High Gain Antenna Pointing on the Mars Exploration Rovers

C. Anthony Vanelli
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA, USA
vanelli@jpl.nasa.gov

Khaled S. Ali
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA, USA
khaled.s.ali@jpl.nasa.gov

Abstract — *This paper describes the algorithm used to point the high gain antennae on NASA/JPL's Mars Exploration Rovers. Each rover's gimballed antenna must track the Earth as it moves across the Martian sky during communication sessions. The pointing algorithm accounts for obstacles to the line-of-sight posed by (1) features on the rover and in the surrounding environment, (2) gimbal range limitations, and (3) kinematic singularities in the gimbal mechanism. The algorithm treats all obstacles with a generalized approach that computes the intercept-times to each obstacle. Where possible, the algorithm takes advantage of pairs of joint-space solutions arising from the mechanism design. The algorithm chooses the solution that provides the longest obstruction-free tracking time. Upon encountering an obstacle, the algorithm automatically switches to the other solution if it is not also obstructed. This algorithm has successfully provided obstruction-free pointing for both rovers throughout the mission.*

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

1 Introduction

The Mars Exploration Rover (MER) communications system consists of three basic elements: a fixed-position low-gain antenna (LGA) 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 realtime communication at high data rates.

Experience from the Mars Pathfinder (MPF) mission showed that its HGA gimbal was vulnerable to "flops" when the Earth was high in the Martian sky. The MPF gimbal was mounted on the stationary lander, with its primary axis horizontal. Figure 1 depicts a typical flop. The HGA is shown enclosed within the celestial hemisphere. The Earth trajectory on the sky is shown with a dashed arc and key positions are numbered 1–4; the respective gimbal positions are labelled *a*–*d* and shown off to the side. As the Earth rose from position 1 to position 2 the gimbal would drop from position *a* to position *b*—the furthest the gimbal could depress. When the Earth proceeded to the neighboring position 3, both gimbal axes would rotate through 180°—temporarily taking the beam off Earth and breaking the link—in order to reacquire "from the flip side" at position *c*. Orbital mechanics compounded the issue: earthrise (as seen from Mars) occurred well before sunrise for several months after landing. By the time the solar-powered vehicle had sufficient power for communication, the Earth was often near zenith—with the HGA almost fully depressed and about to flop. The resulting interruptions were extremely disruptive to mission operations.

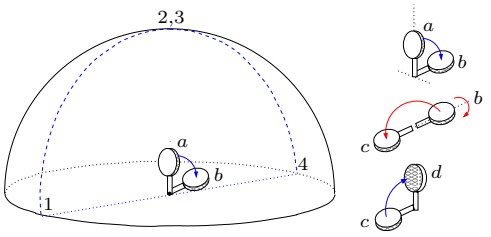

Figure 1: Mars Pathfinder HGA flops.

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 gimbal 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 obstruct the line-of-sight. Some obstructions 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).

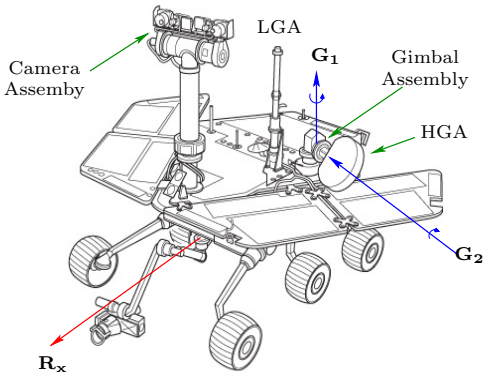

Figure 2: MER HGA and deck obstructions.

A pointing algorithm was designed to address these challenges. The HGA pointing algorithm (1) minimizes the occurence 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.

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

Figure 3 depicts the HGA gimbal assembly. There are two gimbal axes $\mathbf{G_1}$ and $\mathbf{G_2}$. The primary axis $\mathbf{G_1}$ is normal to the deck and the secondary axis $\mathbf{G_2}$ is parallel to the deck. The antenna disk is mounted at the end of a short arm and the beam radiates perpendicular to the disk. The vector $\mathbf{R_x}$ points toward the front of the rover. The HGA gimbal frame $\mathcal{F}_{\mathrm{H}}$ has its origin at the intersection of the gimbal axes, with basis vector $\mathbf{H_z}$ along axis $\mathbf{G_1}$ and basis vector $\mathbf{H_x}$ directed 30° to the right of $\mathbf{R_x}$. The reference position $(g_1, g_2) = (0, 0)$ places the arm along the $\mathbf{H_x}$ axis and points the beam along the $-\mathbf{H_y}$ axis. Note that $g_2 = 90°$ points the beam vertically up. All subsequent computations are in the HGA frame unless explicitly indicated otherwise.

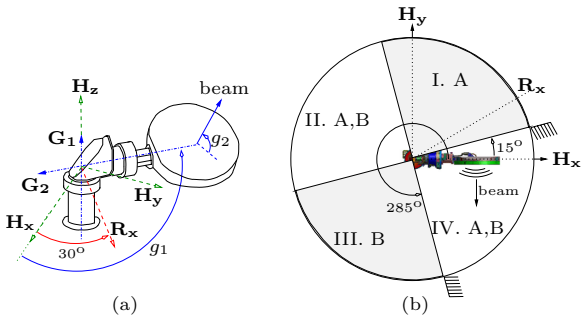

Figure 3: (a) Gimbal assembly and (b) admissible branches.

There are two joint-space configurations to point the antenna at a given target $\mathbf{e} = [e_x, e_y, e_z]^T$ in the HGA frame. The first solution, which we call Branch A, is

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

The second solution, which we call Branch B, is

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

The Branch A solution returns $g_2 \in [-90°, 90°]$ while the Branch B solution returns $g_2 \in [90°, 270°]$. The notion of "branch" is undefined when $g_2 = \pm 90°$; 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 $g_1 \in [15°, 285°]$. The secondary gimbal is software-limited to $g_2 \in [0°, 180°]$. 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 $\mathbf{e}$ lies in region III the only joint-space solution that does not 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. Using the unit celestial sphere to represent the set of all unit vectors, we map each constraint onto the celestial sphere using circles; this mapping defines

an *occlusion*. We also approximate the trajectory of the Earth on the 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 $\boldsymbol{\omega}$ and a point $\mathbf{p}_0$ on the circle, any spherical circle can be expressed parametrically with

$$\mathbf{p}(\varphi) = \exp(\varphi \boldsymbol{\omega}^{\times})\, \mathbf{p}_0, \tag{3}$$

as φ ranges from 0 to 2π. From [2], the matrix exponential $\exp(\varphi\boldsymbol{\omega}^{\times})$ is a rotation matrix that rotates vectors about $\boldsymbol{\omega}$ through an angle φ. The notation $\boldsymbol{\omega}^{\times}$ denotes the constant skew-symmetric matrix formed from the elements of $\boldsymbol{\omega}$:

$$\boldsymbol{\omega}^{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{4}$$

Since from the rover's point of view Mars is fixed and Earth is moving, the cone axis $\boldsymbol{\omega}_E = -(\boldsymbol{\Omega}/\|\boldsymbol{\Omega}\|)$, where $\boldsymbol{\Omega}$ is the angular velocity of Mars. Given an initial position $\mathbf{e}_0$ and setting $\varphi = (\,\|\boldsymbol{\Omega}\|\, t)$, the Earth trajectory is

$$\mathbf{e}(t) = \exp\left(\,\|\boldsymbol{\Omega}\|\, t\, (\boldsymbol{\omega}_E)^{\times}\right)\, \mathbf{e}_0. \tag{5}$$

The resultant circle E is shown in Figure 4.

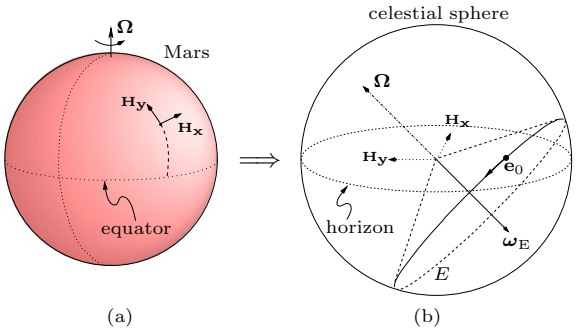

Figure 4: (a) The HGA frame on rotating Mars and (b) the resulting Earth circle E in the celestial sphere.

3.2 Occlusion Circles

We also use circles on the celestial 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* $C := \circledcirc(\boldsymbol{\omega}, \alpha)$ shown in Figure 5. The circle C has cone axis $\boldsymbol{\omega}$ and half-angle α. The center point of this circle is $\mathbf{c} = \boldsymbol{\omega}\cos(\alpha)$. Note that $\mathbf{c}$ lies *within* the unit sphere.

By convention we take the generating axis $\boldsymbol{\omega}$ to be directed *towards* the region of the sphere that is occluded. Thus $\mathbf{p}(\varphi)$ from Equation 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 $\mathbf{q}$ is occluded if it is a member of the region enclosed by C. The membership test is

$$\begin{array}{lcll} (\mathbf{q}-\mathbf{c})\cdot\boldsymbol{\omega} & \geq & 0 & \Rightarrow \mathbf{q} \text{ is occluded.} \\ (\mathbf{q}-\mathbf{c})\cdot\boldsymbol{\omega} & < & 0 & \Rightarrow \mathbf{q} \text{ is not occluded.} \end{array} \tag{6}$$

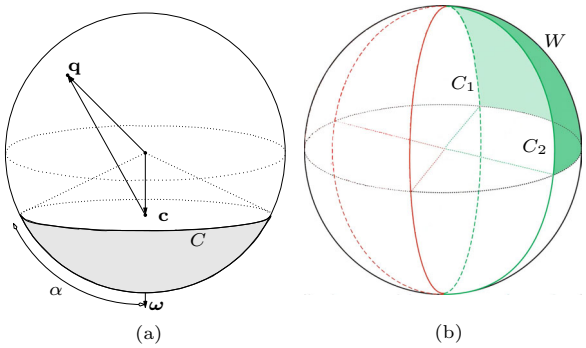

Figure 5: (a) Occlusion circle C and (b) Wedge W.

A *wedge occlusion* is constructed from the intersection of two circles $C_1 := \circledcirc(\boldsymbol{\omega}_1, \alpha_1)$ and $C_2 := \circledcirc(\boldsymbol{\omega}_2, \alpha_2)$. Figure 5 shows the wedge $W := C_1 \cap C_2$; the lower half of the wedge has been omitted for clarity. A point $\mathbf{q}$ 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 $\mathbf{q}$ that satisfy *both*

$$(\mathbf{q}-\mathbf{c}_1)\cdot\boldsymbol{\omega}_1 \geq 0, \tag{7}$$

$$(\mathbf{q}-\mathbf{c}_2)\cdot\boldsymbol{\omega}_2 \geq 0. \tag{8}$$

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

$$(\mathbf{q}-\mathbf{c}_i)\cdot\boldsymbol{\omega}_i \geq 0, \text{ for } i = 1, \ldots, n. \tag{9}$$

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 corresponding incursion time. We present these occlusions in order of increasing complexity.

4.1 Terrain Occlusion

The Terrain Occlusion models the local horizon; 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 $\boldsymbol{\omega}_{\mathbf{T}}$ pointing at the center of Mars and $\alpha_{\mathrm{T}} = \pi/2$. The occluded portion of the celestial sphere represents the ground.

For a given Earth start position $\mathbf{e}_0$, the pointing algorithm computes the incursion time t_{T}—the time until the Earth passes into the occlusion circle T. First it verifies that $\mathbf{e}_0$ 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 the 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 $\mathbf{e}_0$. 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 $\mathbf{e}_0$ to the intersection(s). The incursion time is thus

$$t_{\mathrm{T}} = \theta_{\mathrm{T}} / \|\boldsymbol{\Omega}\|. \tag{10}$$

4.2 Deck Occlusion

The Deck Occlusion models the Rover Equipment Deck, which presents an obstacle if the rover is tilted while the Earth is low in the sky. 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 $\boldsymbol{\omega}_{\mathbf{D}} = -\mathbf{H}_{\mathbf{z}}$, and half-angle $\alpha_{\mathrm{D}} = \pi/2$. The occluded portion of the local celestial sphere represents the sky below the deck. The algorithm computes the incursion time t_{D} in the same manner as t_{T}, i.e., by finding the shortest displacement θ_{D} and taking $t_{\mathrm{D}} = \theta_{\mathrm{D}} / \|\boldsymbol{\Omega}\|$.

4.3 Kinematic Singularity

The HGA gimbal has a singularity at $\mathbf{e} = [0, 0, 1]^T$ (where $g_2 = 90^{\mathrm{o}}$). For Earth trajectories near this point, the required azimuthal tracking speed $\dot{g}_1$ momentarily exceeds the primary gimbal motor's maximum speed $\dot{g}_{1\max}$, causing a temporary pointing error. We define an occlusion around the singularity with sufficient angular radius to ensure $\dot{g}_1$ remains achievable. From the geometry we have $\boldsymbol{\omega}_{\mathbf{K}} = \mathbf{H}_{\mathbf{z}}$ and $\alpha_{\mathrm{K}} = \operatorname{asin}\left(\|\boldsymbol{\Omega}\| \,/\, \dot{g}_{1\max}\right)$.

For MER, α_{K} is only about 0.16^{o}. However the main lobe of the antenna is 2^{o} wide, which is easily large enough to compensate for any induced pointing error. We thus neglect this occlusion for the purposes of this paper.

4.4 Hardstop Occlusions

The motion limit constraints arise from the travel limits of the gimbals. 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 a single branch. Accordingly we define *two* wedge occlusions to capture the specific effects of the hardstops on 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 $W_{\mathrm{HA}} := H_1 \cap H_2$ is composed of two circles

$$H_1 = \circledcirc\left(\begin{bmatrix} -\cos(g_{1\min}) \\ -\sin(g_{1\min}) \\ 0 \end{bmatrix}, \frac{\pi}{2}\right), \text{ and} \tag{11}$$

$$H_2 = \circledcirc\left(\begin{bmatrix} \cos(g_{1\max}) \\ \sin(g_{1\max}) \\ 0 \end{bmatrix}, \frac{\pi}{2}\right). \tag{12}$$

As with the Terrain and Deck Occlusions, the algorithm tests the initial point $\mathbf{e}_0$ for membership in W_{HA} using Equations 7 and 8; if $\mathbf{e}_0$ is within the wedge, then the incursion time t_{HA} is set to zero.

Otherwise, the algorithm computes 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 $t_{\mathrm{HA}} = \infty$.

The remaining intersections are sorted in order of increasing angular displacement from the initial point $\mathbf{e}_0$ along the directed curve E. The first intersection is thus the incursion point, with angular displacement θ_{HA} along E from the initial point $\mathbf{e}_0$. The incursion time is then $t_{\mathrm{HA}} = \theta_{\mathrm{HA}} / \|\boldsymbol{\Omega}\|$.

4.4.2 Branch B Hardstop Occlusion

We likewise construct the Branch B Hardstop Occlusion using two circles to form the wedge $W_{\mathrm{HB}} := H_1' \cap H_2'$, with

$$H_1' = \circledcirc\left(\begin{bmatrix} \cos(g_{1\min}) \\ \sin(g_{1\min}) \\ 0 \end{bmatrix}, \frac{\pi}{2}\right), \text{ and} \tag{13}$$

$$H_2' = \circledcirc\left(\begin{bmatrix} -\cos(g_{1\max}) \\ -\sin(g_{1\max}) \\ 0 \end{bmatrix}, \frac{\pi}{2}\right). \tag{14}$$

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 $t_{\mathrm{HB}} = \theta_{\mathrm{HB}} / \|\boldsymbol{\Omega}\|$.

4.5 Pancam Mast Occlusion

The science camera assembly (known as the pancam mast assembly or PMA) presents a particularly large 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 $(g_1, g_2) = (60^o, 60^o)$.

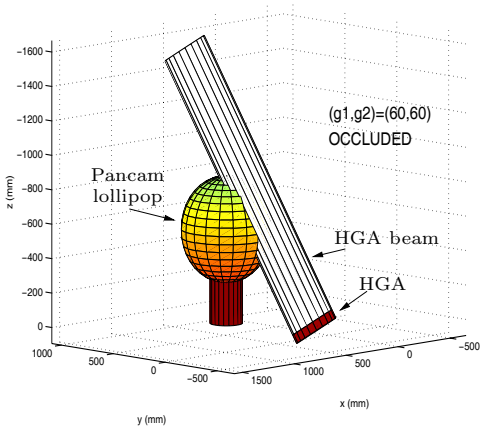

Figure 6: Pancam/HGA interference model.

The HGA model is driven through all the *Branch A* gimbal angles (ignoring hardstops), taking 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" if 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 $M_{\mathrm{PA}} = C_1 \cup C_2 \cup \cdots \cup C_n$.

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

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

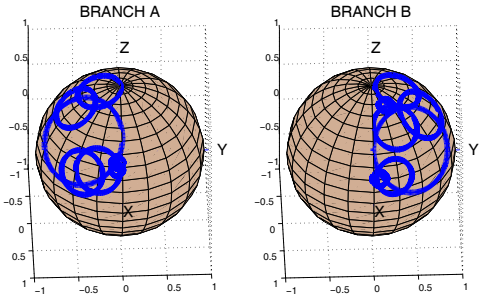

Figure 7: Pancam Melds.

4.5.2 Using the Pancam Occlusions

Based on the Branch A Pancam Occlusion M_{PA}, the pointing algorithm computes the Earth incursion time t_{PA} as follows. The algorithm first tests the initial point $\mathbf{e}_0$ for membership in M_{PA} using Equation 9; if $\mathbf{e}_0$ is within the meld, then the algorithm sets $t_{\mathrm{PA}} = 0$.

Otherwise 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 $\mathbf{e}_0$ along E. The first intersection in the list is the incursion point with displacement θ_{PA}. The corresponding incursion time is then $t_{\mathrm{PA}} = \theta_{\mathrm{PA}} / \|\boldsymbol{\Omega}\|$.

We repeat 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. 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 using

$$\begin{aligned} t_{\mathrm{A}} &= \min(t_{\mathrm{T}}, t_{\mathrm{D}}, t_{\mathrm{HA}}, t_{\mathrm{PA}}), \\ t_{\mathrm{B}} &= \min(t_{\mathrm{T}}, t_{\mathrm{D}}, t_{\mathrm{HB}}, t_{\mathrm{PB}}). \end{aligned} \tag{15}$$

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 (e.g., the Earth is currently below the deck), then the HGA communication session is aborted, and the onboard communications manager takes 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 ($t_A = t_B$), then the algorithm defaults to a pre-programmed branch.

The branch selection step occurs only at the *start* of the communication session. Once the HGA pointing algorithm has selected a branch and started tracking 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 pointing algorithm. Without this rule, the algorithm would immediately flop to the alternate branch—and disrupt a communication session that might have ended before the current branch would have become occluded.

5.2 Handling Occlusions in Midtrack

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

- **Terrain or Deck Occlusion:** The communication session is immediately terminated.
- **Hardstop Occlusion:** The algorithm performs a flop to the alternate branch.
- **Pancam Occlusion:** Rather than flop, for operational reasons the algorithm continues to track but 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 its normal transmission behavior.

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.

The Earth trajectory as a function of time is shown starting at position $\mathbf{e}_0$ at time t_0, entering wedge W_{HB} at $\mathbf{e}_1$ and exiting at $\mathbf{e}_2$. The Earth sets before entering the wedge W_{HA}. Thus

$$\begin{aligned} t_A &= t_{set} \\ t_B &= t_1 \end{aligned} \tag{16}$$

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

6 Conclusions

We have presented the HGA pointing algorithm for the Mars Exploration Rovers. The algorithm maximizes the communications time available to the rovers when using the HGA, subject to multiple pointing constraints.

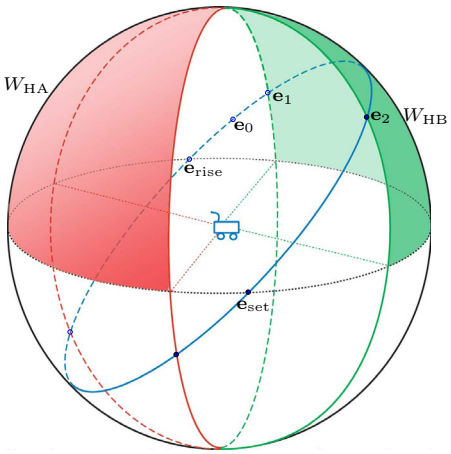

Figure 8: Pointing Strategy Example

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 pancam assembly and to recommend that 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 the interference where possible. 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 approaches 18 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.