Optimized Rotation-Axis Attitude Maneuver Controller for the Space Shuttle Orbiter

D. C. Redding* and N. J. Adams*
The Charles Stark Draper Laboratory, Cambridge, Massachusetts

This paper describes a controller for automatic attitude maneuvers of reaction-control-jet-equipped spacecraft. It is based on a new, real-time algorithm for solution of fuel-optimal maneuvers, assuming on-off control and linearized dynamics. These solutions provide efficient model trajectories that are easily tracked using a feed-forward, feedback controller structure. This approach was developed as a software upgrade to the Space Shuttle Orbiter on-orbit autopilot, for which it shows a very substantial performance benefit in a wide range of simulated maneuver tasks.

I. Introduction

THE design of an efficient, reliable attitude maneuver controller for the Space Shuttle Orbiter is complicated by several factors. Control accelerations, provided by two systems of reaction control system (RCS) jets, are highly cross-coupled and low in magnitude, due to the Orbiter's asymmetric mass distribution and restricted placement of RCS jets (Figs. 1 and 2). Control accelerations can change significantly in direction and magnitude, due to changed mass properties during payload operations, and depending on which RCS jets are selected. The Orbiter is required to perform a variety of different maneuver tasks, including inertial maneuvers and maneuvers-to-a-track, where the Orbiter rotates to point at a particular external object, such as a point on the Earth's surface or an object in Earth orbit. Finally, the controller design is constrained by limited on-board computation capability.

The Optimized Rotation-Axis (ORA) controller presented in this paper overcomes these problems, providing reliable, efficient, and precise attitude maneuver capability for the Orbiter and similar spacecraft. It explicitly accounts for the effects of mass asymmetry and RCS jet placement, cross-axis coupling, and variable thrust levels, to achieve excellent maneuver predictability, allowing efficient, precise maneuver control. It uses optimized maneuver jet firing sequences that do not require jet pulsing to uncouple control accelerations. It works in ten different RCS modes, as required by operational considerations. It has demonstrated significant performance improvement in all Orbiter maneuver tasks, as compared to the current Orbiter maneuver controller,[1,2] at the cost of an increase in computer memory usage.

The main innovation of the ORA controller is a new solution for open-loop, optimized, three-axis maneuver trajectories. The ORA trajectory algorithm solves a two-point boundary-value problem for fuel-optimal (or near-optimal), fixed-end-time maneuvers, using linearized spacecraft dynamics with three-axis coupled, on-off control for arbitrary initial and final rate and attitude. It solves these trajectories in under 100 ms of CPU time using the on-board IBM AP-101 computers. The solutions provide open-loop jet firing commands and set-points for the standard digital autopilot (DAP) attitude control feedback loops[1,3] that track the optimized trajectories. The trajectories are easily and efficiently tracked by spacecraft that operate at relatively low rates, providing that the control torques $\bar{\tau}$ satisfy

$$|\bar{\tau}| \gg |\bar{\omega} \times I \cdot \bar{\omega}| \tag{1}$$

This is the case with the Orbiter, as the jet torques are at least 10-100 times larger than the gyroscopic coupling terms in normal operations. The ORA trajectories are derived in Sec. II, and their solution and implementation discussed in Sec. III of this paper.

The ORA controller uses a feed-forward, feedback control structure. At the start of each maneuver, the trajectory solution algorithm computes an open-loop optimized trajectory to take the spacecraft from its current attitude and rate to the specified final attitude and rate. Following computation, the open-loop firing commands are fed forward to the appropriate jets, while the open-loop trajectory states provide time-varying set-points for the DAP phase-plane attitude controller. Deviations from the nominal maneuver are countered by feedback firings, which are added to the feed-forward firings. A change in the target attitude, due to a change in crew command or kinematic effects, will cause recomputation of the maneuver. Upon completion of the maneuver, the DAP switches to the attitude-hold (or track) mode. Examples of ORA performance in the Orbiter DAP are presented in Sec. IV.

The Orbiter is sketched in Fig. 1 with a large payload held on the Remote Manipulator System (RMS) arm. The vernier reaction control system (VRCS) shown consists of six 24-lb thrust jets; these are capable of providing control angular accelerations of between 0.002 and 0.02 deg/s^2, depending on payload and acceleration direction. The primary reaction control system (PRCS) is sketched in Fig. 2. The PRCS jets produce 870 lb of thrust; this plus the greater number (14 jet groups) and more advantageous placement of the jets combine to produce control accelerations for the PRCS of up to 1 deg/s^2. Maneuvers of the Orbiter are normally limited to 2.0 (PRCS) and 0.2 (VRCS) deg/s.

There is relatively little work published recently on fuel-optimal, RCS jet controlled rigid-body attitude maneuvers. Single-axis fuel-optimal maneuvers appear in textbooks.[4] Dixon et al.[5] give a thorough analysis of nonlinear fuel-optimal rest-to-rest maneuvers for axisymmetric "rod-like" vehicles assuming impulsive thrust. More recent articles have emphasized pure feedback methods and momentum-wheel actuation using quadratic cost functions.[6,7] A feed-forward, feedback model-following controller similar in concept to the ORA was developed and test-flown on the Infra Red Astronomical Satellite (IRAS) spacecraft. This controller flies time-

Received Sept. 25, 1985; revision received March 21, 1986.

*Staff Engineer. Member AIAA.

optimal rather than fuel-optimal trajectories, using momentum wheels rather than RCS jets.[8] Nonoptimal feedback controllers have been designed using Lyapunov functions.[9,10] The standard Orbiter DAP is described in Refs. 1 and 2; it is derived from the single-axis feedback controller developed for the Apollo.[11] A further refinement of the single-axis approach is given in Ref. 12.

II. Three-Axis Fuel-Optimal Attitude Maneuvers Using Linearized Dynamics and Coupled On-Off Control

This section presents a discussion of the general problem of optimal attitude maneuvers for spacecraft that satisfy Eq. (1). The necessary conditions for fuel-optimal, fixed-end-time solutions, showing general characteristics of optimal three-axis maneuvers, are derived. Solution of these maneuvers is addressed in Sec. III.

The attitude motion of a rigid body about its center of mass can be expressed in terms of the body-fixed attitude rate vector $\bar{\omega}$ and the spacecraft inertia matrix I as

$$\dot{\bar{\omega}} = I^{-1} \cdot [\bar{T} - \bar{\omega} \times I \cdot \bar{\omega}] \tag{2}$$

In the case of a spacecraft controlled by n distinct RCS jets, the ith of which produces torque vector $\bar{\tau}_i$, the external torque $\bar{T}$ is expressed (neglecting disturbance torques) as

$$\bar{T} = \sum_{i=1}^{n} u_i \bar{\tau}_i \tag{3}$$

Here $\bar{u}$ is a control vector with n elements u_i, such that

$$u_i = \begin{cases} 1, & \text{if the } i\text{th jet is firing} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

We are concerned with the case where $|T| \gg |\bar{\omega} \times I \cdot \bar{\omega}|$, so that we drop these nonlinear coupling terms from Eq. (2). The rate equation becomes

$$\dot{\bar{\omega}} = \sum_{i=1}^{n} \bar{\alpha}_i u_i \tag{5}$$

where the column vectors $\bar{\alpha}_i$ represent the angular acceleration produced by the ith jet,

$$\bar{\alpha}_i = I^{-1} \bar{\tau}_i \tag{6}$$

We choose body-attitude states $\bar{\theta}$ to represent attitude motion, so that the second state equation [with Eq. (5)] is

$$\dot{\bar{\theta}} = \bar{\omega} \tag{7}$$

Note that using $\bar{\theta}$, rather than Euler parameters or another general representation of attitude, introduces potential difficulties in determining target states $\bar{\theta}_{\text{targ}}$ from an inertial target attitude. The value of $\bar{\theta}_{\text{targ}}$ depends on the path taken to achieve it. This is discussed at the end of this section.

The attitude maneuver problem is to find functions $\bar{u}(t)$ that take $\bar{\omega}$ and $\bar{\theta}$ from an arbitary initial state $(\bar{\omega}_0, \bar{\theta}_0)$ to a target final state $(\bar{\omega}_f, \bar{\theta}_{\text{targ}})$. In keeping with standard Orbiter practice, the maneuver is to be done at a specified total rate magnitude ω_m, where

$$\max |\bar{\omega}| \cong \omega_m \tag{8}$$

Perhaps the most direct formulation of the optimal control problem for these maneuvers would be to strictly constrain the maneuver rate to satisfy Eq. (8). Then the cost function J would contain a fuel term and a time term. The resulting solutions could involve complicated discontinuities in the control, however.

A simpler approach is to enforce Eq. (8) indirectly, by adopting a fixed-end-time formulation and then choosing t_f to give approximately the desired rate. In this formulation, which is the one we have chosen, the cost function is the integrated jet on-time (which is proportional to the total fuel use for jets with the same mass flow rate)

$$J = \int_{t_0}^{t_f} \left(\sum_{i=1}^{n} u_i \right) \mathrm{d}t \tag{9}$$

The optimal time history for the control $\bar{u}$ is determined by minimizing the Hamiltonian H^*, formed from the state vari-

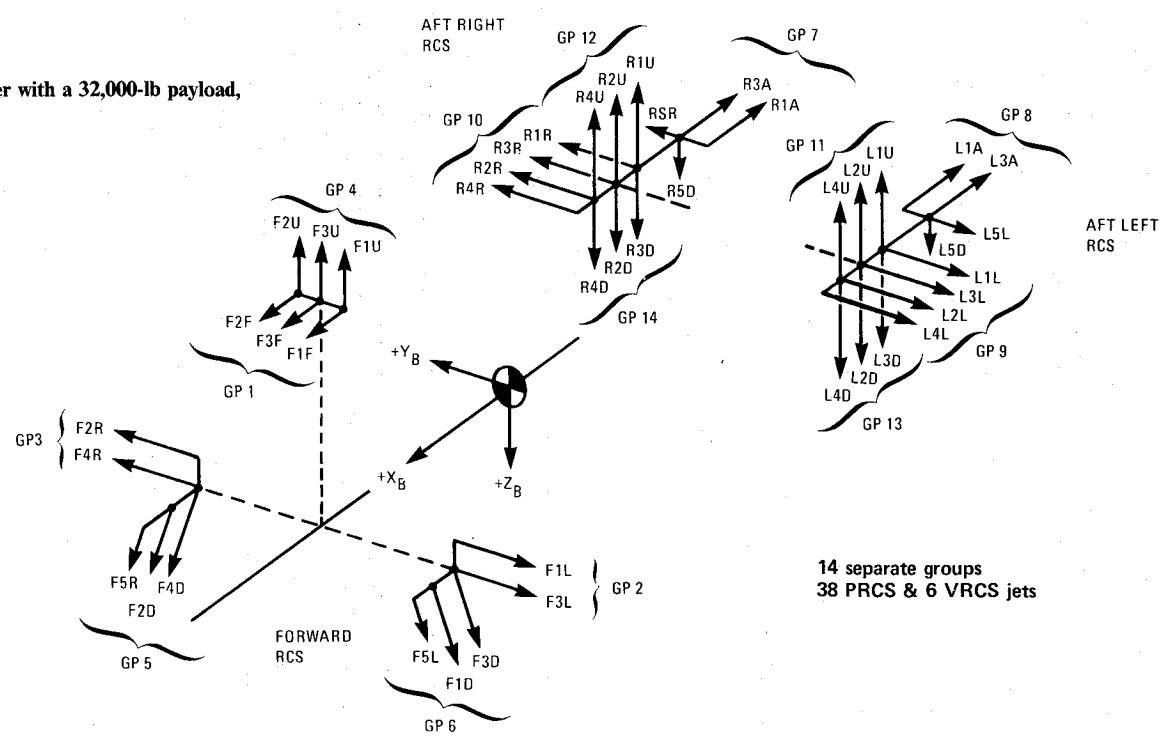

Fig. 1 STS Orbiter with a 32,000-lb payload, showing the VRCS.

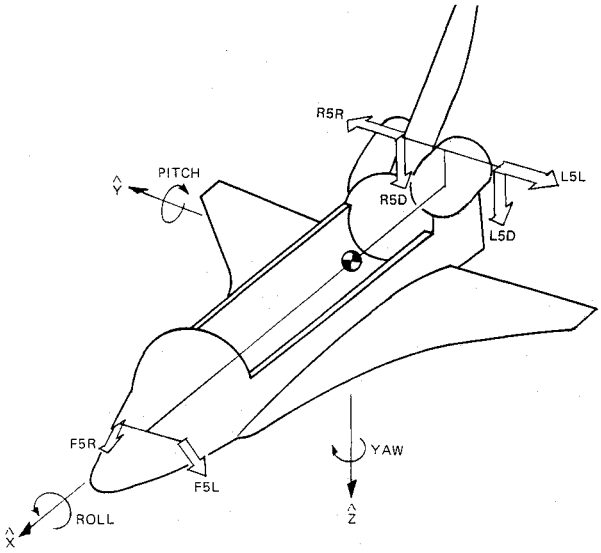

Fig. 2 PRCS jet location and direction.

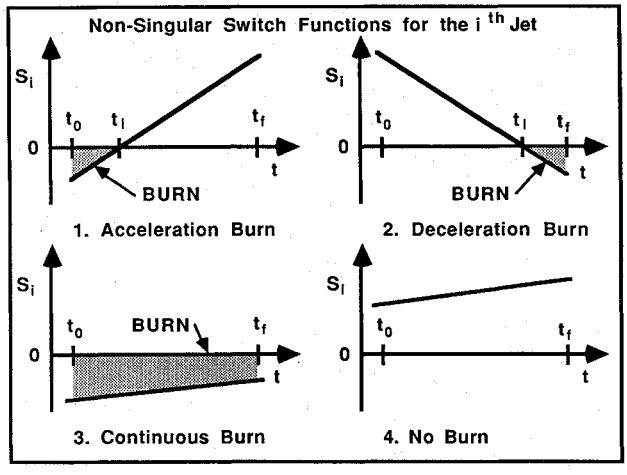

Fig. 3 Behavior of the switch function S_i.

ables and adjoint vectors $\bar{\lambda}_\omega$ and $\bar{\lambda}_\theta^4$,

$$H^* = \bar{\lambda}_\theta \cdot \bar{\omega} + \sum_{i=1}^{n} [\bar{\lambda}_\omega \cdot \bar{\alpha}_i + 1] u_i \qquad (10)$$

Applying the minimum principle to Eq. (10), by inspection

$$\min_{u_i} [H^*] \rightarrow \begin{bmatrix} u = 1 & \text{if } [\bar{\lambda}_\omega \cdot \bar{\alpha}_i + 1] < 0 \\ u = 0 & \text{if } [\bar{\lambda}_\omega \cdot \bar{\alpha}_i + 1] > 0 \end{bmatrix} \qquad (11)$$

Thus, the optimal controls are given in terms of the adjoint variables by switch functions S_i, where

$$S_i = \bar{\lambda}_\omega \cdot \bar{\alpha}_i + 1 \qquad (12)$$

The S_i indicate a burn for the ith jet when $S_i < 0$, and no burn when $S_i > 0$.

The time-dependency of the adjoint vectors is found by applying the Euler-Lagrange equations to H^*. Thus,

$$\dot{\bar{\lambda}}_\theta = 0 \qquad (13)$$

$$\dot{\bar{\lambda}}_\omega = -\bar{\lambda}_\theta \qquad (14)$$

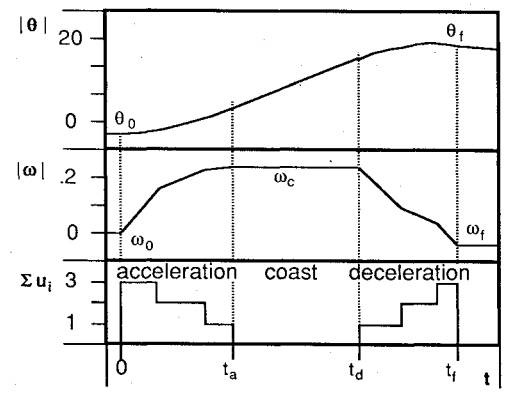

Fig. 4 Typical ORA trajectory.

```
procedure: TRAJECTORY_JET_SELECT(ω̄0,ω̄f) assign(t̄B,TB,J̄B);
    TB=∞;
    do for all eligible i,j,k;
        t̄ijk=[Aijk]^-1 (ω̄f-ω̄0);
        T=ti+tj+tk;
        if T<TB and ti>0 and tj>0 and tk>0 then
            do;
                TB=T;
                t̄B=t̄ijk;
                J̄B=vector(i,j,k);
            end;
    end;
close TRAJECTORY_JET_SELECT;
```

Fig. 5 Trajectory jet selection routine in pseudocode.

The behavior of S_i, and so the time history of the control $\bar{u}$, is fully determined by initial adjoints $(\bar{\lambda}_\omega, \bar{\lambda}_\theta)_0$ through Eqs. (13) and (14). Corresponding attitude motion is determined by $(\bar{\omega}_0, \bar{\theta}_0)$ and Eqs. (11), (12), (5), and (7).

Over the time course of a typical maneuver, there are five possible histories for each S_i, as determined by the adjoint Equations (13) and (14). Four of these are illustrated in Fig. 3. The fifth is singular control, occurring if S_i is identically zero throughout the maneuver, indicating an intermediate value for u_i (chatter). Nonsingular control cases are limited to one switch per jet during the maneuver, a consequence of the linearity of $\bar{\lambda}_\omega$.

The general characteristics of nonsingular, optimal, fixed-end-time, three-axis maneuvers can be deduced from this information. Because there is at most one switch per jet during each maneuver, no jet will burn more than once. Thus, for a total of n jets, there is a maximum of n burns during an optimal maneuver. Furthermore, if a jet is fired at all it burns at the beginning of the maneuver, at the end of the maneuver, or throughout the maneuver. The result is "staircase" optimal jet firing patterns (Fig. 4); with an acceleration phase, where several jets turn on at t_o, turning off in staggered fashion as the maneuver progresses; a coast phase, with no firings; and a deceleration phase, where several other jets turn on as t approaches t_f, all of which turn off at $t = t_f$.

In summary, the necessary conditions for optimality provide that 1) no jet burns twice, and 2) firings will be of the pattern illustrated in Fig. 3. This behavior is sketched, along with the corresponding attitude motion, in Fig. 4 (see also Sec. IV).

It is useful to define jet mapping arrays that identify the acceleration and deceleration jets. Thus, J_{Ai} is the jet ID number of the ith acceleration jet and J_{Di} identifies the ith deceleration jet, $\bar{\alpha}_{J_{Ai}}$ is the acceleration vector of the ith acceleration jet, etc. Similarly, the switch-off time for

the ith acceleration jet is denoted t_{Ai}, and the switch-on time for the ith deceleration jet is denoted t_{Di}.

Optimal trajectories can then be expressed in terms of initial attitude states and jet burn times by integrating the system dynamic equations while following the optimal jet firing patterns. This integration gives expressions for the coast rate $\bar{\omega}_c$ and final conditions $\bar{\omega}_f$ and $\bar{\theta}_f$ that incorporate the conditions for optimality, except for the condition that no jet turns on twice. These are

$$\bar{\omega}_c = \sum_{i=1}^{n_A} \bar{\alpha}_{J_{Ai}} t_{Ai} + \bar{\omega}_0 \tag{15}$$

$$\bar{\omega}_f = \sum_{i=1}^{n_D} \bar{\alpha}_{J_{Di}} (t_f - t_{Di}) + \bar{\omega}_c \tag{16}$$

$$\bar{\theta}_f = \sum_{i=1}^{n_A} \bar{\alpha}_{J_{Ai}} \left[t_f t_{Ai} - \frac{1}{2} t_{Ai}^2 \right] + \sum_{i=1}^{n_D} \frac{1}{2} \bar{\alpha}_{J_{Di}} \left[(t_f - t_{Di})^2 \right] + \bar{\omega}_0 t_f + \bar{\theta}_0 \tag{17}$$

where n_A and n_D denote the number of acceleration and deceleration jets. Note that, in general, $\bar{\omega}_c$ will not parallel the maneuver eigenaxis. Solving for a particular maneuver trajectory amounts to finding values for $\bar{\omega}_c$ that cause $\bar{\theta}_f$ to equal $\bar{\theta}_{\text{targ}}$, as discussed in the next section.

Transformation between the burn switch times t_{Ai} and t_{Di} and initial adjoint parameters follows from the corner conditions, which yield

$$t_i = \frac{\lambda_{\omega_0} \cdot \bar{\alpha}_i + 1}{\bar{\lambda}_\theta \cdot \bar{\alpha}_i} \tag{18}$$

where t_i is the switch time for the ith jet, regardless of whether the jet is for acceleration or deceleration. The initial adjoints are expressed as

$$\begin{bmatrix} \bar{\lambda}_{\omega_0} \\ \bar{\lambda}_\theta \end{bmatrix} = \left[\begin{array}{c|c} -\bar{\alpha}_1^T & \bar{\alpha}_1^T t_i \\ \vdots & \vdots \\ -\bar{\alpha}_6^T & \bar{\alpha}_6^T t_6 \end{array} \right]^{-1} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{19}$$

Several spacecraft, including the Orbiter, use eigenaxis (single axis) rotations directly between the initial and target attitudes as a basis for generating maneuver commands. An advantage of this approach is that the kinematics of true single-axis rotations are linear, enabling simple prediction of inertial maneuver targets in body states. A disadvantage of eigenaxis controllers is that, in order to produce accelerations that track the eigenaxis, jet pulsing may be required. In the case of the current Orbiter controller, this pulsing reduces efficiency, shortens jet lifetime, and makes it difficult to predict acceleration levels and maneuver times. Also, eigenaxis trajectories do not satisfy all boundary conditions for maneuvers between states with rates not parallel to the eigenaxis. To compensate for this, the Orbiter controller continuously recomputes the eigenaxis during each maneuver, changing maneuver commands accordingly.

The min-fuel trajectories discussed in this section allow the maneuver to deviate from the eigenaxis during acceleration and deceleration, making up the difference by using an optimized single-rotation axis $\bar{\omega}_c$ for the coast (hence the name ORA). Using these ORA trajectories as the basis for a maneuver controller offers advantages of nonpulsed, optimal rate changes, satisfaction of body-attitude and rate boundary conditions, and highly accurate prediction of maneuver acceleration levels and times. Because the ORA trajectories differ from the eigenaxis, however, our assumption of linear kinematics for determination of $\bar{\theta}_{\text{targ}}$ becomes a source of error. The magnitude of this error depends on the extent to which the maneuver deviates from the eigenaxis, which is primarily a function of the spacecraft control jet orientation and acceleration level relative to the commanded rate changes. In Orbiter operations, this error exceeds attitude-hold deadbands in extreme cases, such as high-rate VRCS maneuvers with heavy payloads held on the RMS arm. In such cases, targeting errors are recognized by the controller, which commands the

```
procedure: TRAJECTORY_COMPUTATION (q̄0,q̄targ,ω̄mnvr,mnvr mode)
assign(t̄a,J̄a,t̄d,J̄d,tf);
    tf=fcn(q̄0,q̄f,ω̄mnvr);
    irecomp=0;
restart:
    irecomp= irecomp+1;
    call TARGET_COMPUTATION(ω̄0,q̄0,ω̄f,q̄targ,tf,mnvr mode)
    assign(ω̄c,θ̄targ);
    do until Δθ<0.1°;
        call TRAJECTORY_JET_SELECT(ω̄0,ω̄c) assign(t̄a,Ta,J̄a);
        call TRAJECTORY_JET_SELECT(ω̄c,ω̄f) assign(t̄d,Td,J̄d);
        tcoast=tf-Ta-Td;
        if (tcoast<5 seconds) or (ωc>1.2 ωmnvr) then
            do;
                tf=fcn(ω̄0,θ̄0,ω̄f,θ̄targ,t̄a,t̄d,J̄a,J̄d,tf,irecomp,ω̄mnvr);
                go to restart;
            end;
        θf=fcn(ω̄0,θ̄0,ω̄f,t̄a,t̄d,J̄a,J̄d,tf);
        Δθ̄=θ̄targ-θ̄f;
        if |Δθ̄|>0.1° then
            ω̄c =(dω̄c/dθ̄f) Δθ̄+ ω̄c;
    end;
close TRAJECTORY_COMPUTATION;
```

Fig. 6 Trajectory solution algorithm in pseudocode.

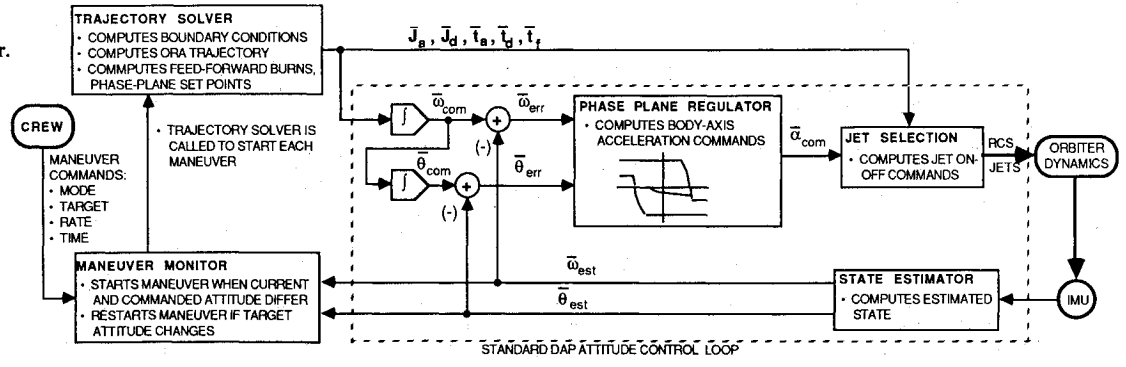

Fig. 7 ORA controller.

maneuver to be recomputed near its conclusion; targets are subsequently achieved in good order (see Sec. IV).

For comparison's sake, we have also computed a few optimal trajectories using fully nonlinear spacecraft dynamics. These solutions were obtained by a relaxation process or continuation method,[13,14] which is neither robust nor practically implemented in a digital autopilot. Starting iterates for initial adjoint variables are those obtained from Eq. (19), which is essentially the solution of Eq. (2) after zeroing the nonlinearities, and is subject to the firing pattern discussed previously. Adjustment of the initial adjoints is done by analytically computing partial derivatives of the final states with respect to the jet switch times, followed by adjustment of the switch times and reapplication of Eq. (19). Participation of the nonlinearities in Eq. (2) is increased by varying a multiplier of them from 0 to 1 in a systematic fashion, thus creating a sequence of neighboring optimal solutions that converge to the solution of the fully nonlinear problem. This technique works best, however, for weakly nonlinear cases; i.e., those involving low rates or rotations about principal axes. These nonlinear solutions differ from the ORA trajectories by having nonconstant coast rates connecting the acceleration and deceleration firings.

III. Solving for Model Trajectories

In the preceding section, Eqs. (15-17) were derived, which express the final body-attitude $\bar{\theta}_f$ for an ORA trajectory in terms of boundary conditions $\bar{\omega}_0$, $\bar{\theta}_0$, and $\bar{\omega}_f$, and jets and burn times J_{Ai}, J_{Di}, t_{Ai}, t_{Di}, and t_f. These equations imbed a necessary condition for optimality; namely, the optimized jet firing pattern (Fig. 4). The second necessary condition is satisfied for any solution that does not fire the same jet for both acceleration and deceleration. Closed-form analytical solution of these equations, in terms of $\bar{\theta}_{\text{targ}}$, cannot be obtained for the fully coupled, asymmetric, three-axis problem. Our solutions are obtained using an iterative numerical algorithm developed for this application. This algorithm is very fast and highly reliable—characteristics achieved by dividing the nonlinear problem into two linear subproblems.

The three elements of the coast rate $\bar{\omega}_c$ are used as optimization variables. The first step of the algorithm is to estimate a value for t_f that will allow Eq. (8) to be approximately satisfied, and to compute a starting iterate for $\bar{\omega}_c$, based on a single-axis approximation of the maneuver. Having chosen a value for $\bar{\omega}_c$, the first, inner subproblem is to select the jets, J_{Ai} and J_{Di}, and burn times, t_{Ai} and t_{Di}, that will satisfy Eqs. (15) and (16) while minimizing total jet on-time. These are used in Eq. (17), together with the boundary conditions and t_f, to calculate the final body-attitude $\bar{\theta}_f$, which will differ at first from $\bar{\theta}_{\text{targ}}$. The second, outer subproblem is to find values for $\bar{\omega}_c$ that cause $\bar{\theta}_f$ to be driven to equal $\bar{\theta}_{\text{targ}}$.

The jet selection problem is simplified by assuming that three or fewer jets will be used for any acceleration or deceleration burn. This assumption loosely reflects a physical plumbing constraint on the Orbiter jets (in general, it is preferable to use all jets pointing in a desired direction, because this reduces the time required for a given rate change). Equations (15) and (16) can be rewritten to solve for burn times in terms of an acceleration matrix A_{ijk} whose column vectors are the angular acceleration vectors for jets i, j, and k. Then the jet selection can be formulated as the following linear programming problem. Using any eligible VRCS or PRCS jets, solve

$$\left[t_i t_j t_k\right]^T = \left[A_{ijk}\right]^{-1}\left(\bar{\omega}_f - \bar{\omega}_0\right) \tag{20}$$

subject to

$$t_i, t_j, t_k > 0 \tag{21}$$

while minimizing

$$T = t_i + t_j + t_k \tag{22}$$

There exist several methods for solving this problem. A modified search approach that provides a good compromise between fast computation and compact code is used herein. This algorithm is summarized in Fig. 5.

The speed of this search procedure is improved by restricting it to examine only those three-jet combinations that are optimal for some rate change. Which three-jet combinations are optimal depends on the mass properties and RCS geometry of the spacecraft: Jets that oppose each other, or that produce little net acceleration, will never be chosen together. Thus for the VRCS, with six jets, there exists a total of $6!/3!(6-3)! = 20$ possible three-jet combinations. For normal Orbiter mass properties, only eight of these possible combinations are actually used.

In PRCS modes, the search for the best three-jet-group combination for a particular rate change is reduced further, by screening out combinations based on the direction of the rate change request. Thus combinations that are never selected for + pitch commands are not searched when the rate change request includes a + pitch component, and so on. The result is that the number of three-jet-group combinations eligible for a particular rate change is reduced from a maximum of 19 to between 3 and 8, so that the selection of PRCS jets for ORA maneuvers is, in general, faster than the selection of VRCS jets.

The outer loop uses Newton's method, as shown in Fig. 6, to find values for $\bar{\omega}_c$, which cause $\bar{\theta}_f$ to be driven to equal $\bar{\theta}_{\text{targ}}$. Successive iterates of $\bar{\omega}_c$ are computed, based on the error due to the previous $\bar{\omega}_c$, as follows:

$$\bar{\omega}_c = \frac{\mathrm{d}\,\bar{\omega}_c}{\mathrm{d}\,\bar{\theta}_f}\left(\bar{\theta}_{\text{targ}} - \bar{\theta}_f\right) + \bar{\omega}_{c_{\text{previous}}} \tag{23}$$

where

$$\frac{\mathrm{d}\,\bar{\omega}_c}{\mathrm{d}\,\bar{\theta}_f} = \left\{\left(\left[A_{ijk}\right]\begin{bmatrix}(t_f - t_i) & 0 & 0\\ 0 & (t_f - t_j) & 0\\ 0 & 0 & (t_f - t_k)\end{bmatrix}\left[A_{ijk}\right]^{-1}\right)_{\text{accel}} - \left(\left[A_{ijk}\right]\begin{bmatrix}t_i & 0 & 0\\ 0 & t_j & 0\\ 0 & 0 & t_k\end{bmatrix}\left[A_{ijk}\right]^{-1}\right)_{\text{decel}}\right\}^{-1} \tag{24}$$

The new $\bar{\omega}_c$ is used in the subsequent iteration to compute a new $\bar{\theta}_f$ closer to $\bar{\theta}_{\text{targ}}$; the process is repeated until it converges, usually in two or three iterations.

Determination of the starting values of t_f and $\bar{\omega}_c$ to start the outer loop depends on the initial and projected final inertial attitudes, $\bar{q}_o$ and $\bar{q}_f$, on the requested maneuver rate ω_{mnvr}, and on the current maneuver mode. Standard Orbiter maneuver modes are inertial (rest-to-rest) maneuvers relative to the local vertical (LVLH), and maneuvers-to-a-track. The latter mode rotates the Orbiter to an attitude and rate that point a specified body vector at a specified external object. Such objects include: objects in an Earth orbit, the center of the Earth, Earth landmarks, the sun or a star.

For an inertial maneuver, the starting value of t_f is estimated as the time required for a coast at ω_{mnvr} about the eigenaxis connecting the specified $\bar{q}_0$ and $\bar{q}_f$, plus a simple estimate of the time required to accelerate and decelerate. This

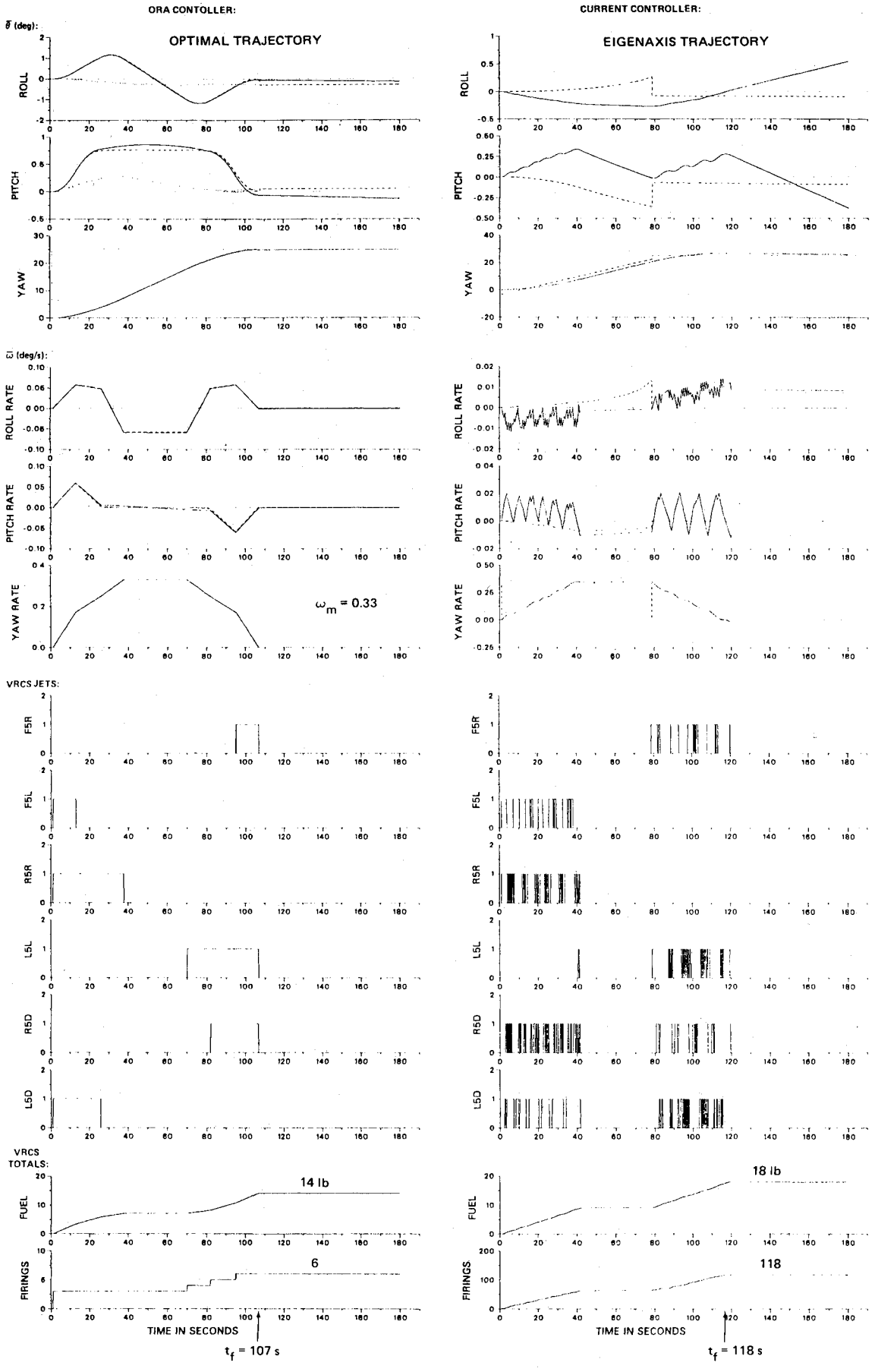

Fig. 8 A 25-deg VRCS yaw maneuver with $\omega_{mnvr} = 0.33$ deg/s.

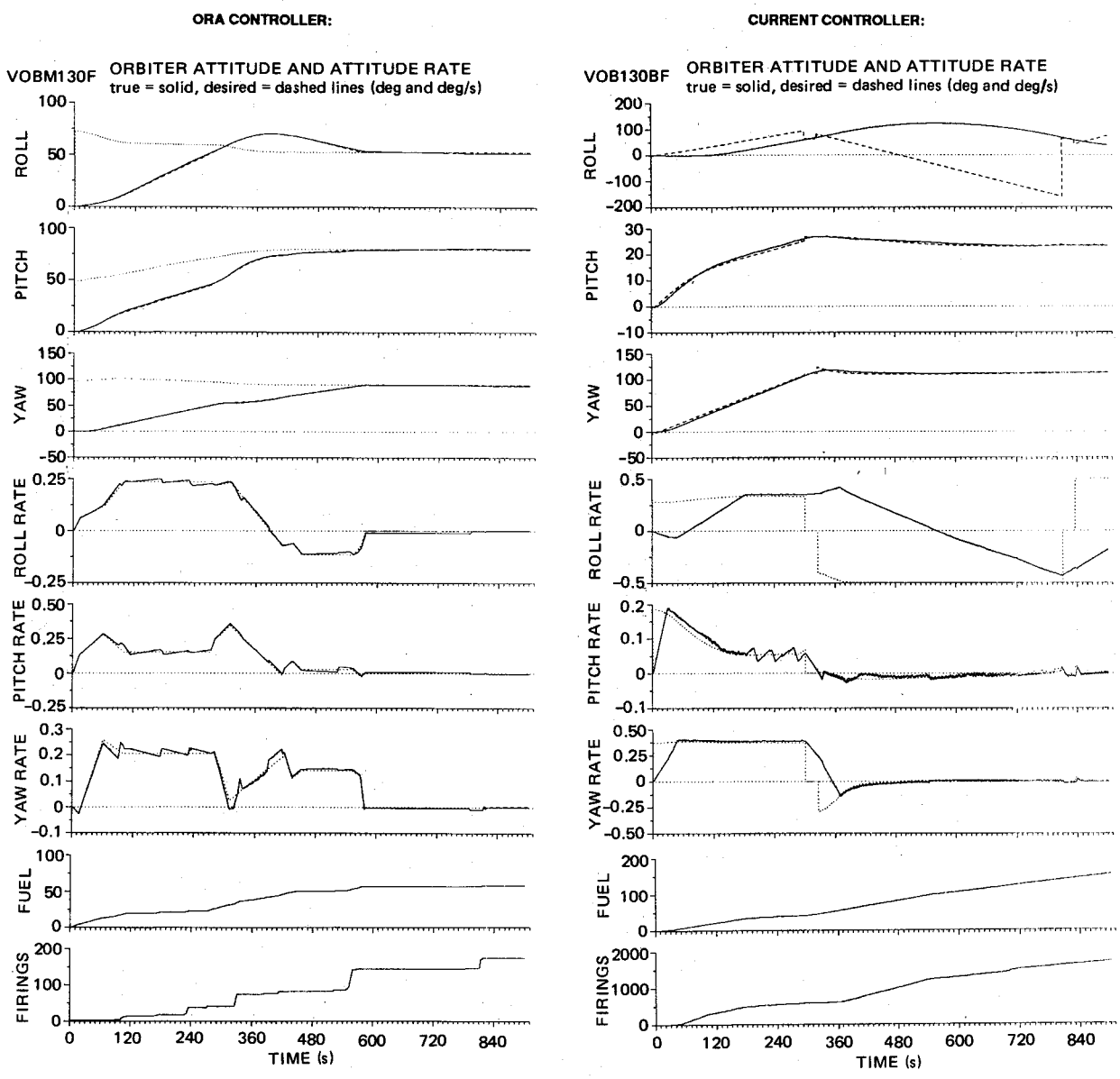

Fig. 9 A 130-deg VRCS maneuver with a 32,000-lb payload and $\omega_{mnvr} = 0.5$ deg/s.

estimate is usually accurate enough so that Eq. (8) is satisfied within 10%. In some more taxing VRCS maneuvers, Eq. (8) is not satisfied to within 20%. In these cases, t_f is recomputed using much more accurate estimates of acceleration level obtained from the first round of jet selections, and the algorithm is restarted (Fig. 6). A second condition for restarting the algorithm is attained when ω_{mnvr} is too large for the maneuver angle and available acceleration, so that the acceleration and deceleration periods overlap and there is no coast period. When this occurs, t_f is recomputed to allow a lower maneuver rate and at least a 5-s coast period. This condition also has the effect of enforcing the linearity of the outer loop, as $\bar{\theta}_f$ is a more linear function of $\bar{\omega}_c$ when burn times are short.

For maneuvers to track moving targets, the final attitude $\bar{q}_f$ itself depends on t_f. In these cases, the starting value for t_f is computed based on the line of sight to the object at the maneuver start time. Then $\bar{q}_f$ is determined by extrapolating the position of the target relative to the Orbiter at time t_f. This is straightforward for all but orbiting objects; these require solving Kepler's equation in an iterative fashion.[15,16] When the relative position of the target at t_f is known, an inertial rotation to intercept the desired track attitude and rate is computed, defining final conditions for the maneuver. The remainder of the trajectory calculation proceeds as for inertial maneuvers. If the computation is restarted, new, more accurate target conditions, based on improved estimates of acceleration levels and target location, are used.

Experience with this algorithm shows it to be highly reliable. By automatically checking that the acceleration and deceleration phases do not overlap, the robustness of the outer loop is assured. While not guaranteeing global optimality, the inner, jet selection problem is linear, robust, and very fast. The result is that, in roughly 200,000 test maneuver calculations, estimated maximum computation time was 128 ms of CPU time on the Orbiter on-board computers. Real time required for trajectory computation is greater than this, however. In order to keep total DAP CPU time requirements below about 15 ms per 80 ms DAP cycle, the algorithm is split into several subfunctions. These are executed sequentially, over three or more DAP cycles. Average real time required ranges from 320 to 960 ms, depending on RCS option and maneuver mode. Maximum real time observed thus far is 1.84 s.

The ORA controller consists of three major modules. The first is the trajectory solution algorithm just described. The

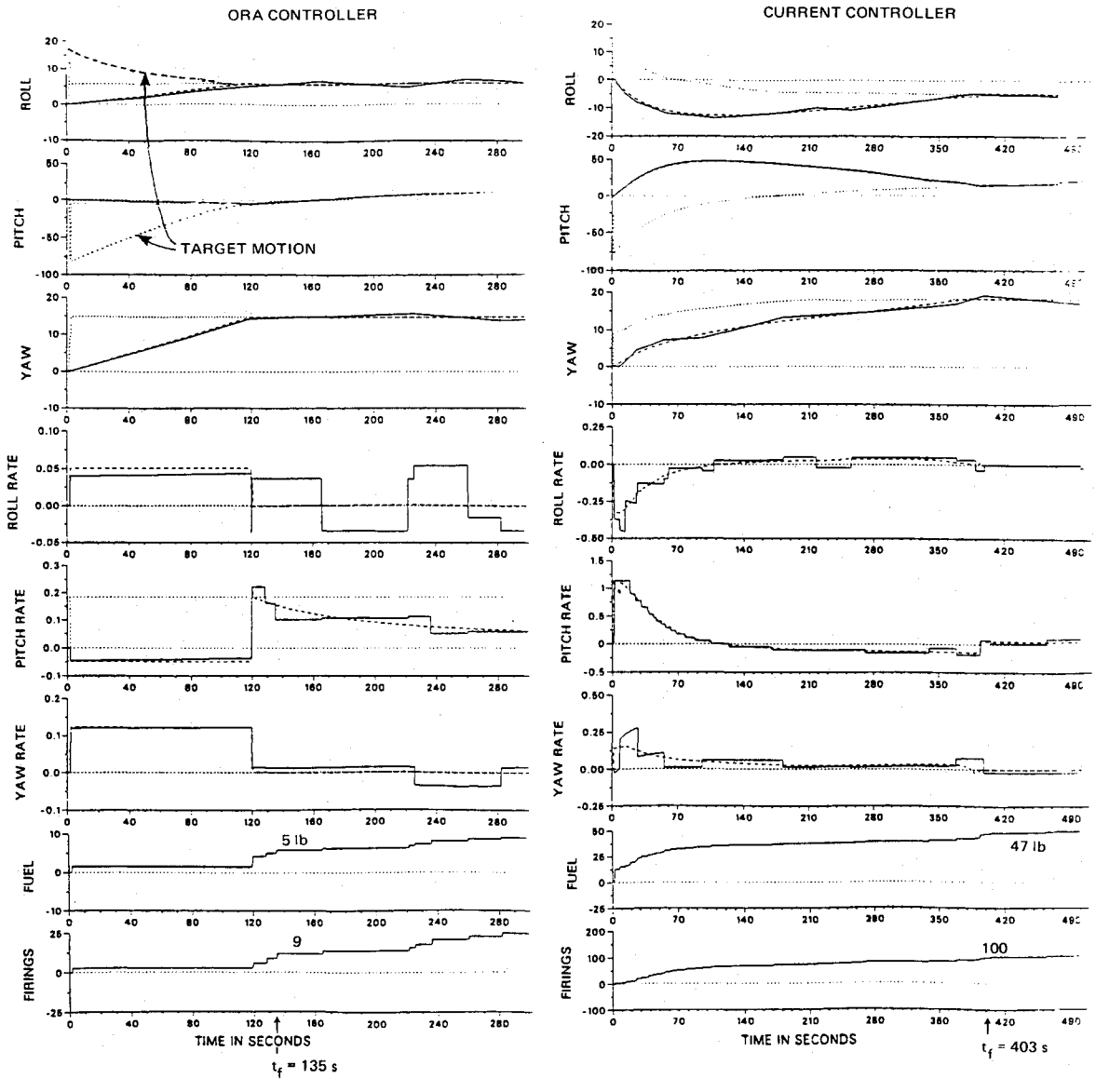

Fig. 10 PRCS maneuver to track a landmark directly under the Orbiter at $t = 0$.

second is an ORA data initialization function. The third is a maneuver monitor procedure, which is executed cyclically whenever the DAP is in automatic mode, comparing current and commanded attitude to determine when a maneuver is required. It initiates maneuvers by calling the trajectory solver, which computes specific maneuver commands (see Fig. 7).

The initialization function is required because the ORA controller supports a total of 10 different RCS options for maneuver firings. These include six VRCS modes, corresponding to normal and alternate mass-properties operating modes, and four PRCS modes. The first of these allows unrestricted jet selection, the second preferentially selects forward jets, and the third selects aft jets only. These modes are used primarily to manage propellant distribution. The fourth PRCS mode selects only those jets that do not point upward, and is used in operations near other spacecraft.

The initialization function is executed whenever the crew changes RCS option. It takes prestored information defining which three-jet combinations can be used in the new RCS mode, and computes an array of jet pointers and an array of the $[A_{ijk}]^{-1}$ matrices for each eligible combination, using standard prestored angular acceleration data. This process converts required data from compact structures used to minimize storage, to structures that allow rapid computation of trajectories. Similar to the trajectory solver, it is split into several functions, each of which is executed in a subsequent 80-ms DAP cycle, until the initialization is complete. The initialization process takes between 2 and 12 80-ms DAP cycles, or from 160 to 960 ms of real time, depending on RCS mode. Total DAP CPU time per cycle during this process remains below 15 ms.

Starting and stopping automatic maneuvers is the function of the maneuver monitor procedure, which is run at a 1.04-Hz rate while the DAP is in automatic mode. When not maneuvering, it continuously compares the current Orbiter attitude with the current target attitude. If these differ by a specified amount, it starts a maneuver by invoking the model trajectory computation function. During maneuvers, it continuously compares the maneuver target attitude with the current target attitude. If the maneuver target has changed by a sufficient amount, the maneuver is restarted at an appropriate time. Following each maneuver, the maneuver monitor switches to attitude hold or track mode.

IV. Results

In this section, three detailed example maneuvers—illustrating typical performance of the ORA controller and comparing it with the current controller—are presented as body-axis time histories, showing the evolution of $\bar{\theta}$ and $\bar{\omega}$ throughout each maneuver. These are shown in Figs. 8-10 as solid lines. The

commanded and target states are also shown as dashed and dotted lines, respectively. The examples were computed using a rigid-body engineering simulation program that does not include atmospheric drag or other environmental disturbance torques. The results compare very closely with results from other, more complete simulations.

The first example (Fig. 8) is a straightforward, 25-deg yaw maneuver chosen to illustrate normal performance of both the ORA and current DAP controllers. The current controller moves the Orbiter along the maneuver eigenaxis, pulsing jets so that the pitch and roll states remain near zero. It completes the maneuver in 120 s using 18 lb of fuel and 118 jet firings. The ORA controller improves these figures, completing the maneuver in 107 s using 14 lb of fuel and only six jet firings. This improvement comes from allowing the roll and pitch states to move away from the eigenaxis in optimal acceleration and deceleration burns, yielding higher overall yaw acceleration at a lower fuel cost.

The second example (Fig. 9) is a three-axis maneuver with a 32,000-lb cylindrical payload held vertically on the RMS arm. It is an extreme maneuver, performed at a 0.5-deg/s rate (well beyond operational limits), and in a direction chosen to provide near worst-case coupling effects. The effect using the ORA controller is to cause the body-attitude target $\bar{\theta}_{\text{targ}}$ to "drift," as the maneuver deviates considerably from the eigenaxis during the 100-s acceleration phase. The target states are shown as dotted lines in Fig. 9. This error is recognized by the controller, which recomputes the maneuver at about 280 s, and again at about 415 s, before achieving the target at 577 s. Fuel expenditure was 57.2 lb; the VRCS jets were cycled 160 times.

The standard controller does not perform nearly as well in this example. The jet pulsing required to track the eigenaxis has the effect of drastically reducing roll control authority in a way not accounted for by the controller. As a result, the controller is unable to estimate the maneuver deceleration time accurately enough to avoid a track overshoot, and the maneuver is repeatedly recycled. Final time for the maneuver was well in excess of 900 s, fuel use was in excess of 155 lb, and firings were in excess of 1800.

The improved modeling of the Orbiter by the ORA allows greatly improved performance in all payload-extended cases, not just this extreme case.

The final example (Fig. 10) is a maneuver to track an Earth landmark using the PRCS. At the start of the maneuver the landmark to be tracked is directly under the Orbiter, which has a large, mostly pitch, initial error to overcome before achieving the commanded track by pointing the payload bay at the target. The track is specified as a three-axis track, with the nose of the Orbiter pointing nominally forward in the orbital plane.

The current controller commands the Orbiter to rotate around the instantaneous eigenaxis between the current attitude and the current maneuver target; as shown, the maneuver target changes significantly as the Orbiter moves past the landmark. The commanded rate is the maneuver rate magnitude times the eigenaxis, added to the target line-of-sight rate. As a result, the maneuver starts out in the wrong direction, delaying completion of the maneuver to 403 s. Fuel cost to complete the maneuver and establish the landmark track is 47 lb; 100 firings were required.

The ORA controller performs much better, mainly because it extrapolates the target line of sight to the maneuver end time and computes the maneuver to intercept, rather than chase, the desired track attitude. In this case, a relatively small maneuver is required, with time, fuel, and firing figures of 135 s, 5 lb, and 9 firings, respectively.

V. Conclusion

Optimal attitude maneuvers for linearized spacecraft dynamics controlled by on-off jets were considered, deriving an optimal jet firing sequence and showing that these trajectories differ from single-axis maneuvers. A parametric solution algorithm was presented that is capable of solving these trajectories in real time in the Orbiter on-board computers. A feed-forward, feedback controller was developed to exploit this capability. In comparison with the current Orbiter controller, this Optimized Rotation-Axis maneuver controller has shown reduction in average maneuver fuel usage of 40%, reduction in jet firing cycles of 70%, and reduction in maneuver time of 10%, in over 100 simulated maneuvers. We anticipate an overall reduction in vernier jet fuel of 26% per mission, and a reduction in vernier jet firing cycles of 51% per mission.

The Optimized Rotation-Axis controller is applicable to jet-controlled spacecraft for which Euler coupling torques are significantly less than control torques. As demonstrated on the Orbiter, it overcomes problems such as variable, three-axis coupled control jet torques, low-thrust jet torques, and multiple modes and tasks, using limited-capability on-board computers, to provide excellent attitude maneuver performance.

Acknowledgments

The authors wish to thank Ed Kubiak and Ken Lindsay of NASA, and Harvey Malchow, Darryl Sargent, Phil Hattis, Nazareth Bedrossian, and Ed Bergmann of the Charles Draper Laboratory Inc. for their contribution to this work. Thanks also to Karen Mills for preparing the manuscript. This paper was prepared by the Charles Stark Draper Laboratory Inc. under Contract NAS9-16023 with the National Aeronautics and Space Administration. The work was supported by NASA Johnson Space Center. Publication of this report does not constitute approval by NASA of findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

References

[1]Hattis, P., "A Review of the Space Shuttle Orbital Flight Control System," Charles Stark Draper Laboratory Rept. CSDL-P-1786, Oct. 1983.

[2]"Space Shuttle Operational, Level C, Functional Subsystem Software Requirements; Guidance, Navigation and Control, Part C, Flight Control, Orbit DAP," Rockwell International Rept. STS83-0009, Change No. PCN 3, June 15, 1984.

[3]Kubiak, E.T. and Martin, M.W., "Minimum Impulse Limit Cycle Design to Compensate for Measurement Uncertainties," *Journal of Guidance, Control and Dynamics*, Vol. 6, Nov.-Dec. 1983, pp. 432-435.

[4]Bryson, A.E. and Ho, Y.C., *Applied Optimal Control*, revised edition, Hemisphere, New York, 1975.

[5]Dixon, M.V., Edelbaum, T.N., Potter, J.E., and Vandervelde, W.E., "Fuel Optimal Reorientation of an Axisymmetric Spacecraft," *Journal of Spacecraft and Rockets*, Vol. 7, No. 11, Nov. 1970, pp. 1345-1351.

[6]Dwyer, T.A.W., "Exact Nonlinear Control of Large Angle Rotational Maneuvers," *IEEE Transactions on Automatic Control*, Vol. AC-29, No. 9, Sept. 1984, pp. 769-774.

[7]Carrington, C.T. and Junkins, J.L., "Nonlinear Feedback Control of Spacecraft Slew Maneuvers," *Journal of the Astronautical Sciences*, Vol. 32, Jan. 1984, pp. 29-46.

[8]Zwartbol, T., Hameetman, G.J., Slippers, C.P.R.C., and Terpstra, A.P., "Experiments in Modern Control On-Board IRAS," AIAA Paper 84-1881, *Proceedings of the AIAA Guidance and Control Conference*, Aug. 1984, pp. 302-315.

[9]Wie, B. and Barba, P.M., "Quaternion Feedback for Spacecraft Large Angle Maneuvers," *Journal of Guidance, Control and Dynamics*, Vol. 8, May-June 1985, pp. 360-365.

[10]Vadali, S.R., "Feedback Control of Spacecraft Large-Angle Maneuvers Using Reaction Wheels and On-Off Thrusters," AIAA Paper 84-1031-CP, *Proceedings of the AIAA Dynamic Specialists Conference*, Feb. 1984.

[11]Crisp, R. and Keene, D., "Attitude Maneuver Optimization to Conserve Reaction Control Propellants," Charles Stark Draper Laboratory Rept. CSDL-E-1832, Aug. 1965.

[12]D'Amario, L.A. and Stubbs, G.S., "A New Single-Rotation-Axis Autopilot for Rapid-Spacecraft Attitude Maneuvers," *Journal of Guidance and Control*, Vol. 2, July-Aug. 1979, pp. 339-346.

[13]Turner, J.D. and Junkins, J.L., "Optimal Large-Angle Single-Axis Rotational Maneuvers of Flexible Spacecraft," *Journal of Guidance and Control*, Vol. 3, Jan.-Feb. 1980, pp.

[14]Turner, J.D., Chun, H.M., and Junkins, J.L., "Optimal Large-Angle Maneuvers with Vibration Suppression," Charles Stark Draper Laboratory Rept. 83-056, 1983, pp. 578-583.

[15]Adams, N.J., "The Target State Prescriber for the ORA Automaneuver Algorithm," Charles Stark Draper Laboratory Rept. CSDL-R-1779, May 1985.

[16]Battin, R.H. and Fill, T.J., "Extension of Gauss' Method for the Solution of Kepler's Equation," *Journal of Guidance and Control*, Vol. 2, May-June 1979, pp. 190-195.