

High Gain Antenna Pointing on the Mars Exploration Rovers

C. Anthony Vanelli and Khaled S. Ali

23rd February 2005

Abstract

This paper describes the algorithm used to point the high gain antennae on NASA/JPL's Mars Exploration Rovers. The gimballed antennae must track the Earth as it moves across the Martian sky during communication sessions. The algorithm accounts for (1) gimballed range limitations, (2) obstructions both on the rover and in the surrounding environment, (3) kinematic singularities in the gimballed design, and (4) up to two joint-space solutions for a given pointing direction. The algorithm computes the intercept-times for each of the occlusions and chooses the joint-space solution that provides the longest track time before encountering an occlusion. Upon encountering an occlusion, the pointing algorithm automatically switches to the other joint-space solution if it is not also occluded. The algorithm has successfully provided flop-free pointing for both rovers throughout the mission.

Keywords: Inverse kinematics, hazard avoidance, motion planning, tracking.

1 HGA Heritage and Flops

The Mars Exploration Rover (MER) communications system consists of three basic elements: a fixed-position low-gain antenna providing direct-to-Earth, semi-omnidirectional communication at very low data rates (~ 10 bps); a UHF-band antenna for short range, very high data rate communication with other spacecraft orbiting Mars, which then relay the data to Earth after some delay; and a gimballed high-gain antenna (HGA) for direct-to-Earth real-time communication at high data rates.

Experience with the Mars Pathfinder (MPF) mission showed that its HGA design was vulnerable to "flops" when the Earth was high in the Martian sky. A typical flop is illustrated in Figure 1. The HGA is shown enclosed within the celestial hemisphere,

and the Earth positions on the sphere are numbered 1–4, while the respective gimballed configurations are labelled *a*–*d* (shown off to the side). As the Earth rose from position 1 to position 2, a flop would occur as the primary gimbal (on the horizontal axis) dropped the HGA from position *a* to position *b*—the furthest the gimbal could depress. As the Earth proceeded to the neighboring position 3, the vehicle would simultaneously rotate both gimbal axes through 180° in order to continue tracking "from the flip side" at position *c*, but severing the communications link in the process. Orbital mechanics compounds the issue: under the Earth-to-Mars trajectory used for MPF, earthrise (as seen from Mars) occurred well before sunrise for several months after landing. As a result, by the time the solar-powered vehicle was able to communicate, the Earth was near zenith and thus the HGA is almost "on the floor" and about to flop. Such interruptions are costly and extremely disruptive to mission operations.

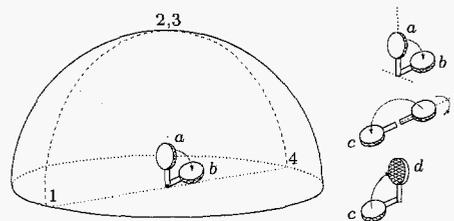


Figure 1: Mars Pathfinder HGA flops.

Since the solar-powered MER vehicles would also face similar orbital geometry, reducing or eliminating flops became a priority for the MER design team. The MER mechanical design places the primary gimbal axis vertically, which alleviates the problem. It also provides two joint-space solutions (known as **branches**) that point the antenna in the same direction. However, the placement of the MER high-gain antenna on the rover deck (see Figure 2) presents

additional challenges. Equipment on the deck (particularly the camera mast) limits the range of motion and also can occlude the line-of-sight. Some occlusions may be avoided by selecting the alternate branch, whereas others may not (such as the deck itself when the rover is tilted and the Earth is low on the horizon.)

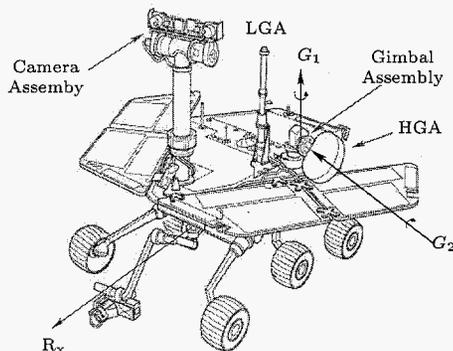


Figure 2: MER HGA and deck obstructions.

A pointing algorithm was designed to address these challenges. The HGA pointing algorithm (1) minimizes the occurrence of flops due to motion limits, (2) avoids local obstructions to the line-of-sight by choosing alternate joint-space solutions where possible, and (3) provides the longest possible tracking time. If a pointing constraint is encountered during the tracking session, the algorithm automatically “flops” to the alternate solution.

The algorithm expands on the approach in [1] to describe all the pointing constraints with circles and combinations of circles on the celestial sphere. The algorithm also approximates the Earth trajectory with a circle on the sphere. For each constraint, the algorithm determines the **incursion time**—the time until the Earth passes into the constraint region—by computing the intersection of the Earth trajectory with each constraint region. While some of the constraints are common to both branches, others are specific to only one branch, so the algorithm determines the overall incursion time for each branch individually. For a given branch, it collects the incursion times of the common constraints together with those specific to that branch; the minimum of these times is the overall incursion time for that branch. In the last step, the algorithm selects the branch with the longest time as the best branch with which to begin

tracking the Earth.

2 Inverse Kinematics

There are two gimbal axes G_1 and G_2 . The primary axis G_1 is normal to the deck and the secondary axis G_2 is parallel to the deck. The antenna dish is mounted at the end of a short arm and the beam radiates perpendicular to the arm, as shown in Figure 3. The vector R_x points toward the front of the rover. The HGA gimbal frame \mathcal{F}_H has its origin at the intersection of the gimbal axes, with basis vector H_z along axis G_1 and basis vector H_x directed 30° to the right of R_x . As shown, the gimbal angles $(g_1, g_2) = (0, 0)$ place the arm along the H_x axis and point the beam along the $-H_y$ axis. Note that $g_2 = 90^\circ$ points the beam vertically up. For the rest of this paper, all computations are in the HGA frame unless explicitly indicated otherwise.

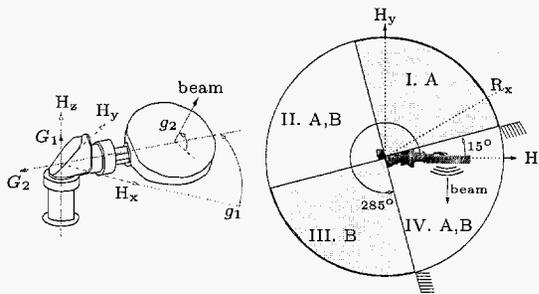


Figure 3: MER HGA Gimbal Assembly.

There are two joint-space configurations to point the antenna at a given target e in the HGA frame. The first solution, which we call Branch A, is

$$\begin{aligned} g_{1A} &= \frac{\pi}{2} + \text{atan2}(e_y, e_x) \\ g_{2A} &= \text{asin}(e_z). \end{aligned} \quad (1)$$

The second solution, which we call Branch B, is

$$\begin{aligned} g_{1B} &= \frac{3\pi}{2} + \text{atan2}(e_y, e_x) \\ g_{2B} &= \pi - \text{asin}(e_z). \end{aligned} \quad (2)$$

The Branch A solution returns $g_2 \in [-90^\circ, 90^\circ]$ while the Branch B solution returns $g_2 \in [90^\circ, 270^\circ]$. The notion of “branch” is undefined when $g_2 = \pm 90^\circ$; then any value for g_1 suffices.

Once the HGA deploys from its launch-stowed configuration, the range of travel for the primary gimbal

is mechanically limited to $g_1 \in [15^\circ, 285^\circ]$. The secondary gimbal ranges through $g_2 \in [0^\circ, 235^\circ]$ (which we further restrict to $g_2 \in [0^\circ, 180^\circ]$ to avoid pointing the antenna beam into the deck). These limits divide the celestial hemisphere into the four regions shown in Figure 3. The admissible solutions for each region are indicated in the figure; for example, when the target e lies in region III the only joint-space solution that doesn't violate the motion limits is Branch B.

Onboard software provides knowledge of the Earth's position in the Martian sky as a function of time. With the rover's position and orientation, also provided by other onboard software, the HGA pointing algorithm transforms the Earth direction vector into the HGA frame. Then the branch solutions are used to generate a set of gimbal angles to point the HGA at the Earth. This process is sufficient to track the Earth during a communication session, but does not guarantee avoidance of flops or other line-of-sight occlusions.

3 Trajectories and Occlusions

There are three categories of constraints: mechanical limits imposed by the gimbal range of motion, kinematic constraints imposed by the gimbal design and the motors' maximum speeds, and physical obstructions to the line-of-sight created by features on the rover deck or the local terrain (such as high canyon walls). Using the celestial sphere to represent the set of all unit vectors, we map each constraint to the celestial sphere using circles drawn on the sphere; this mapping defines an **occlusion**. We also approximate the trajectory of the Earth on the Martian celestial sphere as a circle. Repeated use of such circles and their intersections form the cornerstone of the HGA occlusion avoidance algorithm.

3.1 Earth Trajectory

As Mars rotates, the Earth traces an approximately circular trajectory on the celestial sphere. This circle can be visualized as the intersection of the sphere with a cone whose central axis is the Mars planetary axis. Given the unit axis ω and a point p_0 on the circle, *any* spherical circle can be expressed parametrically with

$$p(\varphi) = \exp(\varphi \omega^\times) p_0, \quad (3)$$

as φ ranges from 0 to 2π . From [2], the matrix exponential $\exp(\varphi \omega^\times)$ is a rotation matrix that rotates

vectors about ω through an angle φ . The notation ω^\times denotes the constant skew-symmetric matrix formed from the elements of ω :

$$\omega^\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (4)$$

Since from the rover's point of view Mars is fixed and Earth is moving, the cone axis ω_E is

$$\omega_E = -\frac{\Omega}{\|\Omega\|}, \quad (5)$$

where Ω is the angular velocity of Mars. Given an initial position e_0 and setting $\varphi = (\|\Omega\| t)$, we define the Earth trajectory

$$e(t) = \exp(\|\Omega\| t \omega_E^\times) e_0. \quad (6)$$

The resultant circle E is shown in Figure 4.

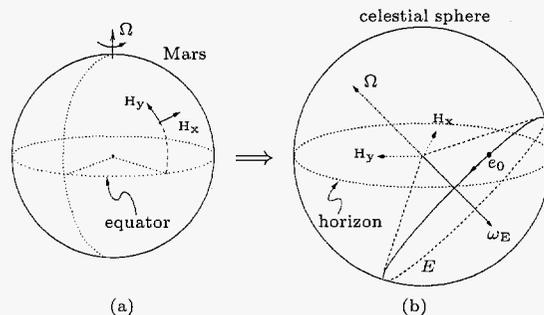


Figure 4: (a) The HGA frame on rotating Mars and (b) the resulting Earth circle E in the celestial sphere.

3.2 Occlusion Circles

We also use circles on the unit sphere to enclose constraint regions—those parts of the sky which are off-limits because of the constraint. We again make use of the cone-sphere analogy to define the **occlusion circle** $C := \odot(\omega, \alpha)$ shown in Figure 5. The circle C has cone axis ω and half-angle α . The center point of this circle is $c = \omega \cos(\alpha)$. Note that c lies *within* the celestial unit sphere.

By convention we take the generating axis ω to be directed *towards* the region of the sphere that is occluded. Thus $p(\varphi)$ from Eq. 3 is a directed curve with the standard definition of **enclosure**: a point is enclosed if it lies to the *left* of all points on the directed

curve [3]. A test point q is occluded if it is a member of the region enclosed by C . The membership test is

$$\begin{aligned} (q - c) \cdot \omega &\geq 0 \Rightarrow q \text{ is occluded.} \\ (q - c) \cdot \omega &< 0 \Rightarrow q \text{ is not occluded.} \end{aligned} \quad (7)$$

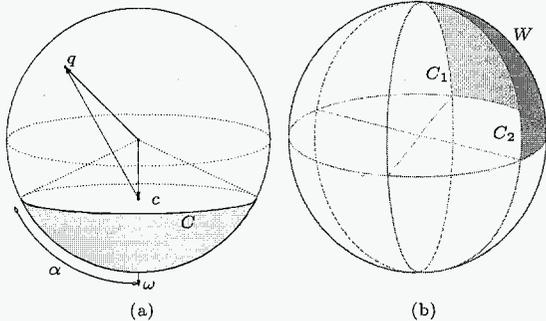


Figure 5: (a) Occlusion circle C and (b) Wedge W .

A **wedge occlusion** is constructed from the intersection of two $C_1 := \odot(\omega_1, \alpha_1)$ and $C_2 := \odot(\omega_2, \alpha_2)$. Figure 5 shows the wedge $W := C_1 \cap C_2$; the lower half of the wedge has been omitted for clarity. A point q is occluded by the wedge W if it is a member of *both* C_1 and C_2 . Thus the occlusion region is the collection of all points q that satisfy *both*

$$(q - c_1) \cdot \omega_1 \geq 0, \quad (8)$$

$$(q - c_2) \cdot \omega_2 \geq 0. \quad (9)$$

A **meld occlusion** is constructed from the union of two or more circles. The occluded region of the meld $M := C_1 \cup C_2 \cup \dots \cup C_n$ is the collection of all points q that satisfy *any* of the membership tests

$$(q - c_i) \cdot \omega_i \geq 0, \text{ for } i=1, \dots, n. \quad (10)$$

4 Occlusion Implementation

As noted, we can describe all three classes of occlusions using circles, wedges, and melds. In the following, we describe the specific implementation of each occlusion and how the HGA pointing algorithm computes the incursion time until the Earth passes into that occlusion. We present these occlusions in order of increasing complexity.

4.1 Terrain Occlusion

The Terrain Occlusion models the local horizon; it is unique in that it is the *only* occlusion that is tied to

the surface of Mars rather than the rover. We define the Terrain Occlusion as a single occlusion circle T with axis ω_T pointing at the center of Mars and $\alpha_T = \pi/2$. The occluded portion of the celestial sphere represents the ground.

For a given earth start position e_0 , the pointing algorithm computes the incursion time t_T —the time until the Earth passes into the occlusion circle T . First it verifies that e_0 is not *already* in T (i.e., below the horizon). If it is, then t_T is set to zero, and the algorithm proceeds to the Deck Occlusion.

If Earth is above the horizon, the algorithm next computes the intersections of the Earth circle E with the Terrain circle T . There are four possibilities: the two circles overlap, they do not intersect, they intersect at two points, or they intersect at one point (they are tangent). The overlap case is handled by the earlier test on e_0 . If the circles do not intersect, t_T is set to infinity. Otherwise, the algorithm computes the shortest angular displacement θ_T along the path E from the current Earth position e_0 to the intersections. The incursion time is thus

$$t_T = \theta_T / \|\Omega\|. \quad (11)$$

Note that since T is tied to the surface of Mars, t_T does not depend on rover attitude. One may also change the terrain half-angle α_T to model cases where the rover is within a deep crater or canyon.

4.2 Deck Occlusion

The Deck Occlusion models the Rover Equipment Deck, which presents an obstacle if the rover is tilted while the Earth low on the horizon. For example, with the rover tilted so that the high point of the deck is easterly, the effective “earthrise over the deck” is delayed until later in the morning; under a high westerly tilt, the effective earthset comes earlier in the evening.

In a manner exactly analogous to the Terrain Occlusion, we define a single occlusion circle D with axis $\omega_D = -H_z$, and half-angle $\alpha_D = \pi/2$. The occluded portion of the local celestial sphere represents the sky below the deck. The algorithm computes the incursion time t_T in the same manner as t_D , i.e., by finding the shortest θ_T and taking

$$t_T = \theta_T / \|\Omega\|. \quad (12)$$

4.3 Kinematic Singularity

The HGA gimbal has a kinematic singularity at $e = [0, 0, 1]^T$ (where $g_2 = 90^\circ$). As the Earth approaches this point, the required azimuthal tracking speed \dot{g}_1 exceeds the ability of the gimbal motor to keep the HGA beam on target. We define an occlusion around the singularity with sufficient angular radius to ensure \dot{g}_1 remains achievable by the primary gimbal motor. From the geometry we have $\omega_K = H_z$ and

$$\alpha_K = \sin^{-1} \left(\frac{\|\Omega\|}{\dot{g}_{1\max}} \right). \quad (13)$$

For MER, this region is only about $\alpha_K = 0.16^\circ$ in half-angle. Therefore the required \dot{g}_1 exceeds and then becomes less than $\dot{g}_{1\max}$ well before the induced pointing error passes outside the antenna's main lobe (which is about 2°). We thus neglect this occlusion for the purposes of this paper.

4.4 Hardstop Occlusions

The motion limit constraints arise from the range of travel of the gimbals. Because the rover deck occludes the lower half of the celestial sphere while the secondary gimbal can reach the entirety of the remaining upper half, we ignore motion limits from the secondary gimbal and consider only the motion limits of the primary gimbal, g_1 .

While these motion limits—or hardstops—do not prevent the HGA from reaching any part of the sky, not all portions of the sky are reachable on both branches as shown earlier. Accordingly we define *two* wedge occlusions to capture the effects of the hardstops, one for each branch.

4.4.1 Branch A Hardstop Occlusion

We construct the Branch A Hardstop Occlusion to occlude only region III from Figure 3. The wedge $W_{\text{HA}} := H_1 \cap H_2$ is composed of the circles

$$H_1 = \odot \left(\left[\begin{array}{c} -\cos(g_{1\min}) \\ -\sin(g_{1\min}) \\ 0 \end{array} \right], \frac{\pi}{2} \right) \quad (14)$$

and

$$H_2 = \odot \left(\left[\begin{array}{c} \cos(g_{1\max}) \\ \sin(g_{1\max}) \\ 0 \end{array} \right], \frac{\pi}{2} \right). \quad (15)$$

As with the Terrain and Deck Occlusions, the algorithm tests initial point e_0 for membership in W_{HA}

using Eqs. 8 and 9; if e_0 is within the wedge, then the t_{HA} is set to zero.

Otherwise, the algorithm computes the incursion time t_{HA} using intersections of circles. It computes the intersections of the path E with *each* of the constituent circles H_1 and H_2 . Since it is possible that an intersection point between E and, say, H_1 may not be in the wedge (because this intersection point is not also in H_2), the algorithm tests *all* intersections for membership in the wedge. Those that are not members are discarded. If no intersections remain, or if there are no intersections, then the algorithm sets $t_{\text{HA}} = \infty$.

The remaining intersections are sorted in order of increasing angular displacement from the initial point e_0 along the directed curve E . The first intersection is thus the incursion point. The incursion time is then

$$t_{\text{HA}} = \theta_{\text{HA}} / \|\Omega\|, \quad (16)$$

where θ_{HA} is the angular displacement along E from the initial point e_0 to the incursion point.

4.4.2 Branch B Hardstop Occlusion

We likewise construct the Branch B Hardstop Occlusion using two circles to form the wedge $W_{\text{HB}} := H'_1 \cap H'_2$, with

$$H'_1 = \odot \left(\left[\begin{array}{c} \cos(g_{1\min}) \\ \sin(g_{1\min}) \\ 0 \end{array} \right], \frac{\pi}{2} \right) \quad (17)$$

and

$$H'_2 = \odot \left(\left[\begin{array}{c} -\cos(g_{1\max}) \\ -\sin(g_{1\max}) \\ 0 \end{array} \right], \frac{\pi}{2} \right). \quad (18)$$

Thus W_{HB} occludes region I from Figure 3. Note how changing the signs of the cone axes reverses the sense of the wedge.

The same computations done for W_{HA} are repeated for W_{HB} , this time recording the Branch B hardstop incursion time

$$t_{\text{HB}} = \theta_{\text{HB}} / \|\Omega\|. \quad (19)$$

As usual, the incursion time is set to infinity if the path E never intersects the wedge W_{HB} , or to zero if the initial point lies within the wedge.

4.5 Pancam Mast Occlusion

The science camera assembly (known as the pancam mast assembly or PMA) presents a particularly large line-of-sight obstruction to the HGA line-of-sight. The HGA arm can “peer around” the pancam mast, so that in general if one branch is occluded, the other one is clear. Nonetheless, there is a small region of sky that is obstructed on both branches. Due to the structural complexity of the PMA, we use a meld to conservatively model the occlusion.

4.5.1 Constructing the Pancam Melds

To define the melds, we construct models of the pancam assembly, the HGA gimbal, and the HGA beam. We sweep the HGA gimbal model through its range of motion, recording where the beam model intersects the PMA model; these points are marked as occluded. The union of these points defines the meld.

The pancam mast can swivel and the head can tilt, so the PMA obstruction varies with pancam pointing. To simplify the model, we use a worst-case volume of the PMA. This volume is formed by swivelling the pancam assembly in both of its gimbal axes, forming a “lollipop” that encloses the PMA. We model the beam emanating from the HGA as a cylinder emerging from the radiating face of the HGA disk. Figure 6 illustrates the complete model at gimbal angles $(g_1, g_2) = (60^\circ, 60^\circ)$.

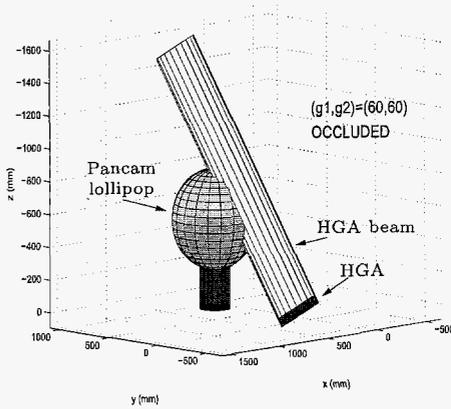


Figure 6: Pancam/HGA interference model.

The HGA is driven through all the *Branch A* gimbal angles (ignoring hardstops), driving the boresight around the unit sphere. At each boresight location on

the sphere, we record an “X” if any part of the cylindrical beam passes through any part of the pancam lollipop; we record an “O” where the beam clears the lollipop. We then draw circles C_i on the unit sphere so that the union of these circles contains all the X’s (and as few O’s as possible). This union defines the meld $M_{PA} = C_1 \cup C_2 \cup \dots \cup C_n$.

We repeat the process for Branch B, generating the meld $M_{PB} = C'_1 \cup C'_2 \cup \dots \cup C'_n$. Figure 7 shows the resultant occlusions as drawn on the unit sphere.

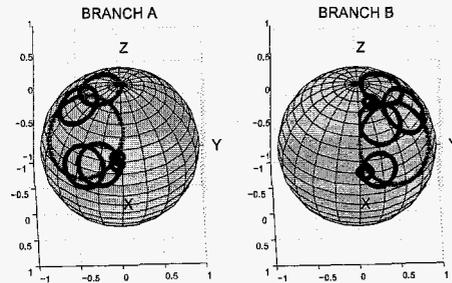


Figure 7: Pancam Melds.

This approach has two advantages: the HGA algorithm does not require knowledge of pancam’s relative orientation, and the science operations team can use the pancam without interfering with HGA communication.

4.5.2 Using the Pancam Occlusions

Based on the pancam meld M_{PA} , the pointing algorithm computes the Earth incursion time t_{PA} as follows. The algorithm first tests the initial point e_0 for membership in M_{PA} using Eq. 10; if e_0 is within the meld, then we set $t_{PA} = 0$.

Next the algorithm computes the intersections of the Earth path E with each of the constituent circles in the meld, and these intersections are sorted in order of increasing angular displacement from the initial point e_0 along E . The first intersection in the list is the incursion point. The corresponding incursion time is then

$$t_{PA} = \theta_{PA} / \|\Omega\|. \quad (20)$$

The algorithm repeats this procedure for the Branch B Pancam Occlusion M_{PB} to compute an incursion time t_{PB} .

5 Execution Strategy

We now describe the final two steps of the HGA pointing algorithm: branch selection at the start of an HGA communication session, and the handling of occlusions as they are encountered in midtrack. We then provide an example.

5.1 Branch Selection

At the beginning of a communication session, the HGA pointing algorithm computes the incursion times for each of the occlusions and then computes the overall incursion time for each branch from

$$\begin{aligned} t_A &= \min(t_T, t_D, t_{HA}, t_{PA}), \\ t_B &= \min(t_T, t_D, t_{HB}, t_{PB}). \end{aligned} \quad (21)$$

Note that the Terrain and Deck Occlusions are common to both branches. Any of the occlusions can be disabled by setting its incursion time to infinity, which has the effect of making the HGA pointing algorithm ignore that occlusion.

The algorithm selects whichever branch provides the longest possible time until incursion, and begins tracking the Earth using that branch. If both branches return zero time (for example, the Earth is currently below the deck), then the HGA communication session is aborted, and the onboard communications manager takes appropriate action to contact Earth through other means—typically by reverting to a session on the low-gain antenna.

If both branches return the same time ($t_A = t_B$), which can occur for late afternoon communication sessions in which a relatively low Earth will encounter the Terrain or Deck Occlusions first, then the algorithm defaults to a pre-selected branch.

The branch selection step occurs only at the *start* of the communication session. Once the HGA pointing algorithm has selected the branch and started tracking the Earth with the HGA, the algorithm remains committed to that branch, *even if the alternate branch later becomes unobstructed and would allow longer tracking from that time forward*. This approach is required because the duration of the communication session is not known by the HGA pointing algorithm. Without this rule, the algorithm would immediately flop to the alternate branch—thus disrupting a communication session that might have been scheduled to end before the current branch became occluded.

5.2 Handling Occlusions in Midtrack

If the *current* branch becomes occluded during the course of the track, then the algorithm takes one of the following actions, depending on which occlusion is encountered.

- If the occlusion just encountered is the Terrain or Deck Occlusion, then the communication session is immediately terminated.
- If the occlusion encountered is the Hardstop Occlusion for the current branch, then the algorithm performs a flop to the alternate branch.
- If the occlusion encountered is the Pancam Occlusion for the current branch, then (for operational reasons) the algorithm continues to track. The algorithm alerts the communications manager to send only realtime health and status data. When the Pancam Occlusion is cleared, the HGA algorithm informs the communications manager that it may resume normal transmission.

5.3 Example

Figure 8 presents an example. The rover is flat and level on the Martian surface, so that the Terrain and Deck Occlusions overlap and occlude the lower hemisphere. The Hardstops Occlusions W_{HA} and W_{HB} are indicated as shaded regions on the celestial sphere. The Pancam Occlusions are disabled and are not shown.

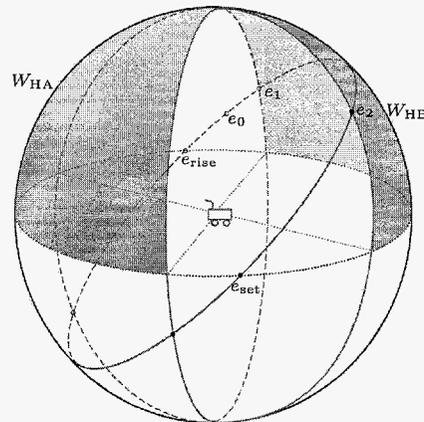


Figure 8: Pointing Strategy Example

The Earth trajectory as a function of time is shown starting at position e_0 at time t_0 , entering wedge W_{HB} at e_1 and exiting at e_2 . The Earth sets before entering the wedge W_{HA} . Thus

$$\begin{aligned} t_A &= t_{\text{set}} \\ t_B &= t_1 \end{aligned} \quad (22)$$

Since $t_A > t_B$, the HGA begins tracking on Branch A and can continue until $t = t_{\text{set}}$.

6 Conclusions

We have presented the HGA pointing algorithm for the Mars Exploration Rovers. The algorithm maximizes the communications time available to rovers when using the HGA, subject to multiple pointing constraints. This increases the operability of the vehicles and thus increases their science return. Furthermore, a ground-tool version of the algorithm has been developed to predict interference from the pan-cam assembly and to recommend that, where possible, the onboard algorithm be allowed to take action to avoid the occlusion, or that the rover be driven and parked in such a way to avoid such interference. The latter practice is standard procedure for new flight software uploads, which require very long, flop-free communication sessions.

The algorithm has been proven in flight: since both MER vehicles landed in January 2004, not one flop has been necessary on either vehicle during the entire mission, which at the time of writing now exceeds 12 months.

7 Acknowledgements

This work was prepared at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

- [1] C. Anthony Vanelli, *Autonomous Reorientation of a Maneuver-Limited Spacecraft Under Simple Pointing Constraints*. Ann Arbor, Michigan: University Microfilms, 1997.
- [2] Richard M. Murray, Zexiang Li, S. Shankar Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Ann Arbor, 1994.

[3] Erwin Kreyszig, *Advanced Engineering Mathematics*, John Wiley & Sons, New York, 1967.

[4] Peter C. Hughes, *Spacecraft Attitude Dynamics*, John Wiley & Sons, New York, 1986.