A Novel Attitude Guidance Algorithm for Exclusion Zone Avoidance

Jesse D. Koenig
SpaceDev, Inc.
13855 Stowe Dr.
Poway, CA 92064
858-375-2032
jesse@spacedev.com

Abstract—The Trailblazer spacecraft was designed to use an optical sensing instrument to track objects moving through space. This instrument would be damaged if pointed to within a certain angle of the sun. Therefore, a novel attitude guidance algorithm was developed to avoid pointing the instrument within this exclusion zone, while minimizing disruptions to target-tracking. Traditional obstacle-avoidance and path-planning algorithms do not suffice because such methods generally do not provide for adherence to an intrinsic desired continuous path. The algorithm's computational demands do not present a problem for Trailblazer, owing to its high-performance computer. [1,2]

TABLE OF CONTENTS

1. INTRODUCTION

Trailblazer, a high-performance microsatellite designed and built by SpaceDev, Inc., was launched on a SpaceX Falcon 1 on August 2nd, 2008 as the primary payload of the Jumpstart mission, sponsored by the U.S. Office of Operationally Responsive Space. Unfortunately, the mission was lost due to a launch failure.

Trailblazer was designed to use an optical sensing instrument to track objects moving through space. It was built according to SpaceDev's philosophy of leveraging Commercial Off-The-Shelf (COTS) technology wherever possible to obtain high performance at low cost. Thus, Trailblazer's main computer was quite powerful by spacecraft standards, with 1856 MIPS @ 800 MHz. Trailblazer also had a capable suite of Guidance, Navigation, and Control (GNC) hardware, including a star tracker, an Inertial Measurement Unit (IMU) containing fiber-optic gyros, a 3-axis magnetometer, a GPS receiver, 4 reaction wheels, and 3 magnetic torque rods. Meanwhile, the ample processing capacity allowed for implementation of sophisticated GNC algorithms, giving Trailblazer even higher performance than would be typically commensurate with its hardware package.

For example:

1. In the area of attitude determination, a more expensive IMU and star tracker would have given lower attitude uncertainties, but Trailblazer achieved this result by employing a 19-state Extended Kalman Filter (EKF).
2. For communications with other spacecraft flying in formation, more expensive antennas and/or higher-power radios would have given omni-directional coverage, but Trailblazer instead used an attitude guidance algorithm to maintain the inter-satellite link by pointing antennas through control of the single degree of attitude freedom available while the instrument was acquiring or tracking a target.
3. The attitude guidance algorithm that was used to optimize inter-satellite communication has another component that maximized the star tracker's utility by keeping it pointed away from the Earth, Moon, and Sun during instrument tracking or acquisition, thereby obviating the need for an expensive two-headed star tracker system.
4. In slewing to acquire a target, traditional attitude control methods would have been able to achieve sufficiently short acquisition times with large enough reaction wheels (or control moment gyros), but a slew guidance algorithm allowed smaller, less expensive wheels to do the job.
5. Expensive modifications could have been made to the instrument to prevent its sensor from being damaged if pointed close to the sun, but instead an attitude guidance algorithm was developed to avoid sun-pointing while minimizing target-tracking disruptions.

This paper describes the sun avoidance attitude guidance algorithm cited in example #5 above.

1

[1] 978-1-4244-2622-5/09/$25.00 ©2009 IEEE.
[2] IEEEAC paper #1456, Version 1, Updated October 28, 2008

2. BACKGROUND

It may be desirable for a spacecraft or some other body or vehicle in space, air, water or on land to track an object that is moving relative to the body, where tracking is defined as maintaining a body axis (such as the boresight of a sensing instrument) pointed at the object. In some of these cases there may be additional axis-pointing constraints that supersede the tracking goal, such as a need to avoid pointing at a predetermined object or location. Specifically, Trailblazer's optical instrument would have been damaged if its boresight were pointed too close to the sun; therefore the boresight had to be prevented from pointing within an exclusion zone, defined as the set of vectors that are less than a certain angle from the sun vector. (Trailblazer's instrument was fixed to the body of the spacecraft, and therefore was pointed by controlling the spacecraft's attitude.)

The target object may have a spatial trajectory relative to that of the vehicle such that to track the target continuously, the boresight has to pass through the exclusion zone. If the highest priority is to prevent the boresight from pointing within the exclusion zone, then the tracking goal must be temporarily sacrificed in this situation. In robotic applications wherein the vehicle's guidance navigation and control (GNC) are handled autonomously without a human in the loop, there is desirably an automatic attitude guidance algorithm to achieve the exclusion zone avoidance.

Thus, the problem is one of autonomous obstacle avoidance, an area that has received significant attention in prior work. Typical approaches using potential field functions [1] and iterative randomized path-planning algorithms [2], [3] generally accomplish transition from an initial state to a final goal state while avoiding collisions with obstacles. Such methods are quite capable in the appropriate applications, able to handle time-variance of both obstacles and goal states. However, they do not generally provide for adherence to an intrinsic desired path between the initial and goal states. In contrast, the tracking application at hand has a fully defined, continuous desired pointing trajectory, which should be followed to within a fraction of a degree unless proscribed by exclusion zone avoidance constraints.

Several novel approaches have been previously developed for specialized applications. Hablani [4] has described a spacecraft attitude guidance algorithm that simultaneously avoids pointing within exclusion zones and maintains antenna pointing to allow ground communication, while performing a slew. Again though, the slews considered in that application are defined by initial and goal states, as opposed to continuous trajectories. Also, the algorithm primarily employs a versine function to generate the avoidance path around an exclusion zone, but that path may in fact intersect the exclusion zone, in which case, it is simply replaced by the exclusion zone boundary. It seems that this discrete change in the path would create a commanded rate discontinuity that would cause an attitude error, likely resulting in an exclusion zone incursion. Meanwhile, generally a spacecraft's dynamic limits will not allow realization of a trajectory that both follows the exclusion zone boundary exactly and keeps up with the desired pointing trajectory so as to merge smoothly after the exclusion zone is passed. Furthermore, even in the case that the versine path does completely clear the exclusion zone, its shape is quite different from that of a conical exclusion zone boundary, causing greater deviation from the nominal path and thus greater deviation duration than is desired in the present application. The algorithm described herein generates an avoidance path that can more closely hug the exclusion zone boundary, thereby minimizing the deviation duration. Finally, it appears that the referenced algorithm is not dependent on the angular rate at which the desired pointing approaches the exclusion zone. In the present application, it is critical that the avoidance path is a function of this rate, to preclude both exclusion zone incursion and unnecessarily high deviation duration, as will be explained more fully subsequently.

Singh et al. [5] have described an algorithm designed for the Cassini spacecraft, to monitor and modify as necessary the nominal attitude commanding to avoid violation of pointing constraints. This algorithm is able to be applied to complex multi-segment slews, but still, each segment comprises an initial state and a goal state, not consistent with the problem at hand in the present work. Perhaps the referenced algorithm could be extended for the present application, such that at each time step, the algorithm's goal state comprised simply the current commands for attitude and angular rate. Still, that algorithm escapes exclusion zone incursion by accelerating directly away from the center of the exclusion zone, which would often mean accelerating largely against the direction of the desired slew's angular rate; this would cause the avoidance pointing to fall behind the desired pointing, leading to unnecessarily high deviation duration. In addition, to generate goal attitudes during avoidance maneuvers, the referenced algorithm uses an iterative process which does not always find a solution, leading to commanded attitude discontinuities that would be highly undesirable in the present application.

With regard to the present application, a relatively simple algorithm would be as follows: for each target-tracking attitude command that would point the boresight inside the exclusion zone, modify that command such that the boresight points along the vector which is as close as possible to the original vector while also being outside the exclusion zone. For a target trajectory requiring the boresight to pass through the exclusion zone, this algorithm would result in a commanded attitude profile (a set of attitude commands over time) for which the boresight-pointing approached the exclusion zone right up to the boundary, traced the boundary around to the point where the target-tracking pointing would exit the exclusion zone, and then resumed tracking. However, this algorithm would

not suffice. The actual vehicle has dynamic limits, that is, limits on its capabilities for angular rates and angular accelerations. The algorithm could cause high commanded rates, and would certainly cause high commanded accelerations at the intersections with the exclusion zone boundary, exceeding the vehicle's dynamic limits. Then, while the vehicle's feedback control system would attempt to make the vehicle follow the commands, it would not be able to, and the boresight would enter the exclusion zone.

To successfully avoid the exclusion zone, the avoidance maneuver must be initiated before the point where the tracking-pointing would enter the exclusion zone. Also, the faster the tracking-pointing is approaching the exclusion zone, the further in advance the maneuver must be initiated in order to generate a pointing profile that avoids the exclusion zone, but remains within the vehicle's dynamic limits. So then, another algorithm could be designed with a conservative strategy such that the boresight avoided the exclusion zone always using a pointing path that would remain within dynamic limits for the maximum possible approach rate (the maximum possible angular rate at which the tracking-pointing could approach the exclusion zone), and thus for all possible approach rates. However, this is an unfavorable strategy because although the highest priority is to keep the boresight out of the exclusion zone, it is also a high priority to minimize the duration of compromised tracking. This conservative method would cause unnecessarily long durations of compromised tracking for most avoidance maneuvers.

Also, a predictive algorithm could be designed which attempts to propagate the target-pointing into the future, and calculate an avoidance maneuver based on some optimization rules. This approach is unfavorable due to both its general complexity, and the difficulty of propagating the target position based only on two-dimensional (2-D) measurements.

Thus, a novel attitude guidance algorithm was developed for Trailblazer's application, using only assessment of the present situation at any given time, without propagation, to perform an exclusion zone avoidance maneuver. The algorithm assumes continuous input of the target-tracking attitude commands and modifies these such that the resulting attitude command profile avoids pointing the boresight into the exclusion zone, remains within the vehicle's dynamic limits, and minimizes the duration of compromised tracking.

3. ALGORITHM OVERVIEW

At each time step, Trailblazer's sun avoidance algorithm examines the current instrument boresight pointing command and spacecraft body angular rate command (which come from a separate target-tracking or target acquisition algorithm), and determines whether the boresight is headed toward an incursion into (or already in) the sun exclusion zone. If so, the algorithm applies a modification to the boresight pointing command, rotating it by some angle in a direction normal to that of the rate command. In this way, the algorithm essentially steers the boresight around the exclusion zone and then back onto the target-tracking profile, analogously to a car driver steering around an obstacle in the road.

Figure 1 shows representations of four hypothetical cases, each of which is illustrated by a flattened portion of an attitude sphere, as follows: The spacecraft is located at the center of the sphere. The lines represent the intersection with the sphere of particular sets of pointing vectors originating from the spacecraft, that is, the center of the sphere. The X and Y axes measure the coordinates of an angular position relative to the center of the exclusion zone. X is in the direction of the cross-boresight rate (the angular rate minus any rate components about the boresight axis) required for target tracking, and Y is in the direction perpendicular to X. The sun avoidance algorithm applies an offset only in the Y direction to the target-tracking boresight pointing command.

The circle represents the boundary of the exclusion zone. The dashed line represents the boresight-pointing that would be required to track the target. The heavy line represents the boresight-pointing that is commanded by the algorithm. In each of the four cases, continuous target tracking would cause the boresight to pass through the exclusion zone. Therefore, in each case, the algorithm applies modifications to the tracking-pointing commands, such that target tracking is temporarily compromised in order to steer the boresight around the exclusion zone. These modifications are generated via a formulation incorporating nested Gaussian functions, as will be described explicitly subsequently.

In Figure 1, case A, a high cross-boresight rate is required to track the target. Based on the high approach rate, the exclusion avoidance algorithm begins modifying the pointing commands far from the exclusion zone.

In case B, the approach rate is 1/5 of that in case A, and the algorithm therefore allows the boresight to get much closer to the exclusion zone before modifying the pointing commands.

It is important to note that with the approach rate of case A, if the pointing path of case B were attempted, the required angular rates and accelerations would exceed the spacecraft's dynamic limits. Then the spacecraft would not be able to follow the commands, which would cause the boresight to overshoot the desired path and enter the exclusion zone. Conversely, with the approach rate of case B, the spacecraft would easily be able to follow the pointing path of case A. However, in that scenario it would be deviating from target-pointing for an unnecessarily long duration.

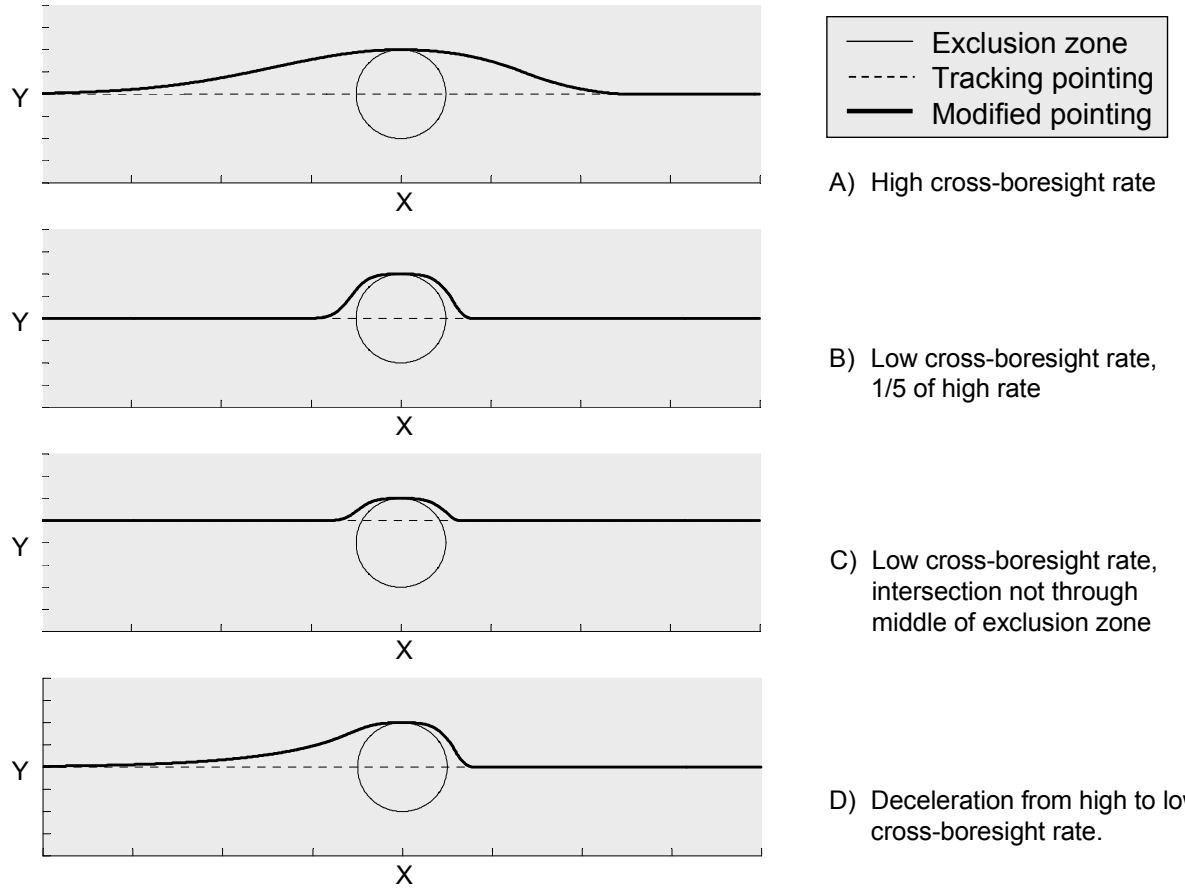

Figure 1: Hypothetical Exclusion Zone Avoidance Cases

In case C, the cross-boresight rate is the same as in case B, but the pointing trajectory crosses through the exclusion zone at a quarter of the zone's diameter, instead of through the center. Therefore, the maximum required deviation from target tracking is smaller. The exclusion avoidance algorithm automatically accounts for this difference, guiding a pointing trajectory that avoids the exclusion zone, stays within the spacecraft's dynamic limits, and does not deviate further than necessary from target tracking.

In case D, the target-tracking pointing decelerates from a high approach rate on the far left side of the plot, equal to that of cases A and C, to a low rate on the right side of the plot, equal to that of case B. The algorithm adapts to the changing rate: the modified pointing commands deviate from target-tracking well in advance of the exclusion zone when the cross-boresight tracking rate is high, so that the spacecraft won't overshoot the commands and point the boresight into the exclusion zone; then as the cross-boresight tracking rate decreases, the pointing commands follow a tighter trajectory around the exclusion zone because with the lower cross-boresight rates this tighter trajectory is within the spacecraft's dynamic limits, and it allows target tracking to resume quickly.

4. DETAILED ALGORITHM DESCRIPTION

Figure 2 shows a flow diagram of the exclusion zone avoidance algorithm. The diagram illustrates a process that is executed at every time step for which an attitude command is given to the Attitude Control System (ACS), whenever the algorithm is enabled. There may be certain operational modes of the spacecraft in which the exclusion zone is not a concern and for which the algorithm is therefore disabled.

It is important to recognize the distinction between a time step and a numbered algorithm step. The entire algorithm is executed within every time step, whereas a step named with a number represents one of the many sequential operational elements that comprise the algorithm.

To facilitate discussion, the term "target-tracking" may be used to express the nominal desired boresight-pointing that does not account for exclusion zones. This nominal desired pointing in fact may be that which maintains instrument-tracking of a target object, but it may comprise some other pointing profile, for example, a spacecraft slew performed to acquire the target (that is, bring it within the instrument's field of view).

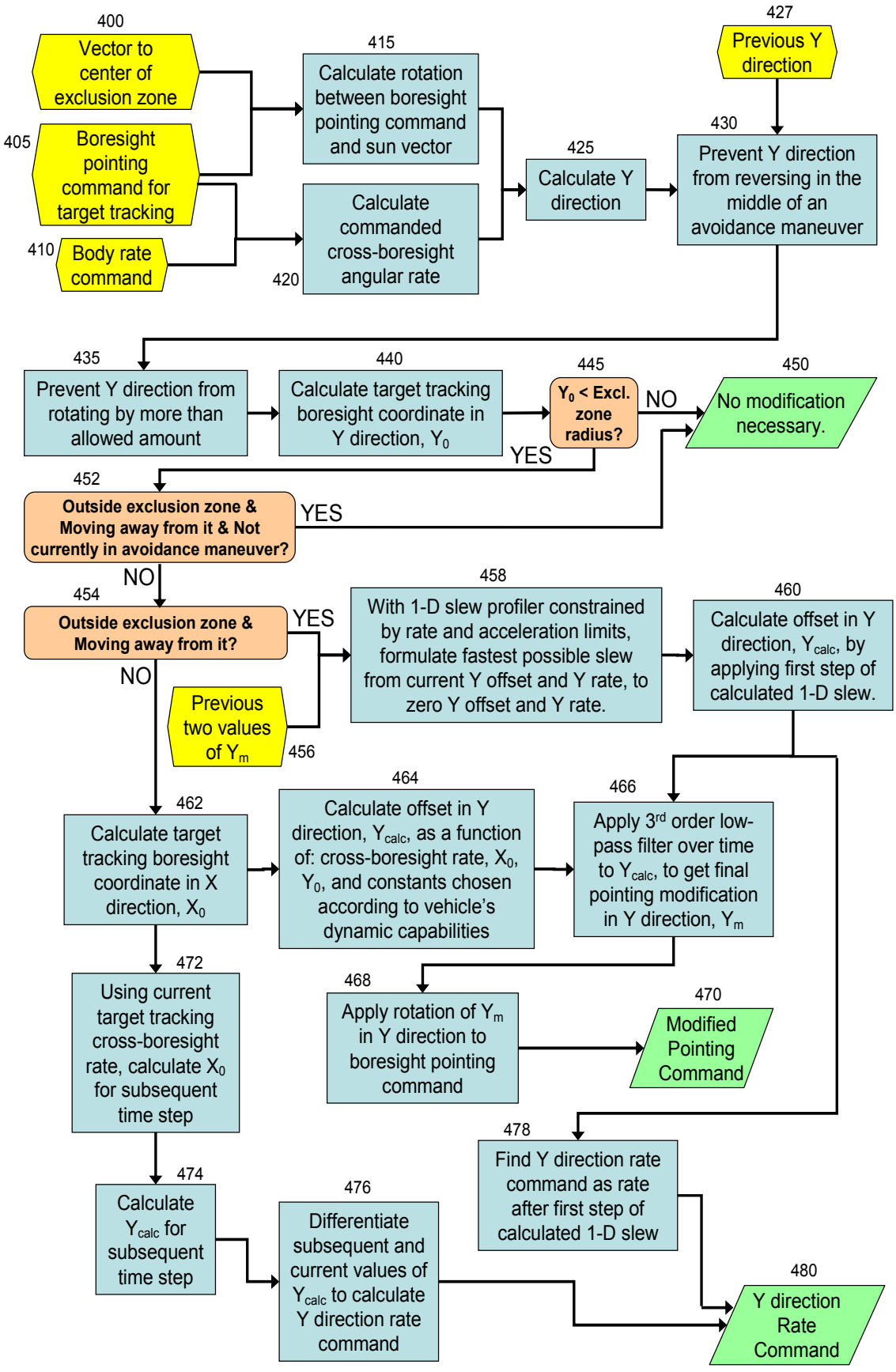

Figure 2: Algorithm Flow Diagram

In Figure 2, the input 400 is the unit vector from the spacecraft to the center of the exclusion zone. For example, the exclusion zone may be around the sun, in which case input block 400 would be the unit vector from the spacecraft to the sun, $\hat{s}$. (Thus, this vector will be referred to as the sun vector henceforth.)

The input 405 is the current boresight pointing command for target tracking, which comes from a separate attitude guidance algorithm for target tracking or target acquisition. The target-tracking boresight pointing command may be expressed as a unit vector, $\hat{b}$.

The input 410 is the spacecraft's current angular rate command for target tracking, $\vec{\omega}$, generated by the same attitude guidance algorithm that gave input 405.

In step 415, the axis $\hat{a}_{sb}$ of the rotation between the boresight pointing command and the sun vector is calculated as follows:

$$\hat{a}_{sb} = \frac{\hat{s} \times \hat{b}}{\left| \hat{s} \times \hat{b} \right|}. \qquad (1)$$

In step 420, the axis $\hat{\omega}_c$ and magnitude ω_c of the commanded target-tracking cross-boresight rate are calculated as follows:

$$\vec{\omega}_c = \vec{\omega} - \hat{b}\left(\hat{b} \cdot \vec{\omega}\right); \qquad (2)$$

$$\omega_c = \left| \vec{\omega}_c \right|; \qquad (3)$$

$$\hat{\omega}_c = \frac{\vec{\omega}_c}{\omega_c}. \qquad (4)$$

In the next step 425, the Y direction, $\hat{Y}$ (as described with reference to Figure 2) is calculated as follows:

$$\hat{Y} = \frac{\hat{a}_{sb} - \hat{\omega}_c\left(\hat{a}_{sb} \cdot \hat{\omega}_c\right)}{\left| \hat{a}_{sb} - \hat{\omega}_c\left(\hat{a}_{sb} \cdot \hat{\omega}_c\right) \right|}. \qquad (5)$$

In step 430, the Y direction is prevented from reversing direction in the middle of an avoidance maneuver. If the target-tracking pointing is heading very close to the sun vector, then a slight change in the target-tracking rate could change the heading from one side of the sun vector to the other, causing $\hat{Y}$ to change by nearly 180°, effectively reversing the Y direction from one time step to the next. This is prevented so that the commanded rates and accelerations from the exclusion zone avoidance algorithm don't exceed the spacecraft's dynamic limits, and also so that the commands from the algorithm don't lead the boresight to cross through the exclusion zone.

To effect the objective of step 430, the algorithm checks whether an exclusion zone avoidance maneuver is underway by checking whether any of the applied Y offsets in a certain number of the immediate past time steps has been above a certain threshold. If this is the case (an exclusion zone avoidance maneuver is underway), the current Y direction, $\hat{Y}$, is not allowed to reverse from the previous Y direction, $\hat{Y}_p$ (input 427), as follows:

$$\begin{aligned} &\text{if } \hat{Y} \cdot \hat{Y}_p < 0, \\ &\qquad \hat{Y} = -\hat{Y}; \\ &\text{end.} \end{aligned} \qquad (6)$$

In the next step 435, an angle $\Delta Y_{\max}$ is defined as follows:

$$\Delta Y_{\max} = \frac{K_1 \omega_{\max} t_{step}}{r}, \qquad (7)$$

where K_1 is a constant to be set for desired performance, $\omega_{\max}$ is the spacecraft's angular rate limit, t_{step} is the time step between attitude commands, and r is the angular radius of the exclusion zone.

Then, still in step 435, the Y direction is prevented from rotating by more than a certain angle, to maintain the attitude commands within the spacecraft's dynamic limits. If the rotation between $\hat{Y}_p$ and $\hat{Y}$ is of an angle greater than $\Delta Y_{\max}$, $\hat{Y}$ is redefined by a rotation from $\hat{Y}_p$ in the same direction, but only by the angle $\Delta Y_{\max}$.

In the next step 440, the Y coordinate, Y_0, of the current target-tracking pointing is calculated as follows:

$$\vec{\omega}_i = \hat{Y} \times \hat{b}, \qquad (8)$$

such that $\vec{\omega}_i$ is a vector roughly parallel to $\hat{\omega}_c$, but normal to $\hat{Y}$, and thus not exactly parallel to $\hat{\omega}_c$ if $\hat{Y}$ was altered in step 435;

$$\hat{i} = \frac{\left(\vec{\omega}_i \times \hat{s}\right) \times \vec{\omega}_i}{\left| \left(\vec{\omega}_i \times \hat{s}\right) \times \vec{\omega}_i \right|}; \qquad (9)$$

$$\begin{aligned} &\text{if } \hat{i} \cdot \hat{s} < 0, \\ &\qquad \hat{i} = -\hat{i}; \\ &\text{end;} \end{aligned} \qquad (10)$$

[$\hat{i}$ is a vector offset from the sun vector in the Y direction, and such that the target-tracking pointing is heading toward $\hat{i}$. In an attitude sphere representation, this would be the intersection closest to the sun vector of: A) the great circle normal to $\vec{\omega}_i$ and containing $\hat{b}$; B) the great circle normal to $\hat{Y}$ and containing the sun vector.]

$$Y_0 = \cos^{-1}\left(\hat{i} \cdot \hat{s}\right); \quad (11)$$

if $\left(\vec{\omega}_i \times \hat{i}\right) \cdot \left(\hat{i} \times \hat{s}\right) < 0$,

$$Y_0 = -Y_0; \quad (12)$$

end.

In decision step 445, if Y_0 is less than r, the algorithm goes to decision step 452; otherwise the algorithm goes to output step 450.

In decision step 452, if the target-tracking pointing is outside of the exclusion zone, and is moving away from the exclusion zone, and an avoidance maneuver is not currently underway (checked as in step 430), the algorithm goes to output step 450; otherwise the algorithm goes to decision step 454.

If the algorithm reaches output step 450, the target-tracking pointing is currently outside the exclusion zone, and not heading toward the exclusion zone, and an avoidance maneuver is not currently underway. In this case, the algorithm is terminated for the current time step, with no modification applied to the target-tracking boresight pointing command.

In decision step 454, if the target-tracking pointing is outside the exclusion zone and moving away from the exclusion zone, the algorithm goes to step 458; otherwise the algorithm goes to step 462.

If the algorithm reaches step 458, the target-tracking pointing is currently outside the exclusion zone, and not heading toward the exclusion zone, but an avoidance maneuver is currently underway. In this case, the goal is to return the boresight to target-tracking (zero Y offset) as rapidly as possible. This is achieved via a one-dimensional (1D) slew profiler function, which calculates the fastest possible slew from the current Y offset and Y rate, to zero Y offset and zero Y rate, constrained by the spacecraft's rate and acceleration limits.

The 1D slew profiler function takes as its inputs: a desired slew duration time T1D; the predefined angular rate and acceleration limits, the initial rate, the final rate, and the range to be traversed. The function then calculates a 1D slew that starts at the initial rate and proceeds with a period of constant acceleration, followed by a period of zero acceleration (constant rate), followed by another period of constant acceleration of the same magnitude and opposite direction as the acceleration in the first period, and ends at the final rate. The duration of the constant rate period may be zero, such that the slew effectively has only two discrete periods, each containing constant acceleration. If the input T1D is left empty, as it is in the present implementation, the 1D slew profiler function will calculate the fastest possible (minimum time) slew that satisfies the remainder of the inputs.

In every time step for which the algorithm reaches step 458, a new 1D slew is calculated, based on current conditions.

In step 460, the current desired Y offset to the target-tracking pointing, Y_{calc}, is taken to be the position of the first time step of the slew calculated in step 458.

If the algorithm reaches step 462, the target-tracking pointing is currently either in or headed toward the exclusion zone, and the algorithm calculates the desired Y offset, Y_{calc}. In this instance, the calculation of Y_{calc} depends upon the value of X_0, the $\hat{X}$ coordinate (as described with reference to Figure 2) of the current target-tracking pointing.
In step 462, X_0 is calculated as follows:

$$X_0 = \cos^{-1}\left(\hat{i} \cdot \hat{b}\right). \quad (13)$$

In step 464, Y_{calc} is calculated as follows:

$$A = r - Y_0; \quad (14)$$

$$n_\omega = \frac{\omega_{\max} - \omega_c}{\omega_{\max}}; \text{ if } n_\omega < 0,\ n_\omega = 0; \text{ end;} \quad (15)$$

$$n_d = K_2\left(1 + n_\omega\left(e^{-\left(\frac{X_0}{K_3 r}\right)^{K_4}} - 1\right)\right); \quad (16)$$

$$n_{Y_0} = \frac{Y_0}{r}; \quad (17)$$

$$d = n_d \frac{A}{\left(1 - n_{Y_0}\right)^{0.5}}; \quad (18)$$

$$Y_{calc} = Ae^{-\left(\frac{X_0}{d}\right)^2}, \quad (19)$$

where e is the base of the natural logarithm, and where K_i {i = 2,3,4} are constants that affect the shape of the exclusion zone avoidance pointing path, to be set for desired performance with consideration of the exclusion zone radius and the spacecraft's dynamic limits.

Both steps 460 and 464 lead to step 466.

In step 466, regardless of how Y_{calc} was calculated, a 3rd order low-pass filter over time is applied to Y_{calc} to smooth its time-profile. The filter constants are chosen based on the spacecraft dynamics and ACS performance. The output from the filter is Y_m, the Y offset from target-tracking pointing of the final modified boresight pointing command.

In step 468, a rotation of angle Y_m, about axis $\hat{Y}$, is applied to the target-tracking pointing to give the final modified boresight pointing command.

The output 470 is the modified boresight pointing command.

For vehicles in which the ACS takes as input a rate command in addition to an attitude command, the algorithm calculates a Y direction rate command, ω_Y, in a manner that often produces improved performance as compared to obtaining it by simply differentiating successive Y_{calc} values. ω_Y is integrated into the full three-dimensional (3D) angular rate command by an algorithm outside of the exclusion zone guidance algorithm, which often produces improved performance as compared to obtaining the 3D angular rate command by simply differentiating successive 3D attitude commands. ω_Y is only calculated in the time steps wherein a value of Y_m is applied to modify the boresight pointing command, i.e. in time steps wherein the exclusion zone guidance algorithm reaches step 470. The calculation of ω_Y begins with step 472 in time steps wherein the algorithm reaches step 462, and with step 478 in time steps wherein the algorithm reaches step 458.

In step 472, ω_c, the magnitude of the commanded target-tracking cross-boresight rate, is used to extrapolate an estimate of the value of X_0 at the next time step, X_{0_Next} as follows:

if [target-tracking pointing is moving away from exclusion zone],

$$X_{0\_Next} = \left| X_0 + \omega_c t_{step} \right|; \qquad (20)$$

else,

$$X_{0\_Next} = \left| X_0 - \omega_c t_{step} \right|; \qquad (21)$$

end.

In step 474, an estimate of the value of Y_{calc} at the next time step, Y_{calc_Next} is calculated in the same way as Y_{calc} is in step 464, but now using X_{0_Next} instead of X_0.

In step 476, Y_{calc_Next} and Y_{calc} are differentiated to give ω_Y, as follows:

$$\omega_Y = \frac{Y_{calc\_Next} - Y_{calc}}{t_{step}}. \qquad (22)$$

In step 478, ω_Y is taken to have the value of the rate reached subsequent to first time step of the 1D slew calculated in step 458.

Both steps 476 and 478 lead to output 480, which is ω_Y.

Note: The exclusion zone guidance algorithm does not provide attitude guidance for the angular degree of freedom about the boresight axis. If it is desired that this angle be controlled, a separate attitude guidance algorithm is required to be applied in each time step, after the exclusion zone guidance algorithm has executed.

5. ALGORITHM SIMULATION

Figure 3 shows a three-dimensional representation of a simulated spacecraft scenario where the exclusion zone avoidance algorithm is active. The sphere in the illustration represents an attitude sphere as described with reference to Figure 1. In Figure 3, the whole attitude sphere is shown instead of the flattened portion of the sphere shown in Figure 1. Again, the spacecraft (or other body) may be imagined to be located at the center of the sphere.

In Figure 3, the small circles on the surface of the sphere represent the pointing directions of the instrument boresight axis over time, showing where the axis would intersect the sphere. These circles are separated by constant time intervals, such that the relative angular rate of the spacecraft may be inferred from their spacing on the sphere. When they are closer together, the spacecraft is moving more slowly, and vice versa. The dashed line represents pointing vectors from the spacecraft to the target over time, showing their intersection with the sphere.

In Figure 3, the large circle on the sphere represents the exclusion zone. In this case, it is an exclusion zone around the sun vector, so the intersection with the sphere of the pointing vector from the spacecraft to the sun would be the center point of the circle. Any boresight pointing that falls within this circle would be within the exclusion zone, and is to be avoided. For extra safety, a 2° margin has been added to the exclusion zone, so the effective exclusion zone used by the avoidance algorithm has a 2° larger radius than the circle as drawn. Thus, the boresight pointing should not come within 2° of the circle.

At initial time t_0, the scenario begins. The spacecraft is in "normal mode" where the attitude guidance is provided by other algorithms. In normal mode, the exclusion zone avoidance algorithm is disabled because the instrument boresight remains pointed far from the exclusion zone due to other factors.

At time t_1, the target commences its trajectory beginning the dashed line showing the target-pointing profile, which soon crosses through the exclusion zone.

At time t_2, the spacecraft is cued with the target trajectory data. This cue may come from another spacecraft or via

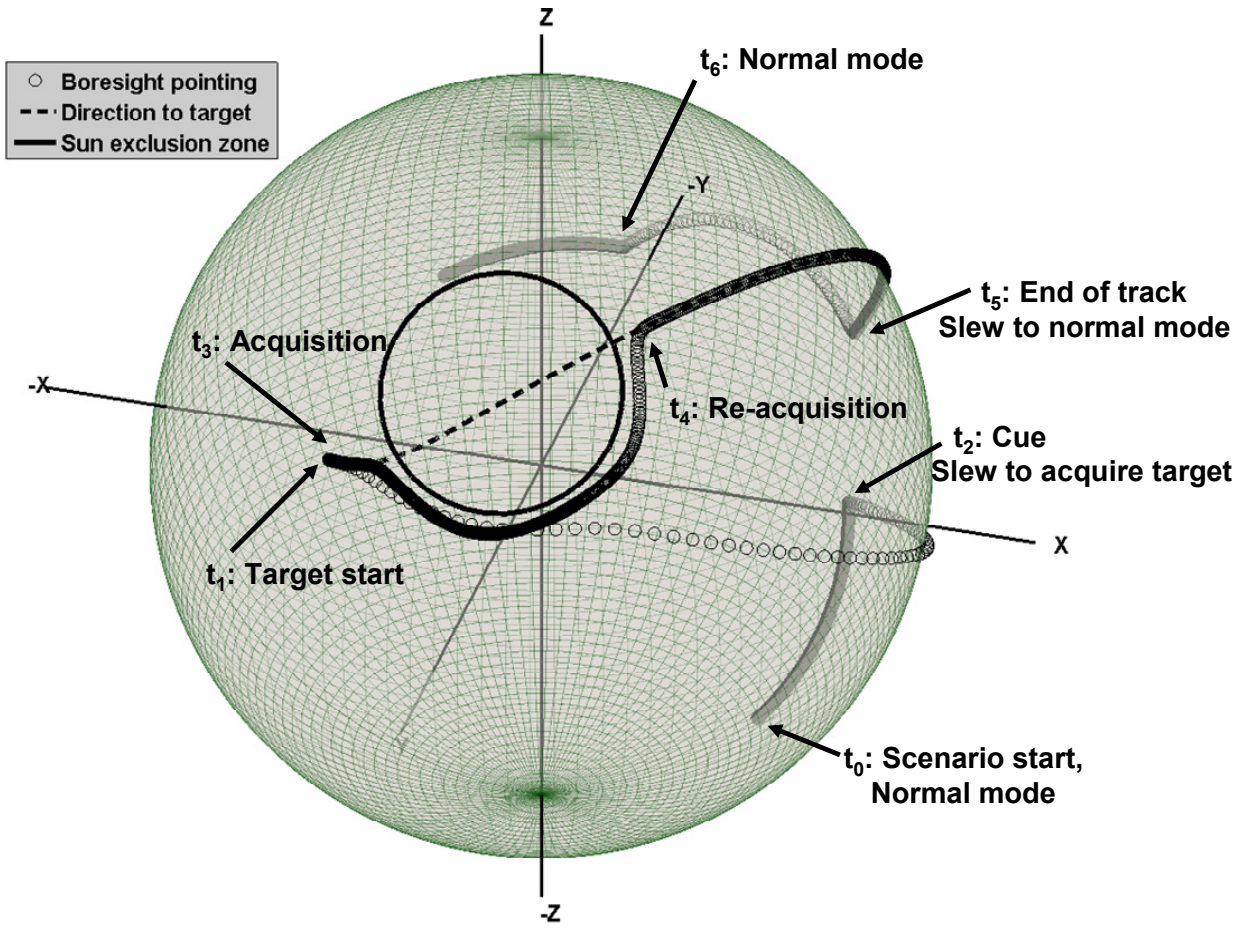

Figure 3: Example Simulation Result

communication from the ground. A separate attitude guidance algorithm then uses this data to provide attitude commands to the ACS, such that the spacecraft slews to point the instrument boresight at the target. When the cue is received and the slew is initiated, the exclusion zone avoidance algorithm is enabled.

The small circles between times t_2 and t_3 represent the boresight-pointing during the slew to acquire the target. The circles in this section are relatively highly spaced because the spacecraft has a relatively high cross-boresight rate during this period. In the scenario represented in Figure 3, the slew to acquisition would cause the boresight to pass slightly through the sun exclusion zone, if the exclusion zone avoidance algorithm were not enabled. However, since the algorithm is enabled, it modifies the pointing commands where necessary to steer the boresight-pointing around the exclusion zone. This can be seen as a slight deviation in the path of the highly spaced small circles shortly before t_3.

It should be noted that in the simulation depicted in Figure 3, the small circles show the actual boresight-pointing directions, not the commanded boresight-pointing directions. If the exclusion zone avoidance algorithm were not performing properly, it would be possible to command the boresight to steer around the exclusion zone, but without keeping the commanded rates and accelerations within the spacecraft's limits. This could lead to the boresight actually pointing into the exclusion zone, even when it was commanded not to. The small circles in Figure 3 show not only that the boresight is commanded to steer around the exclusion zone, but that the avoidance algorithm is performing properly, keeping commands within the spacecraft's dynamic limits such that the spacecraft may successfully follow the pointing commands, preventing the actual boresight from pointing into the exclusion zone.

At time t_3, the slew is complete and the spacecraft has achieved target acquisition, meaning that the instrument boresight has been pointed at the target. Tracking then commences, meaning that the boresight is kept pointed at the target. However, after a short period of tracking, the exclusion zone avoidance algorithm begins modifying pointing commands to again steer the boresight around the sun exclusion. This time, the deviation caused by the algorithm is much larger and more prolonged than the deviation during the slew to acquisition. In this case, it causes the tracking to be compromised, meaning that during the exclusion zone avoidance maneuver, the instrument boresight is not pointed at the target. Generally,

compromised tracking is undesirable, but here it is preferable to the alternative, which is to point the boresight into the exclusion zone and damage the instrument.

At time t_4, the spacecraft smoothly reacquires the target and commences tracking again. The smoothness of this re-acquisition shows again that the attitude commands (including the pointing modifications generated by the exclusion zone avoidance algorithm) remain within the spacecraft's dynamic limits.

Between times t_4 and t_5, the spacecraft tracks the target, until at t_5 the target trajectory ends and the spacecraft slews back to normal mode. The spacecraft reaches normal mode at t_6, and at that time the exclusion zone avoidance algorithm is disabled.

7. CONCLUSION

An attitude guidance algorithm has been developed to prevent a particular body axis, such as an optical instrument boresight, from pointing within a certain exclusion zone, such as an angular region around the sun. The algorithm is appropriate for applications exhibiting an intrinsic desired pointing profile, such as for tracking a target, separate from exclusion zone considerations. Deviations from this nominal desired profile are minimized to the extent possible while preventing exclusion zone incursions, by accounting for the spacecraft's dynamic limits and its current attitude state (3-D attitude and angular rates) relative to the zone, to generate modified attitude commands that hug the zone boundary more tightly when the spacecraft is slewing more slowly, and take a longer path around the zone when the spacecraft is slewing more rapidly (analogous to a car driver's instinct to begin steering around an obstacle further in advance, the faster he is driving). The algorithm operates generally such that the nominal desired attitude profile might not be for target tracking, but might be a slew to acquire the target, or some other maneuver or attitude mode.

This algorithm has performed well in simulations across many scenarios with various spacecraft and target states, preventing the instrument from pointing within the exclusion zone, and allowing high-duration target tracking.

Further work may include development of a rigorous method to determine favorable K values (Eq. 16) as a function of spacecraft dynamic limits and exclusion zone size.

ACKNOWLEDGEMENT

This material is based upon work supported by the U.S. Army Space and Missile Defense Command under Contract No. HQ0006-04-D-0002.

REFERENCES

[1] C. R. McInnes, "Large Angle Slew Maneuvers with Autonomous Sun Vector Avoidance," AIAA Journal of Guidance, Control, and Dynamics, Vol. 17, No. 4, 875–877, 1994.

[2] W. T. Cerven, F. Bullo, and V. L. Coverstone, "Vehicle Motion Planning with Time-Varying Constraints," AIAA Journal of Guidance, Control, and Dynamics, Vol. 27, No. 3, 506–509, 2004.

[3] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-Time Motion Planning for Agile Autonomous Vehicles," AIAA Journal of Guidance, Control, and Dynamics, Vol. 25, No. 1, 116–129, 2001.

[4] H. B. Hablani, "Attitude Commands Avoiding Bright Objects and Maintaining Communication with Ground Station," AIAA Journal of Guidance, Control, and Dynamics, Vol. 22, No. 6, 759–767, 1999.

[5] G. Singh, G. Macala, E. Wong, and R. Rasmussen, "A Constraint Monitor Algorithm for the Cassini Spacecraft," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Reston, VA, 272–282, 1997.

BIOGRAPHY

Jesse Koenig *is a spacecraft GNC engineer, program manager, and business development associate at SpaceDev, Inc. He was responsible for developing much of the GNC simulations, as well as the attitude guidance and control algorithms for the Distributed Sensing Experiment (DSE) and Trailblazer missions. He has managed two programs focused on low-cost robotic lunar landing missions, one of which culminated with a successful flight-test of the first ever soft-landing vehicle to be propelled by hybrid rocket motors. Before SpaceDev, Mr. Koenig received a B.A. in Physics from Middlebury College and an M.S. in Aeronautics and Astronautics from Stanford University, and he did GNC systems engineering at NASA Goddard and Boeing Satellite Systems. He has also performed peer-reviewed published research on kinetic impact asteroid deflection.*