NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-350

A Fourier Transform Technique for the Prediction of Torsional Transients for a Spacecraft From Flight Data of Another Spacecraft Using the Same Booster

M. R. Trubert

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 15, 1967

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-350

A Fourier Transform Technique for the Prediction of Torsional Transients for a Spacecraft From Flight Data of Another Spacecraft Using the Same Booster

M. R. Trubert

Approved by:

M.E. Alper by H J Hollrock

M. E. Alper, *Manager*
Applied Mechanics Section

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 15, 1967

TECHNICAL MEMORANDUM 33-350

Prepared Under Contract No. NAS 7-100
National Aeronautics & Space Administration

PRECEDING PAGE BLANK NOT FILMED.

Contents

Tables

Figures

Abstract

A frequency-domain solution for determining the torsional acceleration transient occurring at booster engine cutoff for the *Surveyor* spacecraft, using the *Ranger* data, is presented in this Memorandum. First, the Fourier transform of the transient acceleration, which occurred at the base of the spacecraft during the *Ranger* flights, is computed. A torque at the gimbal blocks is assumed to be the cause of this acceleration. The structural characteristics of the *Atlas/Agena/Ranger* vehicle are introduced in the form of a transfer function computed from the normal modes of the structure for viscous or hysteretic damping. Then the Fourier transform of the torque at the gimbal blocks is deduced.

The next step is the computation of the transfer function of the *Atlas/Centaur/Surveyor* vehicle, also using the normal modes of the structure. Then the Fourier transform of the transient acceleration at the *Surveyor* field joint is determined, assuming that the same torque is present at the gimbal blocks of the *Atlas* engine for both spacecraft, since the same booster is used. Finally, the time history of the *Surveyor* field joint acceleration is computed by inverse Fourier transformation.

A digital computer program has been written which computes direct and inverse Fourier transforms, and also operates on the transfer functions of the structures. The results are compared, using *Ranger VI, VII, VIII* and *IX* data, with the analog-time-domain solution of the same problem treated in an earlier Memorandum, and the instability that was encountered in this type of solution is explained. Although the method is illustrated for the *Ranger* and *Surveyor* spacecraft, it is quite general and can be applied to any spacecraft for which the same booster is used. The digital-frequency-domain method is much more easily implemented than the analog method exposed in an earlier Memorandum.

A Fourier Transform Technique for the Prediction of Torsional Transients for a Spacecraft From Flight Data of Another Spacecraft Using the Same Booster

I. Introduction

A method for the determination of the torsional transient vibration environment of the *Surveyor* at booster engine cutoff (BECO), Ref. 1, used the flight data of the *Ranger* series as an input and the problem was solved in the time domain. However, an instability occurred on both the analog and digital computers which was resolved by a special scheme: a synthesis of an input resulted in an output equivalent to the flight data. In the present report, the frequency domain solution, which inherently avoids the instability just mentioned, is used for the same problem.

II. Determination of the Gimbal Torque

A. Viscous Damping

The time history of the angular acceleration $\theta_1(t) = u_1(t)$ at the *Ranger* adapter is known for the BECO event and is recorded on magnetic tape. It is assumed that this acceleration is due to an unknown transient torque $T(t)$ applied at the gimbal blocks of the *Atlas* engine at BECO. The first objective is the determination of the Fourier transform $F(\omega)$ of $T(t)$.

The equations of motion for the *Atlas/Agena/Ranger* vehicle, Ref. 1, are

$$\ddot{q}_n + 2\,\xi_n\,\omega_n\,\dot{q}_n + \omega_n^2\,q_n = \frac{\phi_{2n}}{m_n}\,T \tag{1}$$

$$u_1(t) = \frac{T}{m_0} + \sum_{n=1}^{N} \phi_{1n}\,\ddot{q}_n \tag{2}$$

$$n = 1, 2, 3, \cdots, N\,.$$

where

q_n are the generalized coordinates

ω_n the natural frequencies

m_n the generalized masses

ξ_n the reduced modal dampings

ϕ_{1n} the modal displacements at the *Ranger* adapter

ϕ_{2n} the modal displacements at the gimbal blocks

Equations (1) and (2) are unstable when the time domain solution is programmed either on an analog computer or on a digital computer as found in Ref. 1. To solve the problem in the frequency domain let us take the Fourier transform of both sides of Eqs. (1) and (2). The Fourier transforms of the torque $T(t)$ and the modal displacements $q_n(t)$ are:

$$F(\omega) = \int_{-\infty}^{+\infty} T(t)\, e^{-i\omega t}\, dt \tag{3}$$

and

$$Q_n(\omega) = \int_{-\infty}^{+\infty} q_n(t)\, e^{-i\omega t}\, dt \tag{4}$$

where

$$i = (-1)^{1/2}, \quad n = 1, 2, \cdots, N$$

The Fourier transform of the modal velocity and the modal acceleration are readily obtained from Eq. (4) and are $i\omega Q_n(\omega)$ and $-\omega^2 Q_n(\omega)$, respectively. Therefore the frequency domain representation of Eqs. (1) and (2) is:

$$(-\omega^2 + 2i\omega_n \omega \xi_n + \omega_n^2)\, Q_n(\omega) = \frac{\phi_{2n}}{m_n} F(\omega) \tag{5}$$

$$V_1(\omega) = \frac{F(\omega)}{m_0} + \sum_{n=1}^{N} \phi_{1n} \left[-\omega^2 Q_n(\omega)\right] \tag{6}$$

where $V_1(\omega)$ is the Fourier transform of $u_1(t)$

$$V_1(\omega) = \int_{-\infty}^{+\infty} u_1(t)\, e^{-i\omega t}\, dt \tag{7}$$

Solving Eq. (5) for $Q_n(\omega)$ and substituting into Eq. (6) we obtain:

$$V_1(\omega) = F(\omega) \sum_{n=0}^{N} \phi_{1n}\, \phi_{2n} \frac{1}{m_n \left[1 - \left(\frac{\omega_n}{\omega}\right)^2 - 2i\xi_n \frac{\omega_n}{\omega}\right]} \tag{8}$$

where the subscript $n = 0$ corresponds to the rigid body roll, i.e., $\phi_{1_0} = \phi_{2_0} = 1$ and $\omega_0 = 0$.

The summation in Eq. (8) represents the transfer function $H_1(\omega)$ from the torque $F(\omega)$ to the adapter acceleration $V_1(\omega)$ for the *Ranger* vehicle.

$$H_1(\omega) = \sum_{n=0}^{N} \phi_{1n}\, \phi_{2n} \frac{1}{m_n \left[1 - \left(\frac{\omega_n}{\omega}\right)^2 - 2i\xi_n \frac{\omega_n}{\omega}\right]} \tag{9}$$

Equation (8) can be rewritten in matrix form as follows

$$H_1(\omega) = \mathbf{\Phi}_1^T\, \mathbf{Z}^{-1}\, \mathbf{\Phi}_2 \tag{10}$$

where

$$\mathbf{\Phi}_1 = \begin{Bmatrix} 1 \\ \phi_{11} \\ \phi_{12} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \phi_{1N} \end{Bmatrix}, \qquad \mathbf{\Phi}_2 = \begin{Bmatrix} 1 \\ \phi_{21} \\ \phi_{22} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \phi_{2N} \end{Bmatrix}$$

represent the columns of the modal displacements at the adapter and the gimbal blocks, respectively, and $\mathbf{Z}$ is a diagonal matrix constructed with the individual transfer functions of the modes

$$\mathbf{Z} = \begin{bmatrix} \ddots & & \\ & Z_n & \\ & & \ddots \end{bmatrix} \tag{11}$$

where

$$Z_n = m_n \left[1 - \left(\frac{\omega_n}{\omega}\right)^2 - 2i\xi_n \frac{\omega_n}{\omega}\right] \tag{12}$$

$$n = 1, 2, 3, \cdots, N.$$

Finally, the input-output relationship relating the torque to the adapter acceleration, Ref. 2, is

$$V_1(\omega) = F(\omega)\, H_1(\omega) \tag{13}$$

from which we obtain the Fourier transform of the torque.

$$F(\omega) = \frac{V_1(\omega)}{H_1(\omega)} \tag{14}$$

B. Hysteretic Damping

If hysteretic damping is assumed for the structure we simply replace $2i\xi_n\,\omega_n/\omega$ by $2i\xi_n$ in Eq. (12) which becomes

$$Z_n = m_n\left[1 - \left(\frac{\omega_n}{\omega}\right)^2 - 2i\xi_n\right] \tag{15}$$

with no other change in the analysis. The Fourier transform of the torque is still given by Eq. (12).

III. Surveyor Field Joint Acceleration

Although the time domain solution for this part of the problem was found to be stable in Ref. 1, a frequency domain solution has been also worked out which presents definite computational advantages. Let us call $H_2(\omega)$ the transfer function for the *Atlas/Centaur/Surveyor* vehicle relating the torque $T(t)$ to the angular acceleration at the *Surveyor* field joint (Fig. 1). The evaluation of this transfer function $H_2(\omega)$ is done in a manner completely similar to $H_1(\omega)$ from Eqs. (8–11) except that the modes shapes ϕ_1, ϕ_2 and the natural frequencies ω_n correspond to the new structure; namely, the *Atlas/Centaur/Surveyor*.

Applying the Fourier transform of the torque $F(\omega)$ to the transfer function $H_2(\omega)$ we obtain the Fourier transform of the response at the *Surveyor* field joint

$$V_2(\omega) = F(\omega)\,H_2(\omega) \tag{16}$$

Then, using Eq. (13) we have

$$V_2(\omega) = V_1(\omega)\,\frac{H_2(\omega)}{H_1(\omega)} \tag{17}$$

Finally, the time history of the response at the *Surveyor* field joints is obtained by taking the inverse Fourier transform of $V_2(\omega)$, viz,

$$u_2(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} V_2(\omega)e^{i\omega t}\,d\omega \tag{18}$$

A digital computer program has been written that performs all the operations mentioned above and gives plots, as shown in Appendix A, of the various functions involved.

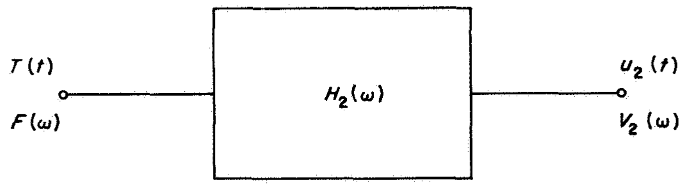

Fig. 1. *Atlas/Centaur/Surveyor* transfer function

IV. Numerical Computation and Comparison With the Analog Method

The frequency domain method just described was used to solve the same problem as in Ref. 1, i.e., deduce the *Surveyor* field joint torsional acceleration from the *Ranger* flight data. The structural characteristics for the *Ranger* and *Surveyor* vehicles were taken from Ref. 1 and were used to compute the transfer functions $H_1(\omega)$ and $H_2(\omega)$ respectively. The four flight torsional accelerations relative to *Ranger VI, VII, VIII,* and *IX* shown in Fig. 2 were obtained from flight tape after transformation from linear acceleration to angular acceleration as indicated in Appendix A of Ref. 1. Figure 3 shows the corresponding *Surveyor* field joint accelerations using the digital computer program described in Appendix A. The analog solution obtained in Ref. 1 with the same structural characteristics and the same pulses as above is indicated in Fig. 4 for comparison.

Comparing the digital frequency domain method (Fig. 3) and the analog time domain method (Fig. 4), we conclude that the agreement between the two methods depends very much on the driving pulses used. The calculated peak amplitude responses at the *Surveyor* field joint for the two methods are indicated in Table 1.

Table 1. Calculated peak amplitudes (rad/sec²) at *Surveyor* field joint

Ranger	Analog method	Digital method	Deviation from digital to analog, %
VI	12	10	+20
VII	12	8.4	+43
VIII	13.5	16	−15.5
IX	8.5	4.7	+81

From this table it is seen that the analog method gives higher peaks than the digital method for *Ranger VI, VII,* and *IX* data while the digital method gives higher peaks for *Ranger VIII* data. However, it should be stated that

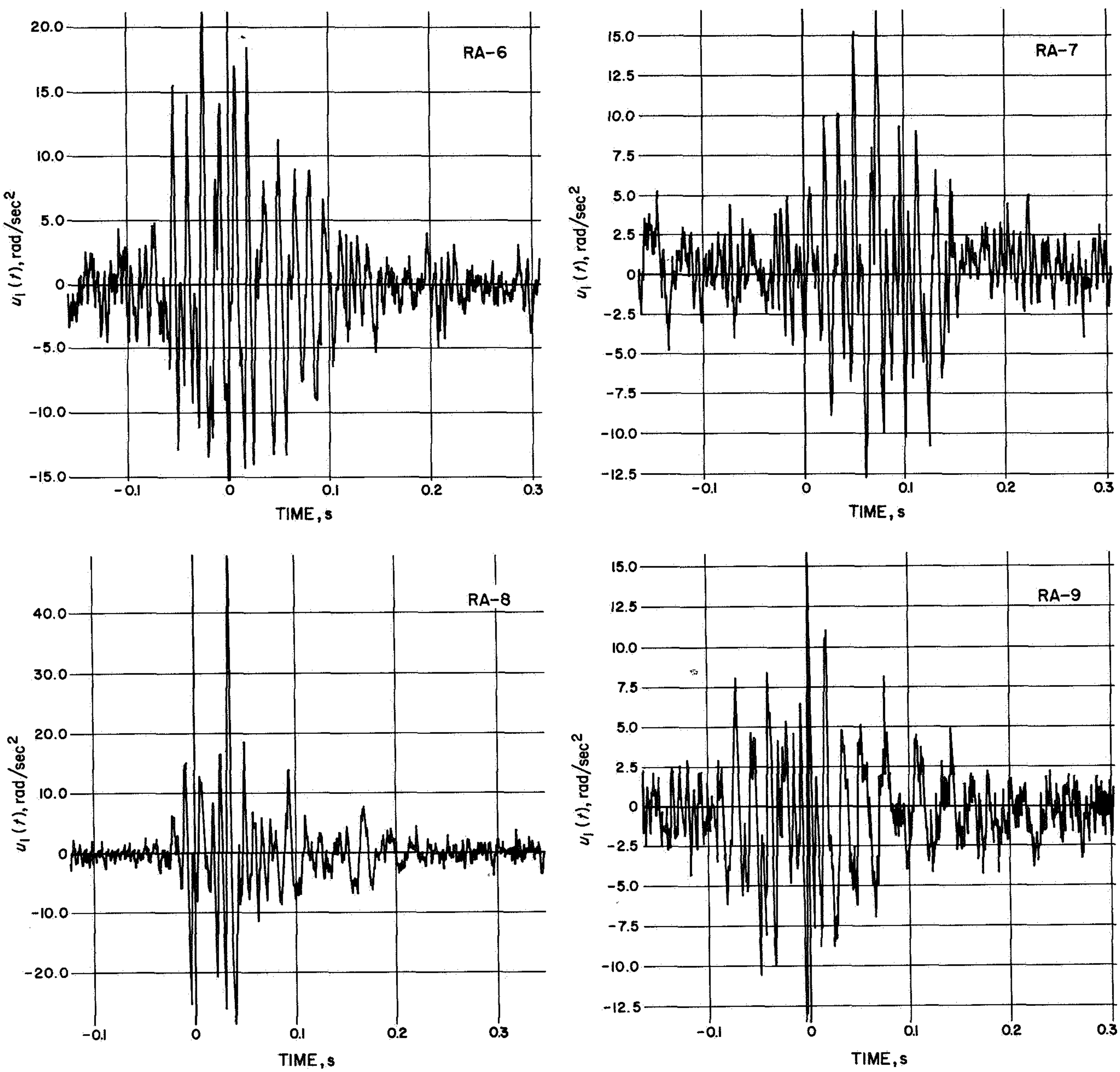

Fig. 2. Flight torsional acceleration

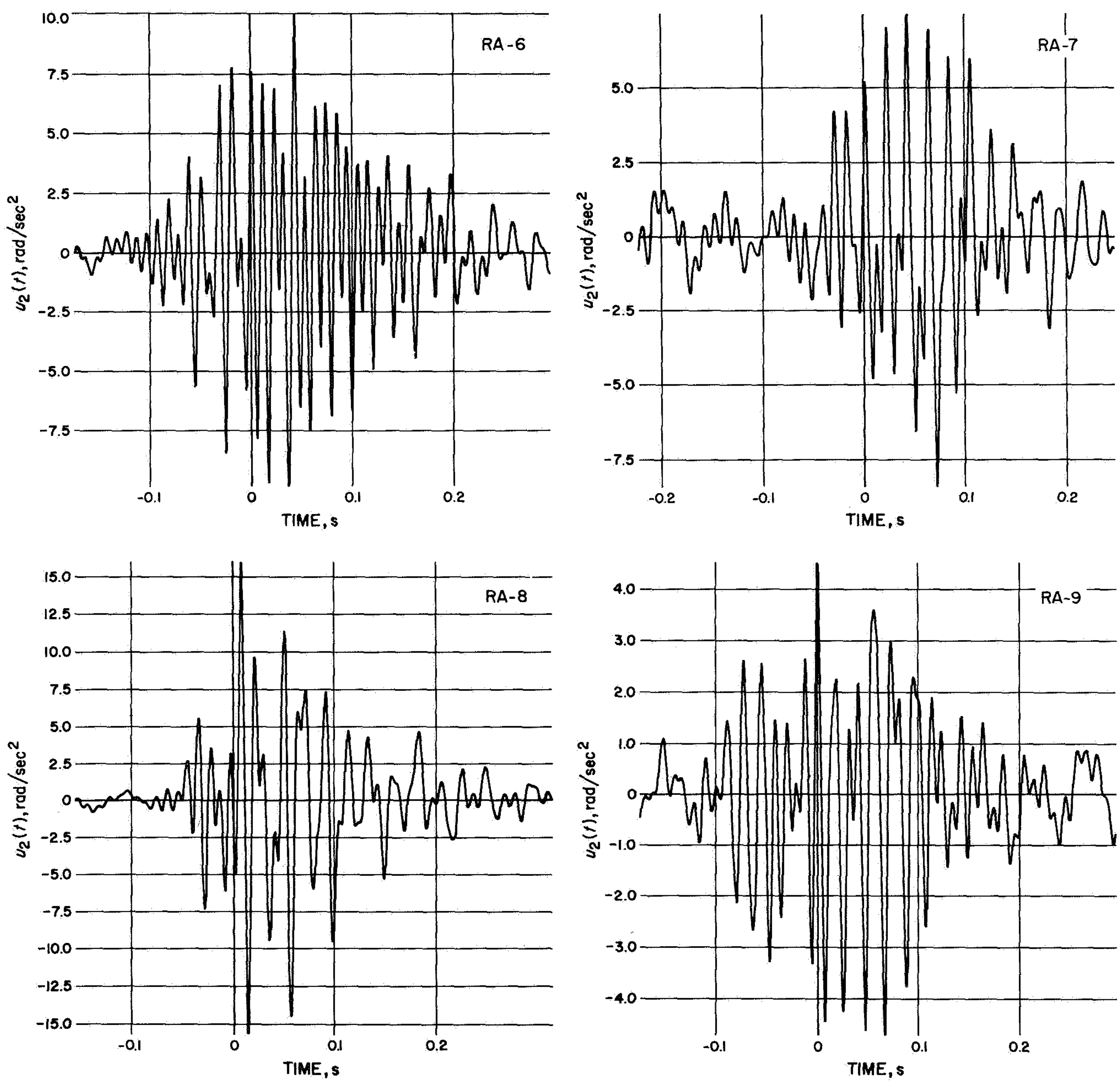

Fig. 3. ***Surveyor*** **field joint torsional acceleration, frequency domain, digital solution**

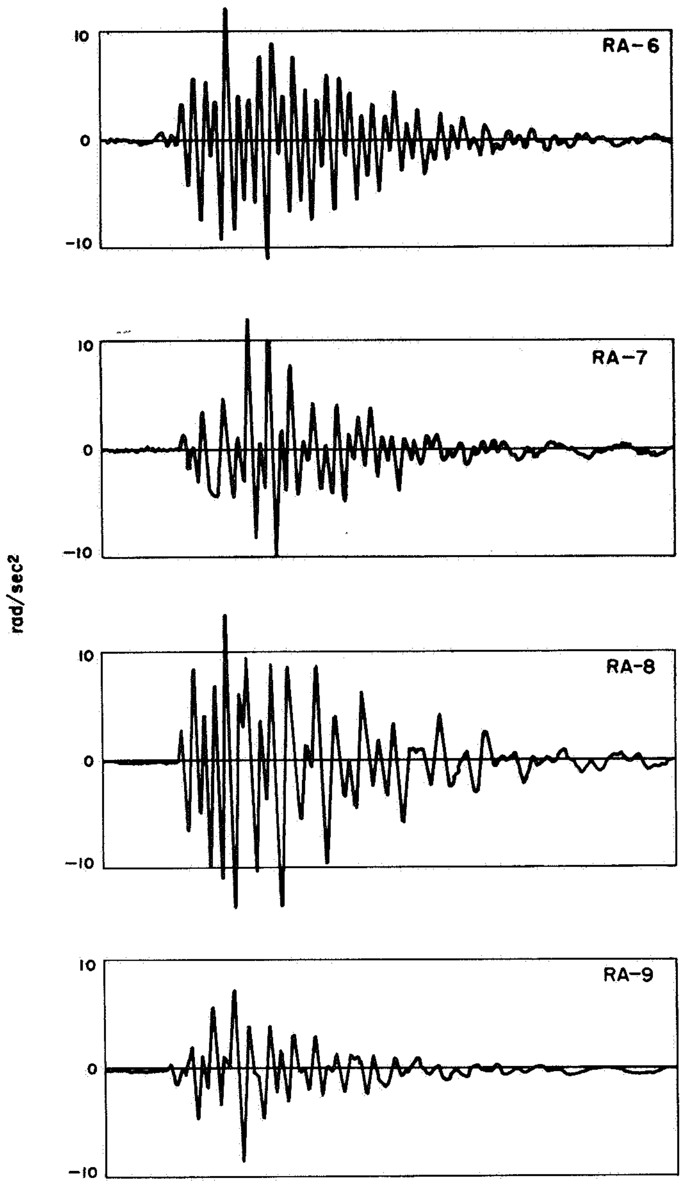

Fig. 4. Surveyor field joint acceleration, time domain, analog solution

this comparison is not intended to choose between the two methods but is simply done here to check the results obtained by the analog method which was only approximate as noted in Ref. 1. The digital frequency domain solution presented earlier is to be preferred since it gives an exact solution with preservation of the phase which is not the case for the analog method.

Finally, we note that the frequency range has been limited to 0–100 cps for the same reasons as in Ref. 1, i.e., the structural model is not representative above 100 cps and the data are not reliable above that same frequency.

Before concluding this Section, it should be noted that upgraded models of the *Atlas/Agena/Ranger* and *Atlas/Centaur/Surveyor* (Ref. 3) were investigated. The results of the computation are shown in Fig. 5. The levels obtained with these models are less than the one obtained previously with the less-refined models.

V. Discussion

As mentioned earlier, the time domain method leading to the determination of $T(t)$ was basically unstable while the corresponding frequency domain method for $F(\omega)$ does not exhibit any stability problem. The reason for this difference is now briefly investigated.

In the Fourier transform approach we consider only the particular solution of Eqs. (1) and (2) which is due to the forcing function $u_1(t)$ excluding the homogeneous solutions of the same system corresponding to a vanishing $u_1(t)$. On the contrary, simulating the system of Eqs. (1) and (2) on an analog computer, or a digital computer, for the time domain solution automatically brings into play both the homogeneous and the particular solutions. In addition, it is only when all the homogeneous solutions are decaying with time that the problem is stable; otherwise, an instability occurs whatever the initial conditions.

We now turn to showing that exploding homogeneous solutions do exist for Eqs. (1) and (2). For convenience we rewrite these equations in matrix form

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \bar{\boldsymbol{\Phi}}_2\, T(t) \tag{19}$$

$$u_1(t) = \frac{T(t)}{m_0} + \bar{\boldsymbol{\Phi}}_1^T\, \ddot{\mathbf{q}} \tag{20}$$

where

$$\mathbf{M} = \begin{bmatrix} \diagdown_{m_n} \diagdown \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \diagdown_{2\xi_n\omega_n m_n} \diagdown \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} \diagdown_{m_n\omega_n^2} \diagdown \end{bmatrix}$$

$$\mathbf{q} = \begin{Bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{Bmatrix}, \quad \bar{\boldsymbol{\Phi}}_1 = \begin{Bmatrix} \phi_{11} \\ \phi_{12} \\ \vdots \\ \phi_{1N} \end{Bmatrix}, \quad \bar{\boldsymbol{\Phi}}_2 = \begin{Bmatrix} \phi_{21} \\ \phi_{22} \\ \vdots \\ \phi_{2N} \end{Bmatrix}$$

To assess the stability of the system of Eqs. (19) and (20) we eliminate the unknown $T(t)$ between these equations. Consequently, we obtain

$$\mathbf{A}\ddot{q} + \mathbf{C}\dot{q} + \mathbf{K}q = \bar{\boldsymbol{\Phi}}_2\, m_0 u_1(t) \tag{21}$$

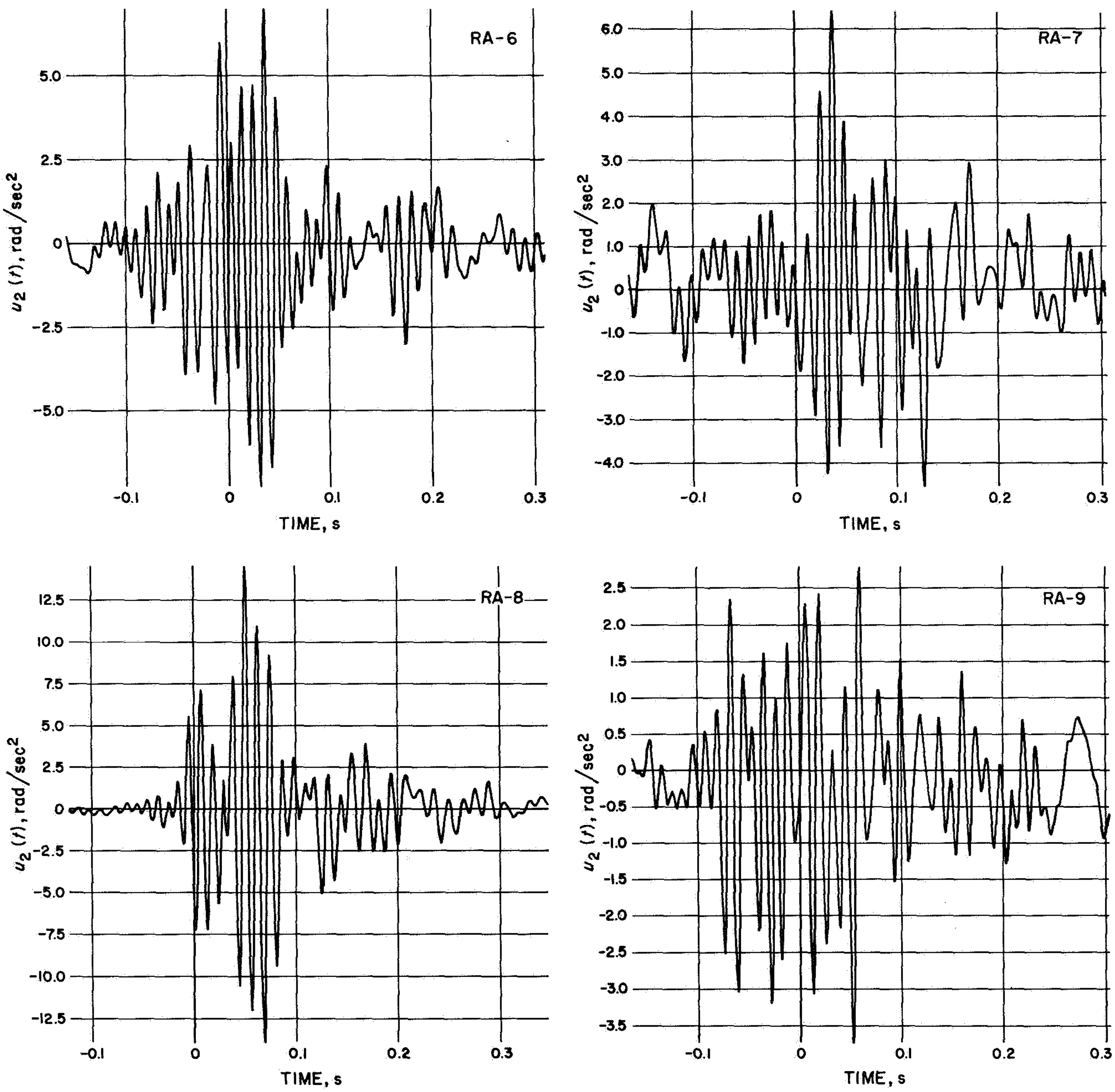

Fig. 5. *Surveyor* field joint acceleration, new model

where

$$\mathbf{A} = \mathbf{M} + m_0 \bar{\boldsymbol{\phi}}_2 \bar{\boldsymbol{\phi}}_1^T \tag{22}$$

We then investigate the homogeneous solutions of the system of equations represented by Eq. (21); to this end, we set the righthand side equal to zero, viz,

$$\mathbf{A}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = 0 \tag{23}$$

Equation (23) represents a complex eigenvalue problem with the particularity that, owing to the term $m_0 \bar{\phi}_2 \bar{\phi}_1^T$, the matrix A is not symmetric. Consequently Eq. (23) is not a classical free vibration problem with positive damping as it may look at first sight. The homogeneous solutions of Eq. (23) are of the form

$$q_n = e^{p_n t} \tag{24}$$

where p_n are either real or complex and $n = 1, 2, 3, \cdots$, i.e., we have either

$$q_n = e^{\lambda_n t} \tag{25}$$

or

$$q_n = e^{\lambda_n t \pm i\omega_n t} \tag{26}$$

where λ_n and ω_n are real. Then the general solution is

$$q_n = \sum_n C_n e^{p_n t} \tag{27}$$

where C_n are arbitrary constants depending on the initial conditions.

The system will be stable if all the real parts λ_n are negative and it will be unstable if any one of the real parts λ_n is positive.

The complex eigenvalue problem was numerically solved for the *Atlas/Agena/Ranger* vehicle of Appendix B of Ref. 1 and two real parts λ_2 and λ_7 were found positive (Table B-1). These two positive real parts give an instability even with a zero initial condition situation since the coefficients C_n are never truly zero due to the computer imperfections.

Appendix A
Computer Program

I. Sequence

A digital computer program has been developed at the Jet Propulsion Laboratory in order to implement the analysis presented in this report. The program (M172) is written in Fortran IV and is explained in detail in Ref. 4. This is a multipurpose program which can perform all or selected portions of the following sequence (see CALC of name list).

1. Given the time history of an acceleration pulse $u_1(t)$ at the base of a given spacecraft (*Ranger*) compute its Fourier transform

$$V_1(f) = a(f) + i\,b(f) \tag{A-1}$$

where

$$a(f) = \int_{-\tau_0}^{-\tau_0+t_N} u_1(t) \cos 2\pi ft \, dt \tag{A-2}$$

$$b(f) = \int_{-\tau_0}^{-\tau_0+t_N} u_1(t) \sin 2\pi ft \, dt \tag{A-3}$$

for equally spaced given discrete frequencies $f = f_1, f_2, f_3, \cdots f_n$ in cycles per second ($\omega = 2\pi f$). The time parameter τ_0 is computed as shown later and t_N is the total duration of the pulse.

2. Compute the transform function $H_1(f)$ for either viscous damping or hysteretic damping according to Eqs. (10–12), or Eqs. (10, 11, and 15).

3. Compute the Fourier transform of the torque $F(f)$ according to Eq. (14).

4. Deduce the Fourier transform of the response $V_2(f)$ at the base of the second spacecraft (*Surveyor* field joint according to Eq. (16).

5. Compute the inverse Fourier transform of $V_2(f)$, i.e.,

$u_2(t) = 2$ real part of

$$\left[\int_{f_1}^{f_n} V_2(f) \left(\cos 2\pi ft + i \sin 2\pi ft \right) df \right] \tag{A-4}$$

which is the computer version of the classical Eq. (18).

II. Determination of τ_0

We first remark that a time translation τ_0 is immaterial for the definition of the pulse, (Fig. A-1). Next, the numerical computation of $V_1(f)$ shows that placing the origin of time at the beginning of the pulse gives a phase angle

$$\alpha_1(f) = \tan^{-1} \frac{b}{a} \tag{A-5}$$

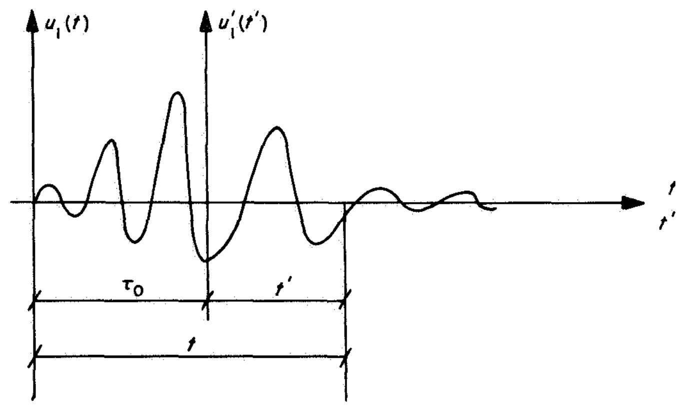

Fig. A-1. Typical pulse

between $-\pi$ and $+\pi$, which varies very rapidly with f. The plotting of this angle $\alpha_1(f)$ is therefore very inconvenient. In order to correct this situation we note that $\alpha_1(f)$ is in fact a superposition of a slow oscillation on a straight line of negative slope. To obtain this slope we straighten out the phase angle by extending $\alpha_1(f)$ beyond the interval $-\pi, +\pi$. To this end we construct

$$k(f) = \alpha_1(f) - 2n\pi$$
$$n = 0, 1, 2, \cdots \tag{A-6}$$

where $n = 0$ for $f = 0$ and n is increased by unity each time there is a sudden jump in $\alpha_1(f)$ from $-\pi$ to $+\pi$. (Fig. A-2). Then, $k(f)$ is written as follows

$$k(f) = \alpha_1'(f) - \beta f \tag{A-7}$$

where β is the slope of the straight line.

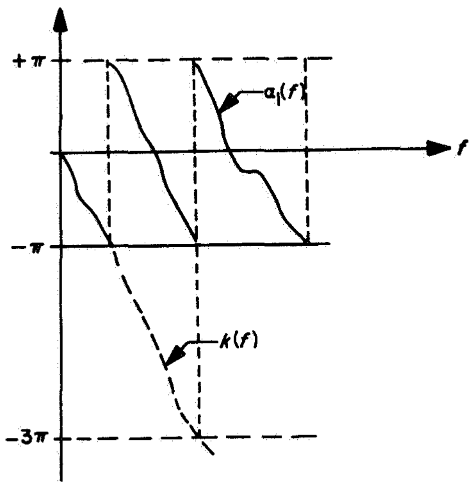

Fig. A-2. Phase angle

Removing the term βf from Eq. (A-7) makes the plotting of the phase angle much more meaningful since only the slowly varying $\alpha_1'(f)$ is plotted. We now show that the slope β is effectively a time translation τ_0. To this end we make the change of variable

$$t' = t - \tau_0 \tag{A-8}$$

therefore $u_1(t)$ is changed into $u_1'(t')$

$$u_1'(t') = u_1(t' + \tau_0) = u_1(t) \tag{A-9}$$

We then take the Fourier transform of $u_1'(t')$, i.e.,

$$V_1'(f) = \int_{-\infty}^{+\infty} u_1'(t')\, e^{-i2\pi f t'}\, dt' \tag{A-10}$$

and finally using Eqs. (A-8) and (A-9) to have

$$V_1'(f) = \int_{-\infty}^{+\infty} u_1(t)\, e^{-i2\pi f(t-\tau_0)}\, dt \tag{A-11}$$

$$V_1'(f) = e^{i2\pi f\tau_0}\, V_1(f) \tag{A-12}$$

Equation (A-12) shows that only the phase angle of the Fourier transform of $u_1(t)$ is affected by a time translation τ_0 with no change for the modulus. If $\alpha_1'(f)$ is the phase angle of $V_1'(f)$ we have, from Eq. (A-12)

$$\alpha_1'(f) = k(f) + 2\pi f\tau_0 \tag{A-13}$$

Comparison to Eq. (A-7) shows that

$$\beta = 2\pi\tau_0 \tag{A-14}$$

For numerical computation the origin of time is first temporarily chosen at the beginning of the pulse and the Fourier transform computed. Then a least-square fitting technique is used to obtain τ_0. For a proper construction of Eq. (A-7) the local slope of $\alpha_1(f)$ must remain relatively small. Hence, the determination of τ_0 is made in two steps:

1. An arbitrary time shift t_1 is applied to the pulse, i.e., $V_1(f)$ is replaced by $V_1''(f)$

$$V_1''(f) = V_1(f)\,(\cos 2\pi f t_1 + i \sin 2\pi f t_1) \tag{A-15}$$

   The slope of the corresponding phase angle $\alpha_1''(f)$ is therefore small.

2. Then the least square fitting is applied to the new phase angle $\alpha_1''(f)$ to give an additional time shift t_0. The total time shift τ_0 is finally

$$\tau_0 = t_0 + t_1 \tag{A-16}$$

   Note that the criterion to choose the arbitrary time t_1 is to place the origin of time near the maximum amplitude of the pulse. An option is built in the program to eliminate the computation of t_0 if so desired, leaving only a time translation t_1.

III. Input

The input function $u_1(t)$ is available either from digital magnetic tape or from cards. When the input is a tape, the analog tape is digitized at a given rate and a 250-words/sec block set tape is prepared. The digital input is identified by a file number and a number of records in the file. All other inputs are self-explanatory in the name list.

IV. Output

The outputs are either printed or plotted. In addition, the function $u_2(t)$ can be stored on magnetic tape. The format for the preparation of this tape is the same as the input tape for $u_1(t)$. Then a special program transforms this digital tape into an analog tape which can be used for environmental vibration testing (Ref. 1).

V. Name-List

The input other than the input tape for $u_1(t)$ and the various options in the program are provided by a name-list, Table A-1. Note that each variable of this name-list has a built-in value which is automatically read by the program if the variable is not taken in the preparation of the name-list. If the variable is indicated, then its value overrides the built-in value. The name-list must start with the word $ PARAMS and end with $.

Table A-1. Name list

Variable	Type	"Built-in" value	Use
CALC	Integer	0	Defines which of the various cases of the sequence are to be performed. *Value* — *Calculation* 0 — Full calculation 1 — Fourier transform only Eq. (A-1) 2 — Inverse transform only Eq. (A-4) 3 — Transfer function only Eq. (10)
INPUT	Integer	0	Must be zero if $u_1(t)$ is an input on tape or equal to number of values of u_1 to be read in from cards. It is also the number of values of t in $u_2(t)$ when making an inverse transform only.
NV1	Integer	0	When performing an inverse transform only, NV1 specifies the number of values to be read in of the modulus of $V_2(f)$.
NV2	Integer	0	As above, NV2 specifies the number of values to be read in of the phase angle of $V_2(f)$. It is also used to specify the number of values to read in when $V_2(f)$ is given in real and imaginary part form.
MODE	Integer	0	Denotes, in an inverse transform only calculation, whether $V_2(f)$ is to be specified by modulus and phase angle (MODE = 1) or given by real and imaginary part (MODE = 0).
POINTS	Integer	2	Number of points to be used in interpolated construction of real and imaginary part of $V_2(f)$ from modulus and phase angle for inverse transform only case (POINTS = 2, linear interpolation).
FILE	Integer	1	Denotes file on input tape which contains $u_1(t)$. If more than one file is asked, the rest of the input being the same, simply add the new file number at end of the name-list. Record number, number of records and calibration factor are also added but only if they are different. However, only one output tape can be written.
REC1	Integer	1	Record number within FILE where first $u_1(t)$ is to be obtained.
NREC	Integer	1	Number of records of $u_1(t)$ to be read.
SCALE	Real	1.0	Calibration factor for $u_1(t)$ (volt/rad/sec^2, or volt/g)
RATE	Real	1000.0	Sampling rate used in digitizing $u_1(t)$.
TYPE	Integer	1	Determines which transfer function to compute, viscous (TYPE = 1) or hysteretic (TYPE = 2).
OUT	Integer	0	Controls the writing of $u_2(t)$ on tape. If OUT = 0, no tape will be written; if OUT $\neq$ 0, a tape will be prepared.
F01	Real	0.0	Initial frequency used for calculating all frequency dependent functions.
DF	Real	0.5	Frequency increment used in calculating frequency dependent functions.
KF1	Integer	241	Number of frequencies. (Thus, if F01 = 0, DF = 0.5, and KF1 = 241, frequency dependent functions will be computed at the points f = 0.0, 0.5, 1.0, 1.5, . . . ,120.)
F02	Real	0.0	Initial frequency to be used in Fourier transform inversion.
KF2	Integer	241	Number of frequencies to be used in transform inversion. (Note that provision has been made to invert $V_2(f)$ over a subset of its domain of definition; hence, F02 $\geq$ F01, and F02 + KF2*DF $\leq$ F01 +KF1*DF. A check on this condition is made by the program and an error message issued if not met.)

Table A-1 (contd)

Variable	Type	"Built-in" value	Use
PRINT	Integer	1	An array of seven numbers which control the printout of the various calculated functions. A nonzero value for PRINT will cause the *i*th function to be listed. The ordered list of functions is: i — *Function* 1 — $v_1(t)$ 2 — $V_1(f)$ 3 — $H_1(f)$ 4 — $F(f)$ — Real, imaginary, modulus, phase angle (for 2–6) 5 — $H_2(f)$ 6 — $V_2(f)$ 7 — $v_2(t)$
PLOT	Integer	1	An array of seven numbers like PRINT which control the plotting of the various functions (as above). A zero value inhibits plotting. For the complex functions, a nonzero value which is also negative causes two plots to be made of the phase angle, one on a linear scale, and the other on a semi-log scale. Only modulus and phase angle are plotted.
DT	Real	0.001	Time increment, required when $v_1(t)$ is read from cards or when performing an inverse transform only to obtain $v_2(t)$.
N1	Integer	0	Number of values to be read in of f_n, M_n, ξ_n, $\emptyset_{1n}$, and $\emptyset_{2n}$ for $H_1(f)$.
MN1	Real	0.0	The array of M_n's for computing $H_1(f)$.
PSIN1	Real	0.0	The array of ξ_n's for $H_1(f)$.
FN1	Real	0.0	The array of f_n's for $H_1(f)$.
A1	Real	0.0	The $\emptyset_1$ column vector for $H_1(f)$.
B1	Real	0.0	The $\emptyset_2$ column vector for $H_1(f)$.
N2	Integer	0	Number of values to be read in of f_n, M_n, ξ_n, $\emptyset_{1n}$, and $\emptyset_{2n}$ for $H_2(f)$.
MN2	Real	0.0	The array of M_n's for computing $H_2(f)$.
PSIN2	Real	0.0	The array of ξ_n's for $H_2(f)$.
FN2	Real	0.0	The array of f_n's $H_2(f)$.
A2	Real	0.0	The $\emptyset_1$ column vector for $H_2(f)$.
B2	Real	0.0	The $\emptyset_2$ column vector for $H_2(f)$.
T1	Real	0.0	Transformation variable t_1 which is applied to the Fourier transform $V_1(f)$ of $v_1(t)$ according to Eq. (A-15); also the initial value of t corresponding to $v_2(t)$ when performing an inverse transform only.
TST	Real	0.5	Refer to straightening out $\alpha_1(f)$ according to Eq. (A-6). The computer parameter TST is used to determine how close to $-\pi$ the angle $\alpha_1(f)$ must be to subtract 2π from $\alpha_1(f)$.
TFLAG	Real	0.0	Used as a flag for computation of τ_0 TFLAG = 0.0 $\quad \tau_0 = \tau_1$ TFLAG = 1.0 $\quad \tau_0 = \tau_0 + \tau_1$
UFLAG	Real	0.0	Determines offset to be applied to the function $v_1(t)$ UFLAG = 0.0 NO OFFSET UFLAG = 1.0 Offset = time average of $v_1(t)$ UFLAG = 2.0 Offset = DC
DC	Real	0.0	Offset to be applied to $v_1(t)$ if UFLAG = 2.0
PFLAG	Real	0.0	Determines if modulus and phase angle should be punched out on cards PFLAG = 0.0 No cards PFLAG = 1.0 Punch cards
SET	Real	0.0	Data set identification (four digits xxxx.)

In addition to the name-list input, the four options of CALC require additional input as shown in Table A-2.

Table A-2. Additional input

Case	Other input
0	If $u_1(t)$ is on tape, this tape is to be mounted on SYSUT5. If $u_2(t)$ is to be written on tape, a tape should be mounted on SYSUT6. If $u_1(t)$ is input on cards, these follow the name-list data and are punched in order $u_1, u_2, \ldots, u_n$ according to the format 6E12.8
1	Same as for Case = 0
2	If $V_2(f)$ is given by real and imaginary parts (MODE=0), then they are punched two points per card according to the format 6E12.8 as follows: f_i, $Re(V_{2i})$, $Im(V_{2i})$, f_{i+1}, $Re(V_{2i+1})$, $Im(V_{2i+1})$ If $V_2(f)$ is given by modulus and phase angle, the moduli are given first in the order: f_i, $Mod(V_{2i})$, f_{i+1}, $Mod(V_{2i+1})$, ... in the format 6E12.8. These are then followed by the phase angles f_i, $Phase(V_{2i})$, f_{i+1}, $Phase(V_{2i+1})$, ... in the same format
3	No other input

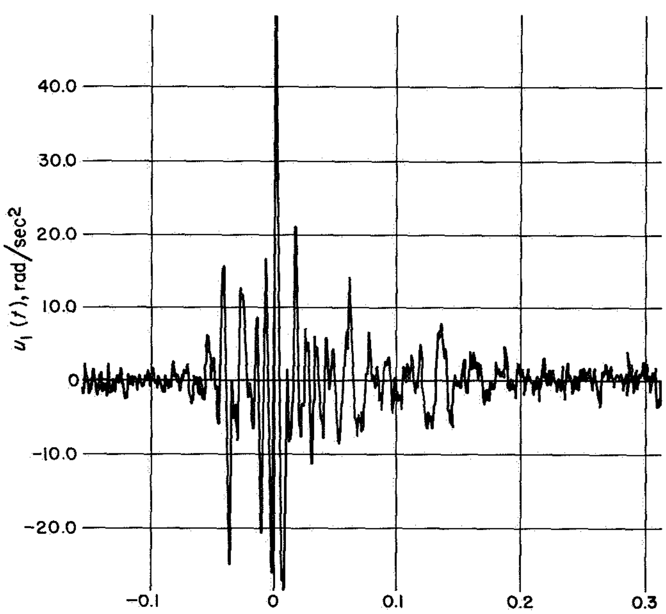

Fig. A-3. Input torsional acceleration, *Ranger VIII*

As an example, Figs. A-3 through A-14 show a typical output calculation corresponding to *Ranger VIII* data. Part of the input is indicated in Table A-3. The imput representing $u_1(t)$ was on digital tape.

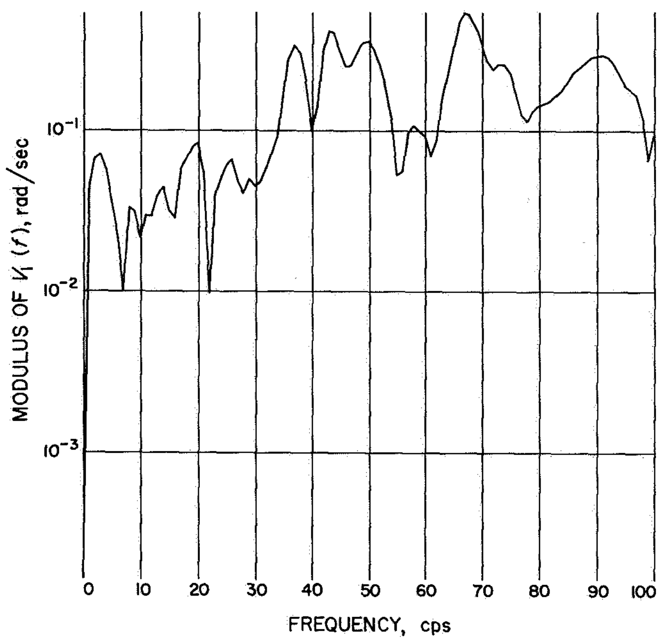

Fig. A-4. Fourier transform of input acceleration, modulus

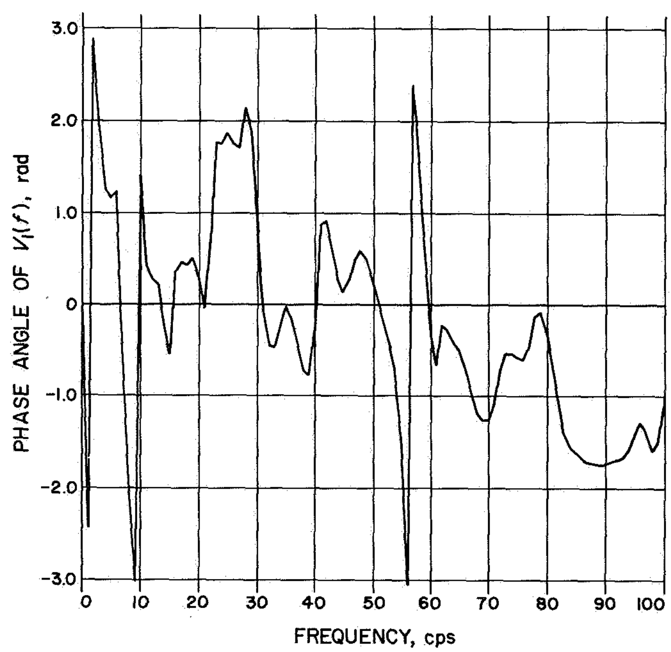

Fig. A-5. Fourier transform of input acceleration, phase angle

Table A-3. Input sample

```
$PARAMS  FILE=1,REC1=1,NREC=3,SCALE=.04975,T1=.12,TFLAG=1,
 UFLAG=1,RATE=3200.,DF=1.0,KF1=101,KF2=101,PRINT=7*1,PLOT=1,-1,5*1,
 N1=9,MN1=105400.,8*1.,PSIN1=0.0,8*.03,FN1=0.,12.58,34.48,55.29,67.62,
 79.82,80.46,110.62,147.18,A1=1.,.126E-2,.141E-1,.343E-2,.578E-2,
 .152E-2,-.467E-2,-.415E-2,.163E-2,B1=1.,.984E-3,-.123E-2,-.604E-3,
 .139E-2,-.825E-3,-.138E-2,.132E-2,-.855E-3,N2=16,MN2=185000.,15*1.,
 PSIN2=0.,15*.03,FN2=0.,11.74,12.17,14.22,16.80,16.98,45.43,54.35,
 67.50,70.19,96.60,110.0,130.0,132.6,145.0,158.1,A2=1.,.108E-2,
 .953E-3,.382E-3,.819E-3,.358E-2,.498E-2,.164E-2,.519E-3,.135E-1,
 .827E-2,.507E-2,-.153E-2,.217E-2,.860E-2,-.341E-2,B2=1.,-.237E-3,
 .219E-3,-.760E-3,.441E-3,-.193E-2,.803E-3,.226E-3,-.212E-2,-.205E-3,
 .595E-3,-.797E-3,.733E-3,.761E-3,.264E-3,.946E-3 $
```

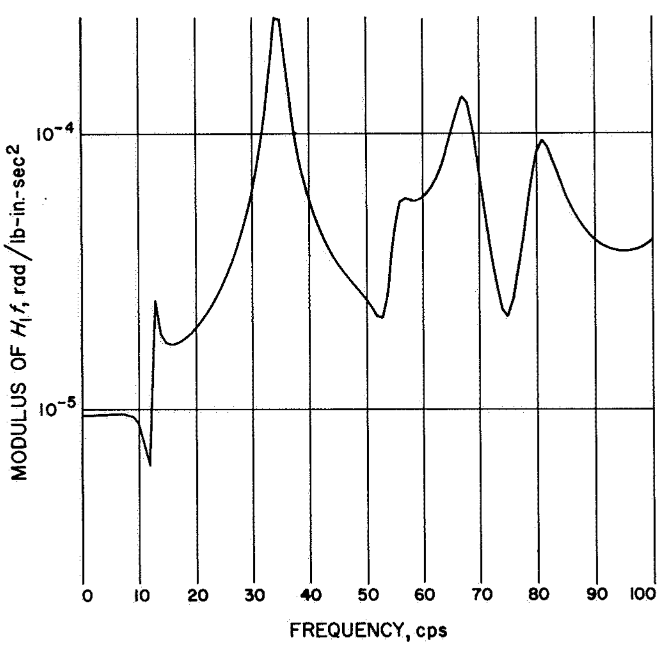

Fig. A-6. Transfer function of *Atlas/Agena/Ranger*, modulus

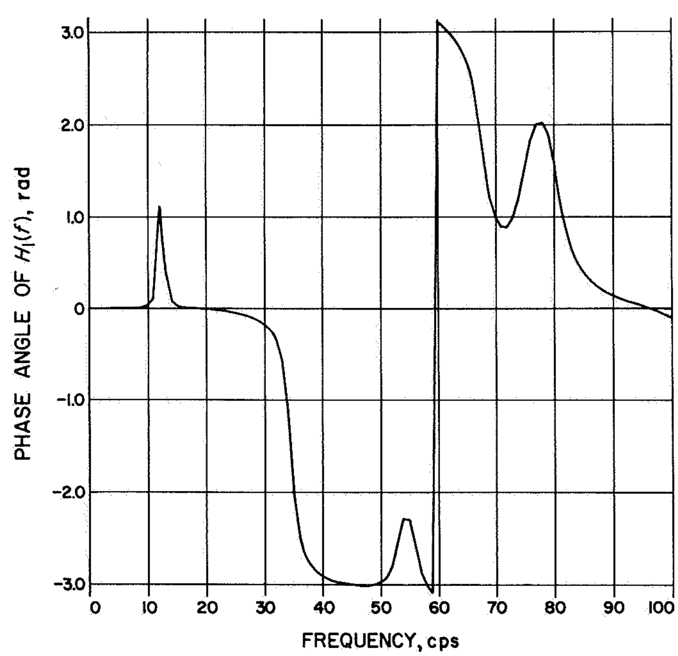

Fig. A-7. Transfer function of *Atlas/Agena/Ranger*, phase angle

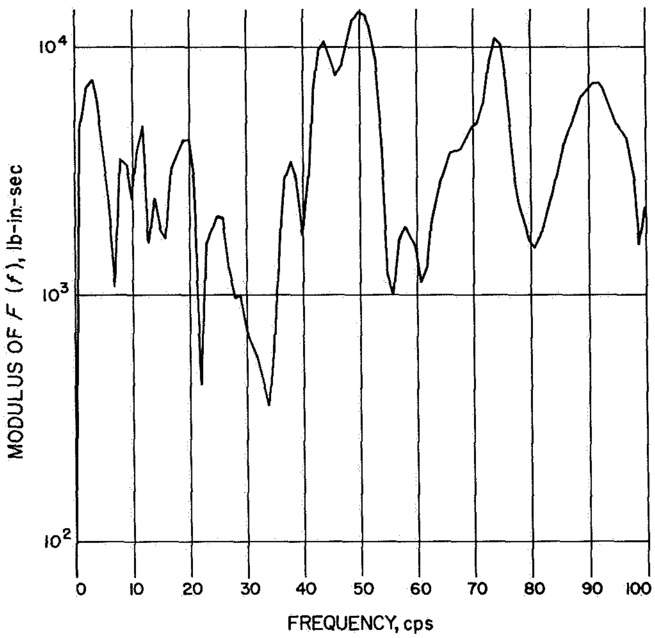

Fig. A-8. Fourier transform of gimbal torque, modulus

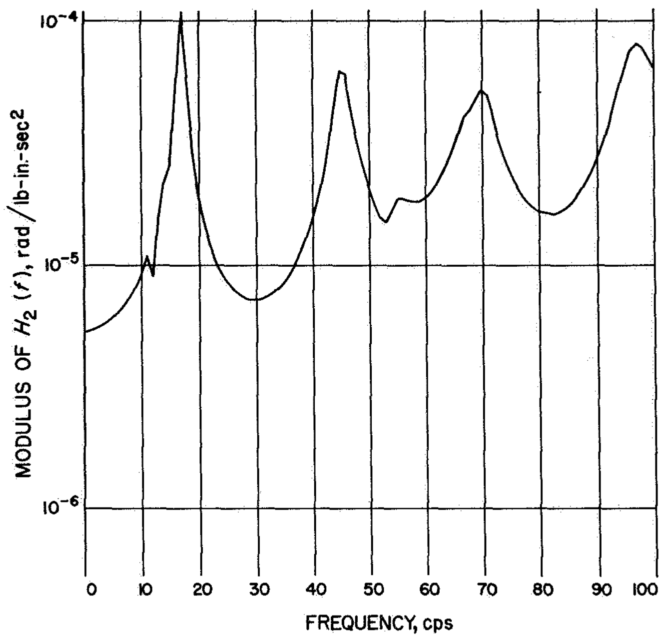

Fig. A-10. Transfer function of *Atlas/Centaur/Surveyor*, modulus

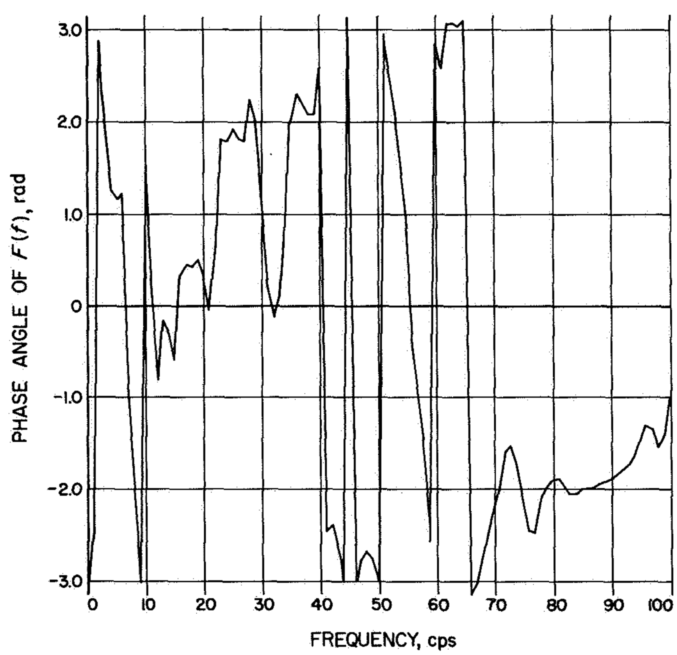

Fig. A-9. Fourier transform of gimbal torque, phase angle

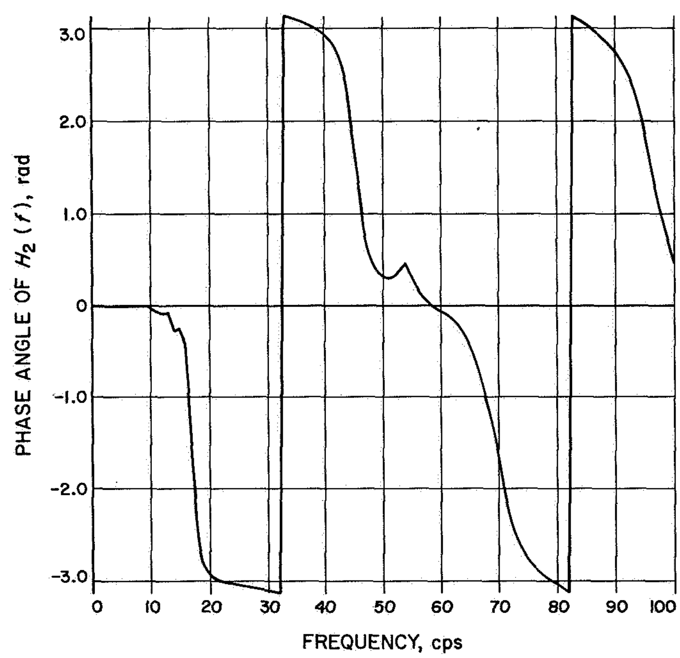

Fig. A-11. Transfer function of *Atlas/Centaur/Surveyor*, phase angle

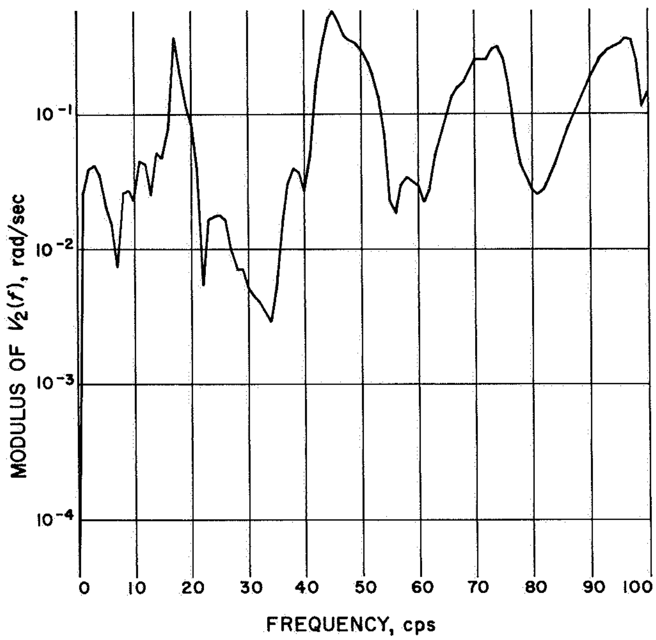

Fig. A-12. Fourier transform of *Surveyor* field joint acceleration, modulus

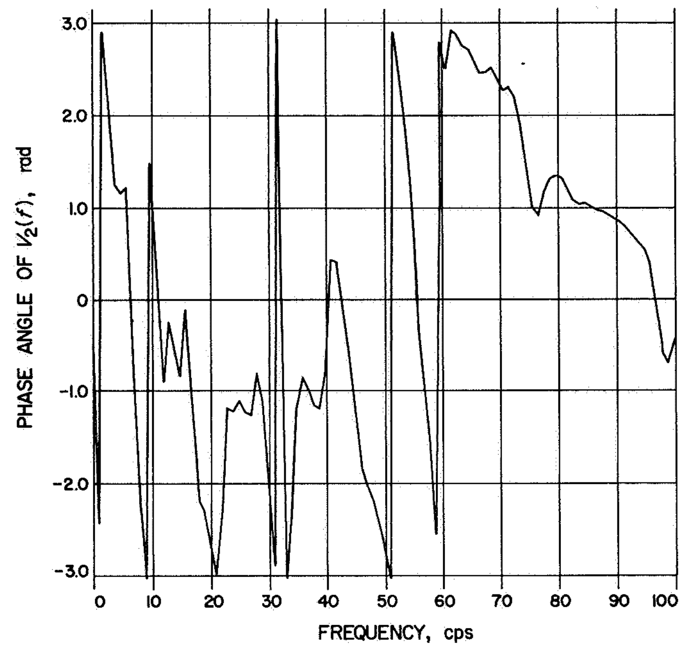

Fig. A-13. Fourier transform of *Surveyor* field joint acceleration, phase angle

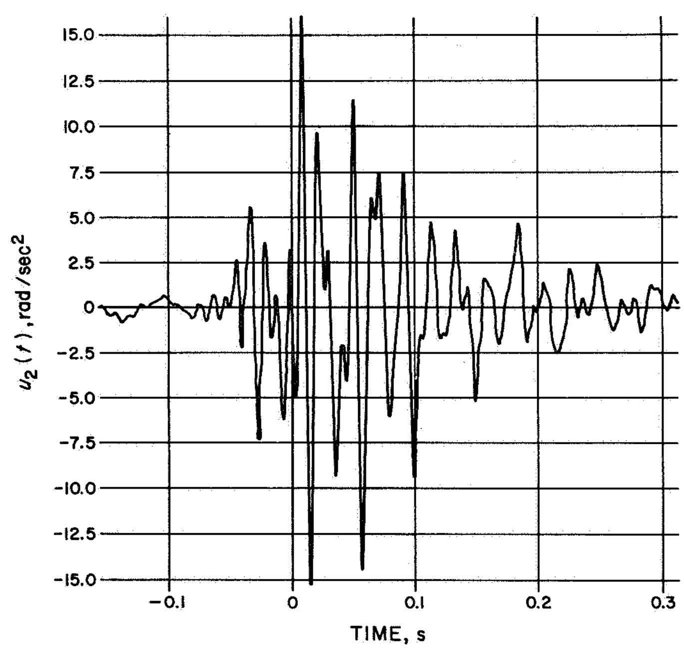

Fig. A-14. *Surveyor* field joint acceleration

Appendix B
Complex Eigenvalue Problem

Equation (23) was solved using the JPL complex eigenvalue program 5363000 ENGSIP (Ref. 5). The input matrix **A** was computed from Eq. (22) in which

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$m_0 = 1.054$$

$$\overline{\boldsymbol{\Phi}}_1 = \begin{Bmatrix} 0.126 \\ 1.410 \\ 0.343 \\ 0.578 \\ 0.152 \\ -0.467 \\ -0.415 \\ 0.163 \end{Bmatrix} \qquad \overline{\boldsymbol{\Phi}}_2 = \begin{Bmatrix} 0.984 \\ -1.230 \\ -0.604 \\ 1.390 \\ -0.825 \\ -1.380 \\ 1.320 \\ -0.855 \end{Bmatrix}$$

The matrices **A**, **C**, and **K** are:

$$\mathbf{A} = \begin{bmatrix} 1.131 & 1.462 & 0.356 & 0.5997 & 0.158 & -0.484 & -0.430 & 0.169 \\ -0.163 & -0.828 & -0.445 & -0.749 & -0.197 & 0.605 & 0.538 & -0.211 \\ -0.0802 & -0.898 & 0.782 & -0.368 & -0.0968 & 0.297 & 0.264 & -0.104 \\ 0.184 & 2.066 & 0.503 & 1.846 & 0.222 & -0.684 & -0.608 & 0.239 \\ -0.1096 & -1.226 & -0.298 & -0.503 & 0.868 & 0.406 & 0.3605 & -0.141 \\ -0.183 & -2.051 & -0.499 & -0.841 & -0.221 & 1.679 & 0.604 & -0.237 \\ 0.175 & 1.961 & 0.477 & 0.804 & 0.212 & -0.649 & 0.422 & 0.227 \\ -0.114 & -1.271 & -0.309 & -0.521 & -0.137 & 0.421 & 0.374 & 0.853 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 4.75 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 13.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20.8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 30.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 30.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 41.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 56.5 \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} 6250 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 46940 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 120700 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 180500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 251500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 255600 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 483100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 855200 \end{bmatrix}$$

Finally the roots of the complex eigenvalue problem are shown in Table B-1.

Table B-1. Roots of the complex eigenvalue problem

Real part λ_n	Imaginary part ω_n
−405.44	0
482.69	0
−2.1737	±75.116
−10.139	±337.22
−14.124	±470.71
−392.82	±474.628
354.311	±480.199
−14.992	±500.140
−26.176	±852.709

References

1. Trubert, M. R., *Use of Ranger Flight Data in the Synthesis of a Torsional Acceleration Transient for Surveyor Vibration Qualification Testing,* Technical Memorandum 33-237, Jet Propulsion Laboratory, Pasadena, California, April 19, 1966.

2. Crandall, S. H., and Mark, W. D., *Random Vibration in Mechanical Systems,* p. 57, Academic Press, Inc., New York, 1963.

3. Garba, J. A., Gayman, W. H., and Wada, B. K., *Computation of Torsional Vibration Modes of Ranger and Surveyor Space Vehicles,* Technical Memorandum 33-277, Jet Propulsion Laboratory, Pasadena, California (to be published).

4. Dalrymple, S. H., *Fourier Transform Technique for Prediction of Torsional Pulse Environment for Surveyor,* Report prepared by Planning Research Corporation for the Jet Propulsion Laboratory, June 17, 1966.

5. Muller, D. E., "A Method of Solving Algebraic Equations Using an Automatic Computer," *Mathematical Tables and Aids for Computation,* Vol. 10, pp. 208–215, 1956.