Gravity Assist Preliminary Research & Design

Iowa State University, Asteroid Deflection Research Center

Justin Graupman

Introduction

The use of a gravity assist maneuver can be very beneficial. It can give a satellite a much needed boost in ΔV in order to reach a certain region or object in space. The gravity assist works by stealing momentum from a flyby planet and thus increasing the heliocentric velocity of the spacecraft. In order to gain velocity, the craft must pass the planet from behind.

This paper will discuss the preliminary methods and design of creating a gravity assist trajectory and present an algorithm to simulate it.

Propagation Methods

In order to simulate the satellite trajectory, a propagation method must first be chosen. Two different types of propagation methods will be discussed.

Ordinary Differential Equation Method

The first and most accurate way of plotting and planning the path of a satellite is to use an ODE solver to directly solve the equations of motion given initial conditions. While this method is ultimately more accurate, it comes with the cost of computation time. This makes the method less friendly but provides a more accurate representation of the satellite's trajectory as well as a higher ability to custom create the path you want the satellite to take. It also allows for the possibility of a continuous thrust solution such as ion propulsion. This method is based upon Newton's Law of Gravitation [5].

$$m_1 * \ddot{r} = -m_2 * \ddot{r} = \frac{G * m_1 * m_2}{r_{12}^3} * \overrightarrow{r_{12}}$$

Newton's Law of Gravitation can be described by the vector form of the two-body force equation above. Where $m_1 and\ m_2$ are the masses of the two gravitational bodies, G is the gravitational constant, and r_{12} is the distance between the two bodies. Acceleration equations can then be simply added together to take into account the force of multiple gravitational bodies acting on a single satellite.

In order to successfully simulate the path of the satellite we will need to at least add the gravitational force bodies that affect the satellite the greatest. These bodies include the Sun, Earth, the flyby planet, and the target Near Earth Object. Given initial conditions, we can then use Runge-Kutta methods or Matlab's ODE45 to solve the differential equations over a certain period of time. NASA's Horizon Database [4] has projected orbits for most known objects within our Solar System. By iterating through initial conditions given by the Horizon Database, we can solve numerous different trajectories until one proves satisfactory.

Due to the computation time required and the complexity of the differential equation method, for this design sequence I will be solving Lamberts problem.

Lamberts Problem

Lamberts problem is characterized by taking two position vectors and the time of flight between them and solving for the incoming and outgoing velocity vectors of the transfer trajectory. In theory, Lamberts problem will need to be solved twice. It will need to be solved once for the departure from Earth and the arrival at the flyby planet, and again for the departure from the flyby planet and the arrival at the target planet. From these departure and arrival velocities, we can then calculate the ΔV requirements that the spacecraft will need to be able to perform those maneuvers. There are many methods that can be used to solve Lamberts problem. For this design we will be using the Universal Variable method.

Universal Variable Method

The algorithm that is used is taken from *Fundamentals of Astrodynamics and Applications* [3]. This universal variable algorithm utilizes a bisection method as opposed to a Newtonian iteration method [1]. While approximately 5% slower, the bisection method is a much more robust solution for a wide variety of transfer orbits.

The formulation of this method begins with the *f* and *g* universal variable expressions that were found to be solutions to Kepler's problem. They are given as follows:

$$f = 1 - \frac{X_0^2}{r_0} c_2 \qquad g = t - \frac{X_0^3}{\sqrt{\mu}} c_3 \qquad \dot{g} = 1 - \frac{X_0^2}{r} c_2 \qquad \dot{f} = \frac{\sqrt{\mu}}{r r_0} X_0 (\Psi c_3 - 1)$$

Where *r* and r_0 are the magnitudes of the initial and final position vectors, X_0 is a universal variable that relates energy to angular momentum, Ψ is the square root of the distance traveled between the two known position vectors, and c_2 and c_3 are Stumpff's coefficients.

Since time of flight is specified at the beginning of the problem, the *f* and *g* expressions that use the change in true anomaly are also required[3]. These are:

$$f = 1 - \frac{r}{p}(1 - \cos(\Delta\nu)) \qquad g = \frac{r r_0 \sin(\Delta\nu)}{\sqrt{\mu p}} \qquad \dot{g} = 1 - \frac{r_0}{p}(1 - \cos(\Delta\nu))$$

$$\dot{f} = \sqrt{\frac{\mu}{p}} \left(\frac{1 - \cos(\Delta\nu)}{\sin(\Delta\nu)} \right) \left(\frac{1 - \cos(\Delta\nu)}{p} - \frac{1}{r} - \frac{1}{r_0} \right)$$

The universal variable X_0 can now be solved for by equating both versions of the *f* equations.

$$X_0 = \sqrt{\left(\frac{rr_0(1-\cos(\Delta\nu))}{pc_2}\right)}$$

This new formulation of X_0 can be substituted into the universal variable form of the $\dot{f}$ expression and used to equate it to the true anomaly angle form. This step is given as:

$$\left(\frac{1-\cos(\Delta\nu)}{\sin(\Delta\nu)}\right)\left(\frac{1-\cos(\Delta\nu)}{p}-\frac{1}{r}-\frac{1}{r_0}\right) = \sqrt{\frac{1-\cos(\Delta\nu)}{rr_0c_2}}(\Psi c_3 - 1)$$

Next, the expression is simplified by multiplying through by the magnitudes of the position vectors:

$$\left(\frac{1-\cos(\Delta\nu)}{\sin(\Delta\nu)}\right)\left(\frac{rr_0(1-\cos(\Delta\nu))}{p}\right)+\left(\frac{rr_0(1-\cos(\Delta\nu))}{\sin(\Delta\nu)}\right)\left(-\frac{1}{r}-\frac{1}{r_0}\right) = \frac{rr_0\sqrt{1-\cos(\Delta\nu)}}{\sqrt{rr_0}\sqrt{c_2}}(\Psi c_3 - 1)$$

And further by dividing through by the first factor of the first term to obtain:

$$\frac{rr_0(1-\cos(\Delta\nu))}{p} = r_0 + r + \frac{\sqrt{rr_0}\sin(\Delta\nu)}{\sqrt{1-\cos(\Delta\nu)}}\frac{(\Psi c_3 - 1)}{\sqrt{c_2}}$$

By defining two terms, *y* and *A*, it is possible to simplify the previous equation further.

$$y = \frac{rr_0(1-\cos(\Delta\nu))}{p} \qquad A = \frac{\sqrt{rr_0}\sin(\Delta\nu)}{\sqrt{1-\cos(\Delta\nu)}}$$

The y and X_0 equations then become,

$$y = r_0 + r + \frac{A(\Psi c_3 - 1)}{\sqrt{c_2}} \qquad X_0 = \sqrt{\frac{y}{c_2}}$$

By setting the two original *g* equations equal to each other it is possible to solve for time. Then, substituting the new variables yields:

$$t = \frac{X_0^3 c_3}{\sqrt{\mu}} + \frac{A\sqrt{y}}{\sqrt{\mu}}$$

The *f* and *g* expressions from the beginning now simplify down to:

$$f = 1 - \frac{y}{r_0} \qquad g = A\sqrt{\frac{y}{\mu}} \qquad \dot{g} = 1 - \frac{y}{r}$$

And finally the velocity vectors can be found from the *f* and *g* expressions via the following expressions:

$$v_0 = \frac{\vec{r} - (f * \vec{r_0})}{g} \qquad v = \frac{(\dot{g} * \vec{r}) - \vec{r_0}}{g}$$

Goals

The goal of this preliminary design is to create a working code that simulates a gravity assist around a flyby planet and arrives at another planet while minimizing the change in velocity required. Perhaps one of the easiest gravity assist maneuvers to perform is that around Venus for an arrival at Mars. In order to properly test the code that is written, it must be tested against known solutions. Once it is tested, I can move on to the calculation of a gravity assist trajectory for arrival at a Near Earth Object. The goals can be summarized in the following lines:

1.) Develop Gravity Assist Maneuver Code
2.) Test Code Against Known Solutions
3.) Begin Analysis for Arrival at Near Earth Object

The known solution that my code will be tested against comes from the AERE 461 design class from the spring of 2012. Although I personally was not enrolled in the class, I chose to take on their class' final project as a side project. The AERE 461 final project required them to develop a minimum change in velocity gravity assist method to arrive at Mars. I will be attempting to be reproducing the following results from the project.

	Earth to Venus	
Initial Julian Date	Final Julian Date	ΔV Required (km/s)
2452489.4485	2452623.3702	3.7751

	Gravity Assist	
Flyby Radius (km)	Turning Angle (deg)	ΔV Gained (km/s)
16986.2937	42.13317	.000931292808

	Venus to Mars	
Initial Julian Date	Final Julian Date	ΔV Required (km/s)
2452623.3702	2452839.5819	2.7288

Problem Description

The main goal of this problem is to utilize a gravity assist from Venus and determine a minimum two impulse mission to Mars. We will begin looking for a solution on January 1st, 2000 at 12:00 PM. The Julian date corresponding to this time is 2451544.5. Therefore:

$$J_0 = 2451544.5$$

Assume that we are leaving Earth from a low Earth orbit altitude of 200 km and will be arriving at Mars and inserting into a 200 km orbit with an eccentricity of 0.8.

Since we are using two Lambert solutions, there are three important dates that need calculated for the minimum delta V solution; the departure date from Earth, the arrival at Venus for the flyby, and the arrival and orbit insertion at Mars. The variables that will need to be optimized are the launch date, the time of flight from Earth to Venus, and the time of flight from Venus to Mars. For this problem we will limit the launch date to be between January 1st 2000 and January 1st 2005. We will also limit the time of flight between Earth and Venus and the time of flight between Venus and Mars to 500 days or less. We will call the variables being optimized T_1, T_2, T_3.

$$T_1 = 0\ to\ 1825 \qquad T_2 = 25\ to\ 500 \qquad T_3 = 25\ to\ 500$$

Incoming and outgoing hyperbolic excess velocities (V_∞) will need to be calculated at the beginning and end of each of the two legs of the journey. This will be accomplished by the two Lambert solver solutions. The difference between the arrival velocity at Venus and the departure velocity needed to reach Mars from Venus will need to be provided by the gravity assist.

Problem Solution Methodology

The problem solution will follow the following pseudo code:

1.) Obtain Ephemeris data for Earth, Mars and Venus from NASA's Horizons database.
2.) Find Lambert Solution from Earth to Venus
3.) Find Lambert Solution from Venus to Mars
4.) Determine the change in velocity necessary at the gravity assist to patch the two Lambert solutions together.

NASA's Horizons Database

The Horizons database is by far the most accurate way of attaining past velocity and position data and future predictions of velocity and position data. It is provided by the Solar System Dynamics Group as well as the Jet Propulsion Laboratory. We will need to first obtain data between the years 2000 and 2005 for this project. The process about to be described in the following pages can be iterated for each date and time between 2000 and 2005 to find an optimal solution.

Lambert Solver

Both legs of the mission will be able to utilize the same Lambert solver. As stated earlier, the solver takes two position vectors and the desired time of flight between them and solves for the incoming and outgoing position vectors of the transfer trajectory. The following pseudo code utilizes the Universal Variable Technique described previously in this paper. This algorithm uses a bisection technique [3] that determines upper and lower bounds and continually readjusts them until the actual value of Ψ for the transfer orbit between the two position vectors is found. The loop exits when the algorithm finds that the current time of flight is close to the specified time of flight.

Given: $(\vec{r_0}, \vec{r}, \Delta t_0) \xrightarrow{yields} (\vec{v_0}, \vec{v})$

The first thing that needs to be done is the calculation of the change in true anomaly angle. To avoid quadrant ambiguity, the following equations can be used [3].

$$\Delta\nu = \cos^{-1}\left(\frac{\vec{r_0}\cdot\vec{r}}{r_0 r}\right) \qquad if\ (\vec{r_0}\times\vec{r})_k \geq 0$$

$$\Delta\nu = (2\pi) - \cos^{-1}\left(\frac{\vec{r_0}\cdot\vec{r}}{r_0 r}\right) \qquad if\ (\vec{r_0}\times\vec{r})_k < 0$$

Where the k represents the third part of the vector in an ijk vector system. Then from the Universal Variable formulation discussed earlier, A can be calculated.

$$A = \sqrt{r r_0 (1 + \cos(\Delta\nu))}$$

Next we will set Initial Conditions for the loop and then walk through the loop using the equations formulated previously.

$\Psi_n = 0.0$, therefore $c_2 = \frac{1}{2}$ and $c_3 = \frac{1}{6}$

$\Psi_{up} = 4\pi^2 \quad \Psi_{low} = -4\pi$

LOOP UNTIL $|\Delta t - \Delta t_0| < 1 * 10^{-6}$

$$y_n = r_0 + r + \frac{A(\Psi_n c_3 - 1)}{\sqrt{c_2}}$$

IF (A > 0) AND (y < 0) THEN

$$\Psi_{low} = \frac{2\sqrt{c_2}(-r_0 - r)}{A c_3} + \frac{2}{c_3} - \Psi_{up} + tolerance$$

$$\Psi_n = \frac{(\Psi_{up} + \Psi_{low})}{2}$$

$$y_n = r_0 + r + \frac{A(\Psi_\mathrm{n} c_3 - 1)}{\sqrt{c_2}}$$

END IF

Where the *tolerance* inside the above IF statement is a very small positive number such that y_n comes out to be slightly above zero.

$$X_0 = \sqrt{\frac{y_n}{c_2}} \qquad \Delta t = \frac{X_0^3 c_3 + A\sqrt{y_n}}{\sqrt{\mu}}$$

IF ($\Delta t \ \leq \Delta t$) THEN

$$\Psi_{low} = \Psi_n$$

ELSE

$$\Psi_{up} = \Psi_n$$

END IF

$$\Psi_{n+1} = \frac{\Psi_{up} + \Psi_{low}}{2}$$

$$\Psi_n = \Psi_{n+1}$$

Stumpff's coefficients can then be recalculated as follows,

$$c_2 = \frac{1}{2!} - \frac{\Psi_{n+1}}{4!} + \frac{\Psi_{n+1}^2}{6!} - \cdots$$

$$c_3 = \frac{1}{3!} - \frac{\Psi_{n+1}}{5!} + \frac{\Psi_{n+1}^2}{7!} - \cdots$$

END LOOP

$$f = 1 - \frac{y_n}{r_0} \qquad g = A\sqrt{\frac{y_n}{\mu}} \qquad \dot{g} = 1 - \frac{y_n}{r}$$

$$\overrightarrow{v_0} = \frac{\vec{r} - f\overrightarrow{r_0}}{g} \qquad \vec{v} = \frac{\dot{g}\vec{r} - \overrightarrow{r_0}}{g}$$

Where $\overrightarrow{v_0}$ and $\vec{v}$ are the departure and arrival velocities respectively.

After the algorithm completes, the final velocity vectors needed at the departure and arrival are obtained. This algorithm will need to be utilized for both the leg between Earth and Venus and the leg between Venus and Mars. Once we have the departure and arrival velocities for both sections of the mission, we can find the ΔVs that will be required. For a better understanding, we will utilize the following positions as subscripts:

Position 1: Departure from Earth
Position 2: Arrival to Venus
Position 3: Departure from Venus
Position 4: Arrival to Mars

Also, take note that in this simulation position 2 and 3 take place at the same time. In other words, the position and velocity of Venus at position 2 and 3 are the same and occur at the same time $(J_0 + T_1 + T_2)$.

Earth Departure

To reiterate, the initial low Earth orbit parameters are:

$$r_p = 200 + R_{Earth} \qquad\qquad e = 0$$

Where R_{Earth} is the radius of the Earth and e is the orbit eccentricity. Next we need to calculate the Earth escape velocity vector.

$$\vec{V}_{\infty_1} = \vec{V}_{0_1} - \vec{V}_{Earth}$$

Where $\vec{V}_{0_1}$ is the departure velocity vector from the Lambert solver and $\vec{V}_{Earth}$ is the velocity vector of Earth at the time T_1 of launch. The departure velocity required to obtain the calculated $\vec{V}_{\infty_1}$ is then:

$$V_{p_1} = \sqrt{V_{\infty_1}^2 + \frac{2\mu_{Earth}}{r_p}}$$

The velocity of the low Earth orbit is then

$$V_{c_1} = \sqrt{\frac{\mu_{Earth}}{r_p}}$$

Now the departure ΔV from Earth can now be calculated.

$$\Delta V_{Earth} = \left|V_{p_1} - V_{c_1}\right|$$

Mars Arrival

Remember, we need to calculate the second leg of the journey before we can calculate the gravity assist maneuver. As a reminder, the orbit insertion parameters at arrival to Mars are as follows:

$$r_p = 200 + R_{Mars} \qquad\qquad e = 0.8$$

Similar to the Earth departure, the following values can be found using similar equations:

$$\vec{V}_{\infty_4} = \vec{V}_{Mars} - \vec{V}_4 \qquad\qquad V_{p_4} = \sqrt{V_{\infty_4}^2 + \frac{2\mu_{Mars}}{r_p}}$$

Where $\vec{V}_{Mars}$ is the velocity of the planet at time $(J_0 + T_1 + T_2 + T_3)$ and $\vec{V}_4$ is the arrival velocity given by the Lambert solution for the second leg of the journey from Venus to Mars. The perigee velocity needed at insertion can now be calculated:

$$p = r_p(1 + e) \qquad\qquad V_{p_{in}} = \sqrt{\frac{\mu_{Mars}}{p}}(1 + e)$$

The ΔV needed at Mars can now be calculated:

$$\Delta V_{Mars} = \left|V_{p_4} - V_{p_{in}}\right|$$

Gravity Assist at Venus

The gravity assist is the part of this problem that patches the two Lambert solutions together. The goal of this section is find a perigee radius at Venus which will give us the required change in velocity to reach Mars. This whole gravity assist formulation was taken from reference [2] page 42. First we must find the $\overrightarrow{v_\infty}$ vectors at arrival and departure from Venus.

$$\vec{V}_{\infty_2} = \vec{V}_2 - \vec{V}_{Venus} \qquad\qquad \vec{V}_{\infty_3} = \vec{V}_3 - \vec{V}_{Venus}$$

Where $\vec{V}_2$ and $\vec{V}_3$ are velocity vectors solved for from the Lambert solvers and $\vec{V}_{Venus}$ is the heliocentric velocity vector of Venus at time $(J_0 + T_1 + T_2)$. The next step is to determine the semi-major axis of the incoming and outgoing trajectories of the satellite. The following equations can accomplish that.

$$a_2 = -\left(\frac{\mu_{Venus}}{V_{\infty_2}^2}\right) \qquad\qquad a_3 = -\left(\frac{\mu_{Venus}}{V_{\infty_3}^2}\right)$$

The radius of perigee at Venus and the turning angle required can be defined as:

$$r_p = a_2(1 - e_2) = a_3(1 - e_3) \qquad\qquad \delta = \cos^{-1}\left(\frac{\vec{V}_{\infty_2} \cdot \vec{V}_{\infty_3}}{V_{\infty_2} * V_{\infty_3}}\right)$$

Unfortunately, the radius at perigee cannot be solved for directly because we have two unknowns within the equation. Both eccentricities are still unknown. We will need another equation to be able to solve for both unknown eccentricities. The turning angle calculated above can also be defined by:

$$\delta = \sin^{-1}\left(\frac{1}{e_2}\right) + \sin^{-1}\left(\frac{1}{e_3}\right)$$

Rearranging the radius at perigee equation and plugging it into the above δ equation yields:

$$\delta = \sin^{-1}\left(\frac{1}{\frac{a_3}{a_2}(e_3 - 1) + 1}\right) + \sin^{-1}\left(\frac{1}{e_3}\right)$$

This equation can then be rearranged as:

$$f = \left(\frac{a_3}{a_2}(e_3 - 1) + 1\right) * \sin\left(\delta - sin^{-1}\left(\frac{1}{e_3}\right)\right) - 1 = 0$$

There is only one unknown in the above equation, but it is impossible to solve for directly. A Newton iteration scheme [1] can be implemented to solve for e_3 in the above equation. First we will need the derivative of the above equation with respect to e_3:

$$\frac{df}{de_3} = \left(\frac{a_3}{a_2}e_3 - \frac{a_3}{a_2} + 1\right)\left(\frac{\cos\left(\delta - sin^{-1}\left(\frac{1}{e_3}\right)\right)}{e_3^2\sqrt{1 - \frac{1}{e_3^2}}}\right) + \frac{a_3}{a_2}\sin\left(\delta - sin^{-1}\left(\frac{1}{e_3}\right)\right)$$

Begin the iteration by choosing an initial guess for e_3, then solve for both f and $\frac{df}{de_3}$. The new initial value for e_3 can then be calculated using the following equation.

$$e_{new} = e_{old} - \frac{f}{\frac{df}{de_2}}$$

Where e_{old} was the initial guess value that was chosen. The e_{new} value then becomes the new guess value for the next iteration. Iterate until e_{new} stops changing and that then becomes the correct e_3. The second value e_2 and the radius of perigee can then be calculated from the following equations:

$$e_2 = \frac{a_3}{a_2}(e_3 - 1) + 1 \qquad r_p = a_2(1 - e_2) = a_3(1 - e_3)$$

Finally the ΔV gained from the gravity assist can now be calculated:

$$\Delta V_{GA} = \left|\sqrt{v_{\infty_2}^2 + \frac{2\mu_{Venus}}{r_p}} - \sqrt{v_{\infty_3}^2 + \frac{2\mu_{Venus}}{r_p}}\right|$$

Solutions Obtained

After iterating through launch dates and time of flights, my code converged on a minimum change in velocity solution. The results are summarized in the following tables.

	Earth to Venus	
Initial Julian Date	Final Julian Date	ΔV Required (km/s)
2452489.04167	2452623.20833	3.77575405897

	Gravity Assist	
Flyby Radius (km)	Turning Angle (deg)	ΔV Gained (km/s)
22503.6240499	42.2778782870	.001033624396324

	Venus to Mars	
Initial Julian Date	Final Julian Date	ΔV Required (km/s)
2452623.20833	2452623.20833	2.72982285006

As you can see, the solutions are extremely similar to the desired solutions discussed earlier in this paper. For reference, here are the velocity vectors from the Lambert solver solutions in km/s.

Earth Departure	20.4263509865	17.2673054757	2.37568575500
Venus Arrival	-35.0943623355	-14.1115907341	-3.19293167526
Venus Departure	-37.2026401942	-15.5929175978	0.143343579142
Mars Arrival	15.5219522206	13.8109688413	0.146263509639

Conclusions

The code was able to reproduce the solutions to the problem therefore proving that the methodology used to find the answer is correct. Given a smaller time step, I would have been able to find more exact solutions. Within the scope of this research, I was able to obtain all the goals except the part that would apply it to arrival at a Near Earth Object.

Future Work

Take the formulation within this paper and instead apply it to an arrival at a Near Earth object. The focus of this research was to ultimately develop a gravity assist solution for arrival at a Near Earth Object. The

Fortran 90 code included for download on the ISGC New Base Program website at Iowa State also needs a great deal of cleanup and optimization. In the time allotted to me, I was unable to clean it up and optimize it. However, it runs and solves the problem correctly nonetheless. Good luck to whoever may decide to continue the research.

Bibliography

References

[1] Bate, Roger R., Donald D. Mueller, and Jerry E. White. "Section 5.3." *Fundamentals of Astrodynamics*. New York: Dover Publications, 1971. 231-41. Print.

[2] Qadir, Khurram. "Multi-gravity Assist Design Tool for Interplanetary Trajectory Optimization." Thesis. Luleå University of Technology, 2008. Web. 19 Feb. 2012. <http://epubl.ltu.se/1653-0187/2009/062/LTU-PB-EX-09062-SE.pdf>.

[3] Vallado, David A., and Wayne D. McClain. *Fundamentals of Astrodynamics and Applications*. Dordrecht: Kluwer Academic, 2001. Print.

[4] Solar System Dynamics Group, and Jet Propulsion Laboratory. "HORIZONS System."*HORIZONS System*. NASA. Web. 19 May 2012. <http://ssd.jpl.nasa.gov/?horizons>.

[5] Chobotov, Vladimir A. *Orbital Mechanics*. 3rd ed. Reston, VA: American Institute of Aeronautics and Astronautics, 2002. Print.