HIRDLS

SW-LOC-293

HIGH RESOLUTION DYNAMICS LIMB SOUNDER

Originator: Lawrence Ames Date: **19 March 1998**

Sub-ject/Title: **Scan Table Interpolation Algorithms**

Description/Summary/Contents:

1. This derives and discusses algorithms that interpolate scan mirror positions, velocities, and accelerations from the scan table entries. It discusses “normal operation”, “safing” (commanded to go to the “safe” position), and “emergency stops”.

Keywords: Software, Scanner, Scan Table Interpolation

Purpose of this Document: (20 char max.) ScanTable Interpol’n

Reviewed/Approved by: Date (yy-mm-dd):	_____ originator	_____ ___ HIRDLS flight s/w manager

Advanced Technology Center CAGE Code 65113
Lockheed Martin Missiles & Space
3251 Hanover Street
Palo Alto, CA 94304-1191
United States of America

EOS

Introduction

Alain Carrier wrote an equation-packed memo on May 27, 1997, deriving the accelerations needed to command the mirror between scan states. Independently, Doug Woodard and I derived comparable results using a graphical analysis on a blackboard in Boulder last spring. This memo relates those two analyses.

A second portion of this memo derives and documents the algorithms for "safing" the mirror, whereby a single command causes the mirror to quickly move to a predefined orientation. A final section discusses "emergency stop", a limit check that will be built into the algorithms to make sure the mirror is never commanded past its limits.

The algorithms will be available to evaluate and pretest proposed scan table entries, allowing the user to determine peak accelerations, average power, maximum angular velocity, and extreme travels, to confirm that the table entries are consistent with design guidelines.

This memo derives the algorithms for the simple "constant acceleration" profile. As described in an accompanying memo (SW-LOC-294), other acceleration profiles are also possible, such as the "sine bell" (a half cycle of the sine wave) and the "versine function" (sine-squared). Each of these has advantages and disadvantages, as is discussed in that memo. Also, each of them scales directly from the parameters derived in this memo.

Normal Operation

The mirror motions are dictated by entries in the scan table. This table has five items per entry: time interval Δt, az and el positions θ_a and θ_e, and az and el angular velocities ω_a and ω_e. The Δt is the time interval allowed for the mirror to reach the specified orientation, at which time it is to have the specified velocity. There is no need to tabulate the angular accelerations α_a and α_e, as they can be derived from the given information. The units used are specified elsewhere: all that is important here is that they are consistent—if Δt is given in seconds and θ in μrad, then α will be expressed in μrad/sec^2.

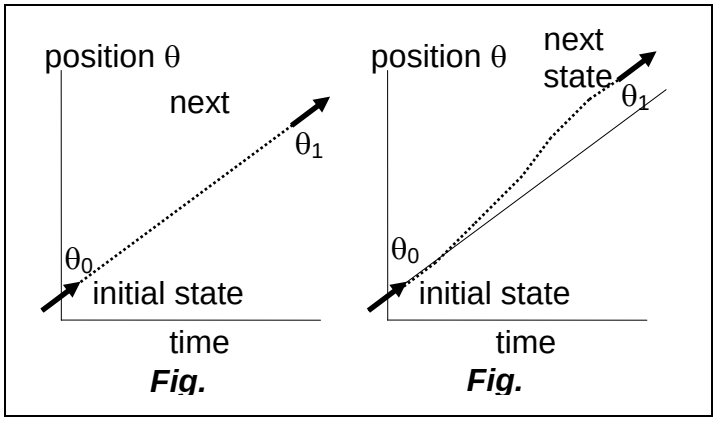

Fig. ***Fig.***

It is possible that the chosen state (Δt, θ_a, θ_e, ω_a, ω_e) can be reached from the previous state with a constant acceleration (case "A", shown in Fig. 1a), where velocity is given by the slopes of the arrows. However, in general, the acceleration needs to change over the time interval in order to meet both the desired endpoint position and velocity (case "B", Fig. 1b). A simple solution is to divide the time interval in two, and use one constant acceleration over the first part of the interval, and a different constant acceleration over the rest.

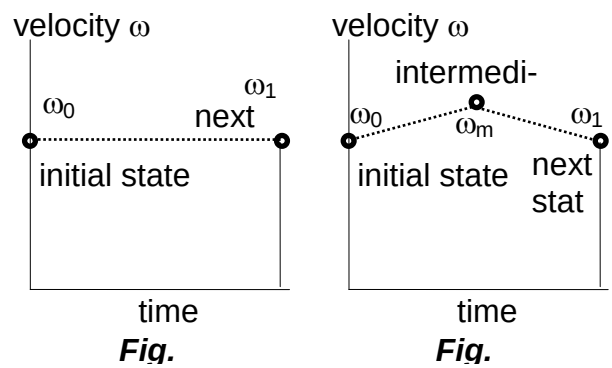

Fig. ***Fig.***

Method for solution: plot velocity as a function of time: by assuming constant accelerations, the velocity plots will be straight lines. Figure 2a shows the velocity as a function of time for case "A": the two dots indicate the initial velocity and the desired velocity at the next state. The numbers "worked out" in case "A": the change in angle (θ_1 - θ_0) is equal to the time interval (Δt) times the mean of the end-point velocities ((ω_1+ω_0)/2): thus the acceleration can be constant (zero in this example) over the full Δt interval. The change in position is the integral of velocity over time, and is equal to the area of the rectangle.

For case "B", the numbers didn't "work out", and so the mirror has to accelerate part of the time and then decelerate the other part. This is apparent in Fig. 2b: the velocity increases part of the time and then decreases. The change in position is again the area under the curve, so nearly any desired state can be made to work out by the appropriate choice of the intermediate value. One of the points of Alain's analysis is that the optimal time for the intermediate point is at the exact midpoint of the time interval (0.5Δt). For constant accelerations, the accelerations are given by the slopes of the line segments: after all, acceleration is the change in velocity per unit time.

Time to derive equations. Both azimuth and elevation are treated the same, so I'll drop the *a* and *e* subscripts. Assume two equal parts to the time interval Δt, with a constant acceleration over the first half and a possibly different acceleration for the second half. Use the subscript '0' to indicate the state at the beginning of the time interval, '1' to indicate at the end, and 'm' at the midpoint (see Fig. 3)

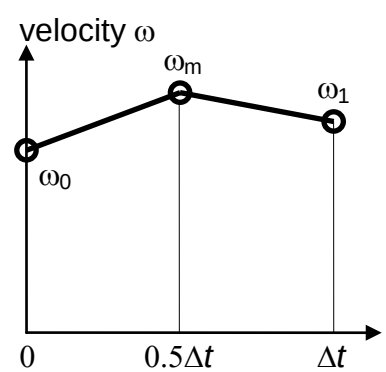

Fig. 3

The change in angle θ_1 - θ_0 = $\int_0^{\Delta t} \omega(t)dt$ = the area under the curve = ½ (ω_0+ω_m)·0.5Δt + ½ (ω_m+ω_1)·0.5Δt. Solve for ω_m in terms of the known ini-

tial state θ_0, ω_0 and final state θ_1, ω_1, and the time interval Δt:

$$\omega_m = 2\frac{(\theta_1 - \theta_0)}{\Delta t} - \frac{1}{2}(\omega_1 + \omega_0)\,. \qquad \text{Eq. 1}$$

The acceleration for the first half of the time interval is

$$\begin{aligned}\alpha_F &= \frac{\omega_m - \omega_0}{.5\Delta t} \\ &= \frac{4}{(\Delta t)^2}(\theta_1 - \theta_0) - \frac{1}{\Delta t}(\omega_1 + 3\omega_0).\end{aligned} \qquad \text{Eq. 2}$$

Similarly, the acceleration for the second half of the time interval is

$$\begin{aligned}\alpha_S &= \frac{\omega_1 - \omega_m}{.5\Delta t} \\ &= \frac{1}{\Delta t}(3\omega_1 + \omega_0) - \frac{4}{(\Delta t)^2}(\theta_1 - \theta_0).\end{aligned} \qquad \text{Eq. 3}$$

(This is the same, after appropriate variable renaming and rearranging, as Alain's Eq. 17a and 17b. Note that either or both α_F and α_S can be positive or negative.)

With the above values for acceleration, which are determined using only known values, the mirror will have the required velocity while at the appropriate angle at the appropriate time. The drawback is that the mirror might be "jerked" (depending on the control-loop algorithm in use), an effect described by the technical term "jerk", referring to the third derivative with respect to time. If one constant acceleration α_F is used for the first half of the time interval, and a second constant acceleration α_S is used for the second, then the mirror may be jerked at the mid-interval transition; it may also be jerked at the end of the interval when the next acceleration is used. (If a feedback control system, which has a finite bandwidth, is used in place of feedforward, the system is not jerked.) Both the sine-bell and the versine acceleration profiles are continuous in acceleration, meaning that they are jerk-free; the versine function is also continuous in higher-order time derivatives, which may be beneficial to the electronics—see SW-LOC-294.

For the constant acceleration profile, it is apparent from Fig. 2b that the extreme velocity during a scan will either be an endpoint of a time interval (which is specified by the user in the scan table), or at the midpoint; it turns out that the other acceleration profiles have the extreme velocities at the same points. I recommend that the user run the proposed scan table through the test algorithm to verify that the peak velocity ω_m is within limits.

Neglecting friction, the acceleration equations accurately reflect the torque needed for the azimuth scan. The elevation scan has a restorative spring to push against, so the torque will not be simply the product of inertia and acceleration. The control algorithms will compensate for friction, spring torque, etc., in calculating the current that needs to be commanded for the motors.

Safing

The mirror may be commanded to go to “safe mode”, which means that the mirror is to move from whatever current initial condition (position θ_I and velocity ω_I in each axis, az and el), to being at rest (ω=0) at the safe position θ_S, in as short a period as practical. Three alternatives to be considered prior to coding: (1) the algorithm can immediately begin moving towards the safe position; (2) to avoid jerks, the algorithm can complete whatever motion is currently in progress (go forward to the next half-interval time) before moving to the safe position; or (3) complete the motion if the magnitude of the acceleration is above some threshold, otherwise start the new command immediately. The initial release version implements alternative (2).

The safing command is similar to a normal motion, except that the time interval Δt is not known, other than that it is to be as short as reasonably possible. The time interval is calculated by minimizing the time interval while keeping the magnitude of the acceleration constrained to be less than plus or minus the maximum acceleration α_X. The time interval for the two angles will in general not be equal: calculate each of them, and then use the longer Δt in the regular equations above. The maximum acceleration α_X is determined by the maximum torque from the motor(s) (which may vary, depending on whether both of the elevation drives are operational), the restorative torque (for elevation), and the moment of inertia.

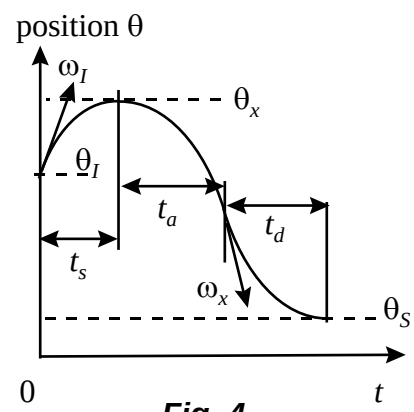

Fig. 4

As before, consider the constant acceleration approach. Plot angular position (representing az or el) as a function of time. It can be seen that there are three cases: (1) initially the mirror is moving the wrong way (away from the safe position); (2) the mirror is moving the right way, but too fast; and (3) the mirror is moving the right way, but not fast enough. In addition, there are a couple borderline cases: the mirror is not moving; the mirror is in the right place, or the mirror is moving at just the right speed—they can be treated as limits of the above cases. The analyses are the same regardless of signs: “too high and moving up” and “too low and moving down” are in the same class.

Figure 4 shows case 1, with the mirror initially moving the wrong direction. The mirror has an initial position θ_I and an initial velocity ω_I. It will take some time t_s to stop: $t_s = -\omega_I/\alpha$, where the minus sign comes from the need to reduce the velocity to stop, and α is $\pm\alpha_X$, with α_X the maximum allowable acceleration, directed downward for positive ω_I and upward for negative ω_I:

$$t_s = |\omega_I/\alpha_X|. \qquad \text{Eq. 4}$$

After stopping, the mirror will be at some extreme position θ_x:

$$\theta_x = \theta_I + \omega_I t + .5\alpha t^2 = \theta_I \pm \frac{\omega_I^{\,2}}{2\alpha_X}, \qquad \text{Eq. 5}$$

using the + sign for positive ω_I and the - sign for negative ω_I.

Once stopped, the mirror then has to be moved from θ_x to the safe position θ_S by accelerating for a time t_a and decelerating for an equal time t_d:

$$t_a = t_d = \sqrt{\frac{|\theta_S - \theta_x|}{\alpha_X}}\,, \qquad \text{Eq. 6}$$

where the absolute value accounts for both the upper and lower cases. The time needed to slow down and reverse directions is then t_s+t_a, and the total time to safe can be t_s+t_a+t_d.

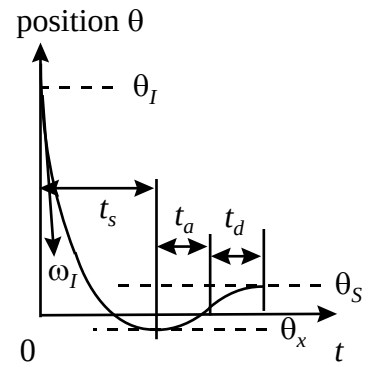

Fig. 6

There are advantages to using two equal half-intervals in the algorithm, as with the normal operation: it is easier to keep track of the two axes if they both have the same timing, and the normal operation algorithms can be reused (minimizing the amount of code to be written). The safing can be made symmetric by allowing an equal time for the decelerate to safe stop phase: give each half-interval t_s+t_a. As can be seen in Fig. 5, this doesn't just mean that a less extreme acceleration can be calculated for the second half-interval: to match slopes and positions, the acceleration for the first half is also moderated (less extreme). Once a time interval is selected, the equations above for normal operation calculate the accelerations, velocities, and positions: the exercise here was to find a reasonably short time interval that wouldn't require excessive accelerations.

Figure 6 shows the second case: the mirror is moving in the correct direction, but going too fast to stop in time. As before, calculate the time t_s needed to come to a stop, determine the stopping position θ_x, and then calculate the time t_a needed to accelerate back to the safe position. Also, as before, once t_s+t_a has been calculated, use the same length for the second half-interval, and then use less than extreme accelerations to stop at the safe position.

Although the situation is different, the equations for case 2 are the same as for case 1: the time to stop is given by Eq. 4, the local extreme position at the overshoot is given by Eq. 5, and the time to accelerate back to the safe position is given by Eq. 6. For case 2, like case 1, the first half-interval is the larger, and the second half-interval will have a more moderate acceleration: in the limiting case where there is no overshoot, the time of the half interval is the stopping time t_s, and the acceleration is zero for the second half interval.

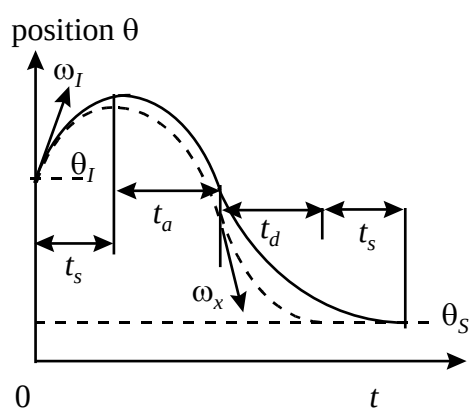

Fig. 5

Figure 7 shows the third case (and this one is different): the mirror is moving the right direction, but it could move faster. In this case, t_s is a negative number, telling when the mirror would have

had to have started moving to reach the observed initial velocity ω_I:

$$t_s = -|\omega_I/\alpha_X|. \qquad \text{Eq. 7}$$

The position θ_x is the position the mirror should have started from, given constant acceleration:

$$\theta_x = \theta_I + \omega_I t + .5\alpha t^2 = \theta_I \mp \frac{\omega_I^{\,2}}{2\alpha_X}. \qquad \text{Eq. 8}$$

This is the same as Eq. 5 above, except that this time the + sign is used for the negative ω_I case, and the - sign is for the positive ω_I case.

The acceleration time t_a and deceleration time t_d are calculated as before, using Eq. 6. However, since t_s was negative, t_d is the defining time: the second half-interval has the limiting acceleration, and the time t_d is used for the length of the half-intervals to calculate the required accelerations: the first half-interval is elongated to equal the second, and so has a less extreme acceleration.

The distinction between case 1/case 2 and case 3 is that for the latter, the mirror is able to "stop in time": cases 1 and 2 both begin with a deceleration (stopping) phase, while case 3 begins with an acceleration. If the mirror is moving in the correct direction, then the stopping time is $t_s = |\omega_I/\alpha_X|$, as given in Eq. 4. The position when stopped is given by Eq. 5: if θ_x is between θ_I and θ_S, then it is case 2. Rephrased, use case 2 equations when

$(\theta_S - \theta_I)$ and ω_I have the same sign (e.g., $\omega_I \cdot (\theta_S - \theta_I) > 0$), and Eq. 9a

$$\left|\frac{\omega_I^{\,2}}{2\alpha_X}\right| < \left|\theta_S - \theta_I\right|. \qquad \text{Eq. 9b}$$

Other considerations for safing:

- Since these paths will be calculated autonomously, they will not have been limit checked for position and maximum speed. There are too many possible conditions that might cause the algorithm to exceed one or the other limits to include them explicitly in the equations, and in some cases the correction for one (e.g., speed) aggravates the other (a longer time interval would allow the mirror to move further past its limit), The solution is to treat the limits independently: upon calculating the "recommended" time interval, the code will determine the maximum speed, and if it exceeds the limit, it will extend the time interval accordingly. This is not a linear relationship: the maximum speed is easily reduced in a case 3 path by increasing the time interval. However, in a case 1 where the initial position θ_I just happens to line up with the inflection point (see Fig. 4), the velocity ω_x will equal ω_I for any possible time interval. (But then, that velocity should not exceed the limit, since ω_I presumably was an acceptable velocity.) The algorithm will try to limit maximum speed, but only to a point (by trying, say, 10-

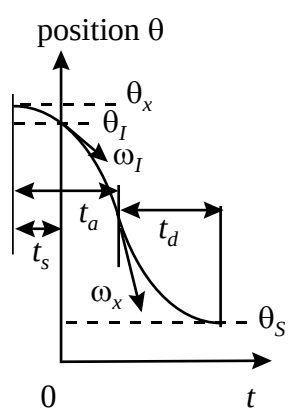

Fig. 7

times the recommended time interval). Position exceedences are handled autonomously by the emergency stop (see below).

- The elevation drive has a restorative spring, which would allow larger torques when moving towards the neutral position. It would further complicate the algorithm to take advantage of possibly increased accelerations in one direction. Also, the effect is smallest in the primary region of interest, as the neutral position is at or near the safe position.

Emergency stop

One last detail to discuss: a limit check on the angular positions. Presumably this limit check will not be needed for normal operations, since the scan table entries will be tested prior to use, but the code will still check each point just to be sure. Also, the emergency stop may be needed for the automatically calculated safing.

The emergency stop is sort of like an algorithmic air-bag: it will deploy just prior to hitting the limit, and will abruptly decelerate the motion. It will use the maximum torque (including the restorative spring force, since that will be strong there and directed in the right direction). It will use peak acceleration, without consideration for smooth derivatives: it will be a bumpy ride.

First calculate the acceleration possible in an emergency: α_E, which includes the restorative torque. Note: the torque varies with angular position—it would needlessly complicate the algorithm to use the full expression: approximate it with a constant that is set to the torque at some angle (say, at 3/4ths of the maximum excursion); don't use the torque at the limit angle.

Use Eq. 5 again, substitute the limiting angle θ_L (use the appropriate upper or lower limit) for the extreme angle θ_x, the current position θ in place of initial θ_I, and then solve for the critical velocity ω_C:

$$\omega_C = \pm\sqrt{\left|2\alpha_E\left(\theta_L - \theta\right)\right|}\ , \qquad \text{Eq. 10}$$

where the + sign is used for the upward motion and upper limit, and the - sign is for the downward and lower limit. If the instantaneous velocity at any time exceeds the limiting critical velocity, then the algorithm jumps to the emergency braking subroutine, slams on the brakes, and then reports back to other software that the mirror has been stopped.

Summary

The equations for normal and safing-mode operations have been calculated for the constant-acceleration profile; with the appropriate modifications given in SW-LOC-294, other acceleration profiles can use the same equations. The emergency-stop algorithm always uses the constant-acceleration profile.

~Lawrence Ames