be plotted. The

third event is the injection burn duration. The injection burn is currently modeled by fitting coefficients to the manufacturer's data, which can be changed through the menus for different vehicles. Event 4 causes the spacecraft to propagate until a selected stopping condition, such as the perigee before the swingby. The next event allows the user to model an impulsive burn by entering three components of the delta-V vector. Event 5 then propagates until the outer loop apogee, where Event 6 models the outer loop apogee maneuver. Event 7 propagates long enough to calculate the B parameters at the second swingby, and Event 8 continues to perigee.

The analyst may run such a mission to generate predicted spacecraft ephemerides, which can be analyzed for station coverage or shadow constraints. If the second swingby is not sufficient for mission goals, then the analyst may use Swingby's targeter to vary some of the values from selected events until the desired conditions at the second swingby are achieved. For example, the analyst may choose to vary the launch time and coast time, along with the normal component of the outer loop delta-V, until the second swingby is acceptable.

Swingby allows the analyst to specify not only the floating end-point goals, as described above, but also "fixed" goals, such as any of the Keplerian elements; orbit period or energy; time of flight or time of arrival; and B·T, B·R, B magnitude, and the angle between B and the T axis.

434

4. CONCLUSIONS

We have described how the Multiconic orbit integrator has been used successfully for rapidly designing gravity-assisted trajectories in the Earth-Moon system. We compared Multiconic with a more sophisticated numerical integrator and showed how the results differed only by about 40 km after 68.5 days, on a trajectory that is heavily influenced by the Moon.

We also presented a method to design gravity-assist trajectories by using a simple Newton method with B-Plane parameters as goals. The analyst, however, describes the goals in terms of the desired semimajor axis by using floating end-point targeting. The targeter will automatically force the outbound trajectory to lie in the Moon's orbit plane.

Currently, we are working on several areas related to this paper. We are increasing Multiconic's accuracy, modeling more perturbations, and increasing its speed. We are also formulating the floating end-point for the more general out-of-plane solutions. Furthermore, Swingby is still in its first stage of development. We plan to increase Swingby's capabilities to include the design of libration point orbits, and trajectories to comets, asteroids, and planets.

REFERENCES

1. Farquhar R W & al 1984, Trajectories and orbital maneuvers for the ISEE-3/ICE comet mission, AIAA Paper No. 84-1976.

2. Uesugi K & al 1990, Japanese first double lunar swingby mission "HITEN", 41st Congress of the International Astronautical Federation, Paper No. 90-343.

3. Farquhar R W & al 1981, A new trajectory concept for exploring the Earth's geomagnetic tail, Journal of Guidance and Control, 4, 2, 192-196.

4. Dunham D W and Devlin (Davis) S A 1990 and 1991, Internal discussions.

5. Byrnes D V & al 1970, , Multi-Conic: a fast and accurate method of computing space flight trajectories, AIAA Paper No. 70-1062.

6. D'Amario L A & al 1980, A new method for optimizing multiple flyby trajectories, AIAA Paper No. 80-1676.

7. D'Amario L A & al 1973, Error Analysis of Multi-Conic Techniques, AAS Preprint, ASS/AIAA Astrodynamics Conference, Vail Colorado.

8. Kwok J H & al 1979, Final report; Jet Propulsion Laboratory contract #955140 and updated user's guide for MULCON, The University of Texas at Austin.

9. Schiff C 1991, Internal GSFC/CSC memorandum.

10. Kizner W 1961, A method of describing miss distances for lunar and interplanetary trajectories, Ballistic Missile and Space Technology, III.

11. Tucker B 1963, Transforming mission constraints into efficient parameters using a floating end-point concept, AIN #40-63.

12. Wertz, J R (Ed.) 1978, Spacecraft attitude determination and control, D. Reidel Publishing Co., Dordrecht, Holland, 363-364.

13. Grubin, C 1977, Simple algorithm for intersecting two conical surfaces, J. Spacecraft, 14.