# Onboard Computer Requirements for Navigation of a Spinning and Maneuvering Vehicle

W. E. VANDER VELDE*
*Massachusetts Institute of Technology, Cambridge, Mass.*

AND

G. K. BENTLEY†
*Avco Corporation, Wilmington, Mass.*

AND

J. H. FAGAN‡
*The Analytic Sciences Corporation, Reading, Mass.*

AND

W. T. MCDONALD§
*North American Rockwell Corporation, Anaheim, Calif.*

**This paper discusses the onboard computer requirements for an inertial navigator that uses a single-axis platform inertial measurement unit and navigates a vehicle during spinning free-fall flight and maneuvering atmospheric flight. The object of the study is to determine design values for the computer parameters which achieve an orientation accuracy of several milliradians and a navigation accuracy of about 1000 ft at the end of this flight. The operating parameters for the coordinate transformation and navigation computers are predicted analytically and verified by use of a flight simulation computer program, which, within the context of a vehicle/trajectory simulation program, permits the simulation of different forms of onboard computers. The computer simulations are performed using fixed-point arithmetic with all variables appropriately scaled, and with controllable word length and cycle time. The principal sources of computation errors in each phase are identified, and the computer design parameters required to constrain the errors to the specified limits are determined. Results indicate that the onboard computer requirements for unaided vehicle navigation from deployment to impact do not pose a significant challenge to current technology.**

## Introduction

THIS paper reports the results of a study conducted to gain insight into the inertial navigator requirements for the onboard computer for maneuvering re-entry vehicle missions. The study was quite general in nature. No attempt was made to determine worst-case trajectory requirements. However, two possible re-entry flight paths were chosen as design points for study purposes. The two trajectories, one in-plane and one out-of-plane, are shown in Fig. 1. The vehicles were assumed to be spin-stabilized at 100 rpm during the free-flight phase, and provision was made to simulate vehicle coning motion caused by a transverse angular momentum component imparted by the spin-up or ejection mechanism at booster cutoff. The purpose of the study was to determine the sensitivity of impact error to computer-induced errors.

Three inertial measurement unit (IMU) configurations were considered: a three-axis stabilized platform, a full strapdown system, and a partial strapdown system employing a single-axis platform stabilized about the vehicle roll (spin) axis. A fixed-point simulation of an onboard computer navigation algorithm was provided, and several candidate digital differential analyzer (DDA) and general purpose (GP) algorithms for the strapdown transformation matrix were simulated. Full control over both DDA and GP word length, parameter scaling, and cycle time was provided.

The computer error propagation was studied for the free-flight phase from booster cutoff to a re-entry altitude of 300,000 ft, and for the atmospheric phase from that point to impact, during which one of the maneuvers in Fig. 1 is executed. The navigation computations were initialized at booster cutoff and continued to impact. A perfect IMU was assumed for the purpose of evaluating computer errors; no instrument error sources or instrument response dynamics enter the results discussed here. The computer error sources were considered singly and in concert. An allowable impact miss contribution, due to computer errors only, of 500 to 1000 ft was arbitrarily assumed.

The results presented here are applicable to two of the candidate IMU configurations, the three-axis platform and the single-axis platform. In the spin-stabilized vehicle, the full strapdown system exhibits a very critical roll error sensitivity to scale factor uncertainty caused by the high vehicle spin rate (100 rpm) and the long free-flight time (approximately 30 min). The full strapdown mechanization was therefore removed from consideration in these studies. The single-axis platform isolates the IMU from the spin motion,

Presented as Paper 68-839 at the AIAA Guidance, Control, and Flight Dynamics Conference, Pasadena, Calif., August 12–14, 1968; submitted October 30, 1968; revision received August 11, 1969.

* Professor, Department of Aeronautics and Astronautics; also, Consultant, Avco Corporation, Wilmington, Mass. Member AIAA.

† Group Leader, Guidance and Control Department. Member AIAA.

‡ Staff Engineer.

§ Senior Staff Scientist.

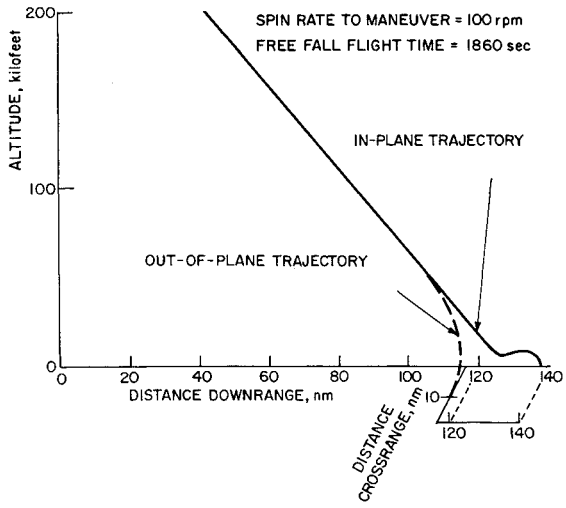


**Fig. 1 Trajectory profiles.**

but the system remains strapped down about the pitch and yaw directions. The transformation matrix computations proceed with the roll rate simply nulled.

Three computation algorithms for the transformation matrix were tested, one DDA and two single-point (only the last values of integrated quantities are stored) GP algorithms. A single algorithm for the navigation computations was used for both candidate mechanizations.

## Computation Algorithms and Parameter Scaling

### Midpoint-$g$ Navigation Algorithm

The vehicle navigation equations to be integrated are given in vector form by

$$\dot{\mathbf{r}} = \mathbf{v}, \quad \dot{\mathbf{v}} = \mathbf{g}(\mathbf{r}) + P\mathbf{a}_a \tag{1}$$

where $\mathbf{r}$ and $\mathbf{v}$ are the vehicle position and velocity in inertial coordinates, $\mathbf{g}(\mathbf{r})$ is the gravitational field expressed in the same coordinates, $\mathbf{a}_a$ is the nongravitational acceleration measured by the accelerometers in their natural coordinates, and $P$ is the transformation matrix relating accelerometer and inertial coordinates. In the three-axis mechanization $P$ is a constant matrix; in the single-axis system $P$ is the solution of

$$\dot{P} = P\Omega \tag{2}$$

where $\Omega$ is the skew symmetric matrix of platform angular velocities

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & 0 \\ -\omega_y & 0 & 0 \end{bmatrix} \tag{3}$$

The single-axis platform is inertially stabilized about its $x$ axis, which is the vehicle roll axis.

Equation (1) is integrated from specified initial conditions by the navigation algorithm. Let

$$\Delta\mathbf{v}_k = \int_{t_{k-1}}^{t_k} P\mathbf{a}_a dt \tag{4}$$

which is the integral of the nongravitational specific force (in inertial coordinates) sustained by the vehicle within the GP navigation cycle time $\Delta t_{\text{nav}}$;

$$\Delta t_{\text{nav}} = t_k - t_{k-1} \tag{5}$$

The accelerometers actually read out quantized indications of the integral in Eq. (4). For the three-axis IMU mechanization in which $P$ is constant, Eq. (4) becomes simply the constant transformation of the instrument output quanta accrued during the cycle time

$$\Delta\mathbf{v}_k = P\Delta\mathbf{v}_{a_k} = P\int_{t_{k-1}}^{t_k} \mathbf{a}_a dt \tag{6}$$

For the single-axis IMU, the integration of Eqs. (4) and (2) is done in the transformation computations.

The midpoint-$g$ algorithm uses a trapezoidal integration rule. The vector $\mathbf{g}$ is computed at the midpoint of the integration interval, with a simple predictor providing the position $\mathbf{r}$ at that point. The equations are

$$\mathbf{g}_{k-1/2} = \mathbf{g}(\mathbf{r}_{k-1/2}) \tag{7}$$

$$\mathbf{r}_{k-1/2} = \mathbf{r}_{k-1} + \mathbf{v}_{k-1}\Delta t/2 + \Delta\mathbf{v}_k\Delta t_{\text{nav}}/8 \tag{8}$$

$$\mathbf{v}_k = \mathbf{v}_{k-1} + \Delta\mathbf{v}_k + \mathbf{g}_{k-1/2}\Delta t_{\text{nav}} \tag{9}$$

$$\mathbf{r}_k = \mathbf{r}_{k-1} + (\mathbf{v}_{k-1} + \mathbf{v}_k)\Delta t_{\text{nav}}/2 \tag{10}$$

The gravity model used in Eq. (7) at least contains an inverse-square, central force term

$$\mathbf{g} = -(\mu/r^3)\mathbf{r} \tag{11}$$

In this study only the central force term was used, although for real missions the model would contain higher-order terms to match the actual earth gravitational field. A recursive computation of $1/r$ avoids the division requirement in Eq. (11),

$$(1/r)_{k-1/2} = (1/r)_{k-3/2}[3 - r_{k-1/2}^2(1/r)_{k-3/2}^2]/2 \tag{12}$$

This is the first cycle of a Newton-Raphson iteration which solves for $1/r$ from $r^2$. Since $r$ changes slowly, sufficient accuracy is obtained. For a change in $r$ of 2000 ft between sample points the corresponding error in $\mathbf{g}_{k-1/2}$, using Eq. (11), is about $5 \times 10^{-8}g$.

Scaling the computer parameters involves choosing the GP word length $N$ (number of bits), the integration interval $\Delta t_{\text{nav}}$, the velocity increment $\Delta v_{GP}$, and the position increment $\Delta r_{GP}$. There are complex interactions among these parameters, which are best revealed in the simulation studies, but simple arguments lead to approximate values which serve as starting points. For any choice of $N$, $\Delta v_{GP}$ and $\Delta r_{GP}$ are specified by the condition that the $N$-bit word must be able to hold the largest velocity and position magnitudes. Corresponding to these values, there exists some optimum $\Delta t_{\text{nav}}$ since the effect of roundoff error dominates for smaller values and the effect of algorithm error dominates for larger values. Since the velocity and position components vary smoothly during free fall and more violently during atmospheric maneuver, the algorithm error per integration step for a given $\Delta t_{\text{nav}}$ can be expected to be smaller during free fall than during maneuver. Since the roundoff error per step is the same, the expected optimum $\Delta t_{\text{nav}}$ will be larger during free fall than during maneuver.

Use of a random walk model for the accumulation of roundoff errors, which is probably optimistic since the errors are not really independent from step to step, gives an rms position error due to rounding the gravity integral to the nearest $\Delta v_{GP}$ of

$$\text{rms}[\epsilon(r)]_g = (t/\Delta t_{\text{nav}})^{1/2}/6 \tag{13}$$

where $t$ is the total flight time. The rms position error due to rounding the velocity integral to the nearest $\Delta r_{GP}$ is

$$\text{rms}[\epsilon(r)]_v = (t/12\Delta t_{\text{nav}})^{1/2} \tag{14}$$

If the rms position error due to each of these effects is set at 100 ft (it is realized that these are optimistic relations and that there are several other sources of computer error) and if $\Delta t_{\text{nav}}$ is considered to be of the order of 1 sec and $t$ to be 1900 sec, the specifications on $\Delta v_{GP}$ and $\Delta r_{GP}$ are found to be $\leq 0.00724$ fps and $\leq 7.95$ ft, respectively.

The maximum values of velocity and position are about 23,650 fps and $25.5 \times 10^6$ ft, respectively, which requires a

dynamic range of 1 part in 3,266,575 for velocity and 1 part in 3,207,547 for position. This dictates a word length of 22 bits to store the magnitudes of these variables, with an additional bit for sign and one for parity check. Thus, a 24-bit word length for the computer is required. With this word length, the velocity and position scaling can be as small as 0.0056 fps, and 6.08 ft, respectively.

One additional constraint is placed on $\Delta v_{GP}$. Integrated specific force enters these computations with the scaling of $\Delta v_k$ from the transformation computations. This pulse count must be multiplied by a constant to rescale it to a least bit of $\Delta v_{GP}$. In this work, the $\Delta v_{GP}$ used in the $GP$ computations is constrained to

$$\Delta v_{GP} = 2^{-m}\Delta v_k \tag{15}$$

for some positive integer $m$, so this multiplication can be accomplished simply by shifting. The integer $m$ is selected as the greatest value which satisfies the constraint that the velocity register be able to hold the maximum velocity magnitude.

Table 1 lists the candidate values of $N$, $m$, $\Delta v_{GP}$, and $\Delta r_{GP}$ for $\Delta v_k = 2.0$ fps (in-plane maneuver) and $\Delta v_k = 1.25$ fps (out-of-plane maneuver).

The remaining task is to optimize $\Delta t_{\text{nav}}$. From prior experience with the midpoint-$g$ integration rule, the best value appears to lie in the range 0.5 to 5.0 sec, with the larger values most favorable during free fall and the smaller values most favorable during atmospheric flight.

## Transformation Computation Algorithms

### *DDA algorithm*

The DDA cycle time is assumed to be the same as the instrument sampling time. The high processing rate makes a simple rectangular integration rule usable for integrating the matrix equation [Eq. (2)] and allows the transformation of the velocity pulses from the accelerometers to be done by simple multiplication. Since the matrix computation must be done throughout the free-fall phase as well as the atmospheric phase, an alternate-order serial DDA mechanization was chosen since it is known to have smaller error propagation than other mechanizations. This mechanization uses the latest available information in the computation cycle and reverses the order of the updating operation on alternate machine cycles to achieve greater precision. The matrix algorithm equations are as follows.

Odd-numbered machine cycles:

$$\begin{aligned} P_{ij}^{(k-1/2)} &= P_{ij}^{(k-1)} + P_{i,j+1}^{(k-1)}\Delta\theta_{j+2}^{(k)} \\ P_{ij}^{(k)} &= P_{ij}^{(k-1/2)} - P_{i,j+2}^{(k-1/2)}\Delta\theta_{j+1}^{(k)} \quad i,j = 1,2,3 \end{aligned} \tag{16}$$

Even-numbered machine cycles:

$$\begin{aligned} P_{ij}^{(k-1/2)} &= P_{ij}^{(k-1)} - P_{i,j+2}^{(k-1)}\Delta\theta_{i+1}^{(k)} \\ P_{ij}^{(k)} &= P_{ij}^{(k-1/2)} + P_{i,j+1}^{(k-1/2)}\Delta\theta_{j+2}^{(k)} \quad i,j = 1,2,3 \end{aligned} \tag{17}$$

For index $i$ the values 1, 2, 3 correspond to inertial $x,y,z$ axes, respectively; for index $j$ the values 1, 2, 3 correspond to body roll, pitch and yaw, respectively; and the index $j$ is treated modulo 3. The angle increments $\Delta\theta_j$ are read from the gyros, and $\Delta\theta_1$ (roll) is zero for the single-axis IMU. The transformation equations are meshed with the matrix operations in the DDA mechanization so that the most recent matrix component values are used to multiply the sampled velocity in accordance with

$$\Delta v_i^{(k)} = P_{i,j+1}^{(k)}\Delta v_{a(j+1)}^{(k)} + P_{i,j+2}^{(k)}\Delta v_{a(j+2)}^{(k)} + P_{ij}^{(k)}\Delta v_{a(j)}^{(k)} \tag{18}$$

Scaling the DDA computations requires the choice of the DDA word length $n$ (number of bits), cycle time $\Delta t_{DDA}$, angle increment $\Delta\theta$, and the velocity increment $\Delta v_a$. The constraints to be satisfied are: 1) $\Delta\theta > |\omega|_{\max}\Delta t_{DDA}$, where $|\omega|_{\max}$ is the maximum angular rate magnitude sustained by the vehicle; 2) $\Delta v_a > |a|_{\max}\Delta t_{DDA}$, where $|a|_{\max}$ is the maximum acceleration magnitude sustained by the vehicle; and 3) all multiplications must be accomplished by simple shifting. The maximum magnitude of any element $P_{ij}$ is unity. Each $P_{ij}$ is carried in two DDA registers each $(n+1)$ bits long, one bit being a sign bit. The most significant part of each $P_{ij}$ is contained in one register, and the least significant part, called the remainder, in the other. The multiplications in Eqs. (16) and (17) can be accomplished simply by adding the most significant part of one $P_{ij}$ into the remainder register for the $P_{ij}$ being updated if the angle scale factor is constrained to have the value $2^{-n}$ rad.

In the free-fall phase the vehicle undergoes coning motion. The transverse (coning) frequency is given by

$$\omega_t = \omega_s(I_s/I_t)\tan\alpha \tag{19}$$

where $\omega_s$ is the spin rate (100 rpm), $I_s/I_t$ is the ratio of the spin and transverse moments of inertia of the body (assumed to be 0.1), and $\alpha$ is the cone half-angle. The cycle time is computed from

$$\Delta t_{DDA} = \Delta\theta/\omega_t \tag{20}$$

where $\Delta\theta$ is 0.488 mrad ($n = 11$) and 0.244 mrad ($n = 12$)—values which, given the dynamics of the in-plane and out-of-plane maneuvers, respectively, satisfy the position accuracy specification. For an $\alpha$ of 10°, $\omega_t$ is 0.185 rad/sec, and $\Delta t_{DDA}$ is 1.25 msec for $\Delta\theta = 0.244$ mrad and 2.5 msec for $\Delta\theta = 0.488$ mrad. If $\alpha$ decreases to 1°, $\omega_t$ decreases and $\Delta t_{DDA}$ increases by a factor of 10.

The body rates during the maneuver may reach magnitudes of the order of 1 rad/sec, necessitating a much shorter DDA cycle time. This suggests strongly that two different cycle times, one for each phase, may be used if an advantage is derived. Otherwise, the cycle time must be set for the worst-case condition.

The precise value of the matrix element scale factor $\Delta P$ should be a little larger than $2^{-n}$ to allow for nonorthogonality error propagation that may make some element magnitude slightly greater than unity. The scale factor $\Delta v_a$ is determined by requirement 2) if the resulting value is consistent with the accuracy specifications, and the scale factor $\Delta v_i$ for the inertial velocity in Eq. (18) has the value

$$\Delta v_i = (\Delta P/\Delta\theta)\Delta v_a \tag{21}$$

The simulation studies used $\Delta v_a$ values of 2.0 and 1.25 fps, corresponding to cycle times of 0.5 and 0.25 msec, for the in-plane and out-of-plane maneuvers, respectively.

**Table 1 Navigation computer scaling**

| | $N$, bits | $m$ | $\Delta v_{GP}$, fps | $\Delta r_{GP}$, ft |
|---|---|---|---|---|
| In-plane maneuver | 24 | 8 | 0.00785 | 6.1 |
| | 25 | 9 | 0.00393 | 3.0 |
| | 26 | 10 | 0.00196 | 1.5 |
| Out-of-plane maneuver | 24 | 7 | 0.00979 | 6.1 |
| | 25 | 8 | 0.00489 | 3.0 |
| | 26 | 9 | 0.00245 | 1.5 |

### *GP transformation computations*

The GP transformation computations trade increased computational complexity for lower solution rate in comparison with the DDA operations. The integration interval $\Delta t_{GP\text{tr}}$ for the solution of the transformation equation

$$\Delta\mathbf{v}_l = \int_{t_{l-1}}^{t_l} P\mathbf{a}_a dt \tag{22}$$

is clearly an integer multiple of the instrument sampling period $\Delta t_{DDA}$ and an integer submultiple of the navigation integration interval $\Delta t_{\text{nav}}$. Ideally, $\Delta t_{GP\text{tr}}$ and $\Delta t_{\text{nav}}$ would be the same, but since the vehicle rotational dynamics are of much higher frequency than the linear dynamics, this condition is not likely to be realized. A single-point, second-order algorithm for Eq. (22) has been used to minimize past data storage requirements; and $\Delta t_{GP\text{tr}}$ which gives sufficient accuracy for the dynamic motions of the vehicle has been determined by the simulation studies. The transformation algorithm is

$$\Delta \mathbf{v}_l = P_{l-1}[I + \Delta\Theta_l/2 + \Delta\Theta_l^2/6]\Delta \mathbf{v}_{al} \tag{23}$$

where $\Delta \mathbf{v}_{al}$ is the pulse count from the accelerometers over the interval $\Delta t_{GP\text{tr}}$, and $\Delta\Theta_l$ is the matrix of gyro pulse counts

$$\Delta\Theta_l = \begin{bmatrix} 0 & -\Delta\theta_{zl} & \Delta\theta_{yl} \\ \Delta\theta_{zl} & 0 & 0 \\ -\Delta\theta_{yl} & 0 & 0 \end{bmatrix} \tag{24}$$

for the single-axis IMU with $y$ and $z$ referring to the pitch and yaw axes, respectively. $P_{l-1}$ is the solution of Eq. (2) and is given by a separate algorithm. The noncommutativity error in (23) is

$$\epsilon[\Delta \mathbf{v}_l]_{n-c} = P_{l-1}[\Omega_{l-1}\dot{\mathbf{a}}_{al-1} - \dot{\Omega}_{l-1}\mathbf{a}_{al-1}]\Delta t_{GP\text{tr}}{}^3/12 \tag{25}$$

and the other algorithm errors are of higher order.

Two algorithms for the generation of $P_l$ have been tested and these are described in the subsections below. The parameter scaling chosen for the instruments and for the GP computations can clearly be used in these transformation computations. These computations were not programmed in fixed-point in the simulation studies, although quantized instrument readouts were provided for.

***Direction cosine integration algorithm***

A well-known single-point, third-order algorithm for Eq. (2) is

$$P_l = P_{l-1}[I + \Delta\Theta_l + \Delta\Theta_l^2/2 + \Delta\Theta_l^3/6] \tag{26}$$

The noncommutativity error in $P_l$ is given by

$$\epsilon[P_l]_{n-c} = P_{l-1}[\Omega_{l-1}\dot{\Omega}_{l-1} - \dot{\Omega}_{l-1}\Omega_{l-1}]\Delta t_{GP\text{tr}}{}^3/12 \tag{27}$$

and other algorithm errors are higher order. An alternative representation is also known

$$P_l = P_{l-1}[I + (\sin\Delta\theta)\Delta\Theta/\Delta\theta + (1 - \cos\Delta\theta)\Delta\Theta^2/\Delta\theta^2] \tag{28}$$

where $\Delta\theta$ is the gyro readout angle scale factor. This form is known to be exact in any region of flight in which the vector cross product $\boldsymbol{\omega} \times \dot{\boldsymbol{\omega}}$ is null.

***Cayley-Klein parameter algorithm***

The Cayley-Klein parameters are a vector $\mathbf{P}$ and a scalar $\lambda$ which satisfy the following differential equations:

$$d\lambda/dt = -\boldsymbol{\omega}^T P/2 \tag{29}$$

$$d\mathbf{P}/dt = [\lambda\boldsymbol{\omega} - \Omega\mathbf{P}]/2 \tag{30}$$

where $\boldsymbol{\omega}$ is the body rate vector. The integration algorithms for Eqs. (29) and (30) are

$$\lambda_l = \lambda_{l-1} - \Delta\boldsymbol{\theta}_l{}^T\mathbf{P}_{l-1}/2 - \lambda_{l-1}\Delta\boldsymbol{\theta}_l{}^T\Delta\boldsymbol{\theta}_l/8 + \Delta\boldsymbol{\theta}_l{}^T\Delta\Theta_l\mathbf{P}_{l-1}/8 \tag{31}$$

$$\mathbf{P}_l = [I - \Delta\Theta_l/2 + \Delta\Theta_l^2/8]\mathbf{P}_{l-1} + [\lambda_{l-1}/2 - \Delta\boldsymbol{\theta}_l{}^T\mathbf{P}_{l-1}/8 - \lambda_{l-1}\Delta\Theta_l/8]\Delta\boldsymbol{\theta} \tag{32}$$

The transformation matrix is then given by

$$P_l = I + 2\lambda_l M_l + 2M_l{}^2 \tag{33}$$

where the matrix $M$ is formed from the components of $P$,

$$M_l = \begin{bmatrix} 0 & -P_{zl} & P_{yl} \\ P_{zl} & 0 & -P_{xl} \\ -P_{yl} & P_{xl} & 0 \end{bmatrix} \tag{34}$$

Initialization of the algorithm requires further auxiliary relations. If the initial value $P_0$ of the transformation matrix is given, the corresponding $\lambda_0$ and $\mathbf{P}_0$ are computed from

$$\lambda_0 = -(\sin\psi_0)/[2(1 - \cos\psi_0)]^{1/2} \tag{35}$$

$$\mathbf{P}_0 = \left(\frac{(1 - \cos\psi_0)/2}{c_{12}{}^2 + c_{13}{}^2 + c_{23}{}^2}\right)^{1/2}\begin{bmatrix} -c_{23} \\ c_{13} \\ -c_{12} \end{bmatrix} \tag{36}$$

where

$$\psi_0 = \arccos\{[\text{tr}(P_0) - 1]/2\}, \quad 0 \le \psi_0 \le \pi \tag{37}$$

$$C = \begin{bmatrix} 0 & c_{12} & c_{13} \\ -c_{12} & 0 & c_{23} \\ -c_{13} & -c_{23} & 0 \end{bmatrix} = (P_0{}^T - P_0)/2 \tag{38}$$

This algorithm is clearly more complex than the preceding one. Therefore, it will be useful only if it offers improved accuracy.

## Simulation Results

The computer errors revealed by the simulation studies have been grouped into four categories for discussion: 1) transformation matrix errors for the single-axis IMU mechanization accrued during free fall when the vehicle undergoes coning motion; 2) impact errors due to the transformation computations; 3) impact errors due to the navigation computations, and these errors pertain to both the three-axis and

**Table 2 Transformation matrix errors at re-entry**[a]

| | DDA simulation | | | | | GP simulation | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\Delta\theta$ = 0.488 mrad, $n$ = 11, $\alpha$ = 1°, $\Delta t_{DDA}$ = 25 msec | | $\Delta\theta$ = 0.488 mrad, $n$ = 11, $\alpha$ = 10°, $\Delta t_{DDA}$ = 2.5 msec | $\Delta\theta$ = 0.244 mrad, $n$ = 12, $\alpha$ = 1°, $\Delta t_{DDA}$ = 12.5 msec | | Precise gyro readout, $\alpha$ = 1°, $\Delta t_{GP\text{tr}}$ = 0.5 sec | | $\Delta t_{GP\text{tr}}$ = 0.125 sec | |
| | AOS | AU | AOS | AOS | AU | DI | C-K | DI | C-K |
| $\phi_x$, mrad | 2.8 | 2.5 | 10.7 | 0.9 | 1.0 | 13.2242 | 13.2233 | 0.8373 | 0.8372 |
| $\phi_y$, mrad | 3.8 | −0.3 | 3.8 | 2.1 | −0.07 | 0.2001 | 0.2000 | 0.0127 | 0.0127 |
| $\phi_z$, mrad | −1.0 | −0.05 | 48.5 | −0.6 | −0.03 | −0.1154 | −0.1153 | −0.0007 | 0.0007 |
| $O_1$, mrad | 1.6 | 0 | −52.0 | 0.5 | 0 | 0 | 0 | 0 | 0 |
| $O_2$, mrad | 2.6 | 0 | −6.2 | 1.3 | 0 | 0 | 0 | 0 | 0 |
| $O_3$, mrad | 0.1 | 0 | 1.6 | 0.3 | 0 | 0 | 0 | 0 | 0 |
| $N_1$ | 1.013 | 1 | 1.131 | 1.0069 | 1 | 1 | 1 | 1 | 1 |
| $N_2$ | 1.0052 | 1 | 1.075 | 1.0026 | 1 | 1 | 1 | 1 | 1 |
| $N_3$ | 1.0066 | 1 | 1.067 | 1.0032 | 1 | 1 | 1 | 1 | 1 |

[a] AOS = alternate-order serial; AU = analytic update; DI = direct integration; C-K = Cayley-Klein.

single-axis IMU mechanizations; and 4) combined effects of all the computer error sources.

### Transformation Matrix Errors During Free Fall

During the free-fall phase the body undergoes unforced coning motion about the inertially fixed angular momentum vector. The three-axis platform isolates the IMU from this motion, but in the single-axis mechanization the angular velocities about the platform pitch and yaw axes excite errors in the transformation matrix computations which must take place during this phase of flight. The angular velocities about the three rotating body axes for this coning motion may be written as

$$\omega_{xb} = \omega_s + \omega_c,\ \omega_{yb} = \omega_t \sin(\omega_s - \omega_c)t$$
$$\omega_{zb} = \omega_t \cos(\omega_s - \omega_c)t \tag{39}$$

where $\omega_s$ is the spin rate, $\omega_c$ is the coning rate,

$$\omega_c = (I_s/I_t)\omega_s \tag{40}$$

and $\omega_t$ is the transverse rate defined by Eq. (19). The angular velocities about the single-axis platform axes are then given by

$$\omega_x = 0,\ \omega_y = -\omega_t \sin\omega_c t,\ \omega_z = \omega_t \cos\omega_c t \tag{41}$$

A closed-form analytic solution of the transformation matrix equation [Eq. (2)] is possible with the angular velocities given by Eq. (41). This facilitates economical computation of the true $P$ matrix in the simulation program, and a detailed study of the algorithm errors during the free-flight phase becomes feasible. Table 2 lists the transformation matrix errors at the end of the free-flight phase (at 300,000 ft altitude) for both the DDA and GP computation methods.

DDA simulation runs were made for two angular quantization levels (the coarser being considered for the in-plane maneuver and the finer for the out-of-plane maneuver) and for two cone angles ($\alpha$ = 1° and 10°). In addition to the simulated alternate-order serial DDA, an analytic update using gyro outputs was used in order to isolate the effects of quantization and noncommutativity errors. This simulation processes the quantized gyro output data through a high-order algorithm with floating-point arithmetic. The algorithm used is Eq. (28), and the algorithm error is principally the third-order noncommutativity error given by Eq. (27). This term is very small for the short time interval $\Delta t_{DDA}$. The $P$ matrix yielded by this high-precision algorithm contains principally the effects of the quantization error.

The GP runs shown in Table 2 were made for two values of the GP transformation interval $\Delta t_{GPtr}$. Precise readout from the gyros was assumed (no quantization) over each interval, so that the results shown in Table 2 are pure algorithm error (noncommutativity plus higher-order terms). The data allow a precise comparison of the errors in the two algorithms tested.

Table 2 presents the errors in the transformation matrix elements in terms of quantities intended to give more insight into the nature of these errors. The angles $\phi_x$, $\phi_y$, and $\phi_z$ are fictitious platform misalignment angles which are reflected by the computed $P$ matrix. Orthogonality errors $O_1$, $O_2$, and $O_3$ are measures of the nonorthogonality of the $P$ matrix, and the normality terms $N_1$, $N_2$, and $N_3$ are measures of the nonnormality of the $P$ matrix column vectors.

The equivalent platform attitude errors are defined as right-hand positive rotations about the true platform axes which relate the misaligned platform axes established by the computed $P$ matrix to the true platform axes. They are computed from the following formulas: roll attitude error ($\phi_x$) $0.5(\mathbf{Z}_{DDA}\cdot\mathbf{Y}_{\text{ref}} - \mathbf{Y}_{D^2DA}\cdot\mathbf{Z}_{\text{ref}})$, pitch attitude error ($\phi_y$) = $0.5(\mathbf{X}_{DDA}\cdot\mathbf{Z}_{\text{ref}} - \mathbf{Z}_{DDA}\cdot\mathbf{X}_{\text{ref}})$, and yaw attitude error ($\phi_z$) = $0.5(\mathbf{Y}_{DDA}\cdot\mathbf{X}_{\text{ref}} - \mathbf{X}_{DDA}\cdot\mathbf{Y}_{\text{ref}})$.

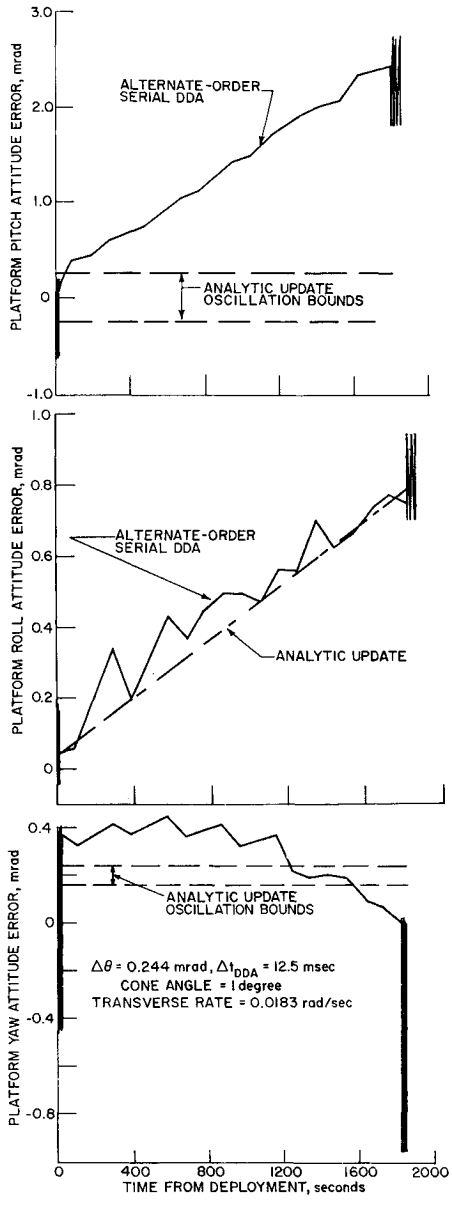


**Fig. 2 Platform attitude errors.**

The platform orthogonality errors are computed from the dot product of nominally orthogonal unit vectors: roll and pitch axes nonorthogonality ($O_1$) = $\mathbf{X}_{DDA}\cdot\mathbf{Y}_{DDA}$, roll and yaw axes nonorthogonality ($O_2$) = $\mathbf{X}_{DDA}\cdot\mathbf{Z}_{DDA}$, and pitch and yaw axes nonorthogonality ($O_3$) = $\mathbf{Y}_{DDA}\cdot\mathbf{Z}_{DDA}$.

The normality terms are calculated from the magnitudes of the nominally unit magnitude platform coordinate vectors: roll axis normality ($N_1$) = $|\mathbf{X}_{DDA}|$, pitch axis normality ($N_2$) = $|\mathbf{Y}_{DDA}|$, and yaw axis normality ($N_3$) = $|\mathbf{Z}_{DDA}|$.

The **X**, **Y**, and **Z** vectors are the columns of the $P$ matrixes, either computed or reference as indicated by the subscript, and nominally represent unit vectors along the platform $X$, $Y$, and $Z$ axes expressed in inertial coordinates.

Figures 2–4 are histograms of the DDA errors for the finer quantization level in Table 2. A fine plot time interval is used at the beginning and end of the runs to show the amplitudes of the oscillations in the data. The behavior shown in the Figs. 2–4 is also typical of the other DDA runs listed in Table 2 except, of course, that the amplitudes are scaled up in the proportions shown in Table 2.

The equivalent platform attitude errors for both the DDA and the analytic update algorithm are shown in Fig. 2. As mentioned previously, the analytic update errors are the effects of quantization and noncommutativity. The analytic

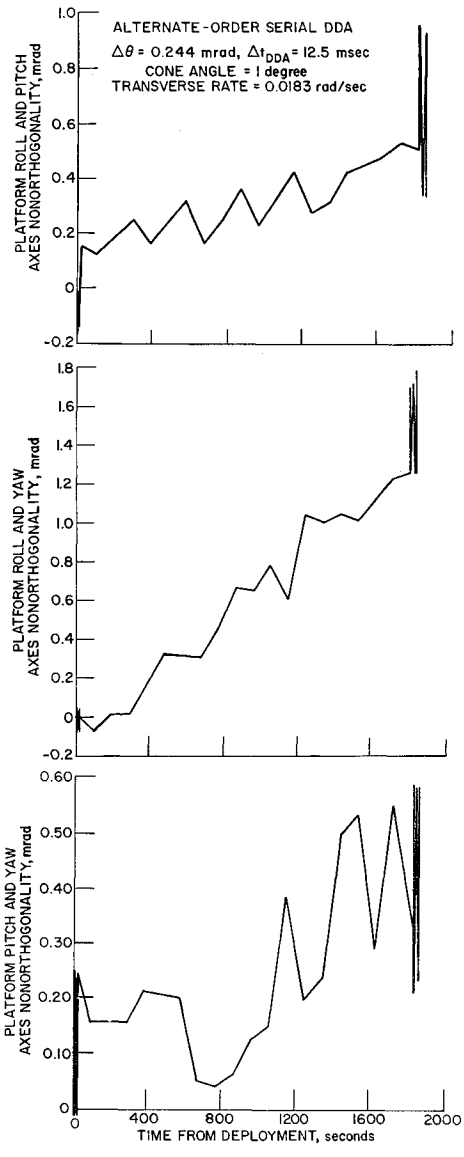


**Fig. 3 Platform orthogonality errors.**

update pitch and yaw attitude errors are essentially all oscillatory quantization errors with amplitudes of plus or minus one $\Delta\theta$ increment. The analytic update roll attitude error shows noncommutativity error as well as oscillatory quantization error, and is characteristic of the oscillatory pitch and yaw motion of the platform. This error propagates secularly because of the persistent character of the input. It is similar to a coning error and is bounded by $\Delta\theta^2$ multiplied by the number of active machine cycles (an active cycle is a cycle in which at least one gyro reads out an increment) within the flight time. In this case that bound is $21\Delta\theta$, and the roll error shown in Fig. 2 is about $4\Delta\theta$.

The analytic update computation suffers no orthogonality or normality error because it is a sequence of updates which is exact for some realizable angular velocity history. The $P$ matrix computed in this way remains perfectly (to the limit of the word length of the computer used to do the simulation) orthogonal and normal. The attitude errors discussed previously result from the fact that the actual angular velocity history will differ from that for which the update would be exact, though both of these are consistent with the observed sequence of gyro outputs.

The alternate-order serial DDA errors are due to computer roundoff and integration rule error as well as the quantization and noncommutativity error represented in the analytic update results. The pitch and yaw attitude errors shown in Fig. 2 now exhibit a secular effect as well as an oscillation of plus or minus $2\Delta\theta$. The increase in amplitude of oscillation compared with the analytic update results is due to the possible storage of $1\Delta\theta$ in the DDA remainder register in addition to the storage of $1\Delta\theta$ in the gyro. The linear terms in these error histories increase to about $2\Delta\theta$ in the case of the yaw error, and $9\Delta\theta$ in the case of the pitch error, both of which are within the bound of $21\Delta\theta$ calculated for the coninglike error due to quantization. The roll attitude error shown in Fig. 2 is almost exactly the same as the analytic update error with small oscillations superimposed. The amplitude of the oscillation, peak to peak, is in this case about $1\Delta P$. This exhibits the effect of quantization in the computer, which is not included in the analytic update computation, as well as quantization in the gyro outputs, which is included in the analytic update computation.

Figure 3 shows that the orthogonality errors are nearly as large as the attitude errors, and Fig. 4 shows that the normality errors can be numerically greater than the attitude errors. The normality error constitutes the principal source of error in transforming the platform referenced velocity increments into inertial coordinates. The problem is particularly severe since the largest normality error terms occur along the platform (body) roll axis which experiences the axial aerodynamic deceleration forces from re-entry to impact. This term alone will cause a position error at impact of thousands of feet for both maneuvers. The results of these atmospheric simulation runs are reported in the following subsection.

The error histories for the individual $P$ matrix elements for the DDA algorithm (not shown) display the same linear and oscillatory components that are observed in the composite attitude error indicators. Note that if $\Delta\theta$ and $\Delta\theta$ are scaled together, all the important error effects tend to scale directly with $\Delta\theta$.

The DDA results for the larger cone angle ($\alpha = 10°$) show errors which are about 10 times greater than for the same computer with $\alpha = 1°$. The cause of this is the larger number of active machine cycles due to the larger excursions of the pitch and yaw oscillations. This multiplies by 10 the bound on coninglike error due to quantization. Even more serious is the fact that with this consistent, cyclic input, certain terms in the $P$ matrix are oscillatory with the same phase as the platform history of angular excursions. This results in consistent truncation error of the same sign in the DDA which accumulates according to the number of active machine cycles. The average truncation error that occurs on each active machine cycle is $0.5\Delta P\Delta\theta$. The number of active machine cycles in time $t$ is $4(\alpha/\Delta\theta)(\omega_c/2\pi)t$, which is 442,000 cycles in this case. Multiplication of the average truncation error by the number of active machine cycles, which varies directly with the cone angle $\alpha$, gives an error accumulation of 50 mrad for $\alpha = 10°$. Reference to Table 2 shows that the

**Table 3 Impact miss due to transformation computation errors**

| | | In-plane maneuver miss, ft | Out-of-plane maneuver miss, ft |
|---|---|---|---|
| Alternate-order serial DDA | | | |
| $n = 11$, $\Delta\theta = 0.488$ mrad, $\Delta v_a = 2.0$ fps, $\Delta t_{DDA} = 0.5$ msec | With initial errors | 5586 | ... |
| | With no initial errors | 203 | ... |
| $n = 12$, $\Delta\theta = 0.244$ mrad, $\Delta v_a = 1.25$ fps, $\Delta t_{DDA} = 0.25$ msec | With initial errors | ... | 1846 |
| | With no initial errors | ... | 58 |
| | With crude normalization | ... | 755 |
| Direct integration GP | | | |
| Full mission, deployment to impact, with $\Delta t_{GPtr} = 0.125$ sec | No quant., precise inst. readout | ... | 74 |
| | With quant., $\Delta\theta = 0.244$ mrad, $\Delta v_a = 1.25$ fps, inst. $\Delta t = 0.25$ msec | ... | 110 |

attitude and nonorthogonality errors are of this magnitude. Increasing the word length of the DDA by 3 bits would decrease this error effect by a factor of 8.

The data shown in Table 2 for the GP algorithms are pure algorithm errors, principally noncommutativity, since no quantization was used and the simulation computations were done in floating-point arithmetic. The results show almost identical errors for both algorithms tested. Since the direct integration method requires less computation than the Cayley-Klein parameter method, if the transformation matrix must be produced at every cycle, it is clearly preferable for the mission considered.

A comparison of the data for the two values of $\Delta t_{GPtr}$ shows that the errors scale with the square of the ratio of these two values, which is $4^2$ or 16. This is the expected scaling for the noncommutativity error for the case of the persistent cyclic driving terms, since the error per step is proportional to the cube of $\Delta t_{GPtr}$, and the number of time steps is inversely proportional to $\Delta t_{GPtr}$. The data show further that the value of 0.5 sec for $\Delta t_{GPtr}$ is clearly too large, since the errors exceed those for the DDA algorithm for $\alpha = 1°$, with quantization error yet to be accounted for. The errors for $\Delta t_{GPtr} = 0.125$ sec, however, appear promising.

It is expected that the quantization errors for the GP algorithms will be considerably less than those for the DDA algorithm, since there are several instrument sample times per $\Delta t_{GPtr}$. It is expected, in particular, that the very important coninglike quantization error propagation will be diminished in the ratio of the instrument sampling time $\Delta t_{DDA}$ to $\Delta t_{GPtr}$.

The GP accuracy expectations are quite pessimistic for larger cone angles. The transverse frequency $\omega_t$ varies directly with $\alpha$, and both the body rates and accelerations are proportional to $\omega_t$ [Eq. (41)]. The noncommutativity error [Eq. (27)] then varies with $\omega_t^2$.

The results for the DDA and GP algorithm simulations for free fall can be summarized as follows.

1) For the smaller cone angle ($\alpha = 1°$) and associated smaller number of active machine cycles, the performance of the simulated alternate order serial DDA is very close to the best obtainable performance that can be achieved by a single-point algorithm which updates on every gyro output, given the sampled and quantized nature of the gyro output. The dominant error sources are: a) the quantization coning error that accumulates secularly in roll, and b) a truncation error of consistent sign which causes a linear accumulation of pitch and yaw attitude error in time, to which is added an oscillation of plus or minus $2\Delta\theta$.

2) With the larger cone angle ($\alpha = 10°$) and larger number of active machine cycles, the accumulated effect of the finite computer word length predominates in the DDA algorithm. The word length would have to be increased by 3 bits for each maneuver to maintain the accuracy exhibited for the smaller angle.

3) Normality error is the principal source of velocity transformation error and will ultimately govern the selection of the DDA computer parameters. A useful alternative to be considered is to use a GP computation to orthonormalize the transformation matrix before reentry.

4) The principal source of error in the GP algorithms is noncommutativity error. This error is quite sensitive to the coning amplitude and may be intolerable for $\alpha \gg 1°$, unless the interval $\Delta t_{GPtr}$ is unreasonably short.

5) Orthonormality errors for the GP algorithms appear to be inconsequential for $\alpha = 1°$.

### Impact Miss Due to Transformation Computation Errors

Impact miss contributions resulting only from errors in the simulated alternate-order serial DDA and direct integration GP velocity transformation computations are presented in Table 3. Ideal accelerometers and perfect navigation computations have been assumed to isolate the transformation errors.

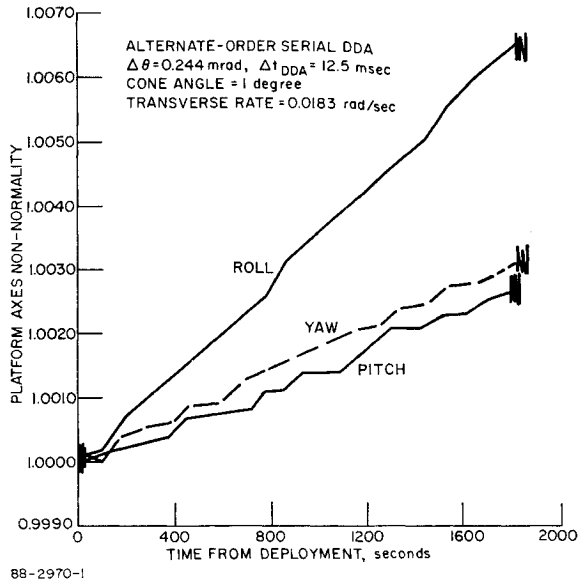


**Fig. 4 Platform normality errors.**

At re-entry the vehicle is oriented so that the single-axis platform $x$ (roll) axis is within about 1° of the vehicle velocity vector, which is in the trajectory plane and about 20° below the local horizontal plane. The platform $y$ (pitch) axis is about 17° above the local horizontal plane. The errors reported here are for the smaller cone angle case, $\alpha = 1°$. The vehicle is assumed spinning to the maneuver initiation point and there is despun instantaneously. It is also found in the aerodynamic environment that the DDA cycle times (instrument sampling times) must be decreased considerably from the free-fall values. The values used for the two maneuvers are listed in Table 3.

The data in Table 3 for the DDA algorithm show clearly that the transformation matrix errors accrued during free fall dominate those accrued during the re-entry phase, and also that the normality error has predominant importance. The maneuvers were "flown" with and without the initial transformation matrix errors accrued during free fall; that is, as if an ideal $P$ matrix correction could take place at the end of free fall.

A crude normalization of the DDA $P$ matrix at the end of free fall was simulated for the out-of-plane maneuver. The column vector elements were simply divided by the correct normality factor $N_j$ listed in Table 2. This operation has the effect of normalizing the column vectors, without orthonormalizing the matrix as a whole. The substantial reduction in target miss indicates the importance of the normality error.

The direct integration GP algorithm was tested with and without instrument quantization, so that comparison of the two miss figures reveals the effect of quantization. The quantization error is evidently about the same magnitude as the algorithm error, so that the quantization choices are about right for the mission.

### Navigation Algorithm Errors

The navigation computations take place during free fall, during which time the accelerometer outputs are null, as well as during the re-entry phase. One of the significant results which emerged from this study is that the navigation cycle time $\Delta t_{nav}$ should be considerably different during the two phases. A reasonable set of GP navigation computer parameters for the two missions were determined by simulation and are shown in Table 4.

**Table 4 Navigation computer parameters**

| | In-plane maneuver | Out-of-plane maneuver |
|---|---|---|
| GP word length $N$, bits | 25 | 25 |
| Integration interval $\Delta t_{nav}$, sec | | |
| Free fall | 3 | 3 |
| Atmospheric flight | 0.75 | 0.75 |
| Velocity scale factor $\Delta v_{GP}$, fps | 0.00393 | 0.00489 |
| Position scale factor $\Delta r_{GP}$, ft | 3.0 | 3.0 |

The values of $N = 25$ bits (23 for magnitude plus sign and parity) and $\Delta t_{nav} = 3$ sec are arrived at by plotting the position error as a function of $\Delta t_{nav}$ for $N = 24$, 25, and 26. In this way, it is possible for each word length to select an optimum $\Delta t_{nav}$ which compromises the dominating effect of roundoff error for smaller values and the dominating effect of integration rule error for larger values. There is a general shape to the position error vs $\Delta t_{nav}$ curve, but there is also considerable scatter of the individual points. The propagated effect of roundoff error especially is sensitive to changes in $\Delta t_{nav}$ and initial conditions. The significant point to be made here is that, while the selected GP navigation computer parameters of $N = 25$ and $\Delta t_{nav} = 3$ sec give position errors for both missions on the order of 100 to 200 ft, the conclusion should not be drawn that this same position error would result for a computer with the same word length but slightly different initial state variables or integration times. However, the errors experienced on neighboring trajectories are expected to remain in the acceptable ranges for this choice of computer parameters. This expectation was verified in the case of one neighboring trajectory simulation, for which the error was 339 ft.

The minimum position error for $N = 24$ occurs for a $\Delta t_{nav}$ between 4 and 5 sec and is on the order of 1000 to 1500 ft for both missions. It is clear from the random walk model for the accumulation of roundoff errors that the effect of roundoff error is greater, since $\Delta v_{GP}$ is larger for the smaller word length, and therefore that the optimum $\Delta t_{nav}$ is larger. Increasing the word length to 26 bits has the effect of reducing the position error for a smaller optimum $\Delta t_{nav}$. It appears that the optimum $\Delta t_{nav}$ for this word length is between 2 and 3 sec, and that the position error is on the order of 50 to 150 ft.

The reduction in $\Delta t_{nav}$ during the re-entry phase is necessary because the trapezoidal integration rule contributes very large errors in the aerodynamic environment for a 3 sec interval. The additional roundoff error for the shorter integration interval is negligible, since the atmospheric flight time is less than a minute. The navigation computer cycle time was reduced by a factor of 4, since an integer power of 2 reduction can be easily implemented. With this selection, the target miss contribution due to navigation algorithm error is determined almost entirely by the propagation during free fall. The misses are 206 ft for the in-plane maneuver and 107 ft for the out-of-plane maneuver.

### Total Computer Error Effects

The instrument and computer parameters presented in Table 5 have been selected for the two maneuvers and two $\alpha$'s. Complete simulation runs have been made for $\alpha = 1°$. For the DDA algorithm, assuming orthonormalization of the $P$ matrix at re-entry and the single-axis IMU, the total impact misses due to computer errors are about 1000 ft for the in-plane maneuver and 800 ft for the out-of-plane maneuver. The use of the GP algorithm in place of the DDA should reduce the misses to about 500 and 400 ft, respectively. The impact misses due to computer errors for the three-axis IMU should be further reduced to between 200 and 300 ft for both missions; that is, only the navigation computation error.

The instrument and computer parameters associated with $\alpha = 10°$ should yield somewhat better results since the velocity quantization levels are considerably smaller. The angular quantization levels and cycle times have been adjusted to afford the same angular accuracies and to accommodate the maximum angular velocities. The GP transformation interval for $\alpha = 10°$, however, remains to be determined, but it should be adjustable to yield tolerable errors in the GP transformation computations.

It must be emphasized that these performance results do not represent limitations on achievable errors. The computer parameters were selected in each case to meet a specification on position error of 500 to 1000 ft. With finer quantization, shorter cycle times, and longer word lengths, more stringent accuracy specifications could be met.

## Conclusions

It should be recognized that none of the computer requirements determined in this study pose a significant challenge to the current technology. The alternate-order serial DDA organization treated in this simulation is perhaps the most convenient organizational form to implement. The nine direction cosine equations can be separated into three independent groups of three equations each. The direction cosine registers for the three cosines in each group are arranged in a three word loop, as are the remainder registers corresponding to these direction cosines. Each of these three sets of loops is then processed through a single serial adder. The two additions required for each update are done in alternating order on successive passes through the adder. The complete cycle time for such a machine is then six words times, and each word time is the bit time multiplied by the number of bits in each word.

Bit rates of 10 MHz are easily achieved by current equipment, and bit rates many times faster than this are commonly achieved in laboratory experiments. The greatest requirement for speed is presented by the DDA designed for the larger cone angle and the out-of-plane maneuver. The word length is 15 bits plus 1 for sign, and the required cycle time is 25 $\mu$sec. Using the organization described above, this would be achieved with a bit rate of 3.84 MHz, a very comfortable figure. Using current-type integrated circuitry, this machine would easily fit in a cigarette package.

It might be noted that if this word length and associated gyro output quantization were used for the smaller cone angle as well, the position error which is due primarily to the transformation matrix non-normality would be reduced by nearly a factor of 16 for the in-plane maneuver and 8 for the

**Table 5 Instrument and computer parameters**

| | In-plane maneuver | | Out-of-plane maneuver | |
|---|---|---|---|---|
| | $\alpha = 1°$ | $\alpha = 10°$ | $\alpha = 1°$ | $\alpha = 10°$ |
| Instrument and DDA parameters | | | | |
| DDA word length, bits | 11 | 14 | 12 | 15 |
| Angular quantization, mrad | 0.488 | 0.0610 | 0.244 | 0.0305 |
| Velocity quantization, fps | 2.0 | 0.20 | 1.25 | 0.125 |
| Free-fall cycle time, msec | 25.0 | 0.3125 | 12.5 | 0.15625 |
| In-atmosphere cycle time, msec | 0.50 | 0.05 | 0.25 | 0.025 |
| GP navigation computer | | | | |
| Word length, bits | 25 | 25 | 25 | 25 |
| Velocity scaling, fps | 0.00393 | 0.00312 | 0.00489 | 0.00391 |
| Position scaling, ft | 3.0 | 3.0 | 3.0 | 3.0 |
| Free-fall integration interval, sec | 3.0 | 3.0 | 3.0 | 3.0 |
| In-atmosphere integration interval, sec | 0.75 | 0.75 | 0.75 | 0.75 |
| GP transformation computer | | | | |
| Transformation interval, sec | 0.125 | | 0.125 | |

out-of-plane case. The resulting position errors due to inaccurate velocity transformation would then be in the range 300–500 ft without normalization or orthogonalization of the computed transformation matrix by the GP computer.

The GP loading requirements to perform the transformation computations every 0.125 sec are very modest. A 400 register memory would be sufficient to store all constants, variables and program words. Even a very conservative computer would have a memory cycle time of about 5 $\mu$sec, an add/subtract time of 2 memory cycle times, and a multiply time of perhaps 4 add times. For these instruction speeds the direction cosine integration and velocity transformation requires 4.12 msec of computation. Since this calculation is performed every 125 msec, the computer duty cycle is only 3.3%.

The speed and memory requirements for a GP computer to process the midpoint-$g$ navigation equations every 0.75 sec are trivial. Only about 100 storage registers are required. The midpoint-$g$ computation involves 22 load or store operations, 21 add or subtract operations, and 20 multiplications. Thus, a computer with the speed indicated above would execute this computation in 1.12 msec. But this computation is repeated every 750 msec, so the navigation computation would occupy the GP computer only 1/670 of the time. Virtually all of the computer's capability remains free to do guidance or other computations. The GP computer to perform the navigation (and transformation) computations cannot be sized without considering these additional tasks, since they may well provide the principal computational requirement.

The study shows clearly that a significant computational accuracy penalty is incurred with the partially strapped down system as opposed to the fully stable platform. The computational accuracy penalty must, of course, be properly weighed in the over-all system design considering instrument errors, electronic and mechanical complexity, and mission objectives.

It is also clear that the computer errors propagating during free fall generally dominate those occurring during the terminal re-entry phase. It is, however, necessary to contain the navigation algorithm error during the atmospheric phase by reducing the integration interval from the optimum value determined for the free-fall phase.

DECEMBER 1969 J. SPACECRAFT VOL. 6, NO. 12

# Attitude Performance of the GEOS-II Gravity-Gradient Spacecraft

J. M. WHISNANT,* P. R. WASZKIEWICZ,† AND V. L. PISACANE‡
*The Johns Hopkins University, Silver Spring, Md.*

**The GEOS-II satellite employs an extendible boom to provide an earth-pointing equilibrium orientation (by gravity-gradient stabilization) and a magnetically anchored eddy-current damper to dissipate librational energy. Effects of the gravity-gradient forces, orbital eccentricity, thermal distortion of the stabilization boom, solar radiation pressure and geomagnetically induced residual dipole and damper torques are included in the prelaunch design and postlaunch performance analyses (digital simulations of the nonlinear differential equations of motion). The attitude determination system utilizes measurements of the orientation of the sun and the geomagnetic field to obtain an overdetermined orientation solution, so that instrument calibration parameters can be determined. Flight data obtained during two different time periods indicate that the attitude is within 5° of local vertical 96% of the time. General agreement between flight data and the results of the digital simulation is obtained.**

## Nomenclature

$a$ = semimajor axis
$c$ = damping coefficient
$I_x, I_y, I_z$ = inertia of spacecraft about body-fixed principal axes
$I_p, I_r, I_y$ = inertia of spacecraft about pitch, roll, and yaw axes
$l$ = length of boom
$M$ = mean anomaly
$O_e(x_{ei})$ = Cartesian frame of reference fixed in inertial space and located at the center of mass of the earth ($i$ = 1 to 3)
$O_s(x_{si})$ = Cartesian frame of reference fixed in the spacecraft ($i$ = 1 to 3)
$O_t(x_{ti})$ = Cartesian frame of reference used to define the local vertical coordinate system ($i$ = 1 to 3)
$\alpha_i$ = set of Euler angles used to define attitude ($i$ = 1 to 3)
$\delta_{vert}$ = total off-vertical libration angle
$\epsilon$ = orbit eccentricity
$\eta$ = angle between the earth-sun line and the normal to the orbit plane
$\varphi$ = pitch motion
$\sigma$ = $(I_r - I_y)/I_p$

Received January 6, 1969; revision received October 7, 1969. This work was supported by NASA, Office of Space Science and Applications under Task I of Contract NOw 62-0604-c.

* Associate Mathematician, Space Analysis and Computation Group, Applied Physics Laboratory.

† Associate Engineer, Space Analysis and Computation Group, Applied Physics Laboratory; now Senior Engineer, Westinghouse Aerospace Division, Baltimore, Md.

‡ Supervisor, Theory Project, Space Analysis and Computation Group, Applied Physics Laboratory. Member AIAA.

## Introduction

THE repeated success of passive gravity-gradient stabilization (GGS) systems at low altitudes has fostered interest in their use for communication and meterological satellites at higher altitudes. At synchronous altitudes the restoring torque (which is inversely proportional to the cube of the distance of the spacecraft from the center of the earth) is two orders of magnitude smaller than at low altitudes. For this reason, verification of the performance of GGS systems at low altitudes, with the intent to improve the modeling of the