REC-ERC-87-4

DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS

May 1987

Engineering and Research Center

U. S. Department of the Interior

Bureau of Reclamation

7-2090 (4-81)
Bureau of Reclamation

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.
REC-ERC-87-4

2. GOVERNMENT ACCESSION NO.

3. RECIPIENT'S CATALOG NO.

4. TITLE AND SUBTITLE
Dynamic Testing of Homogeneous Embankment Models

5. REPORT DATE
May 1987

6. PERFORMING ORGANIZATION CODE
D-1542

7. AUTHOR(S)
R. W. Luehring

8. PERFORMING ORGANIZATION REPORT NO.
REC-ERC-87-4

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Bureau of Reclamation
Engineering and Research Center
Denver, CO 80225

10. WORK UNIT NO.

11. CONTRACT OR GRANT NO.

13. TYPE OF REPORT AND PERIOD COVERED

12. SPONSORING AGENCY NAME AND ADDRESS
Same

14. SPONSORING AGENCY CODE
DIBR

15. SUPPLEMENTARY NOTES
Microfiche and/or hard copy available at the E&R Center, Denver, Colorado

Editor:RDM(c)

16. ABSTRACT

Physical model tests were performed on homogeneous cohesionless embankments subjected to idealized ground motion. During the testing, the response of model embankments to a sinusoidal input motion was monitored, and acceleration and displacement data were collected. An analytical model, based on Newmark's sliding block theory, was used to estimate the permanent displacement of sliding blocks subjected to input dynamic motion. Analytical results were compared with physical model test results to assess the validity of the mathematical model.

It was concluded that the mathematical model did not accurately predict the deformation of homogeneous cohesionless embankment models because of difficulty in determining the effective shear strength of the material at low normal stresses.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. *DESCRIPTORS*-- deformation/ homogeneous/ cohesionless/ embankments/ models/ displacement/ sliding block/ dynamic loading/ laboratory shaking experiments/ impact/ static shear strength/ dynamic properties/ embankment stability/ seismic safety/ limit equilibrium/ acceleration/ sliding mass/ failure plane/ earthquakes/ earth dams/ sinusoidal input

b. *IDENTIFIERS*--

c. *COSATI Field/Group* 08M *COWRR:* 0813 *SRIM:*

18. DISTRIBUTION STATEMENT
Available from the National Technical Information Service, Operations Division, 5285 Port Royal Road, Springfield, Virginia 22161.
(Microfiche and/or hard copy available from NTIS)

19. SECURITY CLASS (THIS REPORT)
UNCLASSIFIED

20. SECURITY CLASS (THIS PAGE)
UNCLASSIFIED

21. NO. OF PAGES
191

22. PRICE

REC-ERC-87-4

DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS

by

R. W. Luehring

May 1987

Geotechnical Branch
Division of Research and Laboratory Services
Engineering and Research Center
Denver, Colorado

SI METRIC

UNITED STATES DEPARTMENT OF THE INTERIOR ★ BUREAU OF RECLAMATION

ACKNOWLEDGMENTS

The author expresses thanks to Richard A. Young, former Head, Soil and Rock Mechanics Section, for valuable guidance, suggestions, and enthusiasm throughout the course of this study; Fred A. Travers of the Concrete and Structural Branch for his assistance in data acquisition, computer-aided documentation, and plotting; Kiran K. Adhya of the Embankment Dams Branch for using the computer program DYNDSP, which computes displacements, and for conducting the deformation analysis; Paul C. Knodel, Chief, Geotechnical Branch, and Dr. Francis G. McLean, Chief, Division of Research and Laboratory Services, for their encouragement and valuable suggestions and information about similar studies conducted by the Public Works Research Institute. Thanks are also due to James M. Boernge of the Soil and Rock Mechanics Section, Henry J. Hoff of the Soil and Rock Testing Station, and Stephen N. Gavlick of the Computer Graphics Section for their computer programming and plotting assistance; Michael W. Knipps and Douglas Roberts for their help in constructing the models; Stephen A. Coman of the Soil and Rock Mechanics Section for plotting and gathering literature; and Mary Ann Trujillo of the Soil and Rock Mechanics Section for her editorial assistance.

As the Nation's principal conservation agency, the Department of the Interior has responsibility for most of our nationally owned public lands and natural resources. This includes fostering the wisest use of our land and water resources, protecting our fish and wildlife, preserving the environmental and cultural values of our national parks and historical places, and providing for the enjoyment of life through outdoor recreation. The Department assesses our energy and mineral resources and works to assure that their development is in the best interests of all our people. The Department also has a major responsibility for American Indian reservation communities and for people who live in Island Territories under U.S. Administration.

The information contained in this report regarding commercial products or firms may not be used for advertising or promotional purposes and is not to be construed as an endorsement of any product or firm by the Bureau of Reclamation.

CONTENTS

TABLES

FIGURES

CONTENTS – Continued

INTRODUCTION

The purpose of this investigation was to observe and document the occurrence of deformation of homogeneous cohesionless embankment models subjected to an idealized input motion. The displacements of the physical models were compared with the permanent displacements calculated by a mathematical computer model based on Newmark's sliding block theory. The effectiveness of the analytical model in accurately predicting the permanent displacement of a physical model was evaluated. This study includes:

1. A review of published methods used to compute permanent embankment displacement caused by dynamic loading

2. A review of literature that summarizes laboratory shaking experiments using impact, eccentrically driven low frequency, and hydraulic horizontal shaking tables

3. A discussion of the embankment model setup and testing, and of the analytical deformation analysis performed

4. Results of laboratory tests used to determine static shear strength and dynamic properties of materials used in the physical model study

CONCLUSIONS

Because the transverse length-to-height ratio of the models tested was 2:1, 30 to 15 inches (762 to 381 mm), the restraining effects of the abutments influenced the resultant failure configuration. It would be necessary to increase the ratio to 4:1 or greater to eliminate this effect.

Because of the limited number of model tests and the variation in test parameters, no meaningful statistical correlation of test parameters could be made.

The embankment displacement computed using the analytical model is sensitive to the selection of shear strength parameters ϕ', the angle of internal friction, and c', the apparent cohesion. The angle of internal friction (ϕ') at very low normal stresses is uncertain. This points to the need to test in the centrifuge at higher stress levels.

The effect of water in the form of capillary suction pressure (apparent cohesion) or excess pore pressure could only be estimated.

The results of the analytical model did not agree with physical model test observations except when a friction angle, ϕ', of about 45° was assumed.

LITERATURE REVIEW – STATE OF THE ART

Stability of embankments under dynamic loading is a subject of great practical significance in geotechnical engineering. In the past, the seismic safety of an embankment was evaluated by pseudostatic analysis. By assuming an embankment experiences a constant lateral acceleration, the factor of safety of the embankment with respect to shear strength could be calculated from results of normal static slope stability analyses.

The method of limit equilibrium is one form of static analysis that can be used in a pseudostatic evaluation of dynamic stability. The method of limit equilibrium satisfies all equations of static equilibrium and is generally used in estimating the stability of natural slopes and embankment structures along potential slide surfaces (Chugh, 1982 [1]*). Numerical solution procedures used to account for deformation properties of materials are limited for slope stability problems. Procedural limitations are due to (a) uncertainties in material properties determined under all stress and boundary conditions encountered in real soil structures, and (b) the lack of a generalized soil model that can realistically consider load deformation properties of soil [1].

Recently, the use of static stability analysis techniques on embankments in seismically active areas has been found inadequate (Prakash, 1977 [2]; and Kutter, 1984 [3]). The essential link between static and dynamic slope stability analysis is the determination of yield acceleration, which is defined as the threshold average acceleration for a slide mass above which permanent deformations occur.

For dynamic slope stability analysis, several analytical methods for computing permanent deformations have been proposed. Five methods that are commonly used or have been proposed are (a) Newmark's method [4], (b) Ambrasey's method [5], (c) Makdisi and Seed's method [6], (d) Wilson and Clough's step-by-step integration method [7], and (e) Kutter's modification to Newmark's method, which uses a decoupling technique [3]. These methods are described in limited detail in this report. However, in-depth discussions of the approaches used to compute permanent deformations and their origins are found in the references cited. Portions of the following discussion were obtained from these references and from a paper by Shieh and Huang [8].

Newmark's Method

Newmark [4] has shown that the permanent displacement of a sliding mass relative to the base is the sum

* Numbers in brackets refer to entries in the bibliography.

of increments of displacement occurring during a number of individual pulses of ground motion. Whenever the ground acceleration exceeds the yield acceleration, sliding will occur along the failure plane, and the magnitude of the displacement is computed by double integration of the acceleration-time history. By assuming resistance to sliding to be rigid-plastic and asymmetrical for an embankment that suffers a slope failure from seismic ground motions, the average earthquake-induced horizontal displacement, U_m, is given by:

$$U_m = \frac{V^2}{2gN}\left(1 - \frac{N}{A}\right) \quad (1)$$

where:

V = maximum ground velocity (LT^{-1}),
A = maximum resistance coefficient (dimensionless),
N = maximum earthquake acceleration coefficient (dimensionless), and
g = acceleration due to gravity (LT^{-2}).

The relative displacement will be permanent if no further motion occurs. Furthermore, freeboard loss, L, can be calculated using the following relationship:

$$L = U_m \tan\delta \quad (2)$$

where δ = the angle of the sliding plane with the horizontal (degrees).

The Newmark charts (see fig. 1) for computing the permanent displacement were developed for normalized earthquakes with maximum acceleration of 0.5 g and maximum velocity of 30 in/s (762 mm/s).

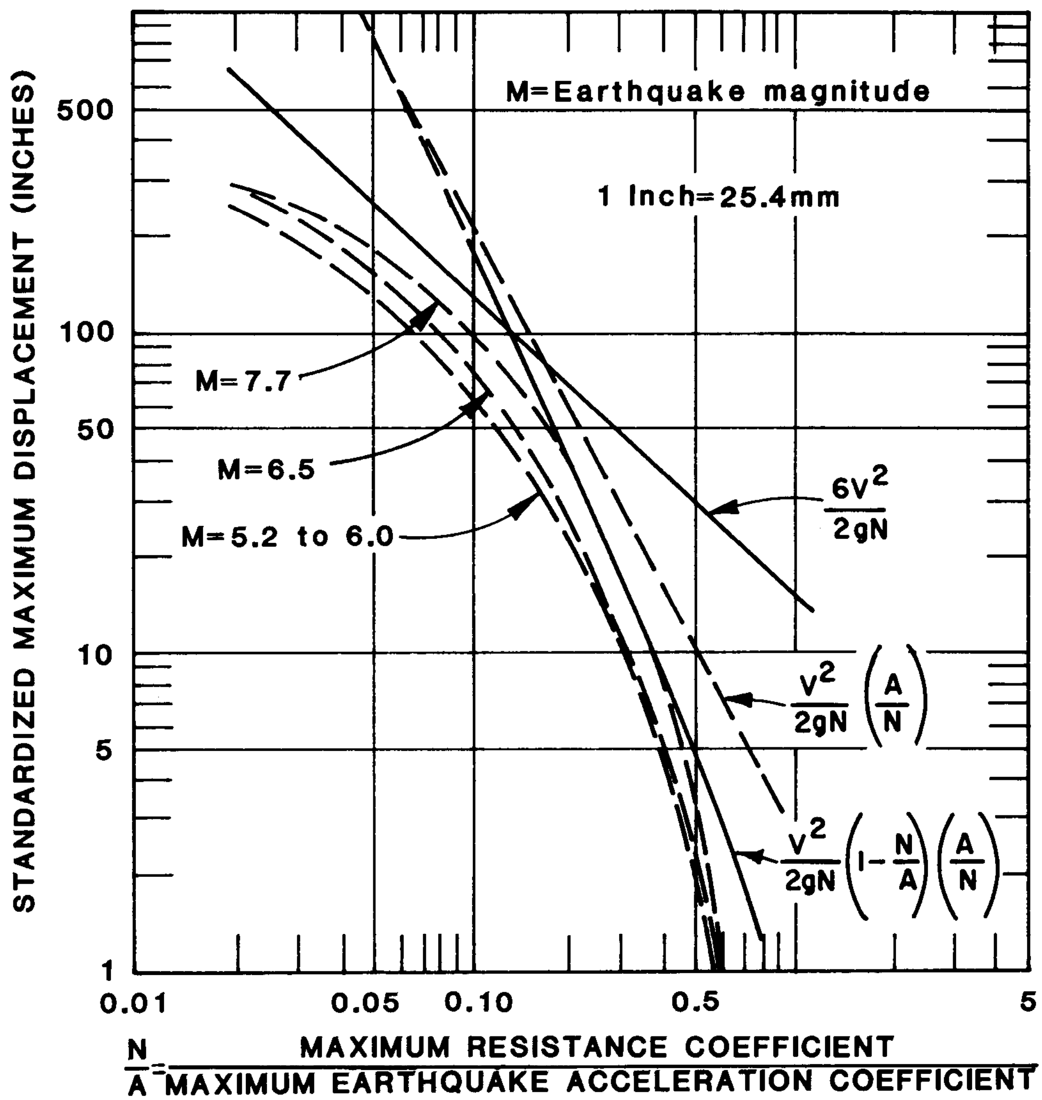


Figure 1. – Standardized maximum displacement for normalized earthquakes. After Newmark, 1965 [4]; from Shieh and Huang, 1981 [8].

Ambrasey's Method

With respect to residual displacements in an earthfill dam, Ambrasey (1974) [5] developed an upper bound empirical equation for a crude evaluation of residual displacement. The equation is:

$$\log_{10}U = 2.3 - 3.3\left(\frac{k_c}{k_m}\right) \tag{3}$$

where:

U = residual displacement in centimeters (L),
k_c = critical acceleration needed to reduce the factor of safety to 1 (LT^{-2}), and
k_m = maximum input acceleration (LT^{-2}).

Equation (3) is useful only for earthquake magnitudes, M, less than 6.5; for $0.1 < (k_c/k_m) < 0.8$; and for surfaces sloped less than 2:1.

Ambrasey also proposed the following equation to compute the critical acceleration:

$$k_c = \frac{\tan\phi' - \tan\beta}{1 + \tan\phi' \tan\beta} \tag{4}$$

where:

ϕ' = effective angle of internal friction of the material (degrees), and
β = slope angle of the dam (degrees).

Equation (4) shows that k_c is a function of the geometry of the mass, soil properties, and static safety factor of the mass profile.

Makdisi and Seed's Method

Procedures for computing deformations of earthfill dams during earthquakes have also been proposed by Makdisi and Seed (1978) [6]. Their proposed approach is equivalent to Newmark's approach except earthquake excitation is obtained from the dynamic response of the embankment using either shear-beam or finite element models. Makdisi and Seed's method assumes perfectly elastoplastic soil behavior. Values of yield acceleration are functions of the embankment geometery, of the undrained shear strength of the material (or the reduced shear strength caused by shaking), and of the location of the potential sliding mass. The numerical application of this method can be carried out using figures 2 and 3.

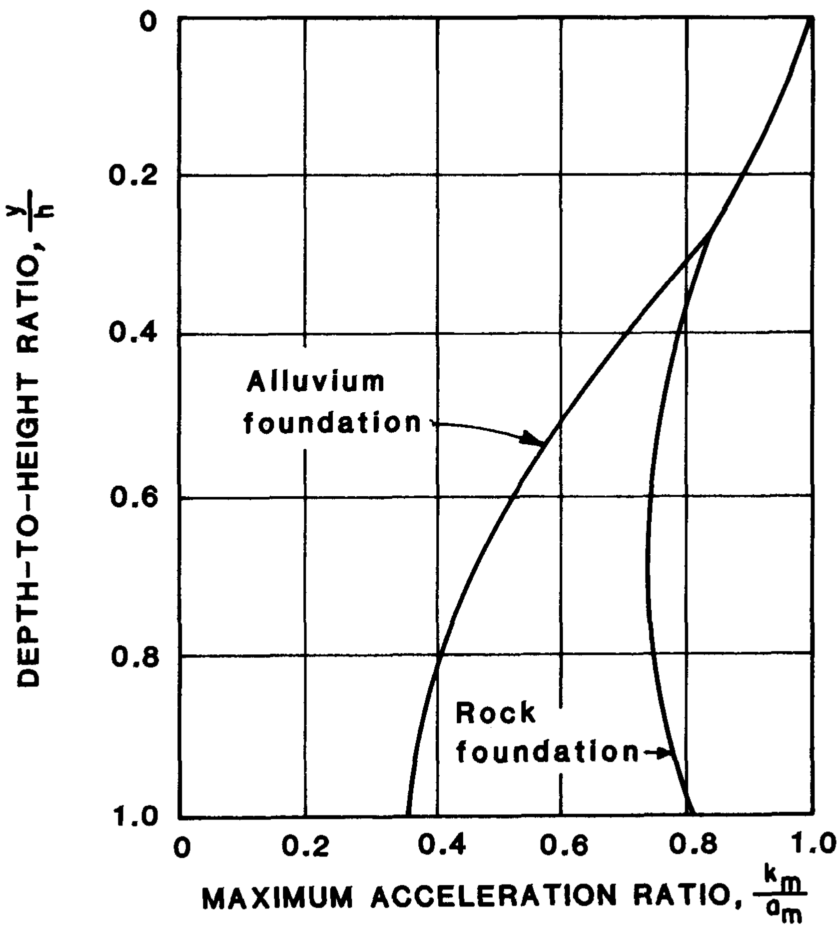


Figure 2. – Average maximum acceleration ratio versus depth-to-height ratio of sliding mass. After Makdisi and Seed, 1978 [6], from Shieh and Huang, 1981 [8].

Yield accelerations are obtained from the pseudostatic slope stability analyses. The basic steps required in the computation are:

a. Determine the yield acceleration from the pseudostatic stability analysis.

b. Determine the maximum acceleration ratio (k_m/a_m) from figure 2 for various depths of the sliding mass. In the ratio, k_m is the average maximum acceleration of the sliding mass, and a_m is the maximum crest acceleration. The ratio y/h is the ratio of the specific depth, y, of a potential mass to the embankment height, h.

c. Evaluate the magnitude of U, the average normalized displacement, from figure 3.

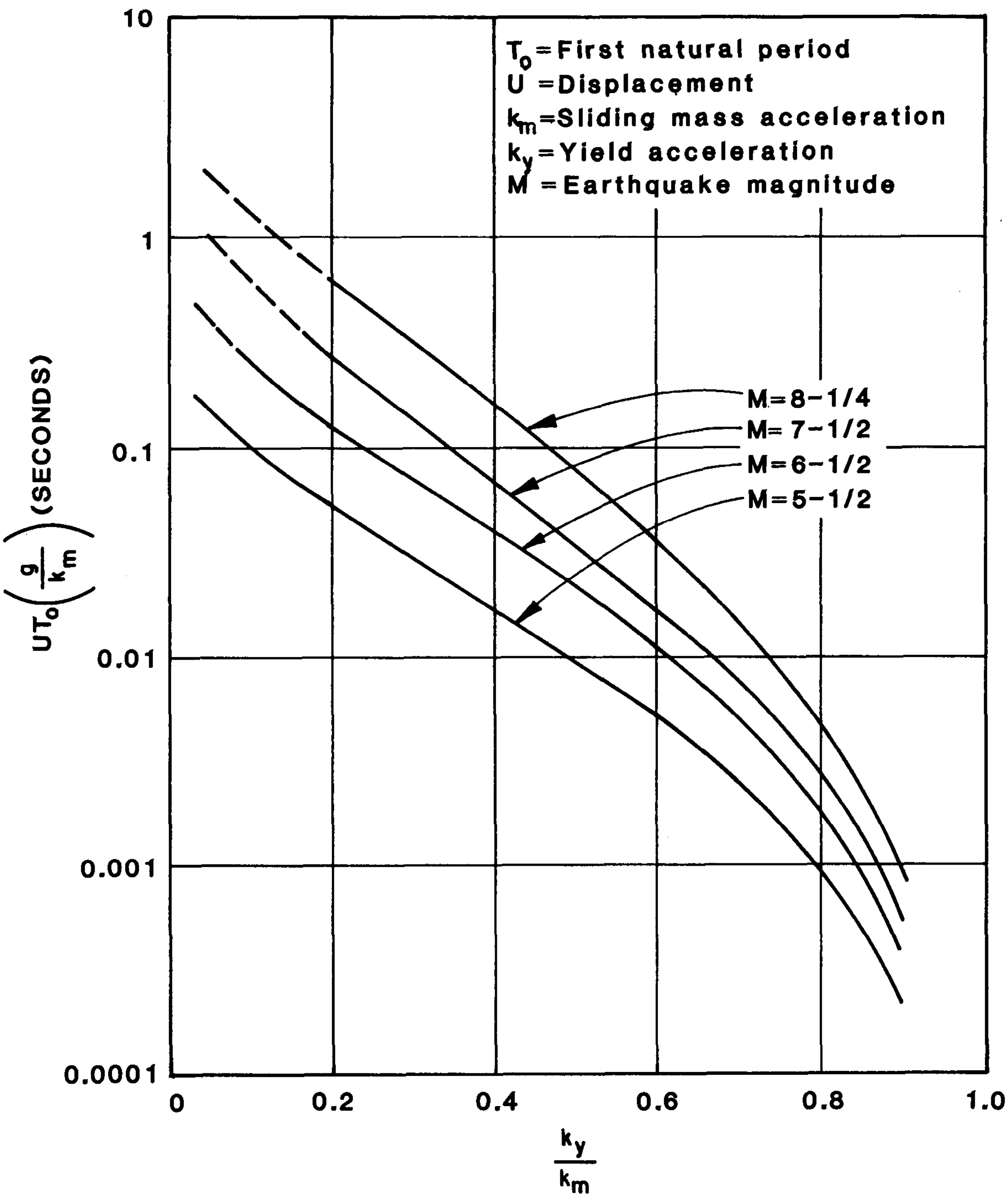


Figure 3. – Average normalized displacement versus normalized yield acceleration. After Makdisi and Seed, 1978 [6]; from Shieh and Huang, 1981 [8].

Wilson and Clough's Step-by-Step Integration Method

Wilson and Clough (1962) [7] have shown that the equilibrium of a single degree system at time t is expressed by the following equation for a viscous form of damping:

$$a_t + 2\lambda\omega V_t + \omega^2 X_t = a_g \tag{5}$$

where:

X_t = relative displacement of the system with respect to the foundation (L),
V_t = velocity of the system (LT^{-1}),
a_t = acceleration of the system (LT^{-2}),
λ = damping ratio (dimensionless),
ω = natural frequency in radians per second (T^{-1}), and
a_g = ground acceleration (LT^{-2})

The step-by-step solution of equation (6) proceeds as follows: The initial displacement, X_o, the initial velocity, V_o, and initial ground acceleration a_{g_o}, are given as the initial conditions of the problem. The initial system acceleration, a_o, is obtained from equation (6) as:

$$a_o = a_{g_o} - 2\lambda\omega V_o - \omega^2 X_o \tag{6}$$

The step-by-step response of the system is obtained by repeated application of several equations. During these applications, the sliding mass acceleration, a_{tg}, is compared with the yield acceleration, a_{yn}. If $a_{tg} \geq a_{yn}$, a_{tg} is set equal to a_{yn}. To simulate the gradual decrease in the shear strength of soil under dynamic loadings caused by pore-pressure building up when $a_{tg} > a_{yn}$:

$$a_{y_n} = \delta n a_{y_m} \tag{7}$$

where:

a_{y_m} = maximum yield acceleration obtained from the pseudostatic analysis of embankment (LT^{-1}), and
δ_n = shear strength reduction factor for the nth cycle (dimensionless).

Permanent displacement was taken to be the difference between displacement spectrum values of the nonlinear and linear systems. The basic steps required in the computations are:

a. Determine the yield acceleration from the pseudostatic stability analysis as proposed by Makdisi and Seed [6].

b. Determine the average maximum sliding mass acceleration k_m from figure 2 as recommended by Makdisi and Seed [6].

c. Develop the acceleration and displacement spectrum curves from the step-by-step integration of an earthquake record selected for the damsite. These curves for several examples are shown on figures 4, 5, and 6.

d. Draw a line horizontally at k_m until it intersects the acceleration spectrum, and then draw a vertical line until it intersects the permanent displacement spectrum. The point of intersection indicates the permanent displacement for the sliding mass (see figs. 4, 5, and 6).

Kutter's Decoupled Method

An improved version of Newmark's sliding block technique for prediction of permanent displacements has been developed from centrifuge testing.

Makdisi and Seed [6], Franklin and Hynes-Griffin [9], Ambrasey and Krinitsky [10], and others previously mentioned, have accounted for nonlinear and resonant effects by carrying out an elastic analysis before a rigid-plastic sliding block analysis. The embankment is first modeled as a visco-elastic shear beam or by finite elements. The visco-elastic model is subjected to design earthquake base motion, and the time history of strains and accelerations in the embankment are calculated. The visco-elastic properties used should be consistent with the level of strains predicted, so that proper selection of material properties for the analysis is an iterative procedure [3].

When the strain-compatible material properties have been determined, the acceleration history is calculated using the visco-elastic model.

Rigid-plastic sliding block analyses are then conducted to determine the permanent deformations of several trial wedges. The base acceleration input used in the rigid-plastic analyses is *not* the design bedrock motion; it is the elastically amplified motion calculated in the visco-elastic shear beam or finite element analysis [3].

The method of predicting permanent displacements is called decoupled analysis because the elastic analysis is separated from the plastic analysis. The decoupling is not rigorous because nonlinear problems *do not* obey the principle of superposition. Lin [11] has shown for a simple system of springs, dashpots, and sliders, that a decoupled analysis usually yields predictions within about a factor of 2 of the exact elastoplastic displacements.

The rigid-plastic sliding block analysis proposed by Newmark [4] has been extended to include elastic and nonlinear effects. The elastoplastic model parameters can be selected rationally so that the sliding

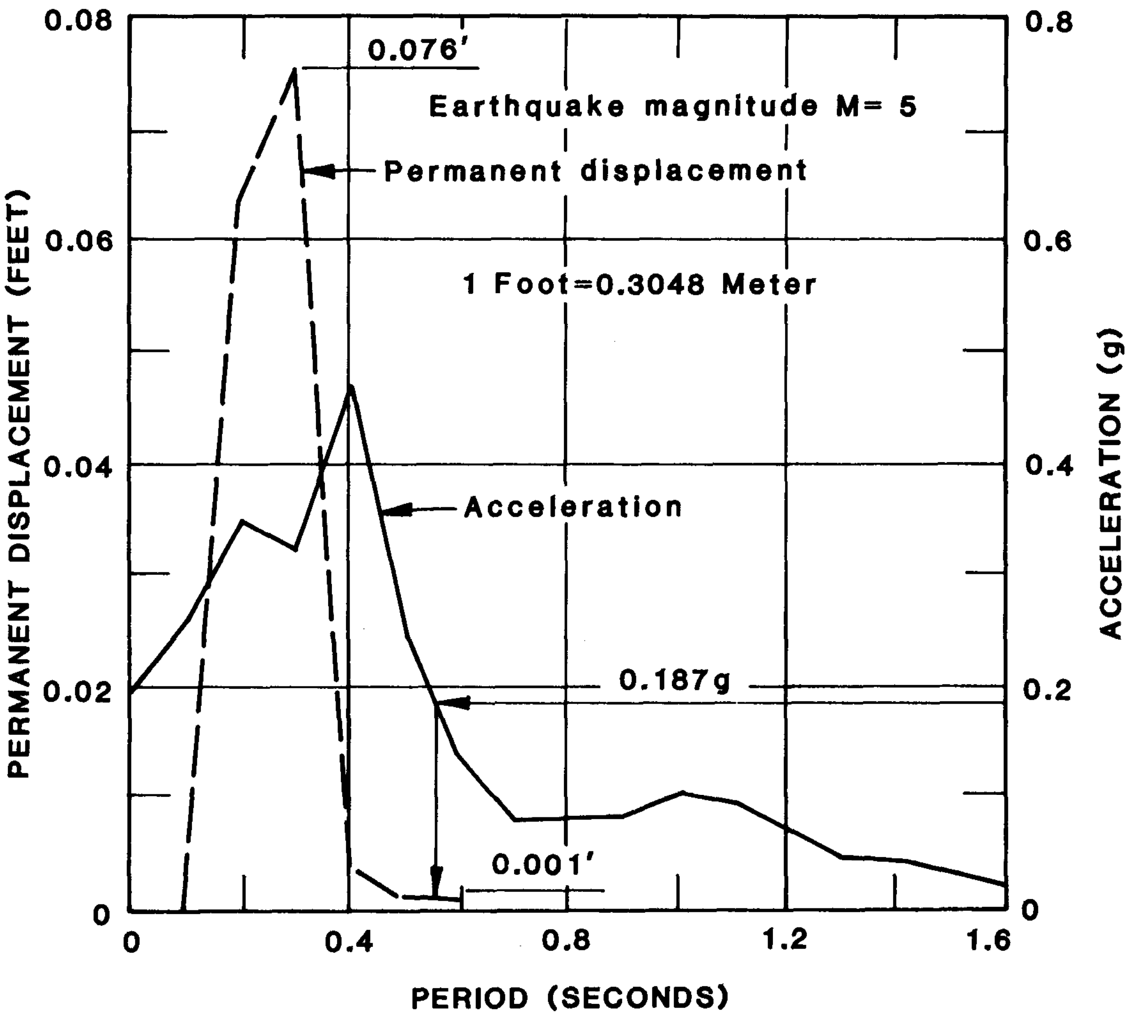


Figure 4. – Guri Main Dam permanent displacement and acceleration response spectrums. After Wilson and Clough, 1962 [7]; from Shieh and Huang, 1981 [8].

block has the same yield acceleration as the embankment and a natural frequency equal to that of the first mode of the embankment during low-amplitude oscillations. The degradation of shear modulus and increase of damping with strain amplitude is reasonably approximated by the nonlinear visco-elastoplastic model. The validity of decoupling the elastic and plastic analyses need not be assumed, and no iterative procedure is required because the required soil properties do not depend on the intensity of the earthquake.

The visco-elastoplastic response of a block on a slope can be calculated in an attempt to obtain improved predictions of permanent displacements. The sliding mass of an embankment resting on a wide shear band is simulated by a rigid block supported on an inclined plane by a nonlinear spring and dashpot.

During loading, a small nonlinearity has been incorporated in the spring characteristic at loads below the shear strength. Unloading is assumed to be elastic with stiffness, S_e.

The spring characteristic during loading is described by:

$$\frac{T}{T_{max}} = 1 - \left[1 - \frac{S_e(D - D_o)}{ET_{max}} \right]^E \quad (8)$$

$$\text{for } D - D_o < \frac{T_{max}E}{S_e}$$

$$\frac{T}{T_{max}} = 1, \text{ for } D - D_o > \frac{T_{max}E}{S_e} \quad (9)$$

where:

T = spring force (MLT^{-2}),
T_{max} = maximum (perfect-plastic) spring force (MLT^{-2}),
S_e = (elastic) tangent stiffness at T = 0 (MT^{-2}),

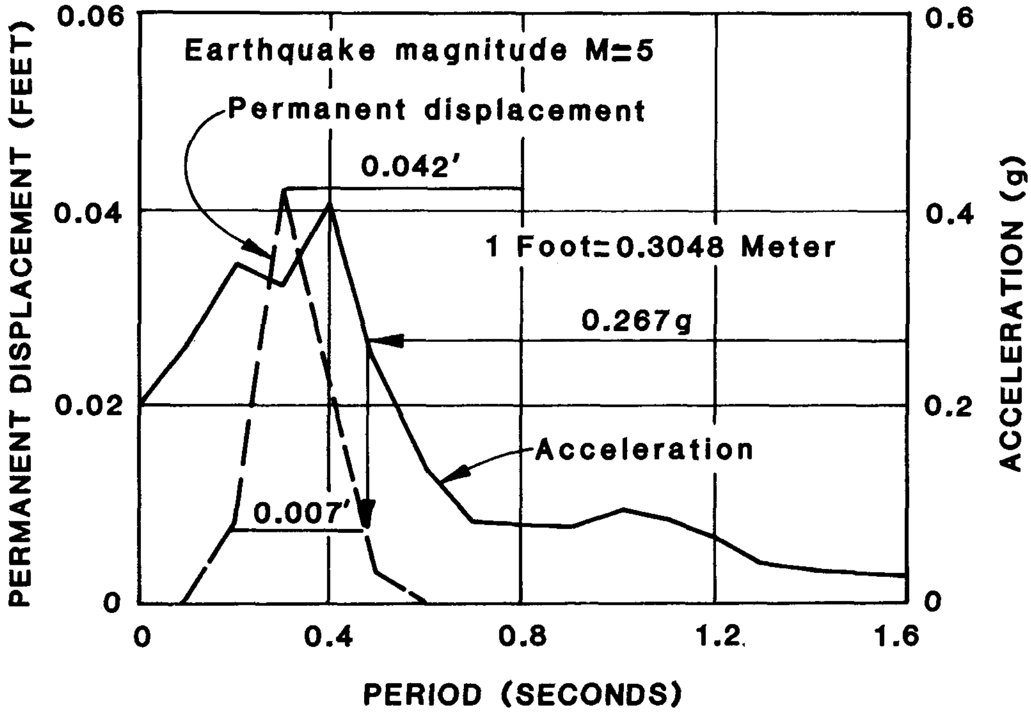


Figure 5. – Guri Dike permanent displacement and acceleration response spectrums. After Wilson and Clough, 1962 [7]; from Shieh and Huang, 1981 [8].

D = displacement (L),
D_o = displacement corresponding to the intersection of the load deformation curve and the T = 0 axis (L), and
E = exponent indicating the degree of nonlinearity (dimensionless).

For E = 1, equations (8) and (9) describe a linear-elastic, perfect-plastic characteristic, respectively. For E = 2, equation (8) defines a parabola. A normalized plot of these equations is shown on figure 7.

The analytical model has been used to predict the behavior of physical centrifuge models shaken during centrifugal flight using the bumpy road apparatus by Kutter [12].

The beauty of sliding block models is their simplicity. Nonlinear and resonant effects can be accounted for and can be mentally visualized and easily understood.

Experimental Research – Model Testing

Prompted by the extensive damage caused by the 1906 California earthquake, Professor F. J. Rogers (1906) [13] initiated an experimental investigation that he hoped would explain why greater earthquake destruction occurs in regions where the structural foundations are soft and not solid rock.

Rogers' shaking experiments with dry and wet sand were conducted using a 1000- by 860- by 300-mm (40- by 34- by 12-inch) box mounted on a car with steel rollers. Rogers concluded that, for dry sand, acceleration with depth was uniform and the soil moved as a rigid unit over a large range of amplitudes and accelerations. However, as the moisture content of the sand increased, an increasing acceleration gradient was observed from bottom to top.

The observation that test amplitude and acceleration increased or was magnified in wet sand enabled Rogers to investigate the greater intensity effects noted in specific regions during the 1906 earthquake.

Jacobsen (1930) [14] built upon the work by Rogers by measuring soil displacements at several heights, using Monterey Sand with moisture contents that varied from dry to saturated. Jacobsen observed that two types of vibrations occur in sand subject to simple harmonic motion:

a. Using partially saturated soil, the entire sand bed moved in phase with the base when the ground motion was not too violent (exhibiting simple elastic vibration).

b. At moisture contents approaching saturation, a chaotic motion of the sand occurred beginning at the top of the bed, and harmonic motion was not achieved. The motion observed was pseudo-elastic vibration in which the shearing rigidity of the sand was no longer in the elastic range.

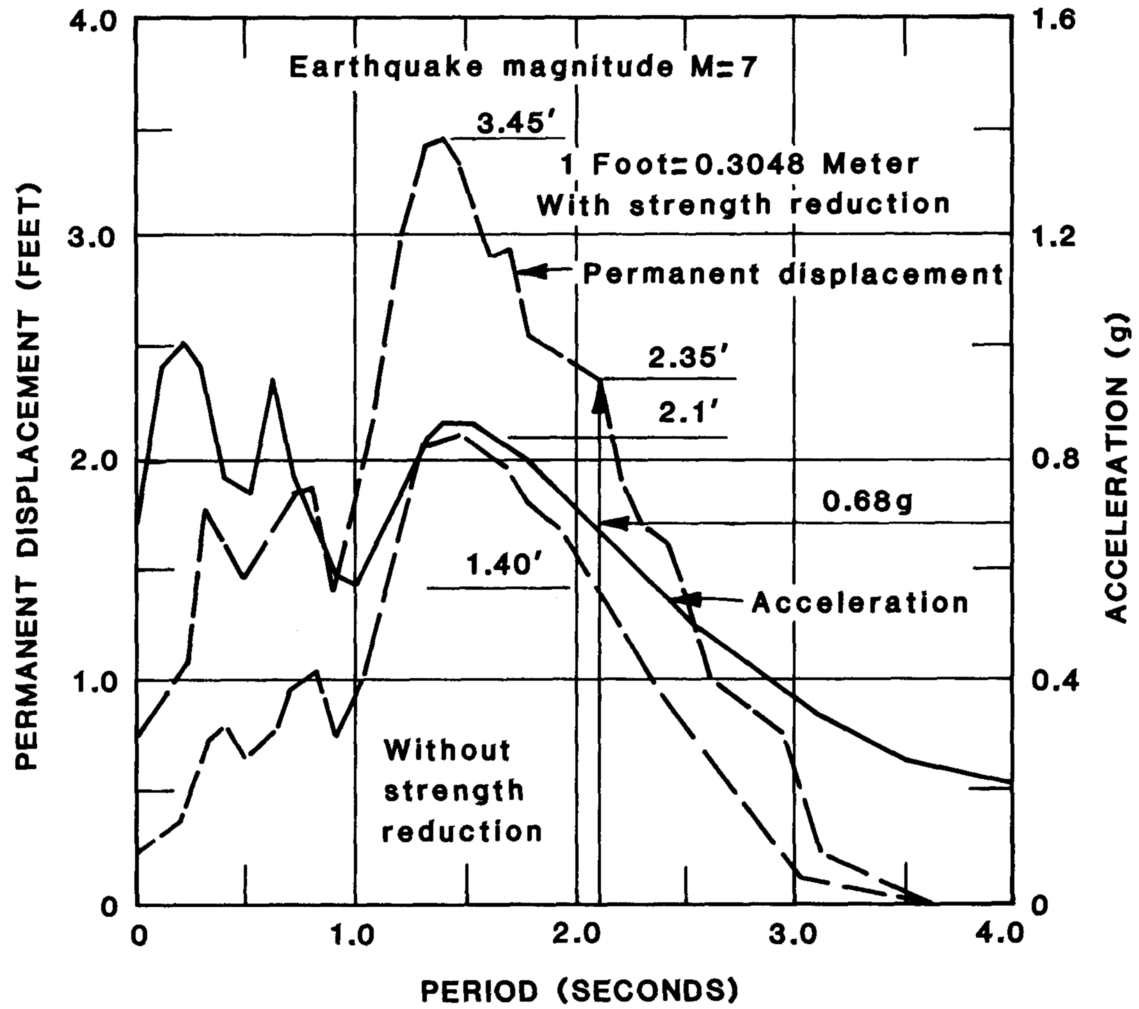


Figure 6. – La Honda Dam permanent displacement and acceleration response spectrums. After Wilson and Clough, 1962 [7]; from Shieh and Huang, 1981 [8].

In a summary of Jacobsen's work, Goodman (1963) [15] described this displacement as "yield displacement," and the corresponding value of acceleration as the "yield acceleration." He also observed displacement magnification factors (a positive acceleration gradient [13]) of 2½ percent for dry sand and 4 percent for sand at an 8-percent moisture content.

In 1936, Casagrande [16] tested model dam sections of "ordinary" saturated beach sand both in very loose and in very dense states. The model dams were subjected to a horizontal oscillation motion. Models made of sand that had been deposited in layers and tamped by hand into a dense state did not change shape even when shaken more violently than models tested in the loose state. However, shaking models built of loose sand resulted in liquefaction of the embankment.

Casagrande described the phenomenon of critical void ratio of a sand. A sand is at its critical void ratio when it can be subjected to any amount of deformation without volume change. Casagrande observed that a loose sand subjected to prolonged shearing would decrease in volume (densify), whereas a dense soil subjected to the same conditions would contain more voids after shearing than before (loosen). Both dense and loose sand would approach the same void ratio.

With regard to his model testing, Casagrande listed four principal factors that influence the reduction of shear resistance with shaking: (1) amount of deformation, (2) intensity of volume decrease with deformation, (3) permeability, and (4) dimensions of the region of shear.

Casagrande concluded, "since the resisting forces increase with only the first power of the height, unless the permeability (i.e., grain size) is reduced in model studies to a small function of what it is in the prototype, the models will be much more resistant to hydrodynamics than will the prototype" [15]. This decrease in grain size would introduce cohesion;

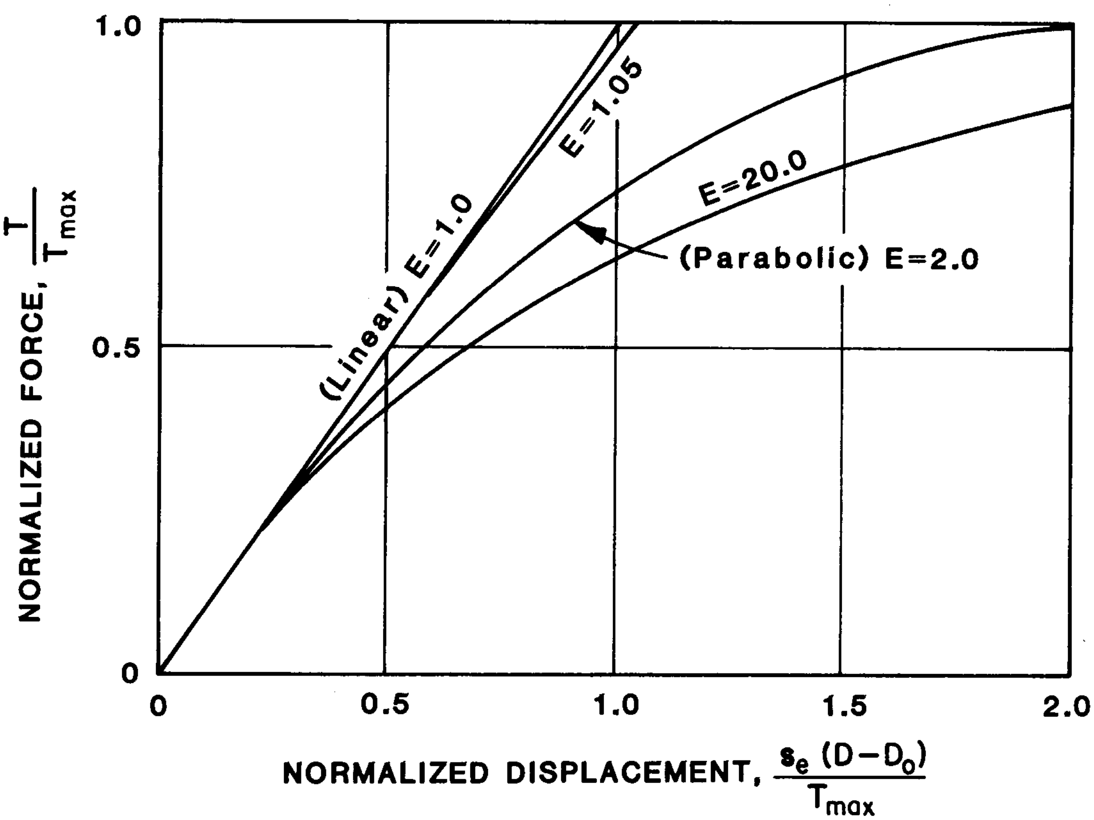


Figure 7. – Family of load-deformation curves. After Kutter, 1984 [3].

therefore, to alleviate influences of excess cohesion, Casagrande performed his model tests in a looser state (higher void ratio) than would normally be found in the prototype.

In 1936, Mononobe, Takata, and Matumura [17] analyzed the theoretical response of earthfill dams having triangular cross sections subjected to dynamic motion. They assumed that the density was uniform with depth, and that the modulus of rigidity and the coefficient of internal friction depended on depth.

The results of their analytical studies verified observations made by Rogers [13] and Jacobsen [14] that an acceleration gradient increases from the bottom to the top of the dam. Mononobe, Takata, and Matmura conducted experiments on model dams using gelatin (agar-agar) and sandy clay. Models were built on a shaking table and subjected to vibration.

Trapezoidal models (263 mm (10.4 inches) in height, 980 mm (38.6 inches) in base width, 13 mm (0.5 inch) in crest width) were tested. Deformation of the top was measured, and the modulus of rigidity was theoretically calculated. It was concluded that the sandy clay model did not give satisfactory results because of a gradient increase in the modulus of rigidity with depth.

In conclusion, their model tests showed that in "dams constructed of sandy clay, the modulus of rigidity (G) varied with dam position and acceleration. A linear increase of rigidity with increasing depth ($G = \rho Z$) lead to a magnification of displacement and acceleration from base to crest of the dam" (for tests conducted with sandy clay models), which was "larger than the magnification effect when G was constant in value" (for tests using the gelatin models) [15].

Heiland (1938) [18] conducted an investigation to assess the dynamic characteristics of Hanson Dam. He assumed that the proposed earthfill dam would behave like an elastic body and have a definite resonant frequency. Heiland incorporated the concept of similitude between the proposed prototype and scaled models. He acknowledged the reduction of scale for the materials and elastic moduli of the dam, and he stated that the geologic section (properties) must be selected so as to preserve a consistent model ratio.

Model tests were conducted using a length scale of 1:2,000 and a frequency scale of approximately 15:1. The results of model testing indicated the first mode of vibration coincided with field measurements.

In an attempt to verify model results, Heiland used an analytical model previously described by Mononobe, Takata, and Matamura (1936) in [17]. In this model the dam was assumed to be a two-dimensional body having a triangular (isosceles) cross section and uniform composition.

When Heiland analyzed a 100-foot (30.5-m) dam with elastic properties similar to those found in the test pits at Hanson Dam, a resonant frequency of 2 hertz was obtained. The results of the analysis conducted by Heiland were thought to be in good agreement with the results of model tests considering the limitations of both methods.

Heiland noted that natural frequency depends only on height and not on base width. He concluded that "only the fundamental mode (natural frequency) is of interest in the present problem." The first mode of resonance for a triangular dam [17] is:

$$f = \frac{V_t}{1.6h} \qquad (10)$$

where:

f = first resonant frequency (T^{-1}),
V_t = transverse wave velocity (LT^{-1}), and
h = height of the dam (L).

In 1940, Jacobsen [19] disagreed with Heiland's exclusive use of the first mode because, "although the higher modes give smaller absolute displacements, the mode shapes are more pronouncedly curved, giving greater shear stresses" (Bustamante, 1964 [20]).

In 1941, J. R. Ramirez [21] conducted a quantitative study of failure mechanisms of granular models subjected to earthquake loadings at Stanford University with the same apparatus used by Jacobsen. He conducted a series of three tests that included models of air-dried sand and models of wet sand at optimum moisture content with and without a reservoir.

The dry models were constructed with 3:1 upstream and 2½:1 downstream slopes and were subjected to horizontal accelerations ranging from 0.99 to 1.42 *g*. Models were constructed on a 1-inch (2.54-mm) layer of sand having the same composition as the embankment. The 6-inch (152.4-mm) height was obtained by rodding in two 3-inch (76.2-mm) lifts. Ramirez observed that test models constructed with dry sand had a tendency for the crest to slump and for the slopes to bulge at the embankment toes. The failure mechanism in the dry embankments was not determined. Test results suggest that "the principal danger from shaking an embankment of dry sand might be from the loss of freeboard and overtopping."

The failure surfaces of the wet models, as described by Ramirez, "show conspicuous curvature" and some approximate arcs of circles, whereas others show a variation in curvature indicating a spiral failure surface. The failure mechanism of the wet models was similar to that of a cohesive soil. This was a result of additional shear strength (apparent cohesion) in the moist sand caused by capillary (suction) pressure.

Failure on wet test models with reservoir loading was through plastic flow or a combination of flow and shear cracks. The surface along which this flow seemed to take place approximated the top flow line (phreatic surface).

Ek-Khoo Tan (1948) [22] described slide phenomena influenced only by gravity for cohesive and cohesionless soils. He investigated the angle of repose phenomenon for dry cohesionless slopes using a device for tilting and measuring the slope angle with a clinometer (a device used to measure inclination).

Failure was observed to begin at the top of the slope and was confined to the surface layers of the slope. Ek-Khoo Tan concluded that the angle of repose (of dry material) is independent of material density.

Ek-Khoo Tan stated that, in cohesive slopes, deep-seated slides involve the rupture and breaking away of a large mass of earth from the slope, and that maximum inclination and height of a stable slope is primarily dependent on the magnitude of the soil cohesion.

To correctly model the capillary forces that could be attributed to the addition of water in cohesionless material, a vacuum was applied to a rubber-membrane-covered slope creating an "equivalent cohesion" in the cohesionless material.

Tests were concluded on models placed in loose and dense states. A vacuum was applied, and the apparatus was tilted until failure occurred. The resultant slope angle was then measured. The angle of slope at failure was observed to increase linearly with equivalent cohesion.

Ek-Khoo Tan devised a procedure to investigate the locations and form of the rupture surface within the slope. He concluded that failure occurs in a soil slope after the strain exceeds a certain well-defined maximum value. This maximum strain appears to be independent of the slope angle and occurs at or near the middle of the sliding curve.

Ek-Khoo Tan concluded that the patterns of shearing stresses and of shearing strains obtained when testing the sand models were similar, and that plastic regions coincided with regions of maximum strain.

Model tests have been conducted to study the behavior of embankments and dams under earthquake loading. Clough and Pritz (1958) [23] reported model tests for a 90-meter (295-ft) high earth and rockfill dam. Models tested were 2 feet (0.6 m) high with a scale of 1:150. From similitude considerations for the same ratio of forces from dead weight, water load, inertial force, and forces associated with elastic deformation and failure, acceleration in the model and in the prototype were shown to be equal. Moreover, if the unit weight of the material in the model and in the prototype was the same, the ratios of cohesion and moduli of shear deformation in the model to those in the prototype was the same as the scale ratio. Based on tests of two models on a shake table, one with a central core and the other with a sloping core, the latter was found to be more earthquake resistant because its structure was more closely bound together. In general, the models suffered no significant changes in section up to a horizontal acceleration of 0.4 *g*. When the table motion was increased so that the acceleration exceeded 1 *g*, the model suffered only minor changes of shape. Seed and Clough (1963) [24] reported tests on sloping core dam models that were 0.65 meter (2.1 ft) high under empty and full reservoir conditions. In a typical model, the crest settlement was approximately 2.9 percent of the height of the dam for a peak earthquake acceleration of 0.68 *g*, and 1 percent of the height of the dam for a peak earthquake acceleration of 0.52 *g*.

Models of the 60-meter (197-ft) high Ram Ganga Saddle Dam were studied on a shake table 5 meters (16.4 ft) long by 2.8 meters (9.2 ft) wide (Krishna and Prakash, 1966 [25]). The problem was to determine the location of the core from seismic considerations.

Models with no core, a central core, and inclined cores were tested. Reservoir conditions tested were both dry and full. The inclined core dam was found to perform better than the central core dam. An important conclusion from this and from a previous study (Krishna and Prakash, 1965 [26]) was that the damage patterns in the model with a core and that of the Ohno Dam damaged in the Kanto earthquake of 1923 were identical. In both cases, typical longitudinal cracks developed along the crest. This observation was substantiated by field data (Seed, et al., 1975 [27]; Lee and Walters, 1974 [28]. This similarity showed conclusively that model tests would give an insight into the behavior of prototype dams, particularly in relation to inelastic deformations.

Another model study of a rockfill dam 61 meters (200 ft) high was performed at Roorkee (Prakash, et al., 1972 [29]) for Pandoh Dam in Punjab, India. A 1:100 scale was used with a model height of 0.61 meter (2.0 ft). The elastic response of the models to a modified Koyna earthquake was studied, and it was found that, even within the elastic range, test conditions in the model were adequately severe. Inelastic response was studied by comparing the damage potential of the table motion with the ground motion expected at the site. Deformation of the dam profile was recorded with a special profile meter. It was found that displacements occurred mainly at the crest. The conventional analysis showed a safety factor of less than unity for the top one-fourth of the slope. Displacements obtained showed the section to be safe. The presence of a berm at a typical level affected the damage pattern in the section in a characteristic manner.

Noda, Tsuchida, and Kurata (1974) [30] tested six models with maximum table accelerations of 200 to 300 gals (0.2 to 0.3 *g*).* The maximum acceleration at the top amounted to about 1,600 gals (1.6 *g*) in sand models and 2,700 gals (2.7 *g*) in clay models. Crest settlements of 4 to 15 percent of the model height were observed with no apparent sliding surface.

Okamoto (1975) [31] reported tests on 1.4-meter (4.6-ft) high models subjected to sinusoidal vibrations. A berm, 500 mm (19.7 inches) wide, was introduced at both 700- and 900-mm (27.5- and 35.4-inch) heights. For equal slopes, it was found that a greater acceleration was required to cause crumbling for a slope with a berm than for a slope without a berm.

On the basis of tests on six models, Watanabe (1977) [32] reported that zoning did not affect the response appreciably and that the amplitude of response acceleration for the full reservoir condition was reduced to about 66 percent of that for the empty reservoir.

Arya et al. (1977) [33] reported model tests on a 99-meter (324-ft) high rockfill dam tested with a scale of 1:130 on the shake table available at Roorkee. Two models, one with a central core and the other with inclined core sections, showed that slumping was more prevalent in an inclined core section although the tendency for separation of the shell from the core was greater with a central core. For the inclined core, both analytical and experimental values of displacements agreed fairly well if the variation of shear modulus proportional to the square root of overburden pressure was included in the analysis.

Arakawa, Kimata, and Kondo (1983) [34] tested 2-meter (6.6-ft) high models using four sandy soils with different embankment and foundation relative densities. Table accelerations of 200, 400, and 600 gals (0.2, 0.4, and 0.6 *g*) were used with a sinusoidal input frequency of 5 hertz. It was found that embankment failures on the loose sandy layers were

* Note: 100 gals ≃ 0.0980665 *g* acceleration.

caused by liquefaction. Embankment failure patterns and degrees of failure on the sandy layers were caused by liquefaction and observed to depend on the relative density of the layers (i.e., the acceleration needed for failure increased with increasing relative density).

V-shaped sloping embankment models 200 mm (7.9 inches) high were tested by Onmachi and Momenzadeh (1984) [35] to clarify dynamic failure characteristics of sloping embankments constructed in narrow valleys. Laws of similitude as described by Clough and Pritz (1958) [23] were incorporated into the material selection and testing as was the use of an impulsive input loading (Okamoto, 1973) [36]. It was concluded that (1) yield acceleration (which initiates the sliding failure) in a three-dimensional embankment is greatly affected by embankment geometry [i.e., yield acceleration significantly increases with a decrease in the width-to-depth ratio (*W/D*)], (2) the restraining effect of valley walls on the yield acceleration appears dependent on the embankment slope angle; and (3) observed settlements of the failure mass relative to the rest of the embankment does not show good agreement with the estimate formulated in terms of the yield acceleration, peak acceleration of the input loading, and the duration.

A model study of a zoned dam was performed (Kikusawa and Hasegawa, 1985 [37]) for Namioka Dam in Japan's Aomori prefecture. The 52-meter (170.6-ft) high embankment was modeled using a scale of 1:100. The authors investigated the effects of input motion, dam type, and reservoir conditions on the response characteristics. The following conclusions were made from their studies: (1) artificial motions produced probabilistically at the site should be used considering the structure spectral features, (2) the seismic dam response was generally greater on the upstream slope with full reservoir conditions than with the reservoir empty, when vibrated with a maximum input motion developing more than 100 gals (0.1 *g*), (3) models with sloped zones exhibited dynamic soil properties that were less linear than those for models with a center zone, and (4) pore-pressure models could be effectively used to simulate the liquefied conditions.

EMBANKMENT MODEL TESTING

Use of Newmark's sliding block method requires strong motion data, material properties, and slip plane and sliding block geometry as input data to determine permanent slope deformation. This method is one of many used [4, 6, 37, 38] in investigating the behavior of embankment dams subjected to seismic loading. Recently, the USBR (Bureau of Reclamation) [39] developed an analytical model that uses Newmark's sliding block procedure to estimate displacements of embankment dams during cyclic loading. The purpose of that study was to obtain strong motion and deformation data from six physical models constructed of cohesionless soil with different moisture contents and to compare acquired data with results from the analytical model.

It was assumed that ground motion in the entire embankment foundation could be represented by one vertical and one horizontal component. The dam was idealized as a two-dimensional body. The USBR study [39] summarizes the experimental and mathematical treatment of the problem.

Embankment Model

Embankment models selected for testing were constructed with a trapezoidal cross section. Models were 15 inches (381 mm) high and 30 inches (762 mm) wide with 2:1 slopes (horizontal:vertical). Crest and base width were 4 and 64 inches (102 and 163 mm), respectively. Transverse length to height ratio was consistently 2:1; therefore, the restraining effects of the abutments could significantly influence the resultant failure configuration. Nevertheless, the model as a whole was treated two dimensionally.

The container in which the embankment models were constructed had inside dimensions measuring 21 inches (533 mm) deep, 30 inches (762 mm) wide, and 90 inches (229 mm) long (fig. 8). To observe slope movement, two walls of the container were composed of 1-inch (25.4-mm) thick plexiglass. The model container was attached to a fixture angle apparatus whose purpose was to (1) redistribute the force of the hydraulic ram actuator on the embankment base, and (2) provide a level surface for the model container to ride upon the hydrostatic bearing tables.

Four models tested were founded directly on a contact cement-sand interface and centered in the testing container. To minimize end effects and effects of the contact cement sand base, the final two models were offset from the center of the container and founded on a 3-inch (76-mm) thick sand base identical in composition to the model embankment soil (figs. 9 and 10).

Soil used in the physical models was medium to coarse, hard, subangular to angular sand and was subjected to laboratory tests as described in appendix A. Effective shear strength parameters of $\phi' = 53°$ and $c' = 0$ lbf/in^2 (0 kPa) best define the relationship of shear stress to effective normal stress at the low normal stress levels in the embankment models.

Models were constructed in eight 1.88-inch (48-mm) lifts. Each lift was vibrator-compacted to a uniform

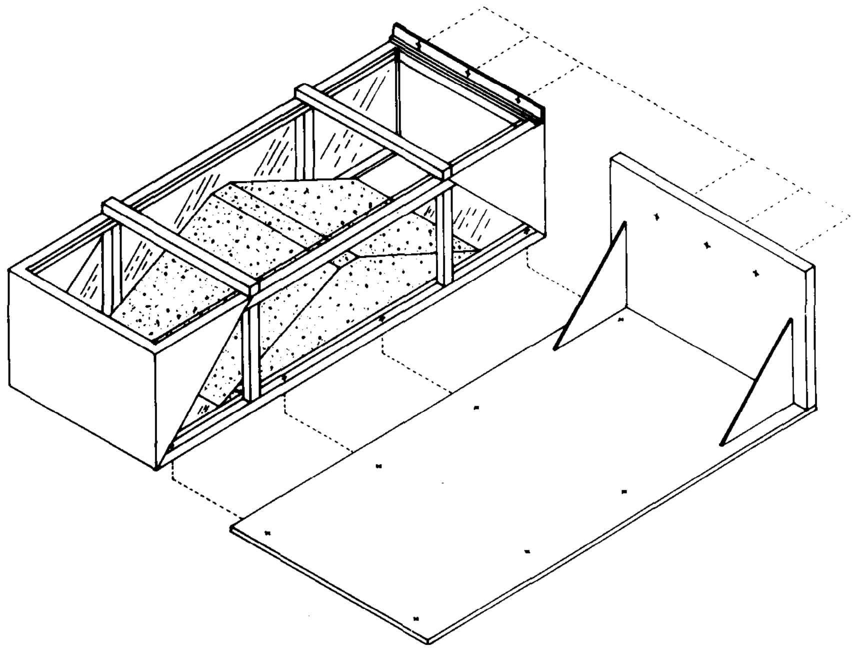


Figure 8. – Test apparatus.

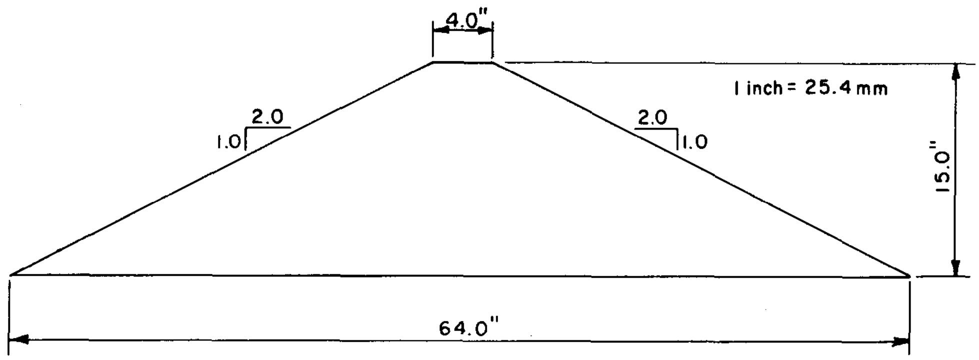


Figure 9. – Cross section for models No. 1 through 4.

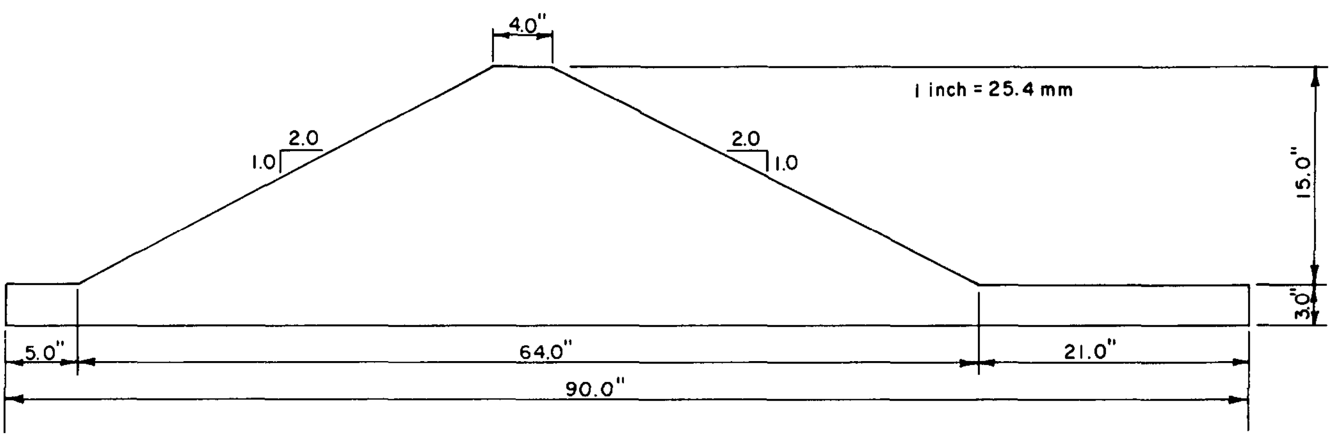


Figure 10. – Cross section for models No. 5 and 6.

density. A thin layer of black sand (same composition as the embankment) was used as an interface between each lift to aid in determining the location of failure planes.

Removable wooden blockouts were used in each lift to minimize the amount of sand processed for each test. Rough slopes were excavated back to the final configuration using a metal trowel and a long straightedge. A small amount of surficial densification and drying occurred during the trimming process (not quantified).

Data Acquisition

Embankment profile measurements were made to determine the model configuration before testing and slope movement and the deformation after testing using a point gauge device. White map pins were placed on the embankment surfaces in a 3- by 3-inch (76- by 76-mm) grid to allow visual monitoring of slope movement. A string grid was mounted on top of the model container to provide a reference by which to observe pin movement.

Slope movement during shaking was monitored and recorded by videotaping from two angles. A camera centered above the model recorded the response during testing, and another camera at floor level recorded the model response through the plexiglass representing one of the model abutments. Videotapes were used in the review, description, and analysis of the model embankment testing. Surficial displacement and particle velocities were easily observed and calculated.

Embankment response during testing was monitored by Entran EGA-125 uniaxial piezoresistive miniature accelerometers embedded in the embankment during construction (fig. 11). Specifications are given in appendix D. Accelerometers were placed at designated elevations and locations in the embankment so that the variation of acceleration with elevation could be investigated (app. B, table B1 and figs. B1 through B6). Initially, it was proposed that acceleration with elevation be investigated (models 1 and 2). Previous investigations have shown that acceleration increases with elevation. Therefore, by placing the accelerometers near or within the slide mass, the time to failure and acceleration increase of the mass could be observed.

Piezoelectric accelerometers were attached to the angle fixture plate to monitor horizontal and vertical baseplate accelerations. Their primary function was to serve as a reference for accelerometers located at various elevations in the embankment. A piezoelectric accelerometer and an LVDT (linear variable differential transformer) were attached to the actuator ram to provide ram displacement data. Monitoring of pore pressure within the partially saturated sand embankments was considered but omitted because it was outside the scope of the investigation.

Signals from accelerometer amplifiers, from the LVDT signal conditioner, and from hydraulic system transducers were sent to an HP (Hewlett-Packard) 2250 measurement and control processor for high-speed scanning and digitization. The HP-2250 provided input filtering to reduce high-frequency noise

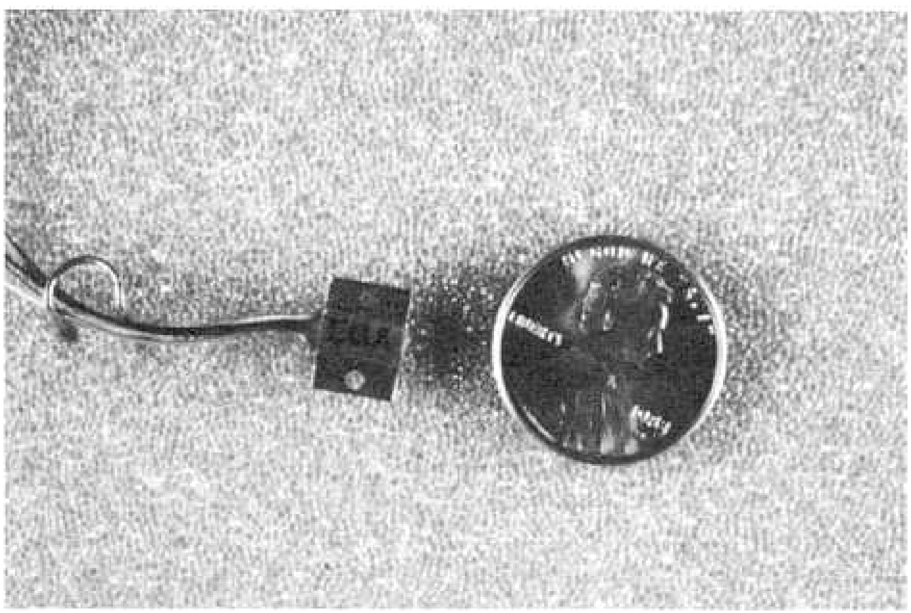


Figure 11. – Miniature accelerometer. P801-D-81081

caused by the bearing tables and actuator ram. Digitized data were transferred to an HP-1000 minicomputer for hard disk storage (fig. 12). For a more detailed description of equipment and procedures used in the model testing, see appendix C.

Raw data were scaled, formatted, processed, stored on tape and hard disk, and transferred to the USBR's Cyber 170/835 main frame computer. The deformation analysis portion of the investigation used the data stored on the Cyber 170/835 main frame.

Loading Parameters

The model testing program was performed using a 25-ton (22 680-kg) hydraulic servocontrolled ram actuator, located in the USBR vibration laboratory (see fig. 13 and app. D for the ram hydraulic performance characteristics). The hydraulic ram actuator was mounted on an isolated 250-ton (226 800-kg) seismic mass. The seismic mass is supported by 25 model AL 255-12 isolators around the bottom edge of its perimeter and in the center of the buttress [40] (fig. 14).

The following test parameters were proposed for embankment model testing:

Parameter	Models No. 1 and 2	Models No. 3 through 6
Frequency	5 Hz	5 Hz
Amplitude	0.20 inch(5 mm)	0.24 inch(6 mm)
Acceleration	0.51 *g*	0.61 *g*

Actual output acceleration varied from test to test, because of amplitude fluctuations. The acceleration was calculated using the following equation [41]:

$$a = 4A\pi^2 f^2 \quad (11)$$

in which:

a = acceleration (LT^{-2}),
A = amplitude (L), and
f = frequency (T^{-1}).

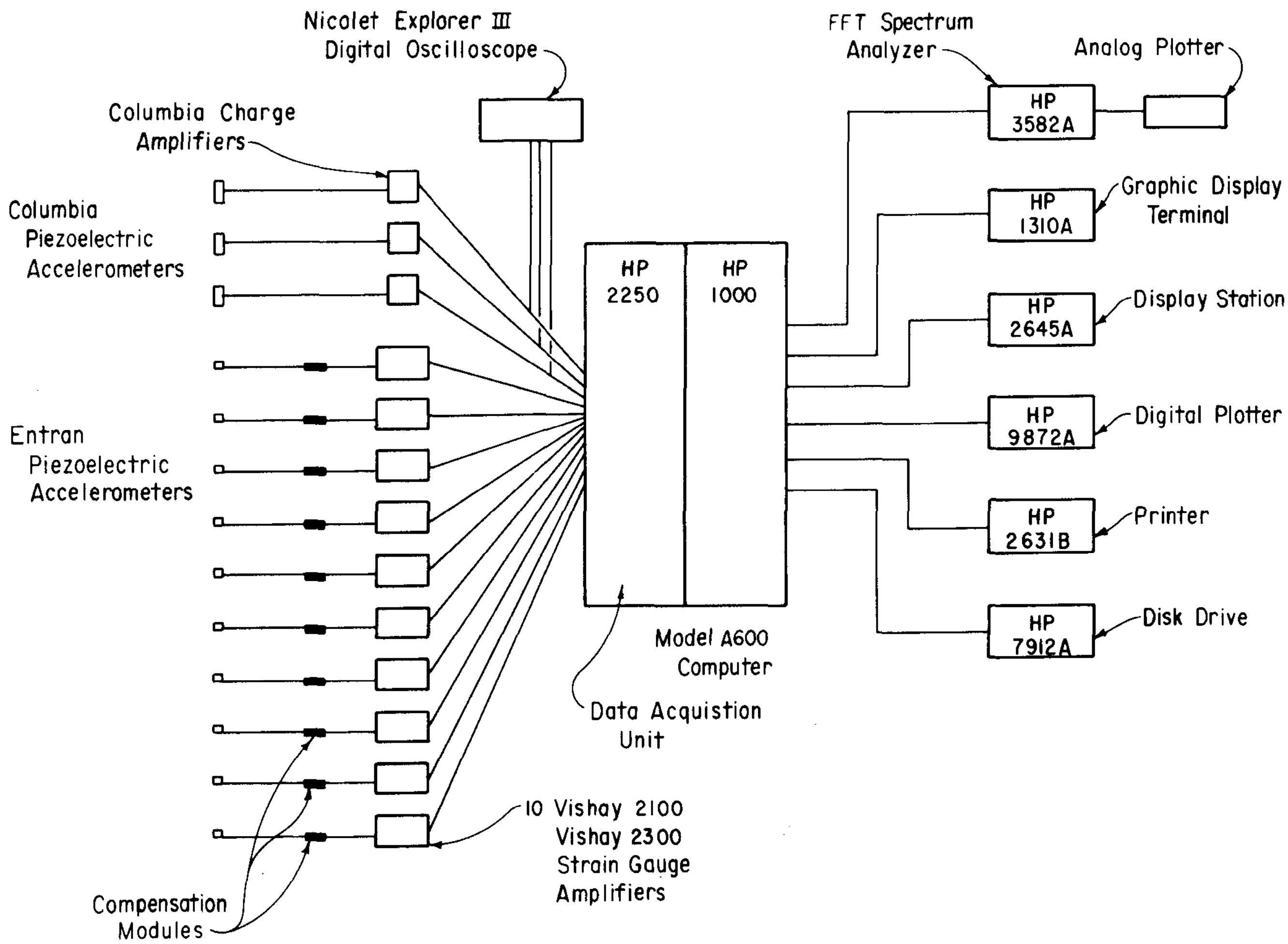


Figure 12. – Instrumentation system block diagram.

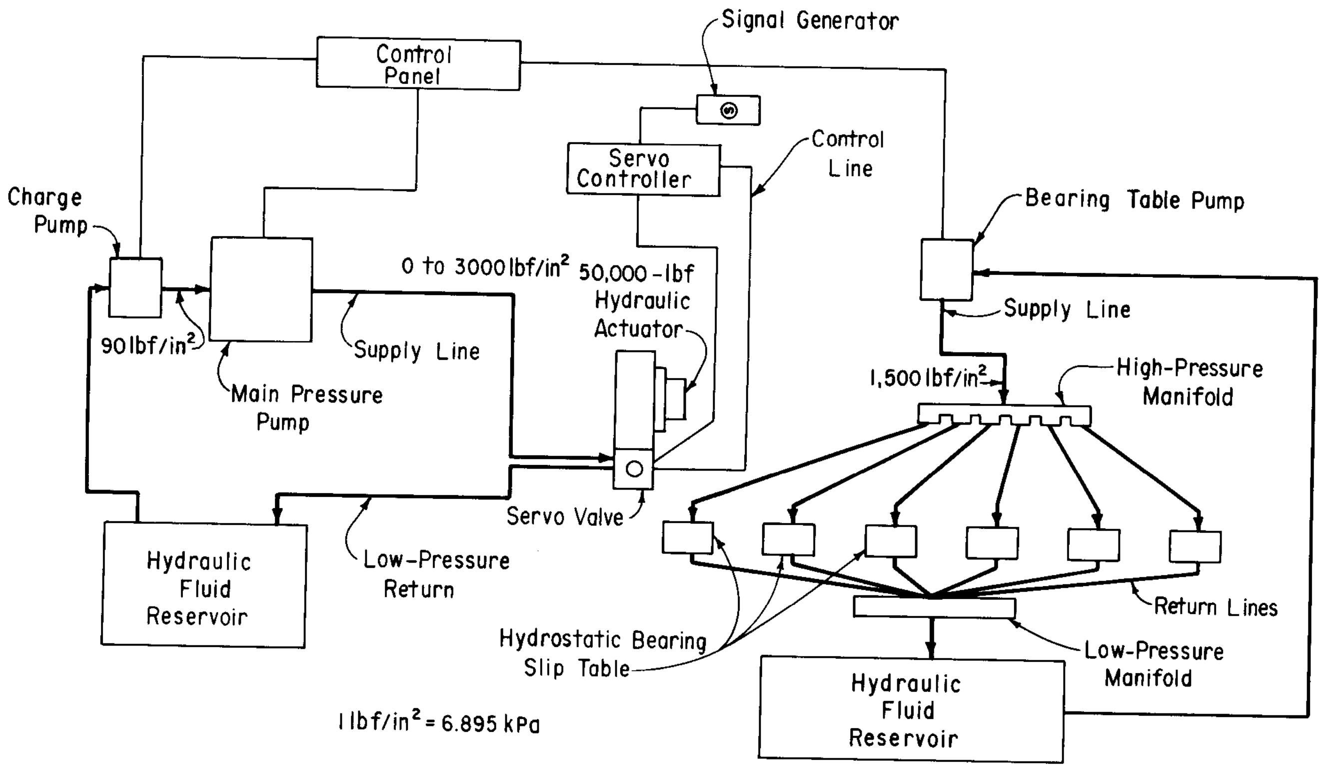


Figure 13. – Hydraulic system block diagram.

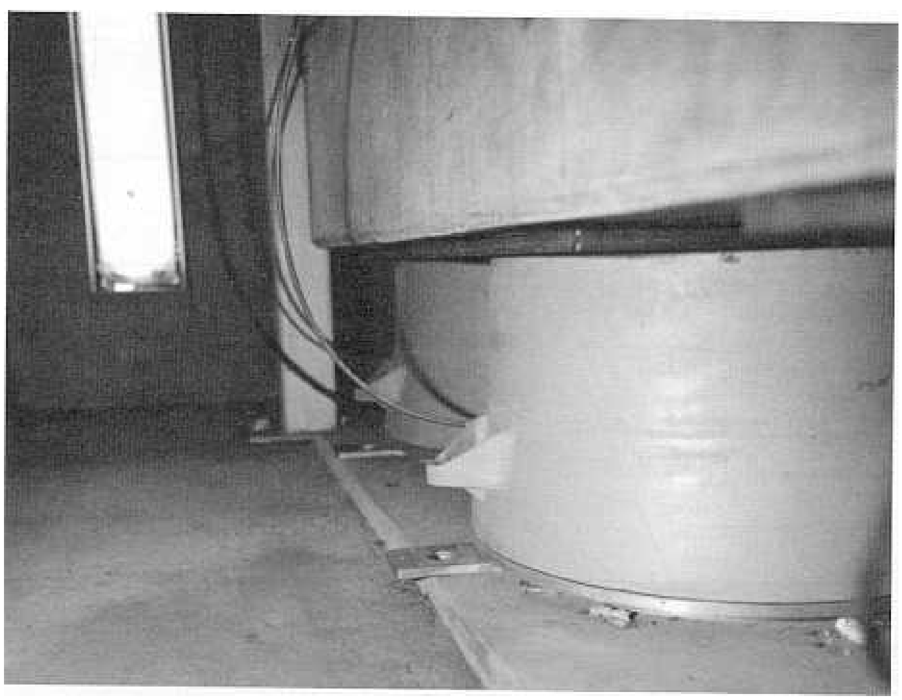


Figure 14. – Isolation system. P801-D-81082

The testing sequence consisted of subjecting the models to a 5-hertz sinusoidal wave form. A total elapsed testing time of 60 seconds was used. Tests were displacement controlled and consisted of a linear increase in amplitude from zero to a designated maximum amplitude over a 5-second time interval. The maximum amplitude was held constant for 50 seconds, then linearly decreased over the remaining 5 seconds of the test (fig. 15). Table 1 is a summary of the pretest embankment parameters.

Test Results

Table 2 summarizes test observations and posttest analyses. Air-dried sand was used in the construction of models No. 1 and 2. Crest particle movement of model No. 1 commenced at 2 to 3 seconds, and particle velocities of 2 in/s (51 mm/s) were calculated. The velocity distribution was typically parabolic across the face of the slope and occurred at a maximum acceleration of 0.45 *g* (see fig. 16). The

Table 1. – Pretest embankment parameters.

Model test No.	Moist unit weight at placement, lbf/ft^3	Moist unit weight at placement, kg/m^3	Moisture content at placement, %	Posttest average moisture content, %	Dry[1] unit weight, lbf/ft^3	Dry[1] unit weight, kg/m^3	Relative density, %
1	103	1650	air[2]	<0.1	103	1650	91.0
2	101	1620	air[2]	<0.1	101	1620	82.1
3	101	1620	7	4.4	96.7	1550	57.6
4	103	1650	7	3.7	99.3	1590	74.0
5	101	1620	9	6.1	95.2	1530	53.4
6	102	1630	4.5	3.9	98.2	1570	68.7

[1]Dry unit weight using posttest average moisture content.
[2]Test used dry sand having moisture content less than 1 percent. All other tests had been placed at predetermined moisture contents.

Table 2. – Summary of test observations and posttest analyses.

	Model test No. 1	2	3	4	5	6
Average moisture content, %	<0.1	<0.1	4.4	3.7	6.1	3.9
Degree of saturation, %	.43	.41	16.3	14.6	21.7	14.9
Time to initial failure, s	3	4	7.8	13.3	14	5.3
Maximum crest or particle velocity, in/s (mm/s)	2 (51)	4 to 6 (102 to 152)	1 (25)	1.8 (46)	2.3 (58)	2.8 (71)
Failure mode or mechanism	Avalanche-type with parabolic velocity distribution		Complex downslope translation circular-arc rotation			
Base acceleration, *g*	.35	.53	.76	.63	.58	.58
Estimated crest acceleration, *g*	.49	.52	.74	.72	.74	.85
Crest/base acceleration ratio	1.40	.98	.97	1.14	1.28	1.47
Crest displacement, inches (mm)						
Horizontal	N/A	N/A	12 (305)	13 (330)	19 (483)	19 (483)
Vertical	1.5 (38)	2 (51)	7 (178)	9 (229)	8.5 (216)	9 (229)
Downslope	N/A	N/A	14 (356)	16 (406)	21 (533)	21 (533)
Estimated volume of failure, in^3 (mm^3)	1,060 17.4×10^7	960 1.57×10^7	3,750 6.15×10^7	3,190 5.23×10^7	3,390 5.56×10^7	3,900 6.39×10^7
Percent of total embankment volume, %	7	6	25	21	22	26

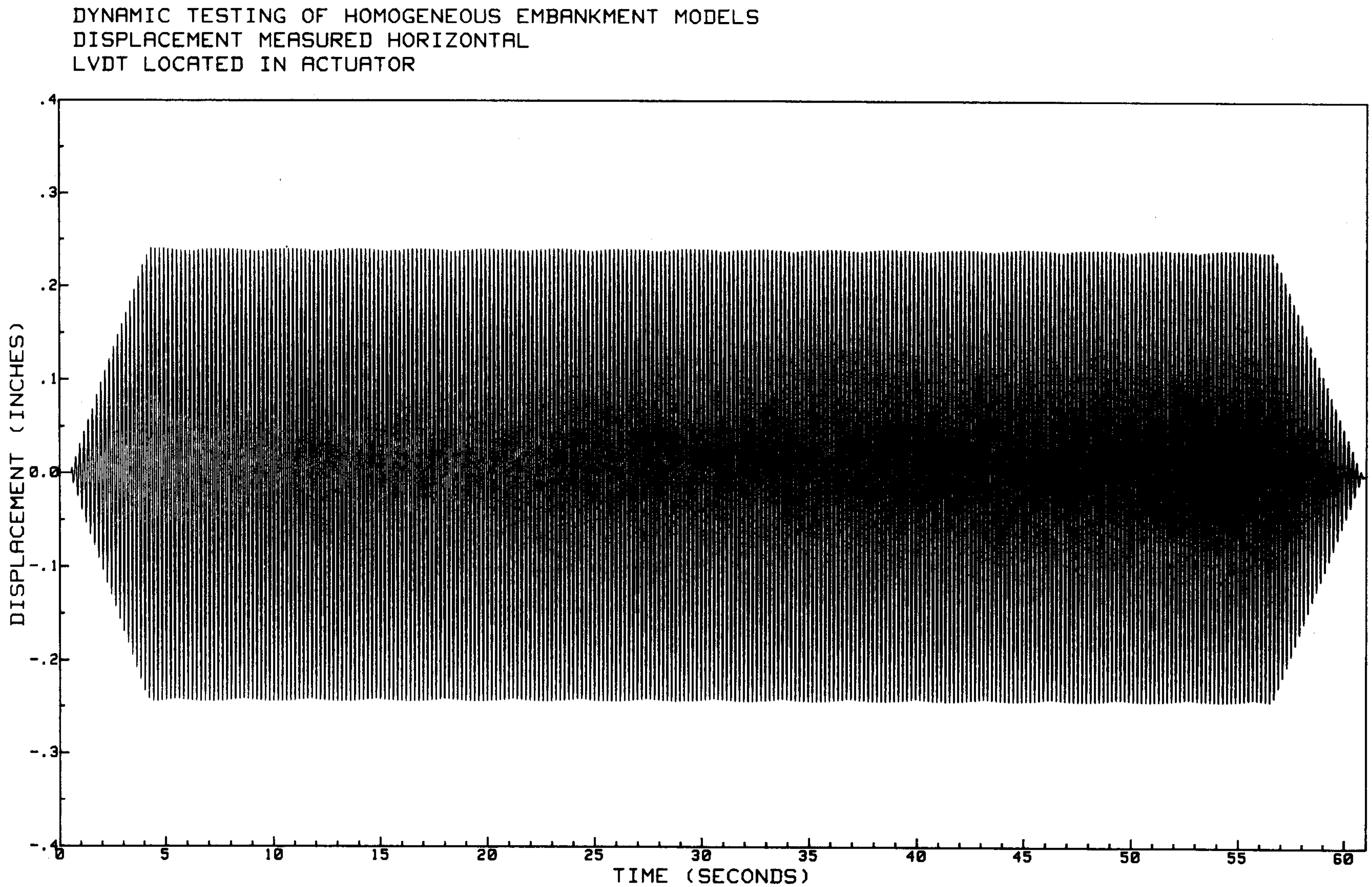


Figure 15. – Displacement versus time plot.

mode of failure was through particle rolling and was classified as an avalanche-type failure [42].

Model No. 2 experienced avalanche-type failure similar to that observed in model test No. 1. Movement began abruptly at 4 seconds, and velocities of as much as two to three times greater than those of model No. 1 (4 to 6 in/s (102 to 152 mm/s)) were calculated. The velocity distribution was also nonlinear across the slope occurring at approximately the maximum acceleration of 0.52 *g*.

Slope failure of models No. 3 through 6 was classified as complex [42], which combined rotation with downslope translation.

Model No. 3 had the deepest circular-arc failure plane (figs. 17 and 18). The concave, spoon-shaped failure plane was deepest at the centerline and became progressively shallower as the plane neared the plexiglass abutments.

The failure plane of model No. 4 appeared to be approximately parallel to the slope typical of an infinite-slope wedge-type failure (or Newmark's sliding block failure), and posttest investigations appeared to support this observation.

The failure mechanism of model No. 5 was a complex, circular-arc type. But it had definite, infinite, or translational slide [43] characteristics. Before movement, the crest began to resonate and a large diagonal crack separated the failure block into two segments. One of the blocks appeared to rotate slightly while moving downward; this caused the other half to move downslope. The toe of the circular-arc failure plane broke out on the slope approximately 3 inches (76 mm) above but not through the model slope toe.

The cause of this two-segment failure is speculative. A moisture gradient may have developed across the slope during placement, vibratory compaction, or testing. The possibility of differential inplace densities across the model embankment also cannot be discounted.

Failure of model No. 6 was a circular arc-type in conjunction with two shallow slides and a shallow block-type failure. This combination of failures contributed to the overall deformation. Both slopes experienced deformation.

Moving in a circular arc, the failure mass moved downslope and appeared to exit on the slope approximately 6 inches (152 mm) above the embankment toe. As the soil was pushed up and over the point of exit, large transverse tension cracks (up to 0.5 inch (13 mm)) developed. The soil flowed over the toe of the model embankment. As the primary failure was occurring, two secondary slides occurred. Approximately 2 seconds after the primary deep-seated failure occurred on one slope, a shallow block-type failure occurred on the opposite slope. The velocity distribution was parabolic and somewhat similar to models No. 1 and 2, except that the failure characteristics were typical of an infinite slope movement.

Dry homogeneous embankments experienced a rapid avalanche-type failure through particle rolling, which caused progressive lowering of the crest. The net decrease in model height that could be attributed to avalanche-type failure was 1.5 to 2 inches (38 to 51 mm) (10 to 13.3 percent) for models No. 1 and 2. The change in the mode of slope failure for models No. 3 through 6 can be attributed directly to the addition of moisture to the soil. Moisture contents ranged from 4 to 6 percent from test to test. Addition of water to the soil contributed an apparent cohesion caused by capillary pressure that helped bond the individual sand particles together. This small apparent cohesion (calculated as less than 1 lbf/in^2 (6.895 kPa) based on particle size) [44] added enough shear strength to the embankment model to overcome acceleration forces on individual particles, which in earlier models caused only surficial particle motion in the form of avalanche-type failure.

For models No. 3 through 6, slope failures were classified as complex. The failure surface configuration varied in depth as the moisture content of the model varied. Downslope crest movement ranged from 14 to 21 inches (356 to 533 mm) along the slope.

Surface measurements (using a point gauge device) before and after testing are summarized in appendix E for each model. Surface measurements are used to plot the model cross sections and as input data for three-dimensional plots. Three-dimensional plots of selected models (before testing and after failure) and cross sections of each model are also summarized in appendix E. All data acquisition (i.e., vertical and horizontal accelerometer time-histories, LVDT displacements, etc.) summarized as plots are located in appendix F.

Regression analyses were performed on various test data, but it was determined that the data were insufficient to develop meaningful correlations.

MATHEMATICAL ANALYSIS

Method of Stability Analysis

An apparent cohesion intercept of the material was computed to take into account the effect of the capillary action of water using the following equation [44]:

$$c' = h_{cr}\gamma_w \tan\phi' \tag{12}$$

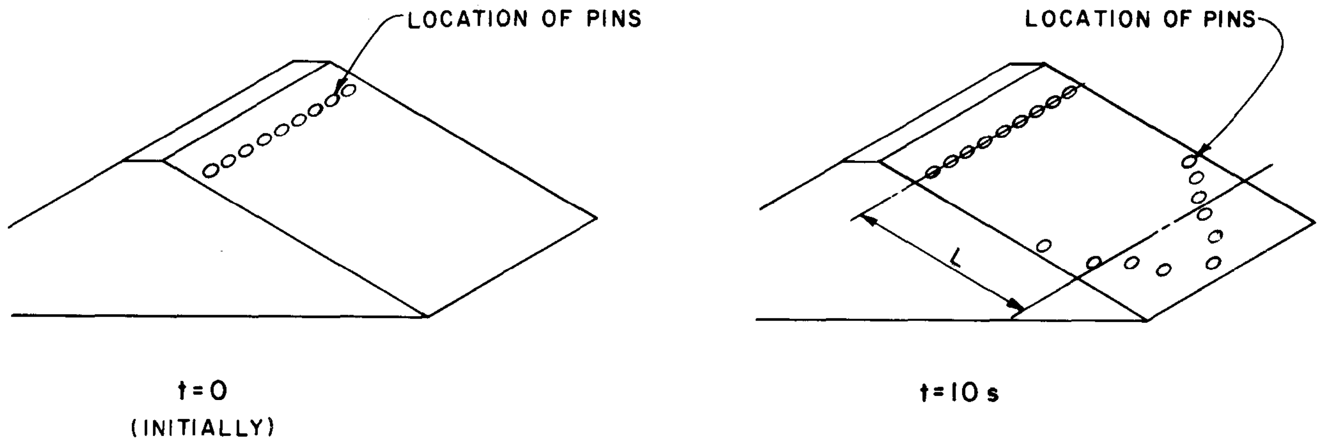


Figure 16. – Parabolic velocity distribution.

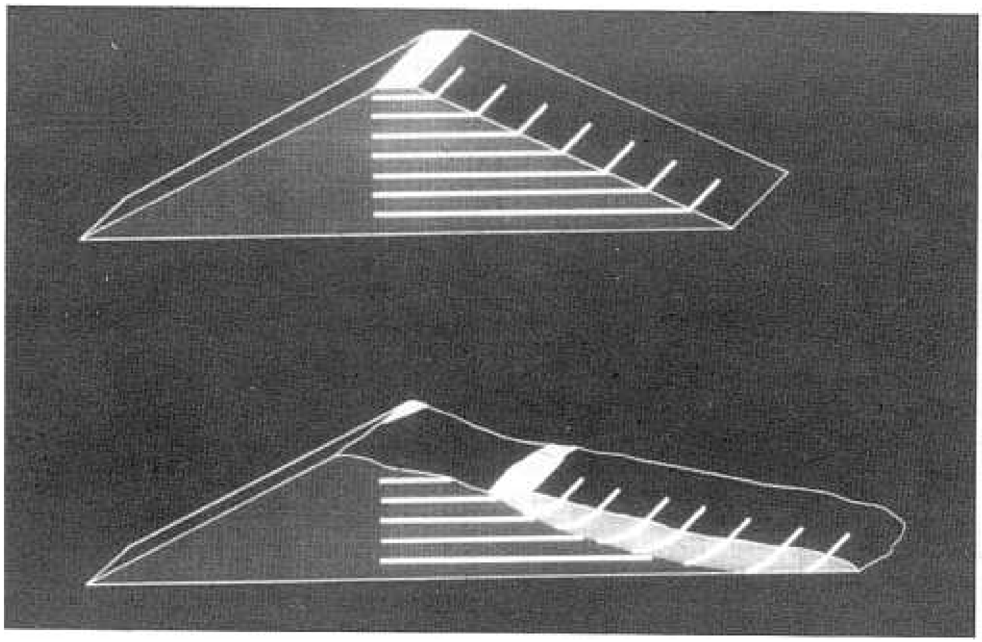


Figure 17. – Model No. 3; cross sections of embankment before and after failure. P801-D-81083

where:

c' = apparent cohesion (FL^{-2}),
h_{cr} = height of capillary rise (L),
γ_w = unit weight of water (FL^{-3}), and
ϕ' = effective angle of internal friction (degrees).

Apparent cohesion was calculated by assuming a capillary rise of approximately 5.5 inches (140 mm) and a friction angle of $\phi' = 45°$. This yielded an apparent cohesion of about 2.0 lbf/in² (1.39 kPa).

Spencer's method of stability analysis was used to search for the critical circular shear plane along which the factor of safety is minimum for static loading conditions. The driving moment of the failure mass was assumed to be caused only by the weight of the material. The critical static circular shear plane was then used in the deformation analysis. Observed failure surfaces were also used as input data to the computer model in an attempt to quantify the apparent cohesion contributed by the partially saturated sand.

Method of Deformation Analysis

The computer program DYNDSP [39] was used to compute deformations along both the critical circular shear plane for dynamic conditions and for a representative infinite-slope shear plane. Vertical and horizontal components of acceleration measured during testing were used in the analysis. The apparent cohesion intercept (c') of the embankment material was reduced to zero after an infinitesimal movement of the failure mass had occurred along the critical plane.

Results of Deformation Analysis

Using horizontal and vertical acceleration history records of the strong motion instruments at the base of the model embankments, deformations (displacements) of the failure mass (rigid block) along the critical circular plane were computed. These displacements are summarized in table 3 and on figure 19.

It can be observed from the results of the deformation analysis that mathematical treatment of the problem is sensitive to variations in the effective angle of internal friction. As ϕ' was varied between 40° and 50° (a reasonable range of ϕ' for the sand tested), predicted block displacements varied from 3 to 61 inches (76 to 1549 mm) along the plane as shown in table 3. Realistic block displacements should be limited to the actual slope length of 33.5 inches (851 mm). For effective friction angles greater than 45°, the displacement results are comparable with actual model performance.

Comparison of Computed Displacements with Test Results

Displacement of failure masses of two models (models No. 4 and 6) due to sinusoidal motion was computed along defined shear planes using the computer program DYNDSP. The following table summarizes computed and measured displacements of two models:

Table 4. – Comparison of computed and measured displacements.

	Displacements	
	Measured	Estimated at $\phi' = 53°$
Model No. 4, inch		0
Model No. 6, inch		0

Because of the nonlinearity of the Mohr failure envelope, strength parameters of $\phi' = 53°$ and $c' = 0$ lbf/in² (0 kPa) best describe the behavior of this sand at effective normal stresses less than 7.7 lbf/in² (53.1 kPa) (app. A, fig.

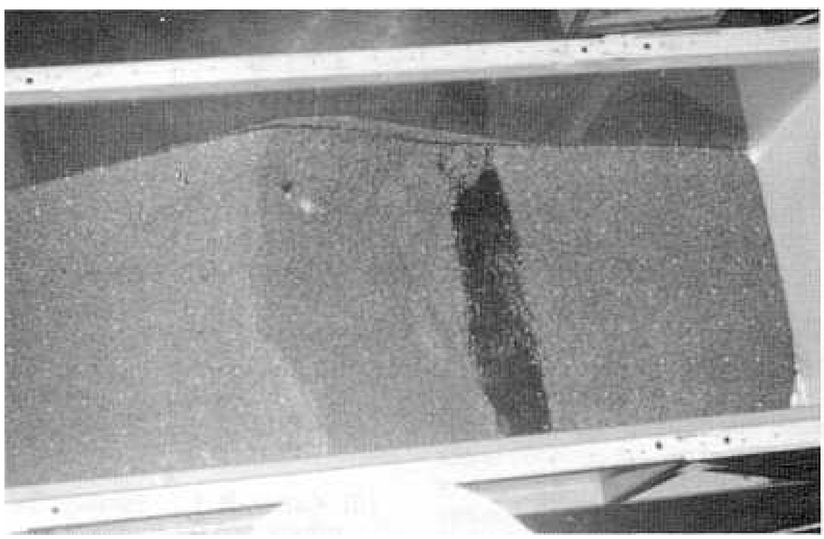


Figure 18. – Model No. 3; postfailure top view of embankment. P801-D-81084

Table 3. – Results of deformation analysis.

	c'		ϕ',	γ_d[1]		Center of Circle X		Center of Circle Y				Displacement along sliding plane		Horizontal displacement		Vertical displacement	
	lbf/ft²	kPa	degrees	lbm/ft³	kg/m³	ft	m	ft	m	ft	m	in	mm	in	mm	in	mm
Test 4																	
Soil type: medium sand	28.8	1.38	42	102	1630	6.7	2.0	4.0	1.2	3.8	1.2	31.4	798	27.7	704	14.7	373
Moisture content: 3.7%	28.8	1.38	45	102	1630	6.7	2.0	4.0	1.2	3.8	1.2	16.4	417	14.5	368	7.7	196
c' = 28.8 lbf/ft² (1.38 kPa)	28.8	1.38	47	102	1630	6.7	2.0	4.0	1.2	3.8	1.2	7.2	183	6.4	163	3.4	86
$\phi' = 45°$																	
Test 6																	
Soil type: medium sand	28.8	1.38	40	101	1620	6.7	2.0	4.0	1.2	3.8	1.2	61.4	1560	54.2	1377	28.8	732
Moisture content = 3.9%	28.8	1.38	43	101	1620	6.7	2.0	4.0	1.2	3.8	1.2	41.2	1046	36.4	925	19.3	490
c' = 28.8 lbf/ft² (1.38 kPa)	28.8	1.38	47	101	1620	6.7	2.0	4.0	1.2	3.8	1.2	14.8	376	13.1	333	6.9	175
$\phi' = 45°$	28.8	1.38	50	101	1620	6.7	2.0	4.0	1.2	3.8	1.2	2.6	66	2.3	58	1.2	30

γ_d = dry unit weight of sand.

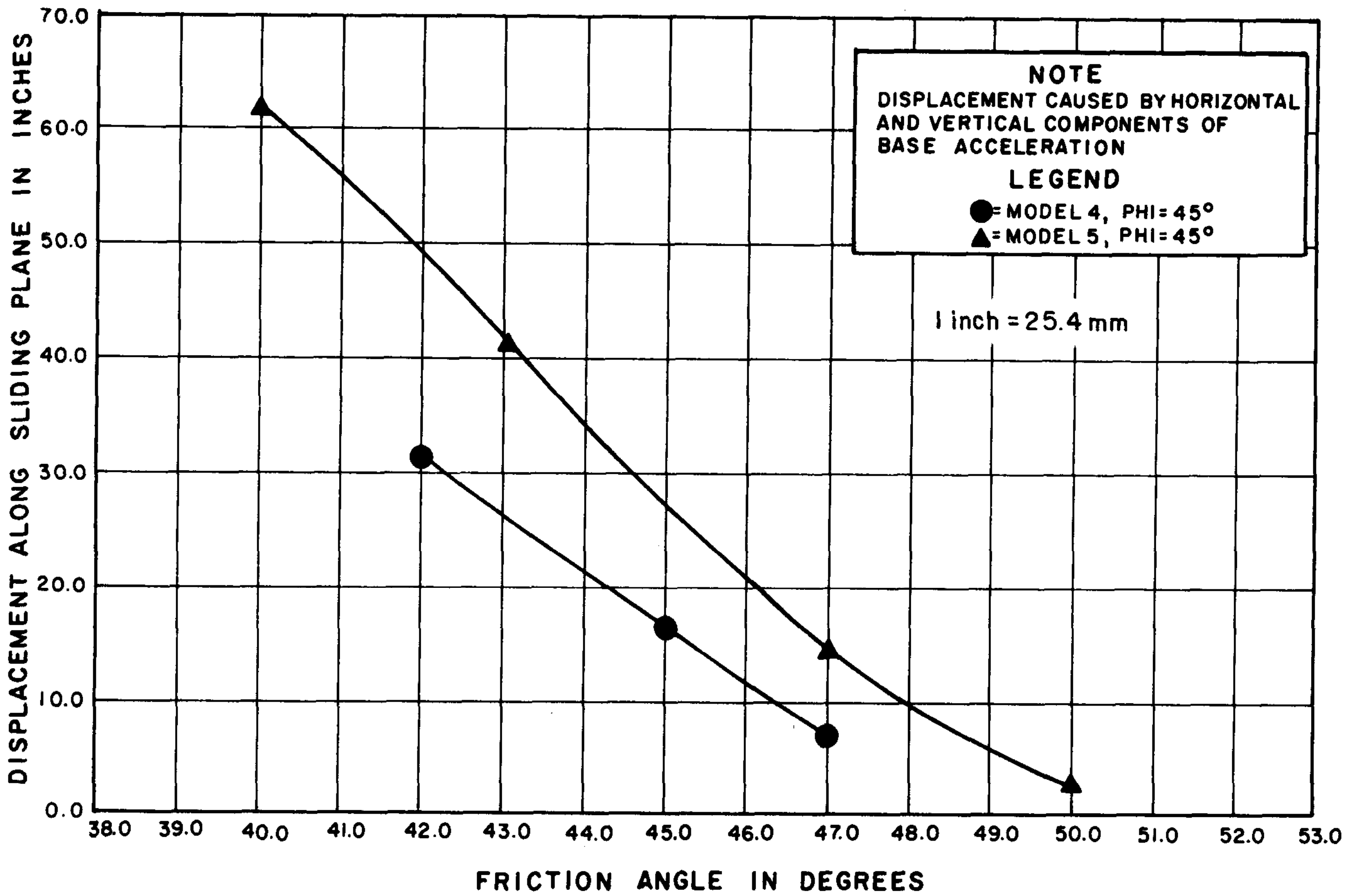


Figure 19. – Correlation of displacement with friction angle.

A-1(b)). The computer model estimated zero displacement using $\phi' = 53°$.

Reasons for the disagreement between measured and computed displacements at $\phi' = 53°$ are currently under investigation. Two possible areas are (1) the measure of shear strength at extremely low confining pressures (currently limited to approximately 1 lbf/in^2 (6.895 kPa)) and (2) the unknown effective stress conditions within the embankment model at failure because of unknown pore pressures in unconfined partially saturated sand. It is interesting to note the close agreement between measured and computed displacements when $\phi' = 45°$ is assumed for the embankment material. A friction angle of about 45° would be obtained if a linear Mohr envelope (app. A, fig. A-1 (a)) is used to evaluate the strength.

BIBLIOGRAPHY

[1] Chugh, A. K., "Slope Stability Analysis for Earthquakes," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 6, pp. 307-322, 1982.

[2] Prakash, S., "Soil Dynamics and its Application to Foundation Engineering," Proceedings of the 9th International Conference on Soil Mechanics and Foundation Engineering, vol. 2, Japan, 1977.

[3] Kutter, B. L., "Earthquake Deformation of Centrifuge Model Banks," *Journal of the Geotechnical Engineering Division*, vol. 110, No. 12, December 1984.

[4] Newmark, N. M., "Effects of Earthquakes on Dams and Embankments," *Geotechnique*, vol. 15, No. 2, pp. 139-160, 1965.

[5] Ambrasey, N. N., "Engineering Seismology and Earthquake Engineering," Miscellaneous paper 70-15, U.S. Army Corps of Engineers, Waterways Experiment Station, Vicksburg, MS, 1974.

[6] Makdisi, F. I., and H. B. Seed, "Simplified Procedure for Estimating Dam and Embankment Earthquake-Induced Deformations," Proceedings, *Journal of the Geotechnical Engineering Division*, vol. 104, GT. 7, 1978.

[7] Wilson, E. L., and R. W. Clough, "Dynamic Response by Step-by-Step Matrix Analysis," Paper No. 45, Symposium on the Use of Computers in Civil Engineering Laboratory, Nacional de Engonhamia Civil, Lisbon, Portugal, 1962.

[8] Shieh, W.Y.J., and R. J. Huang, "Permanent Deformation of Earth Dams Under Earthquakes," Proceedings, International Conference on Recent Advances in Geotechnical Earthquake Engineering and Soil Dynamics, University Missouri-Rolla, St. Louis, MO, vol. 1, pp. 453-458, April 1981.

[9] Franklin, A. G., and M. E. Hynes-Griffin, "Dynamic Analysis of Embankment Sections, Richard B. Russel Dam," Proceedings of the Conference on Earthquakes and Earthquake Engineering – the Eastern United States, pp. 623-642, Knoxsville, TN, 1981.

[10] Ambrasey, N. N., and E. L. Krinitsky, "Duration, Spectral Content, and Predominant Period of Strong Motion Earthquake Records from Western United States," Miscellaneous paper S–73–1, Report No. 8, U.S. Army Corps of Engineers, Waterways Experiment Station, Vicksburg, MS, 1977.

[11] Lin, J. S., *Probabilistic Evaluation of the Seismically Induced Permanent Displacements in Earth Dams*, Research Report No. R82–21, Order No. 729, School of Engineering, Massachusetts Institute of Technology, Cambridge, MA, June 1982.

[12] Kutter, B. L., "Deformation of Centrifuge Models of Clay Embankments Due to Bumpy Road Earthquakes," *Soil Dynamics and Earthquake Engineering*, vol. 2, No. 4, 1983.

[13] Rogers, F. J., *Experiments with a Shaking Machine*, California State Earthquake Investigation Commission, vol. 1, part 2, pp. 326-335, 1906.

[14] Jacobsen, L. S., "Motion of a Soil Subjected to Simple Harmonic Ground Vibration," *Bulletin of the Seismological Society of America*, vol. 20, pp. 160-195, 1930.

[15] Goodman, R. E., "The Stability of Slopes in Cohesionless Materials During Earthquakes," Ph.D. Dissertation, University of California, Berkeley, CA, 1963.

[16] Casagrande, A., "Characteristics of Cohesionless Soil Affecting the Stability of Slopes and Earth Fills," *Journal of the Boston Society of Civil Engineers*, vol. 23, pp. 13-32, 1936.

[17] Mononobe, N., A. Takata, and M. Matumura, "Seismic Stability of the Earth Dam," *Transactions*, Second Congress on Large Dams, vol. 4, pp. 435-444, 1936.

[18] Heiland, C. A., *Seismic Investigations of Hansen Dam and Site*, Report to U.S. Engineers Office, South Pacific Division, San Francisco, CA, 1938.

[19] Jacobsen, L. S., in the report by E. P. Hollis, "A Compilation of Existing Data Relating to the Effects of Earthquakes on Earth Dams," U.S. Army Corps of Engineers, South Pacific Division, San Francisco, CA, 1940.

[20] Bustamante, J. I., "Dynamic Behavior of Non-Cohesive Embankment Models," Ph.D. Dissertation, University of Illinois, Urbana, IL, 1964.

[21] Ramirez, J. R., "Failure of Model Sand Dams Subjected to Simulated Earthquakes," Thesis for Degree of Engineer, Stanford University, Palo Alto, CA, 1941.

[22] Tan, Ek-Khoo, "Stability of Soil Slopes," Proceedings, American Society of Civil Engineers, vol. 113, January 1948.

[23] Clough, R. W., and D. Pritz, "Earthquake Resistance of Rockfill Dams," *Journal of the Soil Mechanics and Foundations Division*, Proceedings of the American Society of Civil Engineers, vol. 82, April 1958.

[24] Seed, H. B., and R. W. Clough, "Earthquake Resistance of Sloping Core Dams," *Journal of the Soil Mechanics and Foundations Division*, Proceedings, American Society of Civil Engineers, vol. 89, No. SM1, pp. 209-242, 1963.

[25] Krishna, J., and S. Prakash, "Behavior of Earth Dam Models Under Seismic Loading," Proceedings of the 3d Symposium on Earthquake Engineering, vol. 2, p. 12, Roorkee, India, 1966.

[26] Krishna, J., and S. Prakash, "Earth Dams Subjected to Earthquakes," Proceedings of the 3d World Conference on Earthquake Engineering, vol. 1, pp. 83-91, 1965.

[27] Seed, H. B., J. Lysmer, and R. Hwang, "Soil-Structure Interaction Analysis for Seismic Response," *Journal of the Geotechnical Engineering Division*, Proceedings, American Society of Civil Engineers, vol. 101, No. GT5, pp. 439-457, 1975.

[28] Lee, K. L., and H. G. Walters, "Earthquake Induced Cracking of Dry Canyon Dam," Proceedings of the 5th World Conference on Earthquake Engineering, vol. 2, pp. 1544-1548, 1974.

[29] Prakash, S., and V. Chandrasekaran, "Model Studies on Rockfill Dam at Pandoh," Paper presented to the Annual Research Session, CBIP, 1972.

[30] Noda, S., H. Tsuchida, and E. Kurata, "Dynamic Tests of Soil Embankments," Proceedings of the 5th World Conference on Earthquake Engineering, vol. 2, pp. 1596-1605, 1974.

[31] Okamoto, S., et al., "Study on Effects of a Berm on the Stability of Rockfill Dams during Earthquakes," Preprint No. 6-14, 6th World Conference on Earthquake Engineering, pp. 79-84, 1975.

[32] Watanabe, T., "A Consideration on the Seismic Coefficients of Rock and Earthfill Dams Through Observed Accelerograms and Model Tests," preprint No. 6-13, 6th World Conference on Earthquake Engineering, pp. 73-78, 1977.

[33] Arya, A. S., and K. Kumar, et al., "Some Aspects of Seismic Design of Rockfill Dams," preprint No. 6-16, 6th World Conference on Earthquake Engineering, pp. 91-97, 1977.

[34] Arakawa, T., T. Kimata, M. Kondo, "Experimental Study on Failure Pattern of Embankment Supported by Sandy Layers During Earthquakes," Public Works Research Institute, *15th Joint Meeting Japan-U.S. Panel on Wind in Seismic Effects*, UJNR, Tsukuba, Japan, May 17-20, 1983.

[35] Ohmachi, T., and M. Momenzadeh, "Impulsive Failure Experiments on V-Shaped Sloping Embankment Models," *Soils and Foundations*, Japanese Society of Soil Mechanics and Foundation Engineering, vol. 24, No. 4, pp. 49-57, December 1984.

[36] Okamoto, S., *Introduction to Earthquake Engineering*, University of Tokyo Press, Tokyo, Japan, pp. 442-444, 1973.

[37] Kikusawa, M., and T. Hasegawa, "Analysis of Model Embankment Dam by Shaking Table Test," *Soils and Foundations*, Japanese Society of Soil Mechanics and Foundation Engineering, vol. 25, No. 1, pp. 1-14, March 1985.

[38] Ambrasey, N. N., "Engineering Seismology and Earthquake Engineering," Miscellaneous Paper S-74-15, U.S. Army Corps of Engineers, Waterways Experiment Station, Vicksburg, MS, May 1974.

[39] Harris, C. W., and J. L. Von Thun, "Estimation of Displacements of Rock Filled Dams Due to Seismic Shaking," International Conference on Recent Advances in Geotechnical Engineering and Soil Dynamics, Proceedings, vol. 1, pp. 417-423, St. Louis, MO, April 25 to May 3, 1981.

[40] McCafferty, R. M., *USBR Vibration Test System*, Division of General Research, Bureau of Reclamation, Denver, CO, preprint of a paper prepared for the 41st Shock and Vibration Symposium, Air Force Academy, Colorado Springs, CO, October 27-29, 1970.

[41] Das, B. M., *Fundamentals of Soil Dynamics*, Elsevier Science Publishing Company, Inc., pp. 7-8, New York, NY, 1983.

[42] Hunt, R. E., *Geotechnical Engineering Investigation Manual*, McGraw-Hill, Inc., pp. 665-695, New York, NY, 1984.

[43] Winterkorn, H. F., and H. Y. Fang, *Foundation Engineering Handbook*, Van Nostand Reinhold Company, pp. 380-393, New York, NY, 1975.

[44] Lambe, T. W., and R. V. Whitman, Soil Mechanics, Massachusetts Institute of Technology, pp. 245-246, John Wiley & Sons, Inc., New York, NY, 1969.

APPENDIX A
RESULTS OF LABORATORY TESTS ON SAND

Laboratory tests were conducted on sand used to construct the model embankments (1) to define material properties to be used as input to analytical methods, and (2) to describe and classify the material.

Gradation, specific gravity, minimum and maximum dry density, direct shear, CD (consolidated-drained) and CU (consolidated-undrained) triaxial shear, resonant column, and petrographic analyses were performed on the model test sand.

Sand used in the research program was prepared by dry sieving to remove the plus No. 8 and minus No. 16 material. This resulted in a clean, uniformly graded, medium- to coarse-grained sand having particle sizes ranging from 1.19 to 2.38 mm with a D_{10} size of 1.2 mm. The specific gravity of the sand was 2.67. An RD (relative density) test resulted in minimum and maximum index unit weights of 86.0 and 105 lbf/ft^3 (1371 and 1680 kg/m^3), respectively.

Petrographic analyses consisted of examining the sample megascopically, microscopically, and by using x-ray diffraction. Lithologic composition of the sand was predominantly granitic rock fragments, quartz, and feldspar, with a trace of mica and amphibole [A1].

CD and CU triaxial shear tests were performed on compacted 2-inch (51 mm) diameter by 5-inch (127 mm) high specimens placed at dry unit weights ranging from 89.5 to 98.8 lbf/ft^3 (1433 to 1583 kg/m^3) (22 to 72 percent RD). Specimens were consolidated at 3.6, 7.0, and 14.0 lbf/in^2 (24.8, 48.3, and 96.5 kPa) effective lateral confining pressures. The sand was tested at moisture contents of 0.0, 4.1, 5.5, and 8.7 percent.

Shear strength of the soil is expressed in terms of the Mohr-Coulomb failure criterion:

$$\tau_f = c' + \sigma_f' \tan\phi'$$

where:

- c' = the effective cohesion,
- ϕ' = the effective friction angle,
- τ_f = shear stress, and
- σ_f' = normal effective stress on the failure plane.

Figure A-1(a) depicts a single "best-fit" linear relationship of shear stress (τ) versus effective normal stress (σ') for the full range of normal stresses tested. It yields an effective friction angle (ϕ') of 43.9° and effective cohesion (c') of 2.9 lbf/in^2 (20.0 kPa). The data are a compilation of 12 triaxial shear test specimens for the sand. Figure A-1(b) approximates the shear strength envelope with two linear relationships. The first relationship is the "best-fit" linear equation for the data points from the tests at the lowest confining pressure and the origin. This yields an effective cohesion of 0 lbf/in^2 (0 kPa) and an effective friction angle of 53.1°, which is valid for effective normal stresses up to 7.7 lbf/in^2 (53.1 kPa). Because the normal stresses in the model embankments are less than 1 lbf/in^2 (6.895 kPa), the use of these parameters was recommended for the mathematical computer model. For normal stresses greater than 7.7 lbf/in^2 (53.1 kPa), ϕ' = 43.9° and c' = 2.9 lbf/in^2 (20.0 kPa) should be used.

A "best-fit" nonlinear shear strength envelope was also developed. It is described by the second order equation: $\tau = 1.38\sigma' - 0.0125(\sigma')^2$ (fig. A-2). "It has been recognized that the failure envelopes of many soils are significantly nonlinear" [A2] (e.g., dense sand [A3, A4] and compacted rockfill [A5, A6]). The nonlinear failure criterion appears to accurately define the shear strength of the sand at the normal stress levels used in the experimental physical models (i.e., $c' < 1$ lbf/in^2 (6.895 kPa)).

A series of direct shear tests were performed, but because the size of the sand particles was disproportionately large compared with the size of the test specimens, the test results were not reasonable.

Resonant column tests were performed on 2.8-inch (71 mm) diameter by 5.6-inch (142 mm) high specimens. Air-dried sand was placed at a dry unit weight of 97 lbf/ft^3 (1554 kg/m^3) and tested in a free-free resonant column apparatus. Modulus and damping were determined at effective confining pressures of 3.5, 7.0, 14.0, and 28.0 lbf/in^2 (24.1, 48.3, 96.5, and 193.1 kPa). G_{max} values were checked against the Hardin-Drnevich [A7] relationship, and shear modulus values obtained during testing were consistent with anticipated values.

Damping values for sands at shear strains $\leq 1 \times 10^{-4}$ typically range from 1 to 3 percent [A8]. The damping values obtained in these tests were less than 1 percent, and lower than anticipated.

Results of the free-free resonant column tests were used to calculate the anticipated range of model embankment shear wave velocities, resonant frequency, and period [A8] (see table A-2).

It was essential that model testing be conducted at input frequencies other than the embankment resonant frequency. Natural resonant frequencies were calculated using a procedure developed by Makdisi and Seed [A8].

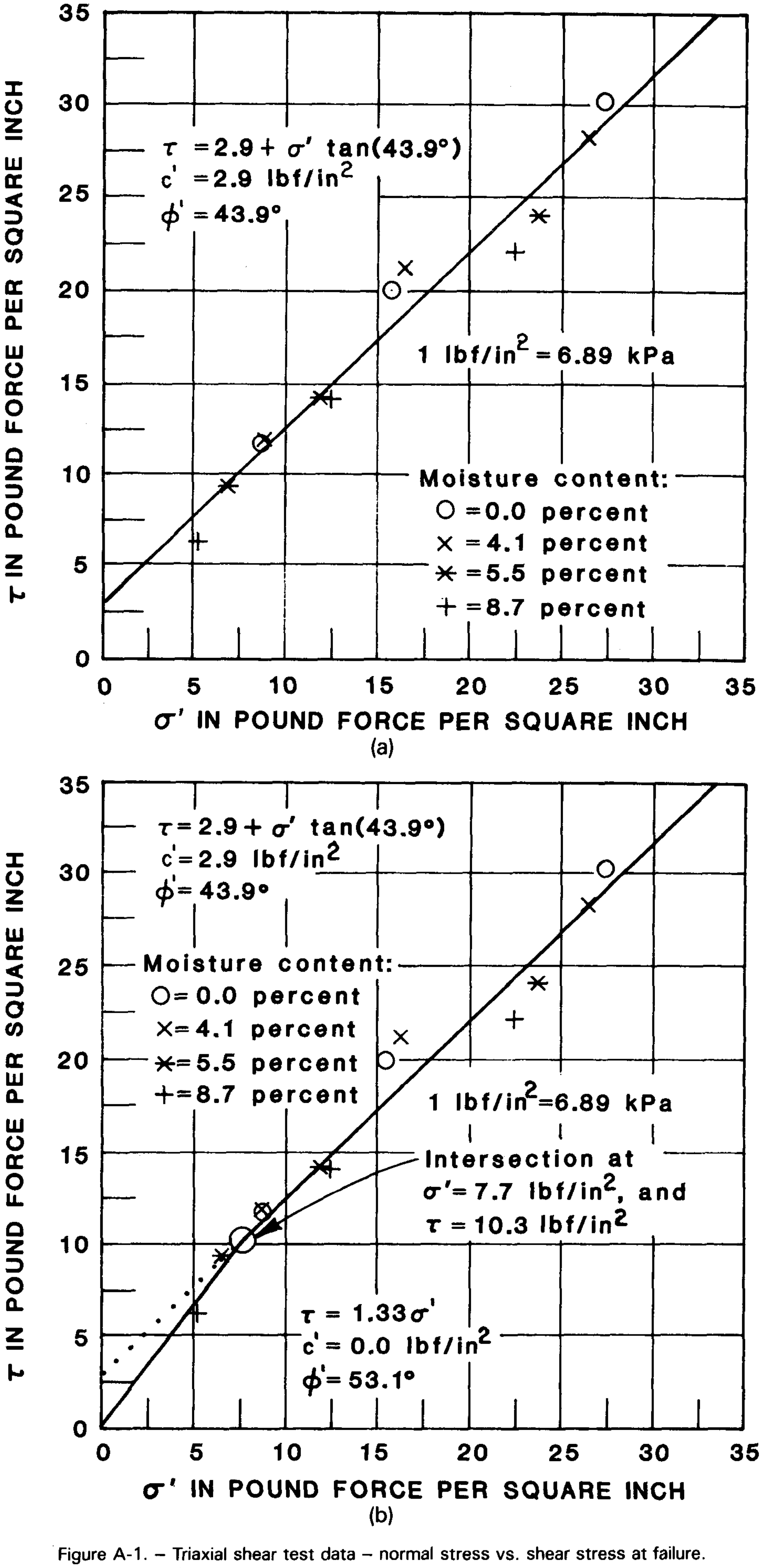


Figure A-1. – Triaxial shear test data – normal stress vs. shear stress at failure.

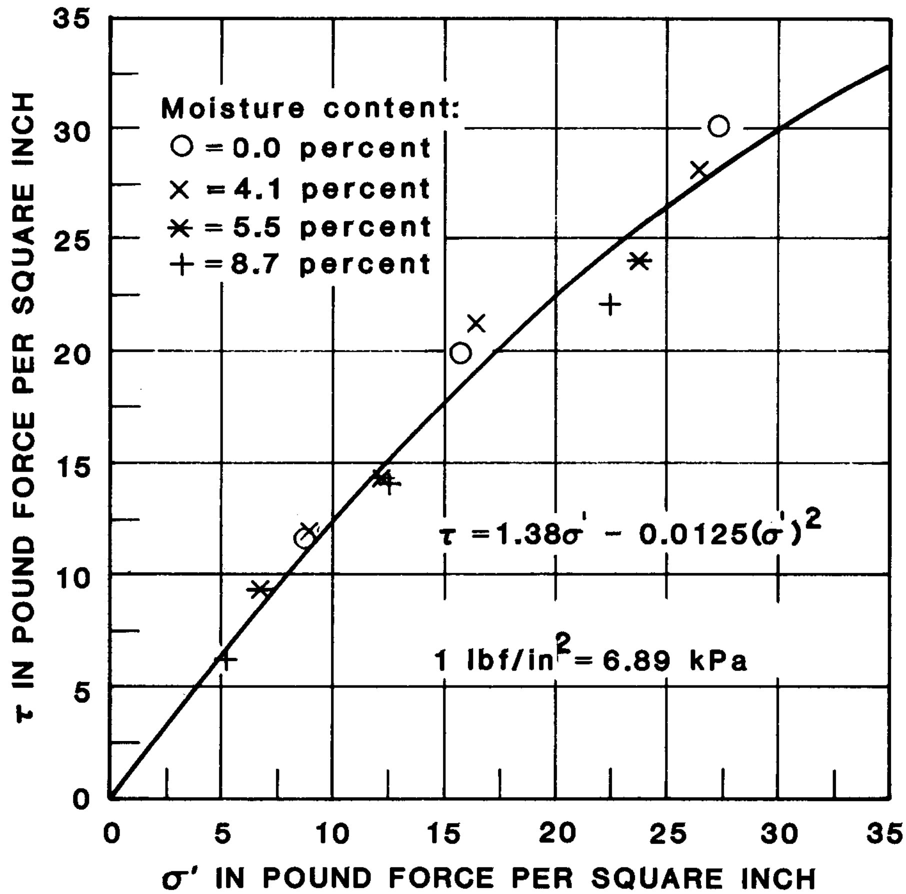


Figure A-2. – Triaxial shear test data – normal stress vs. shear stress at failure.

Table A-1. – Comparison of laboratory and calculated results using the Hardin-Drnevich relationship.

$\bar{\sigma}_3$		σ'_m		$G_{max}{}^*$		G_{max}	
lbf/in²	kPa	lbf/in²	kPa	lbf/in² × 10³	GPa	lbf/in² × 10³	GPa
3.5	24.1	3.5	24.1	6.81	0.047	6.5	0.045
7	48.3	7	48.3	9.63	.066	9.5	.066
14	96.5	14	95.6	13.6	.094	13	.090
28	193.1	28	193.1	19.3	.133	18	.124

Notes:

$\bar{\sigma}_3$ = effective confining pressure.
σ'_m = mean principal effective stress (assumed that $\bar{\sigma}_3 = \sigma'_m$).
$G_{max}{}^*$ = maximum shear modulus calculated from the Hardin-Drenevich relationship.
G_{max} = maximum shear modulus obtained from laboratory test results.

Table A-2. – Model embankment properties.

Shear modulus range		Shear wave velocity		First three natural frequencies, Hz			Embankment period (1/Hz)		
lbf/ft² × 10⁵	GPa	ft/s	m/s	1st	2d	3d	1st	2d	3d
9.8	0.047	554	169	169	389	609	0.006	0.003	0.002
27.7	.133	931	284	284	654	1,025	.004	.002	.001

BIBLIOGRAPHY

[A1] Bechtold, C. A., Head, Chemistry, Petrography, and Chemical Engineering Section, Memorandum to Head, Soil Mechanics Section, Attention: R. Luehring, Subject: "Petrographic Rock and Mineral Identification of Sand Sample – Engineering Properties of Rockfill Dams – Geotechnical Research Project DB–31," Petrographic referral code: 84-18, Bureau of Reclamation, Denver, CO, March 8, 1983.

[A2] Charles, J. A., and M. M. Soares, "The Stability of Slopes in Soils with Nonlinear Failure Envelopes," *Canadian Geotechnical Journal*, J. 21, pp. 397-406, 1984.

[A3] Vesic, A. S., and G. W. Clough, "Behaviour of Granular Materials Under High Stresses," *Journal of the Soil Mechanics and Foundations Division*, American Society of Civil Engineers, 94 (SM3), pp. 661-688, 1968.

[A4] Ponce, V. M., and J. M. Bell, "Shear Strength of Sand at Extremely Low Pressures," *Journal of the Soil Mechanics and Foundations Division*, American Society of Civil Engineers, 97(SM4), pp. 625-637, 1971.

[A5] Marachi, N. D., C. K. Chan, and H. B. Seed, "Evaluation of Properties of Rockfill Materials," *Journal of the Soil Mechanics and Foundations Division*, American Society of Civil Engineers, 98(SM1), pp. 95-114, 1972.

[A6] Marsal, R. J., *Mechanical Properties of Rockfill in Embankment Dam Engineering*, Casagrande volume, John Wiley & Sons, New York, NY, pp. 109-200, 1973.

[A7] Seed, H. B., and I. M. Idriss, "Soil Moduli and Damping Factors for Dynamic Response Analysis," Report No. EERC 70–10, Earthquake Engineering Research Center, University of California, Berkeley, CA, December 1970.

[A8] Makdisi, F. I., and H. B. Seed, "A Simplified Procedure for Evaluating Embankment Response," *Journal of the Geotechnical Engineering Division*, No. G12, American Society of Civil Engineers, December 1979.

APPENDIX B
ACCELEROMETER LOCATIONS

Table B-1. – Accelerometer locations.

Model No.	Accelerometer No.	Offset from major axis ℄, inch (mm)	Offset from minor axis ℄, inch (mm)	Accelerometer elevation, inch (mm)
1	7	10 (254) E	3 (76) S	10 (254)
	8	℄	6 (152) S	14 (356)
	13	℄	℄	5 (127)
	18	℄	3 (76) S	10 (254)
	17	did not operate	℄	1 (25)
2	7	0.5 (13) E	℄	12 (305)
	8	1.5 (38) W	1.25 (32) N	14.5 (368)
	13	2 (51) E	1.25 (32) S	5 (127)
	18	1.5 (38) E	0.5 (13) S	10 (254)
	6	11.5 (384) E	not located	13.5 (343)
3	7	7 (17.8) W	5 (127) S	12 (305)
	8	3 (76) E	9 (229) S	14 (356)
	9	8 (203) E	1 (25) N	11 (279)
	18	℄	2 (51) N	14 (356)
4	8	14 (356) W	11 (279) N	6 (152)
	9	7.5 (190) E	2.5 (63) N	11.4 (290)
	10	5 (127) W	3 (76) S	13 (330)
	13	18 (457) W	1 (25) S	7 (178)
5	8	18 (457) W	8 (203) N	7 (178)
	9	4 (102) E	9 (229) N	12 (305)
	10	3.5 (889) W	6 (152) S	17 (432)
	13	10 (254) W	0.5 (13) N	14 (356)
6	1	5 (127) E	℄	9.6 (244)
	8	3 (76) W	3 (76) S	12.1 (305)
	9	11.5 (292) W	6.5 (165) N	8.7 (221)
	17	11.5 (292) W	1 (25) N	7.2 (183)

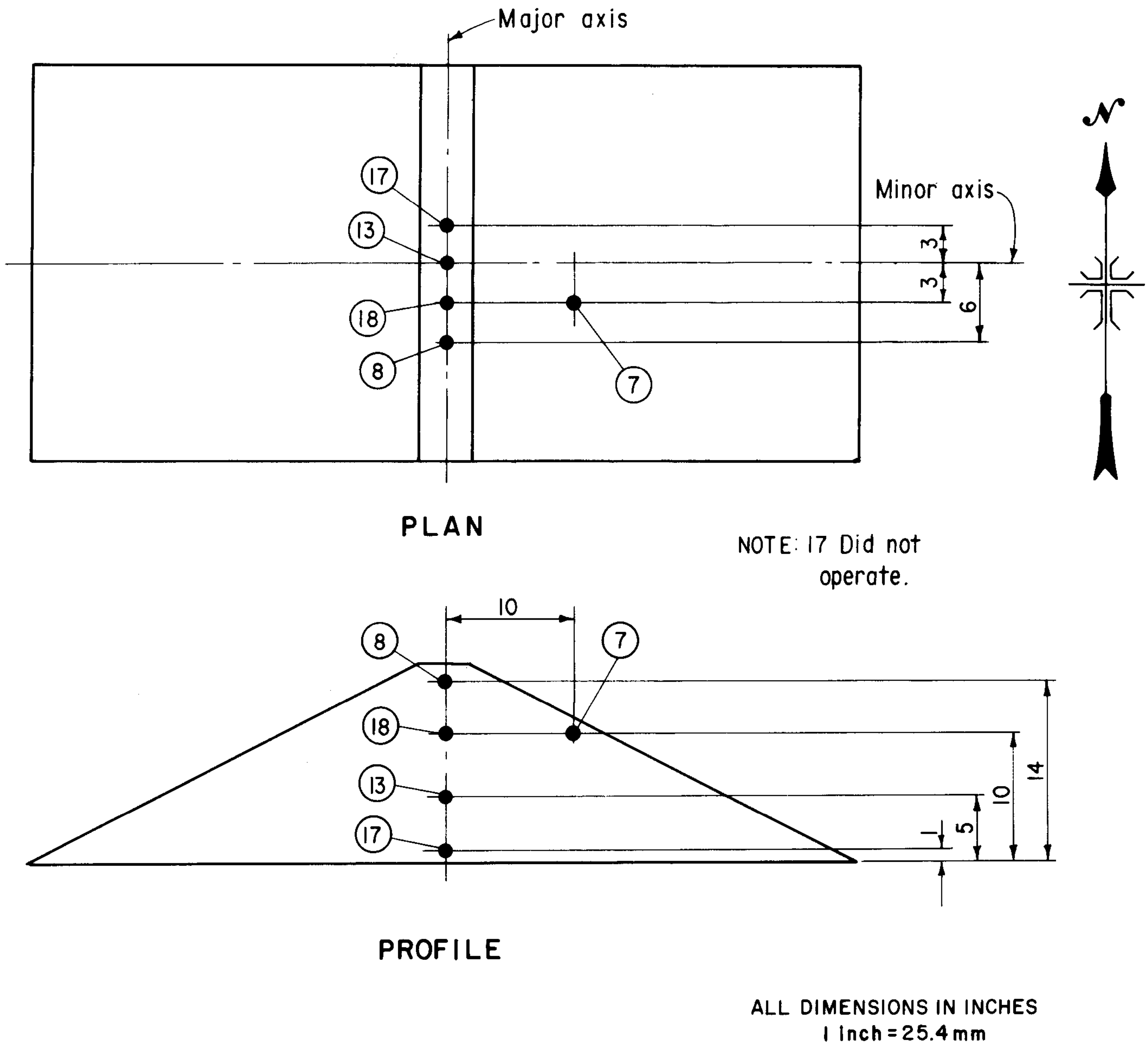


Figure B-1. – Accelerometer locations for model No. 1.

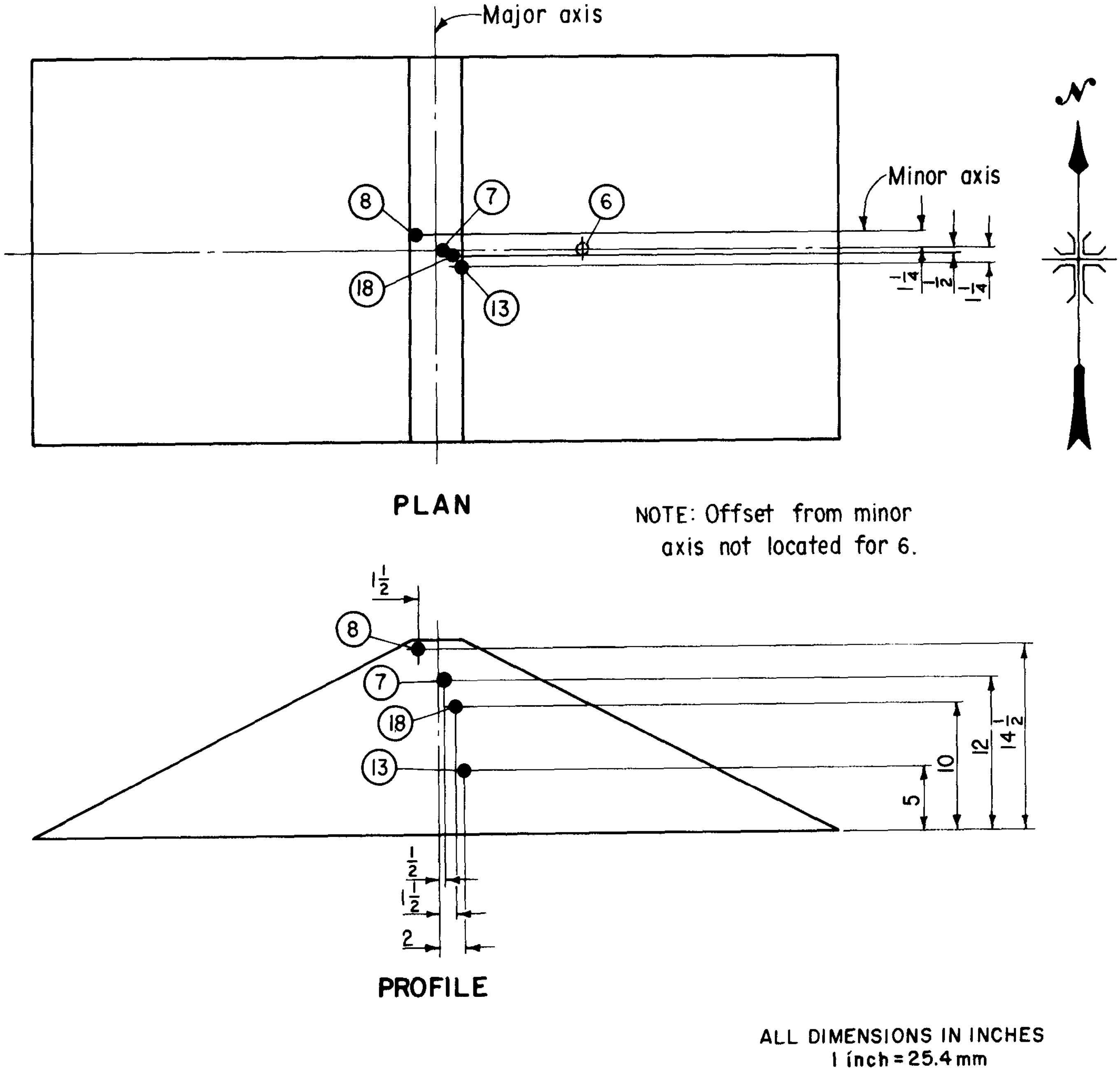


Figure B-2. – Accelerometer locations for model No. 2.

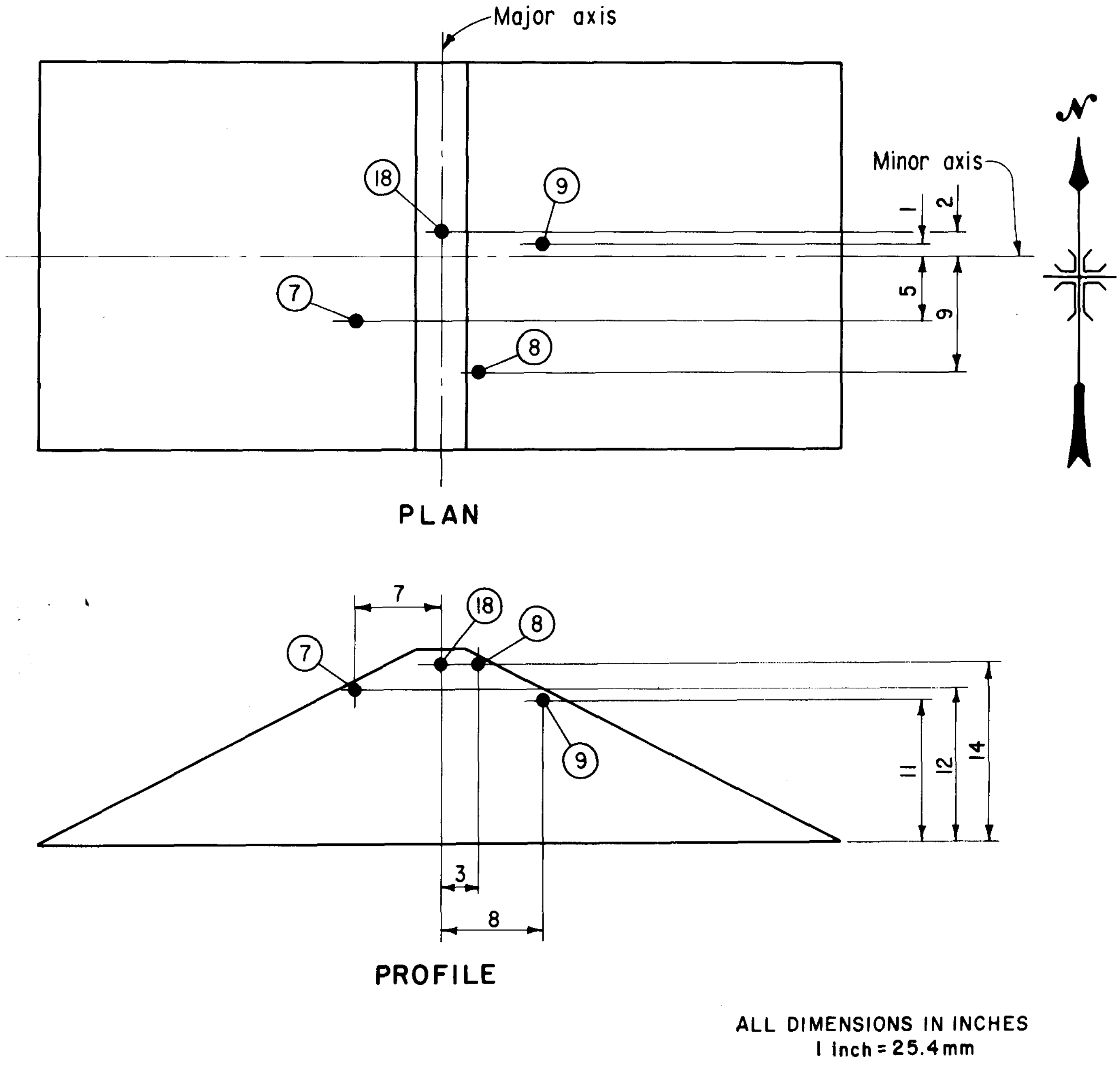


Figure B-3. – Accelerometer locations for model No. 3.

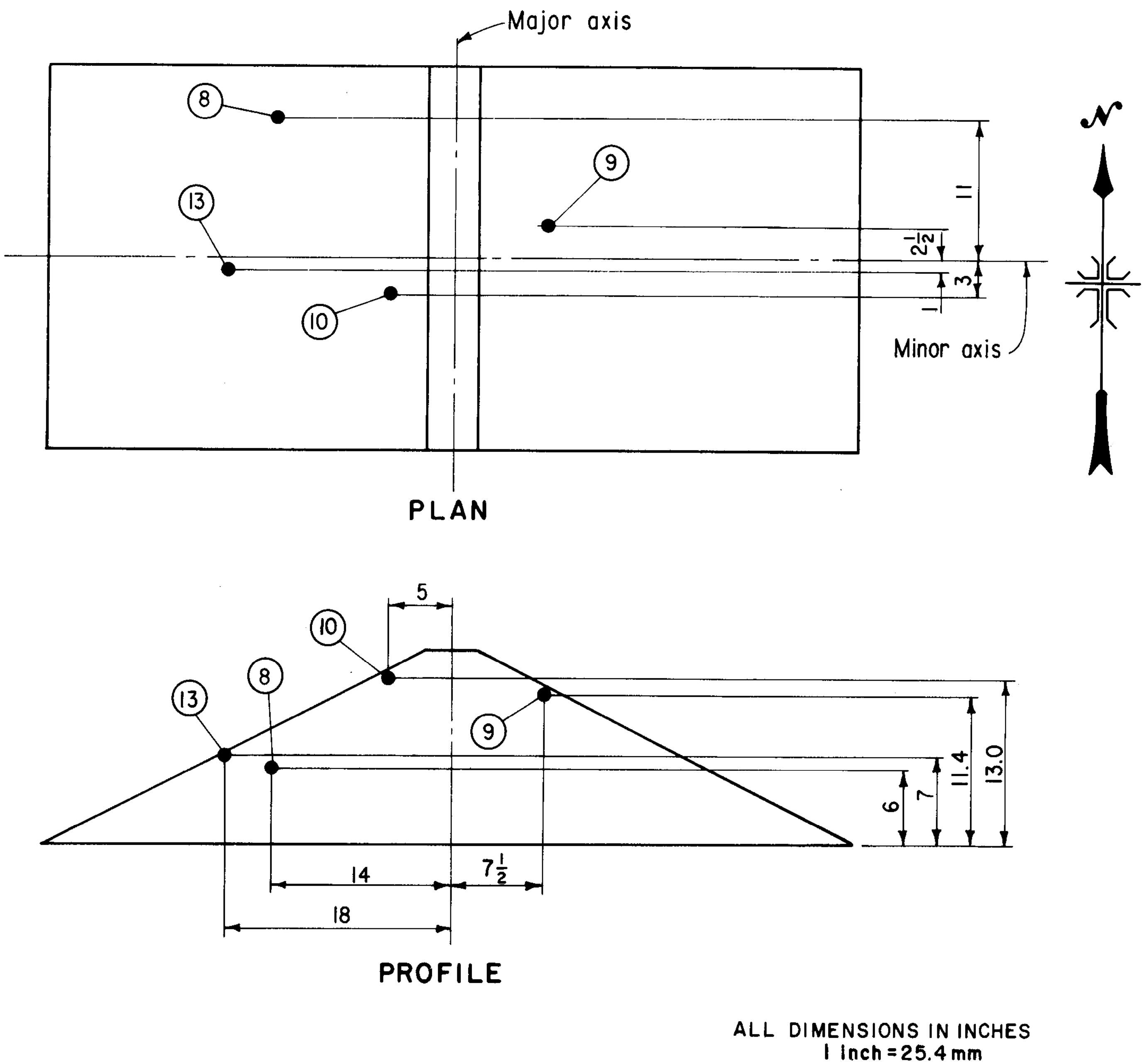


Figure B-4. – Accelerometer locations for model No. 4.

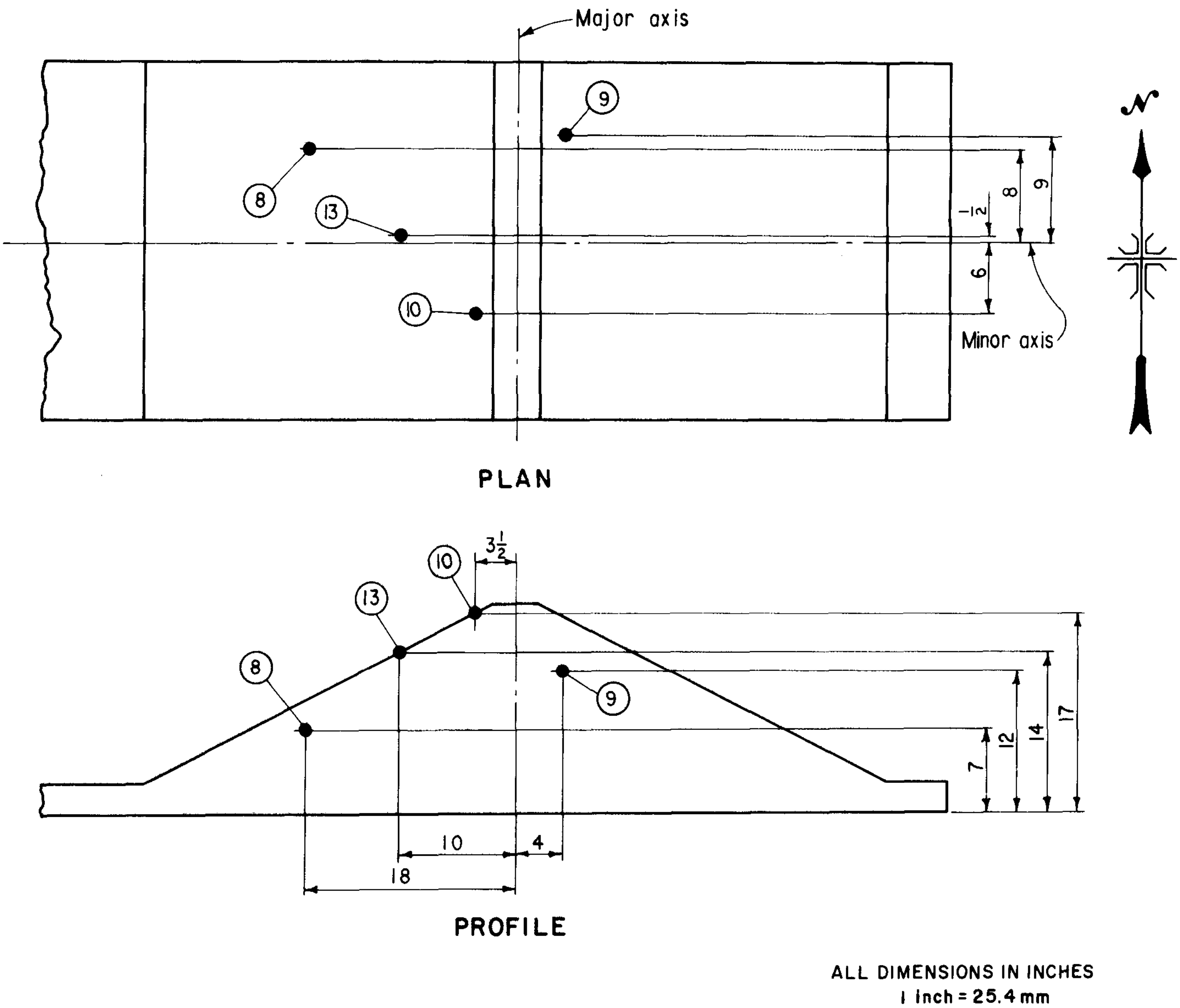


Figure B-5. – Accelerometer locations for model No. 5.

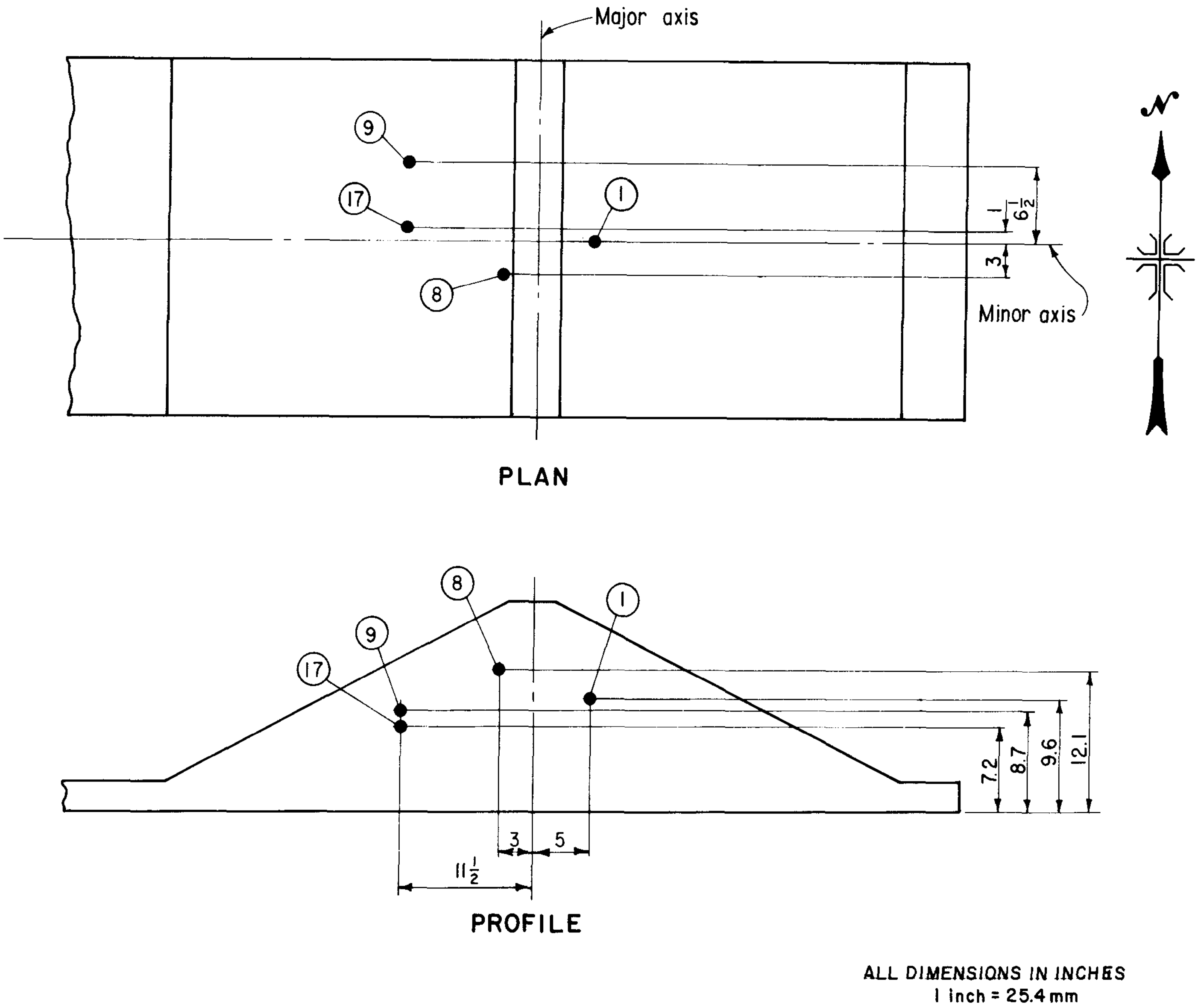


Figure B-6. – Accelerometer locations for model No. 6.

APPENDIX C
EQUIPMENT DESCRIPTION AND DATA ACQUISITION DOCUMENTATION

CONTENTS

ROCKFILL RESEARCH PROJECT CONTROL AND INSTRUMENTATION SYSTEM*

INTRODUCTION

The facilities of the Concrete and Structural Branch's Vibration Laboratory were used for testing scale models of rockfill dams as part of the Rockfill Research Project, DB-31. This testing took place in August through November 1983. A block diagram of the system is shown on figure C-1.

Scale model dams were constructed using sand of varying moisture contents in a steel-framed box with plexiglass sides. This box was rigidly mounted on a steel table, which in turn was mounted on hydrostatic bearing tables, allowing the table to move in one axis. One end of the table was attached to a hydraulic actuator mounted on a fixed buttress that moved the assembly in a sinusoidal manner.

As the dam was shaken, its movements were recorded on video tape and accelerations were measured using embedded accelerometers. This acceleration data and accelerations measured on the fixture were fed to a high-speed data acquisition system for processing by computer and storage on disk. The models were shaken at 5 hertz with a peak acceleration of 0.5 to 0.7 *g* for a 1-minute duration. After the test, acceleration data were retrieved from the disk, and plots of the data were generated. Selected accelerometer data were then transmitted to the Cyber system for processing.

SYSTEM DESCRIPTION

The control and instrumentation system consists of three parts, which will be described individually: (1) shaker system, (2) data acquisition system, and (3) data processing software.

Shaker System

The hydraulic actuator used to shake the models is a 25-ton (22 680-kg), 1-inch (25-mm) stroke ram equipped with a high-performance two-stage Team SV-200 servovalve. The actuator has an integral LVDT for stroke control. It is mounted on a buttress at one end of the 250-ton (226 800-kg) seismic mass. The mass isolation system was disabled for these tests because of the low frequency being used. Hydraulic power was supplied to the actuator from a 2,500-lb/in^2 (17.2-MPa), 70-gal/min (265 L/min) hydraulic power supply.

The actuator was electronically controlled by the servo valve and an MTS 442 servo controller. The servo controller was configured to use position feedback (supplied by the LVDT in the ram) to control the actuator. A voltage input to the servo controller controlled movement of the actuator.

The required input to the test specimen was a 5-hertz sinusoidal waveform ramped over 5 seconds from zero amplitude to the amplitude corresponding to the required acceleration. This amplitude was held for 50 seconds, then ramped back to zero amplitude in 5 seconds for a total test duration of 60 seconds. This is graphically illustrated on figure C-2. This voltage waveform was generated as follows: A program running on the HP-85 computer sent commands to the digital to analog converter card through the HP-3497 data acquisition and control unit to cause its output to go from zero volts to 10 volts in 5 seconds, hold it for 50 seconds, and ramp back down in 5 seconds. The stepped output from the D/A converter was then fed into a Krohn-Hite 3323R filter configured as a 1 hertz low-pass filter of unity gain. This provided a smooth voltage ramp that was fed to one input of a multiplier. The other multiplier input was fed by an Exact 605 function generator set for 5 hertz and 10 volts peak output. The resulting output from the multiplier was fed to the servo controller and the ram displacement was as shown on figure C-2. The ramp generation program is discussed in detail in appendix C1.

A sinusoidal input voltage resulted in an acceleration waveform that appeared to be quite sinusoidal. On closer examination using a spectrum analyzer, the harmonic content of the acceleration waveform was found significant. Tuning the servo controller gains and adding rate stabilization improved the waveform to an acceptable level. The final adjustments yielded a waveform with all harmonics at least 20 decibels below the fundamental with most 40 decibels below.

Data Acquisition System

Test data were collected both visually and electronically. Visual data collection was accomplished through the use of two television cameras connected to two video tape recorders. Electronic data collection used eight accelerometers and an LVDT to measure model and test fixture behavior. The data from these transducers were multiplexed and digitized by an HP-2250 measurement and control processor and were then transferred to an HP-1000 minicomputer that stored the data on hard disk.

Visual data collection used two color television cameras connected to two video tape recorders. One

* By Fred A. Travers, Electronics Engineer, Concrete and Structural Branch, Division of Research and Laboratory Services, Bureau of Reclamation, February 1984.

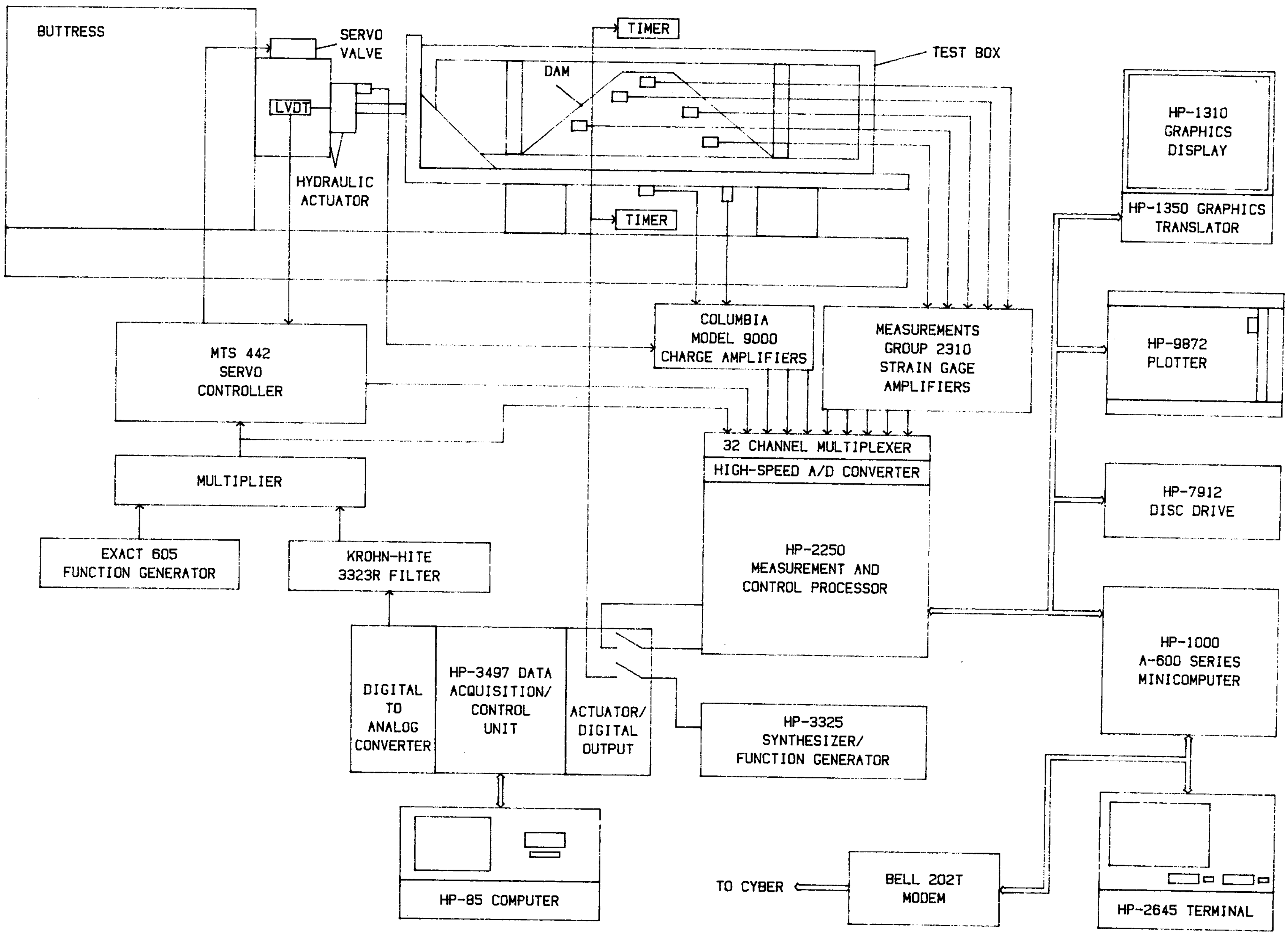


Figure C-1. – Instrumentation system block diagram.

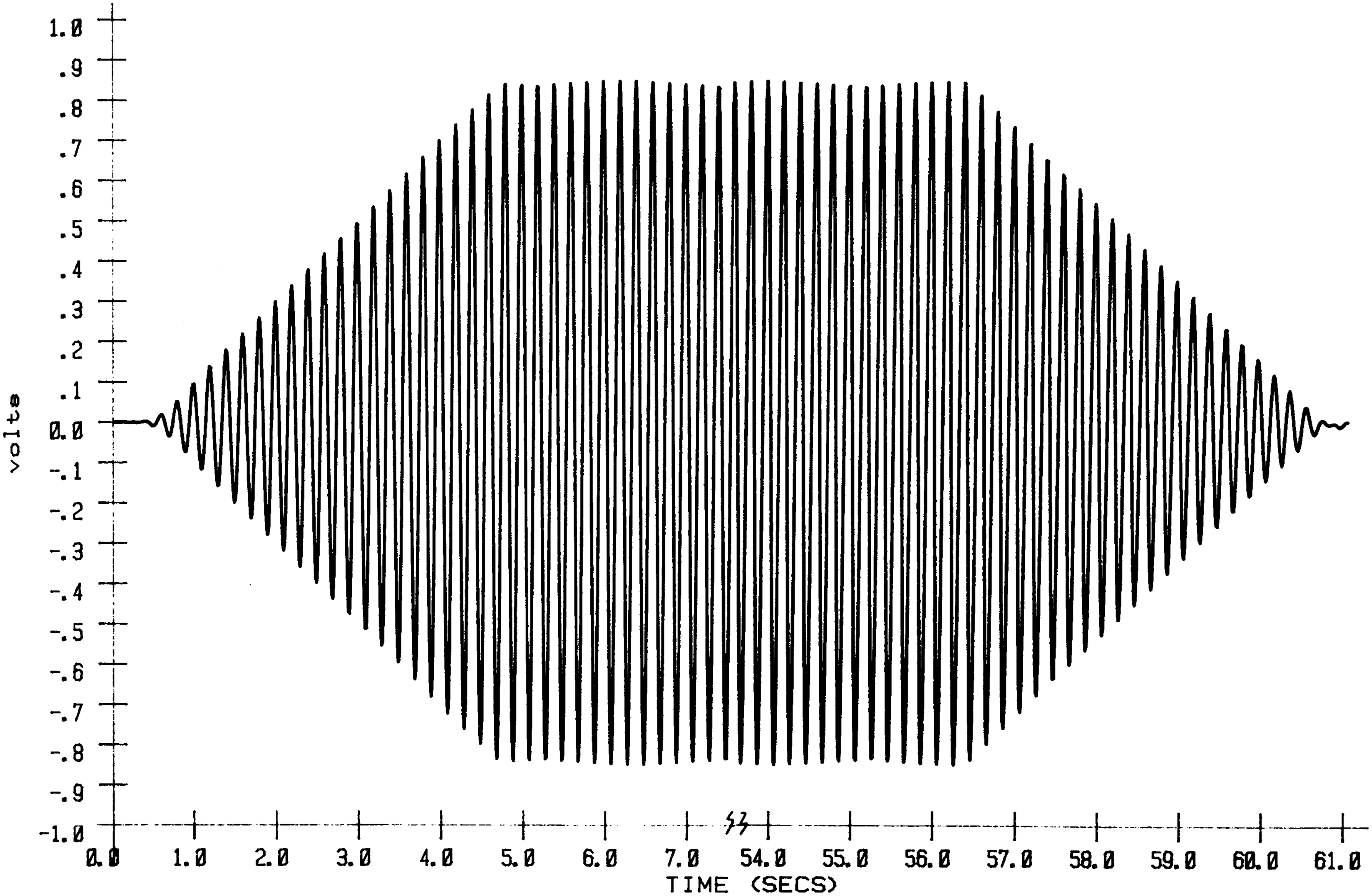


Figure C-2. – Input signal waveform.

camera was positioned above the model and the other on a side. Digital timers were mounted near the model in view of the cameras to provide a time reference on the video tape. The timers were controlled by the ramp generation program on the HP-85 computer. A 1-kHz square wave was generated by the HP-3325 synthesizer/function generator and fed into an actuator/digital output card in the 3497. From there, it fed the digital timers. The program on the HP-85 used the 3497 to turn on the 1 kHz signal to the timers and then started the d-c ramp. Refer to appendix C1 for details of this program. Other visual measurements of the model were made manually before and after each test.

Electronic instrumentation consisted of eight accelerometers, five of which were embedded in the actual model, three were mounted on the test fixture (one vertically and one horizontally) and one on the actuator. The LVDT in the actuator provided displacement information.

The accelerometers embedded in the model were Entran Model EGA–125–50 and EGA–125–100 miniature accelerometers weighing only 0.5 gram. These accelerometers had peak ranges of 5 *g* and 10 *g* and were of the strain gauge type. They were coated in beeswax to alleviate problems from moisture contamination. The signals from these five accelerometers were sent to Measurements Group Model 2310 signal conditioning amplifiers where the signal was amplified and scaled to 5 volts per peak *g*. The accelerometers mounted on the test fixture and actuator were Columbia Model 302–6 charge-type accelerometers. The signals were sent to Columbia Model 9000 change amplifiers for amplification and scaling to 5 volts per peak *g*. All accelerometers were calibrated individually against an accelerometer standard at 5 hertz and 1 *g* peak. The calibration factors for each transducer were entered into a file in the HP–1000 minicomputer for use in processing the data. The program CALFAC creates the calibration factor file and is described in appendix C2. The LVDT in the actuator feeds its signal to the MTS servo controller where it is amplified and scaled to 20 volts per inch (508 volts/mm).

The signals from the eight accelerometer conditioning amplifiers and the LVDT conditioner as well as the input signal to the servo controller (the output from the multiplier) were sent to channels 17 through 26 of the HP–2250 measurement and control processor. The HP–2250 provided input filtering by means of a low-pass filter with its 3-decibel point at 7.5 hertz and a 6 decibels per octave rolloff. This input filtering was necessary to eliminate high-frequency noise caused by the actuator dither and bearing tables. Attenuation of the 5-hertz fundamental was compensated for with an adjustment of the transducer scale factors used to process the raw data after the test.

The input filtering was done by an HP–25540B signal conditioning module located on the HP–25502A 32-channel high-level multiplexer card. The multiplexer card scanned the 10 inputs at a rate of 1000 channels per second, giving each channel 20 samples per cycle of the 5-hertz fundamental. The multiplexed signal was then sent to an HP–25501A 16-channel high-speed analog input card where the signals were digitized. The digitized data were then transferred to the HP–1000 minicomputer, which put the data on hard disk. This was accomplished by programs operating simultaneously in the HP–2250 processor and the HP–1000 minicomputer. This transfer process is described in detail in appendix C3.

The digitizer in the 2250 provides 14-bit resolution with an auto ranging input. Maximum full-scale input is ± 10 volts, and so with the 5 volts per *g* calibration of the accelerometer input signal, the minimum acceleration resolution was 0.00024 *g*. The accelerometers have a linearity of 1 percent making the acceleration data accurate to within about 0.1 *g*.

Data acquisition was triggered by the ramp generation program running on the HP–85 (see appendix C1). When the program closed the contacts on the actuator/digital output card in the HP–3497 to start the digital timers, it also closed a set of contacts to operate the external trigger on the HP–2250. The HP–2250 was in a wait state before this and on this trigger signal it started data acquisition. The HP–1000 minicomputer program was waiting for data from the 2250. The whole data collection and transfer process is described in appendix C3.

The data collected during the test were transferred to the HP–1000 in raw binary form as two 16-bit words to obtain a high transfer rate. The first word and part of the second contain the binary value of voltage and the rest of the second word contains a code for the gain of the A/D converter input. When the data were taken off of disk, they were converted to actual voltage values. In addition, to maintain a high data-transfer rate, zero correction of the readings was not done. To do zero correction, the zero offsets were measured just before the test by shorting the inputs and then reading the resulting voltage. These values were stored for subsequent correction of the data. The program ZEROC measured these zero offsets and stored them. See appendix C4 for details on this program.

A procedure and checklist for testing is included as appendix C5.

Data Processing Software

Processing of the test data was limited to only two functions: (1) plotting on CRT and hardcopy, and (2) transfer of data to the Cyber mainframe. Plotting the data provided feedback after a test so that data validity could be verified. Transfer of the data to the Cyber provided the acceleration data to the Dams Branch for analysis in their displacement dynamics program.

The program PLOT provided the capability to plot out up to 15 seconds of data from up to three selected channels on the HP-1310 graphics display or on the HP-9872 plotter. The program CONVERT converted raw data into a form acceptable for use by the DYNDSP program on the Cyber. Data transfer was then accomplished by writing the converted data on a tape and then logging onto the Cyber and reading the tape using TEXT mode. See appendix C6 for descriptions of the plotting program, conversion program, and the transfer process.

APPENDIX C1
RAMP GENERATION PROGRAM

The ramp generation program serves three functions: (1) triggering of the HP–2250 to begin data acquisition, (2) digital timer clock-gating, and (3) ramp generation. Because of the timing demands on the HP–2250 during high-speed continuous data acquisition, it could not be used to generate the ramp and gate the time block. Therefore, it was decided to use a "stand-alone" system consisting of an HP–85 and an HP–3497 with an HP–44428 actuator/digital output card to trigger the HP–2250 and gate the clock for the digital timer and an HP–44429 D/A converter card to generate a voltage ramp. The following documented program listing gives the details of this process. The HP–2250 was triggered through the use of the external trigger input on the 2250 MCI card.

The procedure for using the program is as follows:

1. Load the program from the disk.
2. Run the program.
3. The operator is prompted for the length of the test (this includes 5 seconds of ramping up and 5 seconds of ramping down).
4. When the test is ready to be run, push any of the function keys to start the test.

```
10 ! ROCKFILL RESEARCH PROJECT - DB-31
20 !
30 ! DC RAMP, DATA TRIGGER AND CLOCK START PROGRAM
40 !
50 !       REV. 0 1/84  F. TRAVERS
60 !
70 !  THIS PROGRAM USES AN HP-3497 TO GENERATE A DC VOLTAGE RAMP FROM 0 TO 10
80 !  VOLTS IN 5 SECONDS.  THE VOLTAGE IS HELD FOR A PROGRAMMED LENGTH OF
90 !  TIME AND THEN IS RAMPED FROM 10 TO 0 VOLTS.  AT THE START OF THE RAMP
100 ! A DIGITAL OUTPUT CARD IS USED TO TRIGGER DATA ACQUISITION AND A CLOCK
110 ! OSCILLATOR IS ENABLED.
130 !
140 ! HP 3497 CONFIGURED AS FOLLOWS:
150 !      SLOT 0 - HP 44428A ACTUATOR/DIGITAL OUTPUT CARD
160 !         CHANNEL 0 - COMMON TO NORMALLY OPEN TO 2250 EXTERNAL TRIGGER
170 !         CHANNEL 2 - COMMON TO NORMALLY OPEN TO CLOCK OSCILLATOR COMMON
180 !         CHANNEL 3 - COMMON TO NORMALLY OPEN TO CLOCK OSCILLATOR HIGH
190 !      SLOT 1 - HP 444291 DUAL OUTPUT 0-+/-10 VOLT D/A CONVERTER
200 !         CHANNEL 0 - TO MULTIPLIER INPUT
210 !
220 ! INITIALIZATION OF DIGITAL OUTPUTS
230 !
240 OUTPUT 709 USING "K" ; "SI"
250 CLEAR
260 !
270 ! ENTRY OF TEST DURATION
280 !
290 DISP "INPUT TEST DURATION IN SECONDS"
300 INPUT S
310 S=S-10
320 CLEAR
330 !
340 ! DEFINE AND LABEL FUNCTION KEYS FOR START
350 !
360 ON KEY# 1,"--------" GOTO 480
370 ON KEY# 2,"-----STA" GOTO 480
380 ON KEY# 3,"RT------" GOTO 480
390 ON KEY# 4,"--------" GOTO 480
400 KEY LABEL
410 !
420 ! HOLD FOR START
430 !
440 GOTO 440
450 !
460 ! TEST START
470 !
480 CLEAR
490 DISP "TEST RUNNING"
500 !
510 ! CLOSE DIGITAL OUTPUTS 0,2,3
520 !
530 OUTPUT 709 USING "K" ; "DW0,15"
540 !
550 ! LOOP TO RAMP UP D/A VOLTAGE
560 !
570 FOR V=0 TO 10010 STEP 110
580 !
590 ! STRING FOR VOLTAGE TO D/A AND OUTPUT OF STRING
600 !
610 O$="AO1,0,"&VAL$(V)
```

```
620 OUTPUT 709 USING "K" ; O$
630 NEXT V
640 !
650 ! WAIT AT FULL AMPLITUDE
660 !
670 WAIT S*1000
680 !
690 ! LOOP TO RAMP DOWN D/A VOLTAGE
700 !
710 FOR V=0 TO 10010 STEP 110
720 V1=10010-V
730 O$="AO1,0,"&VAL$(V1)
740 OUTPUT 709 USING "K" ; O$
750 NEXT V
760 !
770 ! OPEN DIGITAL OUTPUTS 0,2,3
780 !
790 OUTPUT 709 USING "K" ; "DW0,0"
800 CLEAR
810 DISP "TEST COMPLETE"
820 END
```

APPENDIX C2
TRANSDUCER CALIBRATION FACTOR FILE

To translate the raw data taken during a test into meaningful numbers, each data point must be multiplied by an appropriate scale factor to translate the voltage read into actual engineering units. A file was created on the computer for each test that held the scale factor for each data channel. Additional data on each channel were also stored. These included the units of measurement, the date of calibration, and a description of the measurement. A sample printout of a data file is included in this appendix.

To facilitate entry of transducer scale factors, program CALFAC was written to request the required data from the operator and then create a properly formatted file to store this data. The operator is prompted for each piece of data required. For each channel the operator must supply a calibration factor in volts per unit of measurement, the units of the measurement, the date of transducer calibration, and a description of the measurement and a test title for the file. The data are displayed for checking before they are stored, and then they are written on a disk and printed. A documented listing of the program follows.

For successive tests with the same transducers and only a test number change, it is easier to edit the original file and create a new copy rather than to go through the file creation program.

Calibration Factor File C1101
13:05 PM MON, 1 NOV., 1983
HP-2250 Measurement and Control Unit

----- Rockfill Research Project DB-31 -- Test Five 11/1/83 ------

CHANNEL	CALIBRATION FACTOR IN VOLTS/UNIT	UNITS OF MEASUREMENT	DATE OF CALIBRATION	MEASUREMENT DECRIPTION
1	0.000000			NO CALIBRATION FACTOR SPECIFIED
2	0.000000			NO CALIBRATION FACTOR SPECIFIED
3	0.000000			NO CALIBRATION FACTOR SPECIFIED
4	0.000000			NO CALIBRATION FACTOR SPECIFIED
5	0.000000			NO CALIBRATION FACTOR SPECIFIED
6	0.000000			NO CALIBRATION FACTOR SPECIFIED
7	0.000000			NO CALIBRATION FACTOR SPECIFIED
8	0.000000			NO CALIBRATION FACTOR SPECIFIED
9	0.000000			NO CALIBRATION FACTOR SPECIFIED
10	0.000000			NO CALIBRATION FACTOR SPECIFIED
11	0.000000			NO CALIBRATION FACTOR SPECIFIED
12	0.000000			NO CALIBRATION FACTOR SPECIFIED
13	0.000000			NO CALIBRATION FACTOR SPECIFIED
14	0.000000			NO CALIBRATION FACTOR SPECIFIED
15	0.000000			NO CALIBRATION FACTOR SPECIFIED
16	0.000000			NO CALIBRATION FACTOR SPECIFIED
17	3.904950	g's	8/4/83	Entran Serial Number 10
18	3.904950	g's	8/4/83	Entran Serial Number 8
19	3.904950	g's	8/4/83	Entran Serial Number 18
20	0.000000			NO ACCELEROMETER THIS TEST
21	3.904950	g's	8/4/83	Entran Serial Number 9
22	3.904950	g's	8/4/83	Actuator Acceleration (SN-74)
23	3.904950	g's	8/4/83	Horizontal Box Frame Acc. (72)
24	3.904950	g's	8/4/83	Vertical Box Frame Acc. (77)
25	23.825001	inches	8/11/83	Actuator LVDT
26	10.000000	volts	8/11/83	Input Ramp Signal
27	0.000000			NO CALIBRATION FACTOR SPECIFIED
28	0.000000			NO CALIBRATION FACTOR SPECIFIED
29	0.000000			NO CALIBRATION FACTOR SPECIFIED
30	0.000000			NO CALIBRATION FACTOR SPECIFIED
31	0.000000			NO CALIBRATION FACTOR SPECIFIED
32	0.000000			NO CALIBRATION FACTOR SPECIFIED
33	0.000000			NO CALIBRATION FACTOR SPECIFIED
34	0.000000			NO CALIBRATION FACTOR SPECIFIED
35	0.000000			NO CALIBRATION FACTOR SPECIFIED
36	0.000000			NO CALIBRATION FACTOR SPECIFIED
37	0.000000			NO CALIBRATION FACTOR SPECIFIED
38	0.000000			NO CALIBRATION FACTOR SPECIFIED
39	0.000000			NO CALIBRATION FACTOR SPECIFIED
40	0.000000			NO CALIBRATION FACTOR SPECIFIED
41	0.000000			NO CALIBRATION FACTOR SPECIFIED
42	0.000000			NO CALIBRATION FACTOR SPECIFIED
43	0.000000			NO CALIBRATION FACTOR SPECIFIED
44	0.000000			NO CALIBRATION FACTOR SPECIFIED
45	0.000000			NO CALIBRATION FACTOR SPECIFIED
46	0.000000			NO CALIBRATION FACTOR SPECIFIED
47	0.000000			NO CALIBRATION FACTOR SPECIFIED
48	0.000000			NO CALIBRATION FACTOR SPECIFIED

```
&CALFA T=00004 IS ON CR  00023 USING 00032 BLKS R=0000
                                              10:48 AM  MON.,  9  JAN., 1984

FTN7X,Q
$FILES 0,1
      PROGRAM CALFAC

C     This program is used to establish a file of calibration factors for
C     transducers used with the 2250 measurement and control unit.

C     Each factor will correspond to one channel of analog input on the
C     2250 and will be given in the form of volts per unit of measurement.

C     The operator is asked to input the channel number, calibration factor
C     in volts per unit of measurement, the unit of measurement, the
C     date of calibration and a description of the measurement.

C     A maximum of 48 channels of calibration factors can be input.

      DIMENSION lbuf(200)          ! large output buffer

      REAL CalFactor(48)           ! array for calibration factors

      INTEGER logonlu              ! LU of log-on device

      INTEGER ios                  ! I/O status return

      INTEGER time(15)             ! buffer for time and date

      INTEGER printerlu            ! printer LU

      INTEGER start                ! starting point of terminal listing

      INTEGER end                  ! ending point of terminal listing

      INTEGER dummy                ! dummy integer variable

      CHARACTER*1 YORN             ! question response input

      CHARACTER*12 CalFile         ! namr of calibration factor file

      CHARACTER*11 Units(48)       ! calibration factor units

      CHARACTER*8 Date(48)         ! dates of calibration

      CHARACTER*32 Description(48)! measurement description

      CHARACTER*80 TestTitle       ! test descriptor phrase

      LOGICAL ex                   ! file existence checker

      DATA printerlu/6/            ! LU of 2631B printer

      logonlu=LOGLU(dummy)

      CALL LGBUF(lbuf,200)

C     operator check of system time

100   CALL FTIME(time)
```

```
        WRITE (logonlu,110) time
  110   FORMAT ('The system time and date are, ',/,15A2,/,
       1         'Is this correct (Y or N)?')
        READ (logonlu,120) YORN
  120   FORMAT (A1)
        IF (YORN.EQ.'Y') GOTO 200
C       if time is incorrect operator is asked to supply the correct time
  130   WRITE (logonlu,140)
  140   FORMAT ('Enter correct system time as',/,
       1         'HH,MM,SS,MM,DD,YYYY')
        READ (logonlu,*) itime1,itime2,itime3,itime4,itime5,itime6
        iresult=SETTM (itime1,itime2,itime3,itime4,itime5,itime6)
C       check to see that time was properly set
        IF (iresult.EQ.0) GOTO 100
        WRITE (logonlu,150)
  150   FORMAT ('Time was improperly entered')
        GOTO 130
C       operator entry of calibration factor file namr
  200   WRITE (logonlu,210)
  210   FORMAT (//,'Enter the name of the calibration factor '
       1              'file as',/,'FILNAM:SC:CR')
        READ (logonlu,220) CalFile
  220   FORMAT (A12)
C       check to see if file already exists
        iline=220
        INQUIRE (FILE=CalFile,EXIST=ex,IOSTAT=ios,ERR=9000)
        IF (ex) THEN
            WRITE (logonlu,230) CalFile
  230       FORMAT ('File ',A12,' already exists',/,
       1              'Do you want to purge the old one',/,
       2              'and replace with a new one (Y or N)?')
            READ (logonlu,120) YORN
            IF (YORN.EQ.'N') GOTO 200
C           existing file is purged if not wanted
            iline=230
            OPEN (100,FILE=CalFile,IOSTAT=ios,ERR=9000)
            iline=235
            CLOSE (100,STATUS='DELETE',IOSTAT=ios,ERR=9000)
        ENDIF
C       open file by namr in CalFile
        iline=240
        OPEN (100,FILE=CalFile,IOSTAT=ios,ERR=9000)
        WRITE (logonlu,240) CalFile
  240   FORMAT (//,'File ',A12,' created for calibration factors',//)
C       operator input of test descriptor phrase
        WRITE (logonlu,250)
  250   FORMAT (//,'Enter title for test',/,79X,'<')
```

```
      READ (logonlu,260) TestTitle
260   FORMAT (A80)

C     array clearing loop

      DO Channel=1,48
         CalFactor(Channel)=0
         Units(Channel)='           '
         Date(Channel)='        '
         Description(Channel)='NO CALIBRATION FACTOR SPECIFIED'
      END DO

C     data entry loop

400   WRITE (logonlu,405)
405   FORMAT (//,'Enter Channel Number')
      READ (logonlu,*,IOSTAT=ios,ERR=410) Channel

C     check to see if last channel

      IF ((Channel.LT.1).OR.(Channel.GT.48)) THEN
         GOTO 500
      ELSE
         GOTO 420
      ENDIF

C     branch for non-numeric entry

410   iline=406
      IF (ios.NE.494) GOTO 9000
      WRITE (logonlu,415)
415   FORMAT ('****Numeric input only',/)
      GOTO 400

420   WRITE (logonlu,425)
425   FORMAT ('Enter Calibration Factor in Volts/Unit of Measurement')
      READ (logonlu,*,IOSTAT=ios,ERR=430) CalFactor(Channel)
      GOTO 435

C     branch for non-numeric entry

430   iline=426
      IF (ios.NE.494) GOTO 9000
      WRITE (logonlu,415)
      GOTO 420

435   WRITE (logonlu,440)
440   FORMAT ('Enter Units of Measurement',/,11X,'<')
      READ (logonlu,445) Units(Channel)
445   FORMAT (A11)

      WRITE (logonlu,450)
450   FORMAT ('Enter Date of Calibration')
      READ (logonlu,455) Date(Channel)
455   FORMAT (A8)

      WRITE (logonlu,460)
460   FORMAT ('Enter Measurement Description',/,32X,'<')
      READ (logonlu,465) Description(Channel)
465   FORMAT (A32)
```

```
      GOTO 400

C     output of calibration factors for operator inspection

500   start=1
      end=12
510   CALL FTIME(time)
      WRITE (logonlu,520) CalFile,time,TestTitle
520   FORMAT (///,'Calibration factor file ',A12,/,15A2,/,A80,//,9X,
     1             'Calibration',/,9X,'Factor in',5X,'Units of',5X,
     2             'Date of',4X,'Measurement',/,'Channel  Volts/Unit  '
     3             'Measurement  Calibration  Description')

C     display first 12 channels on screen

530   WRITE (logonlu,540) (Channel,CalFactor(Channel),Units(Channel),
     1                     Date(Channel),Description(Channel),
     2                     Channel=start,end)
540   FORMAT (I4,F15.6,2X,A11,4X,A8,3X,A32)

C     ask operator if these factors are OK

      WRITE (logonlu,550)
550   FORMAT ('Are these factors, units and dates correct (Y or N)?')
      READ (logonlu,120) YORN
      IF (YORN.EQ.'Y') THEN
          start=start+12
          IF (start.GT.48) GOTO 600
          end=end+12
          GOTO 510
      ELSE
          GOTO 400
      ENDIF

C     write correct data file to disc

      iline=600
      CALL FTIME(time)
600   WRITE (100,610,IOSTAT=ios,ERR=9000) time,TestTitle
610   FORMAT ('Calibration factor file for HP-2250',/,15A2,/,A80)
      WRITE (100,620,IOSTAT=ios,ERR=9000) (Channel,
     1        CalFactor(Channel),Units(Channel),Date(Channel),
     2        Description(Channel),Channel=1,48)
620   FORMAT (I2,F10.7,X,A11,X,A8,X,A32)
      iline=620
      CLOSE (100,STATUS='KEEP',IOSTAT=ios,ERR=9000)
      WRITE (logonlu,630) CalFile
630   FORMAT (//,'Calibration factors written to file ',A12)

C     write calibration factor file to printer

      WRITE (printerlu,700) CalFile,time,TestTitle
700   FORMAT (26X,'Calibration Factor File ',A12,/,26X,15A2,/,26X,
     1        'HP-2250 Measurement and Control Unit',//,26X,A80,//,
     2        35X,'CALIBRATION',/,35X,'FACTOR IN',5X,'UNITS OF',
     3        5X,'DATE OF',/,26X,'CHANNEL  VOLTS/UNIT  MEASUREMENT  '
     4        'CALIBRATION  MEASUREMENT DECRIPTION',/)
      WRITE (printerlu,710) (Channel,CalFactor(Channel),Units(Channel),
     1        Date(Channel),Description(Channel),Channel=1,48)
710   FORMAT (I30,F15.6,2X,A11,4X,A8,3X,A32)
```

```
      GOTO 9500
C     I/O error handling routine
9000  WRITE (logonlu,9010) ios,iline
9010  FORMAT ('Error encountered = ',I6,' at iline = ',I6)
9500  WRITE (printerlu,'(1H1)')
      STOP
      END
```

APPENDIX C3

DATA ACQUISITION AND TRANSFER SOFTWARE

High-speed continuous data acquisition on the HP–2250 Measurement and Control Processor requires coordinated programs to be running concurrently on the HP–2250 and on the HP–1000 minicomputer. Most of this software was written by Hewlett-Packard and purchased with the HP–25581A Automation Library.

Operation of continuous data acquisition (CDA) on the HP–2250 and HP–1000 is as follows: An MCL (measurement and control language) program is downloaded to the HP–2250 from the HP–1000 using the MCX utility. This program clears the HP–2250 memory, downloads the binary absolute program !CDA, sets scan rate and channels to be scanned, defines type of data transfer, defines start mode, and then waits for the GRAB program to run on the HP–1000. A listing of this MCL program is included in this appendix and has file name INFILE.

The HP–1000 minicomputer had two programs in it that were waiting to receive data from the HP–2250. These programs were supplied by Hewlett-Packard and have the names &GRAB and &GRAB2. Listings are included in this appendix. To use these for data acquisition an ID segment is created for GRAB2 and GRAB is run. After GRAB completes some initial housekeeping tasks, it schedules GRAB2, sends a bus trigger to the HP–2250 and then waits for the HP–2250 to say it has data.

When the HP–2250 receives the bus trigger from GRAB, the binary absolute program is activated. It waits for an external trigger, then starts collecting data and puts it in a buffer. When the buffer is full, it asserts the EOI line on the HPIB and continues to collect data sending it to a second buffer. The HP–1000 minicomputer senses the EOI and GRAB starts again. GRAB empties the first buffer of data. It then writes this data on a disk. While the data are being written on the disk, the HP–2250 fills the second buffer and asserts EOI again. GRAB2 responds by emptying that buffer while GRAB is writing on the disk. These programs keep alternating execution to keep up with the data coming from the HP–2250.

```
INFILE T=00004 IS ON CR  00023 USING 00004 BLKS R=0000
                                        8:15 AM  TUE., 10  JAN., 1984


Documented 2250 CDA input file
     *****CAUTION -- will not run with comments

NTASKS(0)!                    clears 2250 memory
SET RESULTS OFF               turns off output to host computer
WRITE SUBROUTINE !CDA         downloads CDA subroutine from host computer
IF BT=1 ENDIF                 waits here for GRAB to run in host computer
CPACE (0,1,0)                 set channel pace mode at 1 millisecond intervals
CDA (6144,2)                  start CDA with 6144 word buffer when external
                                  trigger occurs
WPACE                         wait for pace pulse
TRANSFER                      use raw data transfer mode starting at slot 2,
AI(2,1,10)                        channel 1 for 10 channels
!                             execute
```

```
&GRAB  T=00003 IS ON CR  00019 USING 00142 BLKS R=0000
                                                 7:59 AM  TUE., 10  JAN., 1984

FTN4X,L
             Program GRAB(3,41),25581-16005 REV.2101 (830722.0758)
*            Source Code Part #:25581-18005     "           "
*            modified by F.T. 7/22/83 to check for opsys=-45

*       This program will perform psuedo class I/O on the disc and HPIB
*       via father/son scheduling scheme. Data transfer from 2250 to
*       disc is implemented with the use of global resource numbers
*       representing semaphores to each of the two routines. Each routine
*       must unlock the resource number it was currently using and then
*       attempt to lock the resource that it wants; effectively this is
*       a wait on a semaphore.  The name of the son being scheduled is
*       GRAB2.


*       Each of the two routines will manage its own track size buffer.
*       Each will attempt to read from the HPIB and then attempt
*       to write its buffer out to the disc. These routines must be
*       core resident to insure high speed operation.


*       Prior to execution of the data transfers, a data collection file
*       is created (if one does not exist) which starts and ends on a track
*       boundary.  This is to assure maximum data rate transfer.



*       the RU command is:-

*       RU,GRAB, 2250 LU, data file, [mode]

*         where namr for each file is: filename:sc:lu:type:size

*         and mode = 0  for just creating data file and not filling file.
*                  = 1  for filling file once & stopping.
*                  = -1 for wrapping around and continually filling file until
*                            a system break or HPIB timeout occurs.


**********************************************************************
**********************************************************************



             implicit integer (a-z)

             dimension    buffer(8192)      ! data transfer buffer
             dimension    buff1(4096)       ! half track buffer 1
             dimension    buff2(4096)       ! half track buffer 2
             dimension    param(14)         ! buffer for parameters to GRAB2
             dimension    data(3)           ! data collection file
             dimension    dcb(144)          ! data control block for "data"
             dimension    dcbt(144)         ! data control block for temp file
             dimension    son(3)            ! name of son program
             dimension    crstat(4,50)      ! cartridge status buffer
             dimension    tempfl(3)         ! name of temporary file to creat hole
             dimension    cmdbuf(40)        ! buffer for command string
```

```
      dimension   prmbuf(10,3)      ! buffer for namr parsing
      dimension   mesage(5)         ! buffer to send system message

      integer*4   tracks            ! number of tracks in data collection fil
      integer*4   size              ! number of records in  "        "       '
      integer*4   actsiz            ! actual size of a created file
      integer*4   tempsz            ! size of temporary file
      integer*4   lastrc            ! last record value
      integer*4   trunc             ! number of records to truncate


*   The following equivalences allow sharing of the large "buffer" array.

      equivalence  (buff1,buffer(1))
      equivalence  (buff2,buffer(4097))
      equivalence (cmdbuf,buffer(1))
      equivalence (prmbuf,buffer(101))
      equivalence   (dcbt,buffer(201))
      equivalence (crstat,buffer(501))


*   The following equivalences point into the dcb of the data collection file
*   to determine information concerning disc access to the file.

      equivalence (track1,dcb(4))        ! first track of data file
      equivalence (sector,dcb(5))        ! first sector of data file
      equivalence (sectrk,dcb(9))        ! sectors per track


*   The following equivalences map parameters needed by the son program into
*   a common data buffer.

      equivalence   (data,param(1))     ! name of data collection file
      equivalence (seccod,param(4))     ! security code of data file
      equivalence (tracks,param(5))     ! number of tracks in data file
      equivalence   (mode,param(7))     ! mode: normal(0) or wrap(1)
      equivalence    (log,param(8))     ! lu of logon device
      equivalence  (maclu,param(9))     ! lu of 2250
      equivalence (disclu,param(10))    ! lu of disc (where data file is)
      equivalence (discwr,param(11))    ! control word for exec disc write
      equivalence  (macrn,param(12))    ! 2250 resource number
      equivalence (discrn,param(13))    ! disc resource number
      equivalence  (class,param(14))    ! class number


*   Constants

      data      son   / 'GRAB2' /       ! name of son program

***************
*   Line 10   *
***************
      data      mesage/ '          '/   ! system message (HPIB timeout)



**********************************************************************
*   Determine parameters.                                            *
**********************************************************************

```

```
*    Pick up RU string and parse.

        call getst (cmdbuf,-80,len)
        call exec (22,1)                   ! do not swap out of memory

        log = loglu (dummy)

        count = 1
        do 100 k=1,3
100     call namr (prmbuf(1,k),cmdbuf,len,count)

        maclu      = prmbuf(1,1)

        data(1)    = prmbuf(1,2)
        data(2)    = prmbuf(2,2)
        data(3)    = prmbuf(3,2)
        seccod     = prmbuf(5,2)
        cart       = prmbuf(6,2)
        size       = prmbuf(8,2)

        mode       = prmbuf(1,3)


        if (maclu.eq.0 .or. data(1).eq.0) then
           write(log,'(//"Insufficient parameters."/)')
           goto 9999
        endif

        maccwd = maclu + 100B              ! set control word for binary read



*    Find correct cartridge for data collection file.

        call fstat (crstat,200,0,0,dummy) ! read cartridge status

        if (cart.eq.0) then
           cart = crstat(3,1)              ! default to top cartirdge
        endif

        do 200 k=1,50
        if ((cart .gt. 0) .and. (cart .eq. crstat(3,k))) goto 210
        if ((cart .lt. 0) .and. (cart .eq. -crstat(1,k))) then
210       disclu = crstat(1,k)             ! found correct cartridge
          goto 300
        endif
200     continue

        write (log,220) cart, cart         ! couldn't find cartridge
220      format(//"Can't find cartridge or LU "I6" ["A2"]."/)
        goto 9999


300     call opsy (opsys)                  ! determine operating system
                                           ! if L or A-series add control bits
        if ((opsys.eq.-29).or.(opsys.eq.-45)) then
           discwr = disclu + 7700B
        else
           discwr = disclu
```

```
      endif



*****************************************************************
*   Creat Data Collection File.                                 *
*****************************************************************


*   The data collection file must be a type 1 file, start on a track
*   boundary (first sector is sector 0), and must have a whole number
*   of tracks (size is evenly divisible by records/track).  If the
*   specified data collection file does not exist, one is created to
*   meet the above requirements.  If the file already exists, it is
*   checked to make sure it meets the above requirements.

      call open (dcb,err,data,1,seccod,-disclu)   ! try and open the file
      if (err.lt.0 .and. err.ne.-6) then
         write(log,'(/"Cannot open data file, FMGR error: "I4/)') err
         goto 9999
      endif

      if (err.ge.0) goto 340           ! file already exists


*   Creat a data collection file.

      write(log,'(/"Creating data collection file.  _")')

      if (size.eq.0) then              ! if specified size is 0 default
         size = -1                     ! size to rest of cartridge
      endif

*   Creat a trial file to determine starting sector & sectors/track.

      actsiz = 0
      call ecrea (dcb,err,data,size,1,seccod,-disclu,128,actsiz)
      call purge (dcb,err,data,seccod,-disclu)

      rectrk = sectrk/2                ! compute records/track
      tempsz = rectrk-(sector/2)       ! compute extra records

*   If file did not start on a track boandary, then creat temporary file.

      if (tempsz.ne.rectrk) then
         call crets (dcbt,err,1,tempfl,tempsz,1,seccod,-disclu,128)
      endif

*   Try creating the data file again.

      call ecrea (dcb,err,data,size,1,seccod,-disclu,128,actsiz)
      call purge (dcb,err,data,seccod,-disclu)

      size= ((actsiz/2 - 1)/rectrk + 1) * rectrk  ! compute whole # of track

*   Creat the real data file with the modified size.

      call ecrea (dcb,err,data,size,1,seccod,-disclu,128,actsiz)
      if(err.lt.0) then
         write(log,'(/"Cannot create file. FMGR error: "I4)') err
```

```
            goto 9999
         endif

         if (tempsz.ne.rectrk) then          ! purge temporary file
            call purge (dcbt,err,tempfl,seccod,-disclu)
         endif

         call open (dcb,err,data,1,seccod,-disclu)


*    Check status of data collection file.

340      call elocf (dcb,err,tempsz,c,d,actsiz,e,ftype,f)

         rectrk = sectrk / 2                 ! compute records/track
         size = actsiz / 2                   ! convert sectors to records
         tracks = size / rectrk              ! compute number of tracks
         trackl = trackl + tracks            ! compute last track boandary
         bsize = rectrk * 128                ! compute data buffer size

         write(log,350) data,ftype,size,bsize
350      format(/"File "3A2" [type "I1"] size: "I6
     1    " records (128 words/record)"/"Data buffer size: "I6" words"/)

         if (sector.ne.0) goto 360           ! check starting sector
         if (size.ne.(tracks*rectrk)) goto 360   !check size
         if (ftype.ne.1) then                ! check type
360         write(log,'(/"Illegal data file."/)')
            goto 9998
         endif

         if (mode.eq.0) goto 9998




***********************************************************************
*    Get ready for CDA.                                               *
***********************************************************************


*     Tell the HPIB driver all about DMA, EOI, and SRQ.

         call cnfg (maclu,1,37000B)

*    Allocate resource numbers and class number for 2250 & disc.

         call rnrq (33B, macrn, stat)
         call rnrq (30B, discrn, stat)
         if (stat.eq.4) then
            write(log,'(/"No resource number available."/)')
            goto 9999
         endif

         class = 0
         call exec (20,0,buffer,0,0,0,class)
         call exec (21,class+20000B,buffer,0)

*    Schedule GRAB2.

```

```
        write (log,'(/"Scheduling GRAB2.")')
        call exec (10,son,0,0,0,0,0,param,14)  ! schedule & pass parameters



******************************************************************
*   CDA collection.                                              *
******************************************************************


*   We're now ready to start CDA.  GRAB2 has already been scheduled and
*   is waiting to run.

        if (mode.lt.0 .and. tracks.eq.1) then    ! set track step size
           step = 1
        else
           step = 2
        endif

        ctrack = track1                     ! set track pointer
        pass = 0                            ! initiallize pass counter

        call exec (21,class+20000B,buffer,0)       ! wait for GRAB2


        write (log,'(/"Start CDA."/)')

        call trigr (maclu)                  ! wait for HPIB trigger
        goto 1010                           ! don't relock macrn


*   This is the main loop.  The loop will lock the 2250 and fill
*   the buffer.  After the buffer is full it will unlock the 2250
*   which should fire up the son.  This routine will then attempt
*   to lock the disc and write out the buffer to it.  When it
*   has finished writing to the disc it will unlock the disc
*   resource number.  The loop will terminate either when the data
*   collection file is filled (mode=1), on a system break command,
*   or an HPIB timeout condition.


1000    call rnrq (3B,macrn,stat)        ! lock 2250

        ctrack = ctrack + step              ! bump track counter
        if (ctrack.ge.trackl) then          ! check for end of file
           pass = pass + 1                  ! bump pass counter
           if (mode.eq.1) goto 2020         ! terminater, if in normal mode
           ctrack = ctrack - tracks         ! reset track counter
        endif

        if (ifbrk (dummy)) 2000,1010              ! check system break

1010    call exec (1,maccwd,buffer,bsize)         ! read a buffer from 2250
        call abreg (a,b)                          ! read length of transfer
        if (b.ne.bsize) goto 2010                 ! if not full, terminate

        call rnrq (7B,macrn,stat)      ! unlock 2250
```

```
      call rnrq (3B,discrn,stat)      ! lock disc

      call exec (2,discwr,buffer,bsize,ctrack,0)  ! write to disc
      write (log,*) ctrack               ! print track just written
      call messs (mesage,10)             ! send message to op system

      call rnrq (7B,discrn,stat)    ! unlock disc



      goto 1000                          ! go back and read the 2250 again




*************************************************************************
*   Program Termination.                                                *
*************************************************************************


*   This is where program terminates because of a system BREAK.

2000  call rnrq (3B,discrn,stat)      ! wait for GRAB2 to finish
      goto 2030


*   This is where program terminates because of incomplete buffer from 2250.

2010  write(log,'(//"Incomplete buffer read.")')
      if (ctrack.eq.track1 .and. pass.eq.0) goto 2040


*   This is where program terminates because file is filled and mode = 1.

2020  call rnrq (7B,macrn,stat)     ! let GRAB2 finish with 2250

      call exec (21,class+20000B,buffer,2)       ! wait till GRAB2 is done

      total1 = (pass*tracks) + ctrack - track1      ! total tracks by GRAB
      total2 = (buffer(1)*tracks) + buffer(2) - track1     ! total by GRAB2

      if (total2.gt.total1) then
         ctrack = buffer(2)              ! take largest track pointer
      else
         pass = buffer(1)                ! take smallest pass counter
      endif

      ctrack = ctrack - 1                ! modify pointer to show last track

2030  if (ctrack.le.track1) then         ! retrack last track if wrap around
         ctrack = ctrack + tracks
      endif

2040  lastrc = (ctrack-track1) * rectrk ! convert to records
      if (pass.eq.0) then                ! truncate file is necessary
         trunc = size - lastrc
         size = lastrc
      else
         trunc = 0
      endif
```

```
      write (log,2050) pass, size
2050  format (//I6" passes through file.  "I9" records written."/)


      if (lastrc.ne.size) then          ! shuffle data into
         step = lastrc / rectrk         !   chronological order.

         do 3000 sectr = 0, sectrk/2, sectrk/2  ! two passes, half tracks.
         count = 0                      ! initiallize counter
         strack = track1                ! set starting track to track 1

2075     ctrack = strack
         call exec (1,disclu,buff1,bsize/2,strack,sectr)     ! save start tr

2080     ptrack = ctrack                ! set previous track to current trk
         ctrack = ctrack + step         ! bump current track
         if (ctrack.ge.track1) then
            ctrack = ctrack - tracks
         endif

         count = count + 1

         if (ctrack.ne.strack) then
            call exec (1,disclu,buff2,bsize/2,ctrack,sectr)
            call exec (2,discwr,buff2,bsize/2,ptrack,sectr)
            goto 2080
         endif

         call exec (2,discwr,buff1,bsize/2,ptrack,sectr) !store start trk
         strack = strack + 1            ! set new starting track
         if (count.ne.tracks) goto 2075
3000     continue
      endif


*   Clean-up and terminate.

      call exec (6,son)              ! terminate the GRAB2

      call rnrq (40B,discrn,stat)    ! de-allocate resource numbers
      call rnrq (40B,macrn,stat)
      call exec (21,class+100000B,buffer,0)      ! de-allocate class #

      call open (dcb,err,data,0,datasc,datacr)     ! open file exclusively
9998  call eclos (dcb,err,trunc)                   ! close (and truncate)

9999  end
```

&GRAB2 T=00004 IS ON CR 00019 USING 00024 BLKS R=0000
8:02 AM TUE., 10 JAN., 1984

```
FTN4X,L
        Program GRAB2(3,41),25581-16006 REV.2101 <830722.0817>
*       Source Code Part #: 25581-18006     "           "

*   This program is scheduled by GRAB to execute simultaneously with
*   GRAB and collect CDA data from the 2250 and transfer it to a disk.
*   While GRAB is reading data from the 2250, GRAB2 is writting it's
*   previously read data to the disk, and vica verca.  This program never
*   terminates itself, but rather is terminated by GRAB.


*************************************************************************
*************************************************************************


        implicit integer (a-z)


        dimension   buffer(8192)     ! data transfer buffer
        dimension   dcb(144)         ! data control block for "data"
        dimension   data(3)          ! data collectin file
        dimension   param(14)        ! buffer to hold parmeters from father

        integer*4   tracks           ! number of tracks in data collection fil



*   The following equivalences point into the dcb of the data collection file
*   to determine information concerning disc access to the file.

        equivalence (track1,dcb(4))        ! first track of data file
        equivalence (sectrk,dcb(9))        ! sectors per track


*   the following equivalences map paramerters sent from the father program
*   into a common data buffer.

        equivalence (data,param(1))        ! name of data collection file
        equivalence (seccod,param(4))      ! security code of data file
        equivalence (tracks,param(5))      ! number of tracks in data file
        equivalence   (mode,param(7))      ! mode: normal(0) or wrap(1)
        equivalence    (log,param(8))      ! lu of logon device
        equivalence  (maclu,param(9))      ! lu of 2250
        equivalence (disclu,param(10))     ! lu of disc (where data file is)
        equivalence (discwr,param(11))     ! control word for exec dixk write
        equivalence  (macrn,param(12))     ! 2250 resource number
        equivalence (discrn,param(13))     ! disc resource number
        equivalence  (class,param(14))     ! class number




*************************************************************************
*   Get ready for CDA.                                                  *
*************************************************************************
```

```
      call exec (14,1,param,14)        ! fetch parameters from father

      call exec (22,1)                 ! do not swap out of memory

      call open (dcb,err,data,1,seccod,-disclu)

      maccwd = maclu + 100B            ! set control word for binary read
      bsize = (sectrk/2) * 128         ! compute data buffer size
      track1 = track1 + tracks         ! compute last track boundary



*********************************************************************
*   CDA Collection.                                                 *
*********************************************************************


      if (mode.lt.0 .and. tracks.eq.1) then
         step = 1
      else
         step = 2
      endif

         ctrack = track1 - 1           ! initialize track pointer

      pass = 0                         ! initialize pass counter

      call exec (20,0,buffer,0,0,0,class)          ! signal GRAB


*   This is the main loop.  The loop will lock the 2250 and fill
*   the buffer. after the buffer is full it will unlock the 2250,
*   lock the disk and write it's buffer to the disc.  Then it
*   will repeat the sequence.

1000  call rnrq (3B,macrn,stat)        ! lock 2250

      ctrack = ctrack + step           ! bumb track counter
      if (ctrack.ge.track1) then       ! check for end of file
         pass = pass + 1               ! bump pass counter
         if (mode.eq.1) goto 2000      ! suspend, if in normal mode
         ctrack = ctrack - tracks      ! reset track counter
      endif

      call exec (1,maccwd,buffer,bsize)            ! read a buffer from 2250
      call abreg (a,b)                             ! read length of transfer
      if (b.ne.bsize) goto 2000                    ! if not full, suspend

      call rnrq (7B,macrn,stat)        ! unlock 2250



      call rnrq (3B,discrn,stat)       ! lock disc

      call exec (2,discwr,buffer,bsize,ctrack,0)  ! write to disc
      write (log,'(10X""I4"")') ctrack   ! write track just written

      call rnrq (7B,discrn,stat)       ! unlock disc

```

```

      goto 1000                             ! go back and read 2250 agian.




***********************************************************************
*   Program Termination.                                              *
***********************************************************************


2000  call rnrq (7B,macrn,stat)             ! unlock disc so father can get it

      buffer(1) = pass
      buffer(2) = ctrack
      call exec (20,0,buffer,2,0,0,class)          ! send track info to GRAB

      call eclos (dcb)
      call exec (7)                         ! suspend until father kills GRAB2

      end
```

APPENDIX C4
ZERO CORRECTION FACTOR MEASUREMENT

The stringent timing requirements placed on the HP–2250 and minicomputer during high-speed data acquisition require that zero offset correction for each channel be done after the test. To correct for zero offsets, a measurement of these offsets can be done just before a test and assumed not to change during the test. The program ZEROC was written to assist in this procedure. This program is run just before a test. It instructs the operator to short the input to each channel to be used successively, then measures the voltage at that input. These voltages are then written on a disk file for zero correction when the data is processed later. A sample output and a documented listing of the program follows.

Zero Correction Factors for HP-2250 Measurement and Control Unit

1:14 PM TUE., 1 NOV., 1983

Channel	Zero Correction Factor
1	10.0000000
2	10.0000000
3	10.0000000
4	10.0000000
5	10.0000000
6	10.0000000
7	10.0000000
8	10.0000000
9	10.0000000
10	10.0000000
11	10.0000000
12	10.0000000
13	10.0000000
14	10.0000000
15	10.0000000
16	10.0000000
17	-.0012500
18	-.0012500
19	-.0012500
20	-.0012500
21	-.0012500
22	-.0012500
23	-.0012500
24	-.0012500
25	-.0012500
26	-.0012500
27	10.0000000
28	10.0000000
29	10.0000000
30	10.0000000
31	10.0000000
32	10.0000000
33	10.0000000
34	10.0000000
35	10.0000000
36	10.0000000
37	10.0000000
38	10.0000000
39	10.0000000
40	10.0000000
41	10.0000000
42	10.0000000
43	10.0000000
44	10.0000000
45	10.0000000
46	10.0000000
47	10.0000000
48	10.0000000

```
&ZERO  T=00004 IS ON CR  00023 USING 00037 BLKS R=0000
                                             7:50 AM  TUE., 10  JAN., 1984

FTN7X,Q
$FILES 0,1
      PROGRAM ZEROC

C     This program is used to develop a file of zero correction factors
C     for each channel to be used with continuous data acquisition (CDA)
C     on the HP-2250.

C     Since CDA does not perform zero correction on the readings that it
C     takes, it is necessary to determine the correct zero correction
C     factor for each channel and then subtract it from each reading after
C     CDA is terminated.  This program should be used immediately prior
C     to the start of CDA.  The data file it generates should be kept with
C     the CDA file to provide zero correction for the CDA data file.

C     This program takes 2 voltage readings from each channel to be used.
C     First the AIC (Analog Input Corrected) function is used to obtain
C     zero corrected readings.  The TRANSFER AI funtion is then used to
C     obtain uncorrected readings.  The corrected reading is then subtracted
C     from the uncorrected reading which yields the zero correction factor.
C     The numbers returned by the 2250 are converted to voltage by the
C     VOLTS function.

C     It is important that the input to each channel be stable during the
C     time that the voltage readings are being taken.  It is advisable to
C     short the inputs or tie them to a fixed voltage during this calibra-
C     tion cycle.


      DIMENSION ZeroCorrection(48)          ! zero correction factors

      CHARACTER*12 ZeroFile                 ! zero correction factor datafile

      CHARACTER*1 YORN                      ! question response input

      INTEGER time(15)                      ! buffer for time and date

      INTEGER Word1C,Word2C                 ! return data words for readings
                                            ! with corrected values

      INTEGER Word1,Word2                   ! return data words for readings
                                            ! with uncorrected values

      INTEGER Slot                          ! beginning slot of ADC

      INTEGER First,Last                    ! range of channel numbers
                                            ! to be calibrated

      INTEGER Channel                       ! current channel number

      INTEGER logonlu                       ! log-on terminal LU

      INTEGER printerlu                     ! printer LU

      INTEGER ios                           ! I/O status return

```

```
      INTEGER ierr                           ! 2250 error return

      INTEGER dummy                          ! dummy integer variable

      LOGICAL ex                             ! file existence checker

      DATA maclu/10/                         ! LU of 2250 measurement
                                             ! and control processor

      DATA printerlu/6/                      ! LU of 2631B printer

      logonlu=LOGLU(dummy)                   ! LU of log on device

C     operator input of data file name

100   WRITE (logonlu,'("Enter data file name as",/,"FILNAM:SC:CR")')
      READ (logonlu,110) ZeroFile
110   FORMAT (A12)

C     check to see if file already exists

      iline=110
      INQUIRE(FILE=ZeroFile,EXIST=ex,IOSTAT=ios,ERR=9000)
      IF (ex) THEN
          WRITE (logonlu,120) ZeroFile
120       FORMAT ('File ',A12,' already exists',/,
     1                'Do you want to purge the old one',/,
     2                'and replace with a new one (Y or N)?')
          READ (logonlu,130) YORN
130       FORMAT (A1)
          IF(YORN.eq.'Y') THEN
              iline=130
              OPEN(100,FILE=ZeroFile,IOSTAT=ios,ERR=9000)
                     iline=135
              CLOSE(100,STATUS='DELETE',IOSTAT=ios,ERR=9000)
              GOTO 200
          ELSE
              goto 100
          ENDIF
      ENDIF

C     open file by namr contained in ZeroFile

200   iline=200
      OPEN (100,FILE=ZeroFile,IOSTAT=ios,ERR=9000)

      WRITE(logonlu,210) ZeroFile
210   FORMAT('File ',A12,' created for zero correction factors',//)

C     operator entry of range of channels to be calibrated

      WRITE(logonlu,300)
300   FORMAT ('Enter range of channel numbers to be ',
     1            'calibrated',/,'first,last')
310   READ(logonlu,*) First,Last
      IF(First.GT.Last) THEN
          WRITE(logonlu,320)
320       FORMAT ('First channel number was greater than last',
     1                ' channel number, reenter',/,'first,last')
          GOTO 310
```

```
        ENDIF

C       calculation of beginning slot and channel number

        IF(First.GT.16) THEN
            Channel=First-16
            Slot=2
        ELSE
            Channel=First
            Slot=1
        ENDIF

C       instruct 2250 to calibrate slots 1 and 2

        inline=400
        WRITE(maclu,400,IOSTAT=ios,ERR=9000)
  400   FORMAT('CLB(1);CLB(2)!')
        iline=405
        READ(maclu,IOSTAT=ios,ERR=420) ierr
  410   IF (ierr.NE.496) THEN
            IF (ierr.NE.0) GOTO 8000
        ENDIF
        GOTO 430
  420   IF (ios.NE.496) GOTO 9000
        GOTO 410

C       loop to initialize zero correction factors, 9.9999999 is stored in
C       each to indicate that it has not been calibrated

  430   DO I=1,48
            ZeroCorrection(I)=9.9999999
        END DO

C       loop to determine zero correction factors one channel at a time

        DO I=First,Last

C           tell operator to short input to current channel

            WRITE (logonlu,500) I
  500       FORMAT (/,'Short input to channel ',I2)      ! input to channel
            READ (logonlu,'(I2)') dummy

C           read current channel with zero correction

            iline=510
            WRITE (maclu,510,IOSTAT=ios,ERR=9000) Slot,Channel
  510       FORMAT ('AIC(',I1,',',I2,',1)!')
            iline=515
            READ (maclu,IOSTAT=ios,ERR=530) ierr,Word1C,Word2C
  520       IF (ierr.NE.496) THEN
                IF (ierr.NE.0) GOTO 8000
            ENDIF
            GOTO 540
  530       IF (ios.NE.496) GOTO 8000
            GOTO 520

C           read current channel without zero correction

  540       iline=540
```

```
            WRITE (maclu,550,IOSTAT=ios,ERR=9000) Slot,Channel
  550       FORMAT ('TRANSFER AI(',I1,',',I2,',1)!')
            iline=545
            READ (maclu,IOSTAT=ios,ERR=570) ierr,Word1,Word2
  560       IF (ierr.NE.496) THEN
                IF (ierr.NE.0) GOTO 8000
            ENDIF
            GOTO 580
  570       IF(ios.NE.496) GOTO 9000
            GOTO 560

C           calculate zero correction factor

  580       ZeroCorrection(I)=VOLTS(Word1,Word2)-VOLTS(Word1C,Word2C)

C           write voltages and correction factor to operator display

            WRITE (logonlu,590) VOLTS(Word1,Word2),VOLTS(Word1C,Word2C)
  590       FORMAT ('Uncorrected voltage = ',F10.6,5X,'Corrected '
     1              'voltage = ',F10.6)
            WRITE (logonlu,600) I,ZeroCorrection(I)
  600       FORMAT ('Channel',I3,' zero correction factor =',F10.6)


C           increment channel number and slot number if necessary

            Channel=Channel+1
            IF ((Slot.EQ.1).AND.(Channel.GT.16)) THEN
                Slot=2
                Channel=1
            ENDIF

        END DO

C       write title and time to data file

        WRITE (100,'("Zero correction factor file for CDA")')
        CALL FTIME (time)
        WRITE (100,700) time
  700   FORMAT (15A2)

C       write data to data file

        WRITE (100,710) (I,ZeroCorrection(I),I=1,48)
  710   FORMAT (I2,F10.7)

        iline=715
        CLOSE(100,IOSTAT=ios,ERR=9000,STATUS='KEEP')

C       print out data file to printer

        WRITE (printerlu,800) time
  800   FORMAT (34X,'Zero Correction Factors for HP-2250 Measurement and '
     1             'Control Unit',//,50X,15A2,//,49X,'Channel',
     2             '      Zero Correction Factor')
        WRITE (printerlu,810) (I,ZeroCorrection(I),I=1,48)
  810   FORMAT (I53,F23.7)
        WRITE (printerlu,820)
  820   FORMAT(1H1)
        GOTO 9500
```

```

C     2250 I/O error handling routine

8000  WRITE(logonlu,8010) ios,iline
8010  FORMAT('2250 error encountered =',I6,' at iline = ',I4)
      GOTO 9500

C     disc read/write error handling routine

9000  WRITE(logonlu,9010) ios,iline
9010  FORMAT('Error encountered =',I6,' at iline - ',I4)

9500  STOP
      END


C     This function takes two words of raw data input from the
C     2250 and converts them to one voltage


      REAL FUNCTION VOLTS(Word1,Word2)

      INTEGER Word1,Word2     ! words to be converted

      IF (IAND(200B,Word2).EQ.0) THEN
          M=IAND(7B,Word2)
          N=IAND(17B,ISHFT(Word2,-3))
          VOLTS=((Word1*256.0)+ISHFT(Word2,-8))*(0.5**(N+1))*(0.1**M)
      ELSE
          VOLTS=-2.0E-9
      ENDIF
      RETURN
      END
```

APPENDIX C5
CHECKLIST AND PROCEDURE FOR TESTING

Checklist

1. All accelerometer inputs should be plugged in as:

Channel #17-Entran Accel. R1-1
Channel #18-Entran Accel. R1-2
Channel #19-Entran Accel. R1-3
Channel #20-Entran Accel. R1-4
Channel #21-Entran Accel. R1-5
Channel #22-Actuator Accel. R1-6
Channel #23-Box Horiz. Accel. R1-7
Channel #24-Box Vert. Accel. R1-8
Channel #25-LVDT from servo-controller front panel
Channel #26-Ramped Sine wave from Mult out

2. Clock for digital timers.

Set HP-3325 synthesizer/function generator to:
1000 hertz
5.0 volts (p-p)
2.5 volts d-c offset
Rear output
Connect output (SIG) to SW-IN.
SW-OUT then connects to R1-24.
This sends a 1-kHz signal to the digital timers during the test.

3. Input signal

Set exact 605 function generator to:
5.0 hertz
10.0 volts
NO OFFSET
Output WG-1 to MULT1-IN.
DAC-OUT (Ramp out) to KH1-I
KH1-0 to MULT2-IN
MULT-OUT to SPAN-2

Set Krohn-Hite filter to:
1 hertz
Low-Pass
Max Flat

This sends the INPUT SIGNAL to the servo-controller during the test.

4. HP–85
a. Insert floppy disk in LEFT drive.
b. Execute LOAD "RAMP" cr
c. execute RUN
d. On the HP–85 CRT, answer question and ENTER the number of seconds for the test duration (a 60-second test duration gives a 50-second full amplitude, a 5-second rise time and a 5-second decay.)

Test Procedure

1. Set the Date/Time Group on the HP–1000 – TM,HH,MM,SS,MM,DD,YYYY cr

2. EXECUTE the following commands

FMGR:TO,8,50000	cr	plotter timeout – 500 seconds
FMGR:TO,10,10000	cr	2250 timeout – 100 seconds
FMGR:RP,GRAB2	cr	GRAB2 ID segment

3. Create a calibration factor file for the test. The best way is:

a. Edit an existing file.

FMGR:EDIT,CO815:FT:23	cr	Edit LINES 2 & 3 to show correct date of test. Make any other necessary changes.
/EC,CO820:FT:23	cr	This creates a new cal-fac file for a test on August 20,SC=FT, and located on cartridge #23

or

b. Run the program CALFAC and answer the questions carefully.

4. Within 10 minutes of running the test, run the 'zero correction factor' program:

Put printer on-line if it isn't already
FMGR:ZEROC cr
Supply file name in this form only
ZO820:FT:23
Remove all inputs to channels 17-26
Short channels as instructed
Plug all inputs back into #17-26 (as shown on checklist 1)

5. Make certain everything is plugged in.

6. Find out how long the test will be.

7. Create a data file on the HP-1000 as follows:
For a 60-s test
a. Multiply test length by 15.63 (60 × 15.63 = 937.80)
b. Round up to 938
c. Select file name TO820:FT:29
d. Execute the command
FMGR:GRAB,10,TO820:FT:29::938,0 cr

e. CRT will display info that a file has been created, and that it is longer than you specified.

8. a. Run the HP-85 program.
   b. Enter test duration.

9. Balance the strain gage amplifiers.

10. Reset the digital timers

11. Start the bearing pressure and recovery pumps.

12. Zero the ERROR on the servo controller.

13. Start the hydraulic Power Supply – TURN ON THE COOLING WATER!!!

14. Adjust Set Point for 0 volts on the digital indicator.

15. Turn up span to 668 for 0.6 *g* or 557 for 0.5 *g*.

16. Execute the following commands on the HP-1000:

    FMGR:MCX,10,INFILE cr
    Some information will be displayed on the terminal with a few pauses between. When it is complete, MCX: will be displayed. When this occurs, enter Q cr, and the 1000 will return to FMGR.

17. Execute the following on the HP-1000:

    FMGR:GRAB,10,TO815:FT:29,1 cr
    After this command is executed, you have approximately 100 seconds to begin the test, or there will be a time-out.
    Once this command is executed, two messages will be displayed on the screen:
    SCHEDULING GRAB2 and START CDA
    When this occurs, the HP-2250 is ready to take data.

18. Start the test by pressing either K1,K2,K3, or K4 on the HP-85. The test will run until completion and the only way to stop it is:
    a. Turn down SPAN on the servo controller.
    b. Turn off pumps.

19. Watch the terminal every 5 seconds or so; a number will be written on the screen in the lower left corner. If this doesn't occur, you are not taking data. The numbers are not listed in a column but are written over each other. These are the track numbers on the disk that have been filled with data.

20. TEST COMPLETE!!!!!

APPENDIX C6
DATA PROCESSING SOFTWARE

The data analysis for these tests was mainly carried out on the Cyber computer system using the Dams Branch displacement dynamics program. Selected data channels from each test were transferred to the Cyber for analysis there. Having no direct communication path to the Cyber, the data was transferred indirectly. The raw data for a selected channel stored on disk were first converted to actual values of acceleration by the program CONVERT (listing follows) and written on a new file on disk. This file was then copied from the disk to the minicartridge on the HP-2645 terminal. The HP-2645 terminal was then disconnected from the HP-1000 and connected to the Cyber. After logging onto the Cyber and creating a file for the data, text mode was entered and the tape containing the data was read. The data were now on the Cyber and ready for analysis. All of the data were not transferred because this process consumed about 20 minutes per channel of data transferred.

In addition to analysis by the Cyber, more immediate results were obtained with plots using the HP-1000. Programs were written to provide plots on the HP-1310 graphics CRT display and as hardcopy plots on the HP-9872 plotter. Because of the size of the program, it was necessary to segment it and to restrict the amount of data to be plotted. The hardcopy plots were limited to 15 seconds of data on each of three channels. The CRT plots were restricted by HP-1350 graphics translator hardware limitations. The CRT plots were limited to 15 total seconds of data for one to three channels (i.e., 15 seconds of data for one channel, 7.5 seconds of data for each of two channels, or 5 seconds of data for each of three channels).

Program PLOT is the main segment that calls segment PLOT1. PLOT1 converts the raw data to usable form, then calls segment CRTPL for plots on the HP-1310 graphics display or segment HRDPL for hardcopy plots using the HP-9872 plotter. A sample plot is shown on figure C-3 and program listings follow.

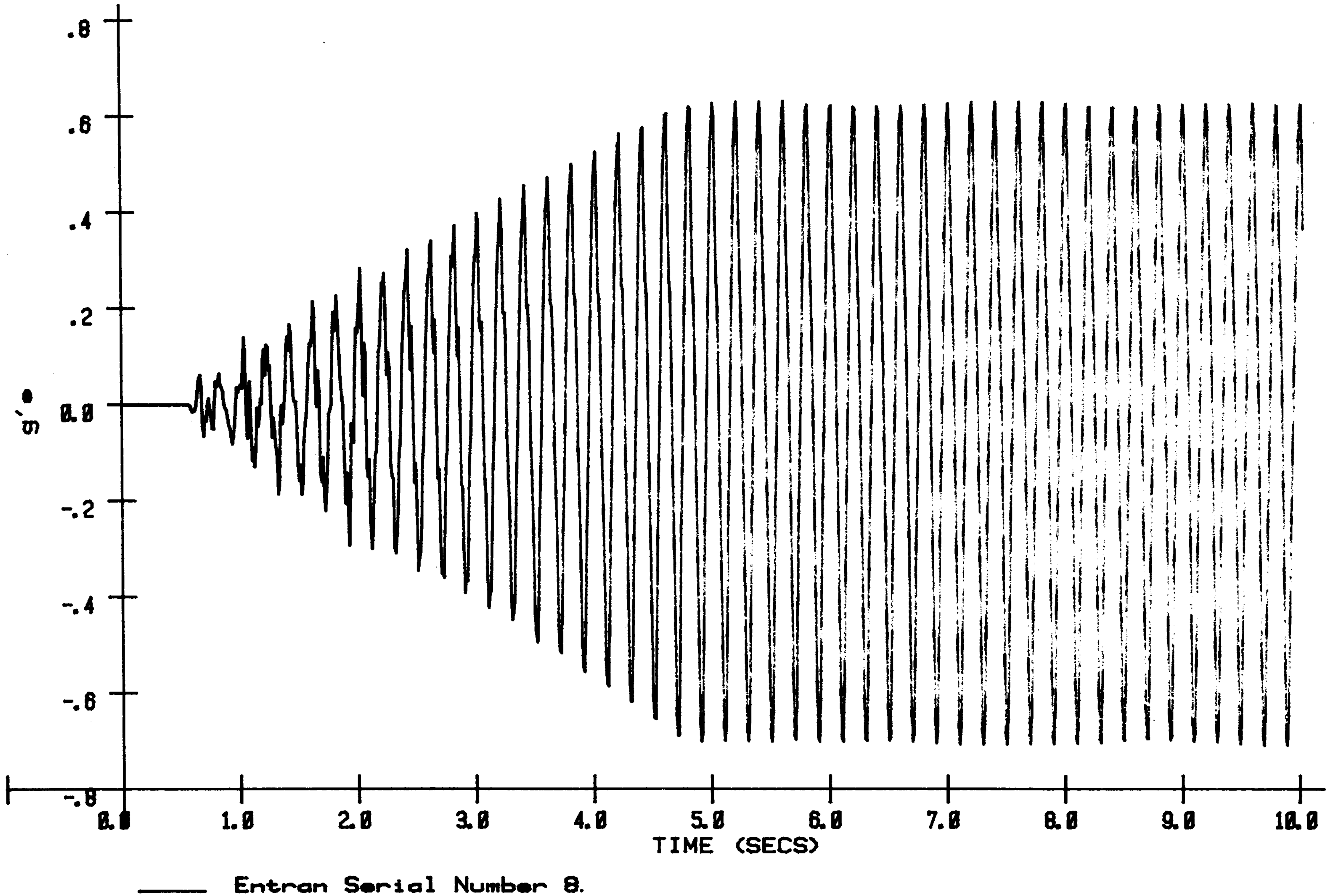


Figure C-3. – Acceleration versus time plot.

```
FTN7X,Q
$FILES 0,4
      PROGRAM CONVERT
C     This is the data conversion program for the Rockfill Research
C     Project DB-31.  Raw data from a test is used by this program
C     to generate a file that can be transferred to the Cyber for
C     analysis.  The data file generated contains the output from one
C     data channel.  Its form is one data point after another in 8F9.6
C     format with each line of data followed by an I7 line number.
C     Three data files are used in this program:
C          DataFile - file where test data is stored.  Written by
C                     CDA programs GRAB and GRAB2 in raw two-word
C                     integer format as taken from the 2250 using
C                     the TRANSFER AI command.  The data is stored
C                     in binary format.  Each data record is 256
C                     words long.
C          ZeroFile - a file of zero correction factors generated
C                     by program ZEROC.  Since the 2250 does not
C                     perform zero offset compensation during CDA
C                     it is done after the test.  Data from the
C                     DataFile is first converted to voltage by
C                     subroutine VOLTS.  The zero correction factor
C                     for the corresponding channel is then sub-
C                     tracted from this voltage yielding the
C                     corrected value.
C          CalFile -  a file containing a calibration factor for
C                     each channel.  These values are used to
C                     scale the zero-corrected data values into
C                     engineering units.  This file also contains
C                     the units of measurement, the date of cali-
C                     bration and a description of the measurement
C                     for each channel.
C     The raw data is taken from DataFile and zero corrected using ZeroFile,
C     then scaled using CalFile.  The data is written to TransFile.
      INTEGER dummy                  ! dummy integer variable
      INTEGER iline                  ! disc read error locator
      INTEGER ios                    ! I/O status return
      INTEGER ChanNum                ! channel to be transferred
      INTEGER NumPoints              ! number of points to be put in file
      INTEGER dbuffer(128)           ! buffer for record of test data
      INTEGER record                 ! record number to be read from
      INTEGER*4 bufferpointer        ! reading position pointer
      INTEGER logonlu                ! LU of log-on device
      INTEGER FileLength             ! length of data file
```

```
      INTEGER StartChan             ! starting channel number
      DIMENSION Data(8)             ! array to hold converted and scaled data
      DIMENSION ZeroDate(15)        ! date and time of zero correction factors
      DIMENSION CalFactor(48)       ! channel calibration factors
      DIMENSION ZeroCorrection(48)! channel zero correction factors
      DIMENSION CalDate(15)         ! date and time of transducer calibration
      DIMENSION lbuf(200)           ! large output print buffer
      LOGICAL ex                    ! file existence inquiry return
      CHARACTER*1 YORN              ! question response variable
      CHARACTER*12 DataFile         ! string for data file name
      CHARACTER*12 CalFile          ! string for cal factor file name
      CHARACTER*12 ZeroFile         ! string for zero factor file name
      CHARACTER*12 TransFile        ! string for transfer file name
      CHARACTER*11 Units(48)        ! channel units of measurement
      CHARACTER*32 Description(48)! measurement description for each channel
      CHARACTER*80 TestTitle        ! string for test title
      CALL LGBUF (lbuf,200)
      logonlu=LOGLU(dummy)
C     operator entry of data file name
100   WRITE (logonlu,105) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
105   FORMAT (4A1)
110   WRITE (logonlu,115)
115   FORMAT ('Enter data file name as',/,'FILNAM:SC:CR')
      READ (logonlu,120) DataFile
120   FORMAT (A12)
C     check to see if file exists
      iline=121
      INQUIRE (FILE=DataFile,IOSTAT=ios,ERR=9000,EXIST=ex,
     1          MAXREC=FileLength)
      IF (ex) GOTO 130
      WRITE (logonlu,125) DataFile
125   FORMAT (//,'File ',A12,' does not exist'//)
      GOTO 110
C     operator entry of zero correction factor file name
130   WRITE (logonlu,105) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
135   WRITE (logonlu,140)
```

```
 140  FORMAT ('Enter name of zero correction factor data file as',/,
     1        'FILNAM:SC:CR')
      READ (logonlu,145) ZeroFile
 145  FORMAT (A12)

C     check to see if ZeroFile exists

      iline=146
      INQUIRE (FILE=ZeroFile,IOSTAT=ios,ERR=9000,EXIST=ex)
      IF (ex) GOTO 150
      WRITE (logonlu,125) ZeroFile
      GOTO 135

C     open zero correction factor data file

 150  iline=150
      OPEN (200,FILE=ZeroFile,IOSTAT=ios,ERR=9000,STATUS='OLD')

C     read zero correction factor data file and close it

      READ(200,200) ZeroDate
 200  FORMAT (/,15A2)
      READ (200,205) (ZeroCorrection(I),I=1,48)
 205  FORMAT (2X,F10.7)

      iline=206
      CLOSE (200,IOSTAT=ios,ERR=9000,STATUS='KEEP')

C     output of zero correction factor data to operator display

      WRITE (logonlu,105) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
      WRITE (logonlu,210) ZeroFile,ZeroDate
 210  FORMAT ('Zero Correction Factor File ',A12,
     1        ' read as follows',/,15A2,//,9X,'Correction',
     2        17X,'Correction',17X,'Correction',/,'Channel    ',
     3        'Factor',9X,'Channel    Factor',9X,
     4        'Channel    Factor')
      WRITE (logonlu,215)(I,ZeroCorrection(I),I+16,
     1        ZeroCorrection(I+16),I+32,ZeroCorrection(I+32),I=1,16)
 215  FORMAT (I4,F14.7,I12,F14.7,I12,F14.7)
      READ (logonlu,220) YORN
 220  FORMAT (A1)

C     operator entry of calibration factor file name

      WRITE (logonlu,105) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
 225  WRITE (logonlu,230)
 230  FORMAT ('Enter name of calibration factor data file name as',
     1        /,'FILNAM:SC:CR')
      READ (logonlu,145) CalFile

C     check to see if CalFile exists

      iline=231
      INQUIRE (FILE=CalFile,IOSTAT=ios,ERR=9000,EXIST=ex)
      IF (ex) GOTO 235
      WRITE (logonlu,125) CalFile
      GOTO 225

C     open calibration factor data file
```

```
235   iline=235
      OPEN (300,FILE=CalFile,IOSTAT=ios,ERR=9000,STATUS='OLD')

C     read calibration factor data file and close it

      READ (300,240) CalDate,TestTitle
240   FORMAT (/,15A2,/,A80)
      READ (300,245)(CalFactor(I),Units(I),Description(I),I=1,48)
245   FORMAT (2X,F10.7,X,A11,10X,A32)

      iline=246
      CLOSE (300,IOSTAT=ios,ERR=9000,STATUS='KEEP')

C     output of calibration factor data to operator display

      DO start=1,33,16
          WRITE (logonlu,105) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
          WRITE (logonlu,250) CalFile,CalDate,TestTitle,(I,CalFactor(I),
     1           Units(I),Description(I),I=start,start+15)
250       FORMAT ('Calibration Factor File ',A12,5X,15A2,//,A80,
     1           'Channel  Calibration Factor  Units        Description'/
     2           (I4,F21.7,4X,A11,2X,A32))
          READ (logonlu,220) YORN
      END DO

C     operator entry of starting channel number

      WRITE (logonlu,105) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
      WRITE (logonlu,300)
300   FORMAT ('Enter beginning channel number')
      READ (logonlu,*) StartChan

C     operator entry of channel to be transferred

305   WRITE (logonlu,310)
310   FORMAT (//,'Enter number of channel to be transferred')
      READ (logonlu,*) ChanNum

C     operator entry and creation of data file name

400   WRITE (logonlu,105) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
      WRITE (logonlu,405) ChanNum
405   FORMAT ('Enter name of channel',I3,' data transfer file as',/,
     1         'FILNAM:SC:CR')
      READ (logonlu,145) TransFile

C     check to see if this file already exists

      iline=406
      INQUIRE (FILE=TransFile,IOSTAT=ios,ERR=9000,EXIST=ex)
      IF (ex) THEN
          WRITE (logonlu,410)
410       FORMAT (///,'File already exists, do you want to purge the '
     1             'existing file',/,'and create a new one by the same'
     2             ' name (Y or N)?')
          READ (logonlu,220) YORN
          IF (YORN.EQ.'N') GOTO 400
          iline=411
          OPEN (400,FILE=TransFile,IOSTAT=ios,ERR=9000,STATUS='OLD')
```

```
          iline=412
          CLOSE (400,IOSTAT=ios,ERR=9000,STATUS='DELETE')
      ENDIF
      iline=413
      OPEN (400,FILE=TransFile,IOSTAT=ios,ERR=9000,Status='NEW')

C     requested data is brought off of disc

500   record=1
      iline=501
      OPEN (100,FILE=DataFile,IOSTAT=ios,ERR=9000,STATUS='OLD',
     1       ACCESS='DIRECT',RECL=256)
      READ (100,REC=record,IOSTAT=ios,ERR=9000)(dbuffer(I),I=1,128)

C     this data is then put into array ChData for ease of transfer

      NumPoints=INT(FileLength*.7997)*8
      WRITE (logonlu,1105) NumPoints
1105  FORMAT ('Number of points =',I5)
      bufferpointer=(ChanNum-StartChan)*2+1

C     loop to load array with zero-corrected and scaled data

      DO Point=1,NumPoints,8
          DO N=1,8
              Data(N)=(VOLTS(dbuffer(bufferpointer),
     1                        dbuffer(bufferpointer+1))
     2                        -ZeroCorrection(ChanNum))/
     3                        CalFactor(ChanNum)
              bufferpointer=bufferpointer+20

C             check to see if past end of buffer

              IF (bufferpointer.GT.128) THEN

C                 if so read next record and reset bufferpointer

                  record=record+1
                  iline=502
                  READ (100,Rec=record,IOSTAT=ios,ERR=9000) (dbuffer(I),
     1                  I=1,128)
                  bufferpointer=bufferpointer-128
              ENDIF
          END DO
          WRITE (400,505) (Data(I),I=1,8),(Point+7)/8
505       FORMAT (8F9.6,I7)
          WRITE (logonlu,510) (Point+7)/8
510       FORMAT ('Line',I4,' written to file')
      END DO
      iline=515
515   CLOSE(400,IOSTAT=ios,ERR=9000,STATUS='KEEP')
      WRITE (logonlu,520) NumPoints,Point/8,TransFile
520   FORMAT (I5,' points in',I4,' lines written to file ',A12)

C     check for additional transfer files to be made

      WRITE (logonlu,525)
525   FORMAT (//,'Do you want to make another transfer file',/,
     1          'from this data file (Y or N)?')
      READ (logonlu,220) YORN
```

```
      IF (YORN.EQ.'Y') THEN
          WRITE (logonlu,105) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
          GOTO 305
      ENDIF
      iline=526
      CLOSE (100,IOSTAT=ios,ERR=9000,STATUS='KEEP')
      WRITE (logonlu,530)
530   FORMAT (//,'Do you want to make another transfer file',/,
     1            'from another data file (Y of N)?')
      READ (logonlu,220) YORN
      IF (YORN.EQ.'Y') GOTO 100
      GOTO 9500

C     disc access error handling routine

9000  WRITE (logonlu,9010) ios,iline
9010  FORMAT (//'****Disc access error ',I4,' in program DATS1',
     1          ' at iline=',I5)
9500  STOP
      END

C     This function takes two words of raw data input from the
C     2250 and converts them to one voltage

      REAL FUNCTION VOLTS(Word1,Word2)

      INTEGER Word1,Word2      ! words to be converted

      IF (IAND(200B,Word2).EQ.0) THEN
          M=IAND(7B,Word2)
          N=IAND(17B,ISHFT(Word2,-3))
          VOLTS=((Word1*256.0)+ISHFT(Word2,-8))*(0.5**(N+1))*(0.1**M)
      ELSE
          VOLTS=-2.0E-9
      ENDIF
      RETURN
      END
```

```
&PLOT   T=00004 IS ON CR   00020 USING 00003 BLKS R=0000
                                        11:55 AM   TUE., 10   JAN., 1984

FTN7X,Q
$FILES 0,3

C      This is the main segment of the program used for plotting for the
C      Rockfill Research Project, DB-31.  This program serves only as a main
C      segment and performs no useful function other than to call
C      the next segment.  This is done because of constraints of DGL.

       PROGRAM PLOT
       COMMON /a/ seg
       INTEGER NAME(3)
       DATA NAME(1)/'PL'/,NAME(2)/'OT'/,NAME(3)/'1 '/
       seg=0

C      call segment DATS1

       CALL SEGLD (NAME,ierr)
       WRITE (1,10) ierr
10     FORMAT ('Segload call error ',I2,' in DATMAIN')
       STOP
       END
```

```
&PLOT1 T=00004 IS ON CR  00020 USING 00066 BLKS R=0000
                                            2:47 PM  TUE., 10  JAN., 1984

FTN7X,Q
      PROGRAM PLOT1(5)

C     This is segment one of the plotting program for the Rockfill
C     Research Project DB-31.  This segment is called by program PLOT
C     and retreives the data file and converts it to actual values using
C     the zero correction factor and calibration factor data files.
C     The data is put in common for use in the next segment.  The next
C     segment is CRTPLT for CRT plots using the HP-1350 graphics translator
C     and the HP-1310 graphics display.  If hardcopy plots are required,
C     the next segment is HRDPLT which plots using the HP-9872 plotter.

C     Three data files are used in this program:
C          DataFile - file where test data is stored.  Written by
C                     CDA programs GRAB and GRAB2 in raw two-word
C                     integer format as taken from the 2250 using
C                     the TRANSFER AI command.  The data is stored
C                     in binary format.  Each data record is 256
C                     words long.
C          ZeroFile - a file of zero correction factors generated
C                     by program ZEROC.  Since the 2250 does not
C                     perform zero offset compensation during CDA
C                     it is done after the test.  Data from the
C                     DataFile is first converted to voltage by
C                     subroutine VOLTS.  The zero correction factor
C                     for the corresponding channel is then sub-
C                     tracted from this voltage yielding the
C                     corrected value.
C          CalFile -  a file containing a calibration factor for
C                     each channel.  These values are used to
C                     scale the zero-corrected data values into
C                     engineering units.  This file also contains
C                     the units of measurement, the date of cali-
C                     bration and a description of the measurement
C                     for each channel.

C     The operator is asked for the data file name, the time-frame he
C     wants to look at and the data channels of interest.  The data is
C     pulled off disc a record at a time, then corrected, scaled
C     and loaded into an array for ease of plotting.
C
C     The operator is also asked for the zero correction factor file name
C     and the calibration factor file name and then these data are
C     retrieved from disc and displayed for the operator to inspect.

      COMMON /a/ seg
      COMMON logonlu,time1,time2,Minplot,Maxplot,Ticplot,Units
      COMMON ChanNum,ChData,NumPoints,NumChan,DataFile,StartChan
      COMMON ZeroCorrection,CalFactor,Description,TestTitle

      INTEGER dummy                   ! dummy integer variable

      INTEGER iline                   ! disc read error locator

      INTEGER ios                     ! I/O status return

      INTEGER ChanNum(3)              ! array of channels of interest
```

```

      INTEGER maxchan              ! maximum number of channels that
                                   ! can be plotted

      INTEGER NumPoints            ! number of points to be plotted
                                   ! for each channel

      INTEGER dbuffer(128)         ! buffer for record of test data

      INTEGER record               ! record number to be read from

      INTEGER NumChan              ! total number of channels to be plotted

      INTEGER*4 bufferpointer      ! reading position pointer

      INTEGER channel              ! channel number index

      INTEGER logonlu              ! LU of log-on device

      INTEGER NAME1(3),NAME2(3)    ! arrays for next segment names

      INTEGER StartChan            ! starting channel number

      REAL Min(3),Max(3)           ! min and max of channel data to be plottec

      REAL Minplot,Maxplot,Ticplot! min, max and tic spacing for plot

      REAL time1,time2             ! time range for plotting

      DIMENSION ChData(3,1500)     ! array to hold converted and scaled data

      DIMENSION ZeroDate(15)       ! date and time of zero correction factors

      DIMENSION CalFactor(48)      ! channel calibration factors

      DIMENSION ZeroCorrection(48)! channel zero correction factors

      DIMENSION CalDate(15)        ! date and time of transducer calibration

      DIMENSION lbuf(200)          ! large output print buffer

      LOGICAL ex                   ! file existence inquiry return

      CHARACTER*1 YORN             ! question response variable

      CHARACTER*1 F                ! CRT/hardcopy flag

      CHARACTER*12 DataFile        ! string for data file name

      CHARACTER*12 CalFile         ! string for cal factor file name

      CHARACTER*12 ZeroFile        ! string for zero factor file name

      CHARACTER*11 Units(48)       ! channel units of measurement

      CHARACTER*32 Description(48)! measurement description for each channel

      CHARACTER*80 TestTitle       ! string for test title

      DATA NAME1(1)/'CR'/,NAME1(2)/'TP'/,NAME1(3)/'LT'/
```

```
      DATA NAME2(1)/'HR'/,NAME2(2)/'DP'/,NAME2(3)/'LT'/

      CALL LGBUF (lbuf,200)

      logonlu=LOGLU(dummy)

      WRITE (logonlu,100) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
  100 FORMAT (4A1)
      IF (seg.EQ.0) GOTO 110
      WRITE (logonlu,105)
  105 FORMAT ('New data file (Y or N)?')
      READ (logonlu,220) YORN
      IF (YORN.EQ.'N') GOTO 1000

C     operator entry of data file name

  110 WRITE (logonlu,115)
  115 FORMAT ('Enter data file name as',/,'FILNAM:SC:CR')
      READ (logonlu,120) DataFile
  120 FORMAT (A12)

C     check to see if file exists

      iline=121
      INQUIRE (FILE=DataFile,IOSTAT=ios,ERR=9000,EXIST=ex)
      IF (ex) GOTO 130
      WRITE (logonlu,125) DataFile
  125 FORMAT (//,'File ',A12,' does not exist'//)
      GOTO 110

C     operator entry of zero correction factor file name

  130 WRITE (logonlu,135)
  135 FORMAT ('Enter name of zero correction factor data file as',/,
     1       'FILNAM:SC:CR')
      READ (logonlu,140) ZeroFile
  140 FORMAT (A12)

C     check to see if ZeroFile exists

      iline=141
      INQUIRE (FILE=ZeroFile,IOSTAT=ios,ERR=9000,EXIST=ex)
      IF (ex) GOTO 145
      WRITE (logonlu,125) ZeroFile
      GOTO 130

C     open zero correction factor data file

  145 iline=145
      OPEN (200,FILE=ZeroFile,IOSTAT=ios,ERR=9000,STATUS='OLD')

C     read zero correction factor data file and close it

      READ(200,200) ZeroDate
  200 FORMAT (/,15A2)
      READ (200,205) (ZeroCorrection(I),I=1,48)
  205 FORMAT (2X,F10.7)

      iline=206
      CLOSE (200,IOSTAT=ios,ERR=9000,STATUS='KEEP')
```

```

C      output of zero correction factor data to operator display

       WRITE (logonlu,100) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
       WRITE (logonlu,210) ZeroFile,ZeroDate
 210   FORMAT ('Zero Correction Factor File ',A12,
      1        ' read as follows',/,15A2,//,9X,'Correction',
      2        17X,'Correction',17X,'Correction',/,'Channel      '
      3        'Factor',9X,'Channel     Factor',9X,
      4        'Channel     Factor')
       WRITE (logonlu,215)(I,ZeroCorrection(I),I+16,
      1        ZeroCorrection(I+16),I+32,ZeroCorrection(I+32),I=1,16)
 215   FORMAT (I4,F14.7,I12,F14.7,I12,F14.7)
       READ (logonlu,220) YORN
 220   FORMAT (A1)

C      operator entry of calibration factor file name

       WRITE (logonlu,100) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
 225   WRITE (logonlu,230)
 230   FORMAT ('Enter name of calibration factor data file name as',
      1        /,'FILNAM:SC:CR')
       READ (logonlu,140) CalFile

C      check to see if CalFile exists

       iline=231
       INQUIRE (FILE=CalFile,IOSTAT=ios,ERR=9000,EXIST=ex)
       IF (ex) GOTO 235
       WRITE (logonlu,125) CalFile
       GOTO 225

C      open calibration factor data file

 235   iline=235
       OPEN (300,FILE=CalFile,IOSTAT=ios,ERR=9000,STATUS='OLD')

C      read calibration factor data file and close it

       READ (300,240) CalDate,TestTitle
 240   FORMAT (/,15A2,/,A80)
       READ (300,245)(CalFactor(I),Units(I),Description(I),I=1,48)
 245   FORMAT (2X,F10.7,X,A11,10X,A32)

       CLOSE (300,IOSTAT=ios,ERR=9000,STATUS='KEEP')

C      output of calibration factor data to operator display

       DO start=1,33,16
           WRITE (logonlu,100) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
           WRITE (logonlu,250) CalFile,CalDate,TestTitle,(I,CalFactor(I),
      1          Units(I),Description(I),I=start,start+15)
 250       FORMAT ('Calibration Factor File ',A12,5X,15A2,//,A80,
      1          'Channel  Calibration Factor  Units         Description'/
      2          (I4,F21.7,4X,A11,2X,A32))
           READ (logonlu,220) YORN
       END DO

C      operator entry of starting channel number

```

```
      WRITE (logonlu,100) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
      WRITE (logonlu,300)
300   FORMAT ('Enter beginning channel number')
      READ (logonlu,*) StartChan
      StartChan=StartChan-1

C     operator entry of time-frame of interest

1000  WRITE (logonlu,100) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
      WRITE (logonlu,1005)
1005  FORMAT ('Do you want CRT plots or hardcopy (C or H)?')
      READ (logonlu,220) F

1010  WRITE (logonlu,1015)
1015  FORMAT ('Enter time-frame of interest as',/,'minsec,maxsec')
      READ (logonlu,*) time1,time2
      IF (time1.GE.time2) THEN
         WRITE (logonlu,1020)
1020     FORMAT (//,'Min time greater than or equal to Max time',//)
         GOTO 1010
      ENDIF

C     Calculation and output of number of channels that can be plotted

      timetotal=time2-time1
      Maxchan=MIN0(INT(15.0/(timetotal)),3)

C     check to see that maximum time for one channel is not exceeded

      IF (Maxchan.LT.1) THEN
         WRITE (logonlu,100) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
         WRITE (logonlu,1025)
1025     FORMAT ('Maximum time of 15 seconds was exceeded',//)
         GOTO 1010
      ENDIF

      IF (F.EQ.'H') Maxchan=3
1030  WRITE (logonlu,1035) timetotal,Maxchan
1035  FORMAT (/,'Maximum number of channels that can be plotted for',
     1         F6.3,' seconds is',I3)

C     operator entry of channels of interest

      WRITE (logonlu,1040)
1040  FORMAT (//,'Enter number of channels to be plotted')
      READ (logonlu,*) NumChan

C     check to see if entered number of channels exceeds maximum

      IF (NumChan.GT.MaxChan) THEN
         WRITE (logonlu,100) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
         WRITE (logonlu,1045)
1045     FORMAT ('Maximum number of channels was exceeded',//)
         GOTO 1030
      ENDIF

      WRITE (logonlu,1050)
1050  FORMAT (//,'Enter channel numbers to be displayed as',/,
     1         'chanA,chanB,chanC, ... ,chanZ')
      READ (logonlu,*) (ChanNum(I),I=1,NumChan)
```

```
C     requested data is brought off of disc

1100  record=INT(time1/0.064+1)
      iline=1101
      OPEN (100,FILE=DataFile,IOSTAT=ios,ERR=9000,STATUS='OLD',
     1      ACCESS='DIRECT',RECL=256)
      READ (100,REC=record,IOSTAT=ios,ERR=9000)(dbuffer(I),I=1,128)

C     this data is then put into array ChData for ease of plotting

      Numpoints=(timetotal)*100     ! number of data points to plot
      bufferpointer=INT(time1*2000.0-INT(time1/0.064)*128.0+
     1              (ChanNum(1)-StartChan)*2-1)

C     loop to load array with zero-corrected and scaled data

      DO Point=1,NumPoints
          DO Channel=1,NumChan

              ChData(Channel,Point)=(VOLTS(dbuffer(bufferpointer),
     1                                  dbuffer(bufferpointer+1))
     2                               -ZeroCorrection(ChanNum(Channel)))/
     3                                  CalFactor(ChanNum(Channel))
              IF (Channel.EQ.NumChan) THEN
                  bufferpointer=bufferpointer+(10-ChanNum(Channel)
     1                +ChanNum(1))*2
              ELSE
                  bufferpointer=bufferpointer+(ChanNum(Channel+1)-
     1                ChanNum(Channel))*2

C                 check to see if negative bufferpointer due to next
C                 channel number less than last near a record boundary

                  IF (bufferpointer.LE.0) then
                      record=record-1
                      iline=1102
                      READ (100,REC=record,IOSTAT=ios,ERR=9000)
     1                    (dbuffer(I),I=1,128)
                      bufferpointer=bufferpointer+128
                  ENDIF
              ENDIF

C             check to see if past end of buffer

              IF (bufferpointer.GT.128) THEN

C                 if so read next record and reset bufferpointer

                  record=record+1
                  iline=1103
                  READ (100,Rec=record,IOSTAT=ios,ERR=9000) (dbuffer(I),
     1                 I=1,128)
                  bufferpointer=bufferpointer-128
              ENDIF
          END DO
      END DO
      iline=1104
      CLOSE(100,IOSTAT=ios,ERR=9000,STATUS='KEEP')
```

```
C      calculation of data minimums and maximums

       DO Channel=1,NumChan
           Max(Channel)=ChData(Channel,1)
           Min(Channel)=Max(Channel)
           DO Point=2,NumPoints
               IF(ChData(Channel,Point).GT.Max(Channel)) THEN
                   Max(Channel)=ChData(Channel,Point)
                   GOTO 1105
               ENDIF
               IF (ChData(Channel,Point).LT.Min(Channel))
     1             Min(Channel)=ChData(Channel,Point)
 1105      END DO
       END DO

C      operator entry of plot min, max and tic spacing

       WRITE (logonlu,100) CHAR(27),CHAR(104),CHAR(27),CHAR(74)
       DO Channel=1,NumChan
           WRITE (logonlu,1110) ChanNum(Channel),Max(Channel),
     1                          Min(Channel)
 1110      FORMAT (/,'Channel',I3,/,5X,'Maximum is',F10.4,/,5X,
     1             'Minimum is',F10.4)
       END DO
       WRITE (logonlu,1115)
 1115  FORMAT (//,'Enter minimum, maximum and tic spacing for plot as',
     1         /,'MIN,MAX,TIC')
       READ (logonlu,*) Minplot,Maxplot,Ticplot

C      call in next segment of program DATS2

       iline=1116

       IF (F.EQ.'C') THEN
           CALL SEGLD (NAME1,ios)
       ELSE
           CALL SEGLD (NAME2,ios)
       ENDIF
       IF (ios.EQ.0) GOTO 9500

C      disc access error handling routine

 9000  WRITE (logonlu,9005) ios,iline
 9005  FORMAT (//'****Disc access error ',I4,' in program DATS1',
     1         ' at iline=',I5)
 9500  STOP
       END

C      This function takes two words of raw data input from the
C      2250 and converts them to one voltage

       REAL FUNCTION VOLTS(Word1,Word2)

       INTEGER Word1,Word2      ! words to be converted

       IF (IAND(200B,Word2).EQ.0) THEN
           M=IAND(7B,Word2)
           N=IAND(17B,ISHFT(Word2,-3))
           VOLTS=((Word1*256.0)+ISHFT(Word2,-8))*(0.5**(N+1))*(0.1**M)
       ELSE
```

```
        VOLTS=-2.0E-9
    ENDIF
    RETURN
    END
```

```
&CRTPL T=00004 IS ON CR  00020 USING 00042 BLKS R=0000
                                           1:44 PM  TUE., 10  JAN., 1984

FTN7X,Q
      PROGRAM CRTPLT(5)

C     This is a segment of the plotting program for the Rockfill
C     Research Project DB-31.  This segment is called by program
C     PLOT1 when the operator requests a plot of data using the
C     HP-1350 graphics translator and the HP-1310 graphics display.

      COMMON /a/ seg
      COMMON logonlu,time1,time2,Minplot,Maxplot,Ticplot,Units
      COMMON ChanNum,ChData,NumPoints,NumChan,DataFile,StartChan
      COMMON ZeroCorrection,CalFactor,Description,TestTitle

      INTEGER dummy                  ! dummy integer variable

      INTEGER ios                    ! I/O status return

      INTEGER logonlu                ! LU of log-on device

      INTEGER NAME(3)                ! array for next segment name

      INTEGER ChanNum(3)             ! array of channels of interest


      INTEGER NumPoints              ! number of points to be plotted for
                                     ! each channel

      INTEGER NumChan                ! total number of channels to be plotted

      INTEGER iline                  ! disc read error locator

      INTEGER Point                  ! point number to be plotted

      INTEGER Channel                ! channel number index

      INTEGER CurveLabel(21)         ! integer variable for curve labeling

      INTEGER TestT(40)              ! test title integer variable

      INTEGER Tlength                ! length of test title

      INTEGER StartChan              ! starting channel number

      REAL Min(3),Max(3)             ! min and max of channel data to be plotted

      REAL Minplot,Maxplot,Ticplot! min, max and tic spacing for plot

      REAL time1,time2               ! time range for plotting

      DIMENSION ChData(3,1500)       ! array to hold converted and scaled data

      DIMENSION lbuf(200)            ! large output print buffer

      DIMENSION ZeroCorrection(48)! channel zero correction factors

      DIMENSION CalFactor(48)        ! channel calibration factors

```

```
      LOGICAL ex                      ! file existence inquiry return
      CHARACTER*1 YORN                ! question response variable
      CHARACTER*11 Units(48)          ! channel units of measurement
      CHARACTER*12 DataFile           ! name of data file
      CHARACTER*32 Description(48)! measurement description for each channel
      CHARACTER*42 Desc               ! channel description
      CHARACTER*80 TestTitle          ! string for test title
      EQUIVALENCE (Desc,CurveLabel),(TestTitle,TestT)
      CALL LGBUF (lbuf,200)
      DATA NAME(1)/'PL'/,NAME(2)/'OT'/,NAME(3)/'1 '/
2     FORMAT(A32)
C     initialization of plotting program
      CALL ZBEGN
      CALL ZDINT (7,0,dummy)
C     labeling of graph with test title
      CALL ZWIND (0.0,1.0,0.0,20.0)
      CALL ZCSIZ (0.01,0.38)
      Tlength=LENGTH(TestTitle,80)
      CALL ZMOVE (0.5-Tlength/2.0*0.01172,15.0)
      CALL ZTEXT (Tlength,TestT)
C     legend for curve 1 ID
      CALL ZMOVE (0.1,5.0)
      CALL ZDRAW (0.193,5.0)
      Desc(1:2)='  '
      Desc(3:34)=Description(ChanNum(1))
      CALL ZTEXT (34,CurveLabel)
      IF (NumChan.EQ.1) GOTO 100
C     legend for curve 2 ID
      CALL ZMOVE (0.1,4.0)
      Desc(1:10)='--------  '
      Desc(11:42)=Description(ChanNum(2))
      CALL ZTEXT (42,CurveLabel)
      IF (NumChan.EQ.2) GOTO 100
C     legend for curve 3 ID
      CALL ZMOVE (0.1,3.0)
      Desc(1:10)='........  '
      Desc(11:42)=Description(ChanNum(3))
      CALL ZTEXT (42,CurveLabel)
C     drawing of axes
```

```
  100  CALL ZVIEW (0.0,1.0,0.3,0.7)
       CALL DRAWAXES (time1,time2,Minplot,Maxplot,1.0,Ticplot,
      1                'TIME (SECS)',Units(ChanNum(1)))

C      plotting of first channel of data with solid line

       CALL ZMOVE (time1+0.001*(ChanNum(1)-1),ChData(1,1))

C           loop to plot individual points

            DO Point=2,NumPoints
                 CALL ZDRAW (time1+0.001*(ChanNum(1)-1)+(Point-1)*0.01,
      1                         ChData(1,Point))
            END DO
       CALL ZMCUR

C      check to see if last channel to plot

       IF (NumChan.EQ.1) GOTO 1000

C      plotting of second channel of data with dashes

       CALL ZMOVE (time1+.001*(ChanNum(2)-1),ChData(2,1))

            DO Point=2,NumPoints,2
                 CALL ZDRAW (time1+.001*(ChanNum(2)-1)+(Point-1)*.01,
      1                         ChData(2,Point))
                 CALL ZMOVE (time1+.001*(ChanNum(2)-1)+Point*.01,
      1                         ChData(2,Point+1))
            END DO
       CALL ZMCUR

C      check to see if last channel to plot

       IF (NumChan.EQ.2) GOTO 1000

C      plotting of third channel of data with dots

       CALL ZMOVE (time1+.001*(ChanNum(3)-1),ChData(3,1))
       CALL ZDRAW (time1+.001*(ChanNum(3)-1),ChData(3,1))

            DO Point=2,NumPoints
                 CALL ZMOVE (time1+.001*(ChanNum(3)-1)+(Point-1)*.01,
      1                         ChData(3,Point))
                 CALL ZDRAW (time1+.001*(ChanNum(3)-1)+(Point-1)*.01,
      1                         ChData(3,Point))
            END DO
       CALL ZMCUR

C      ask operator if finished with CRT plots

 1000  WRITE (logonlu,1005)
 1005  FORMAT(//'Do you want to plot more (Y or N)?')
       READ (logonlu,1010) YORN
 1010  FORMAT(A1)
       iline=1210
       seg=2
       IF (YORN.EQ.'Y') CALL SEGLD (NAME,ierr)
```

```
      IF (ierr.EQ.0) GOTO 9500

C     disc access error handling routine

9000  WRITE (logonlu,9005) ierr,iline
9005  FORMAT (//'****Disc access error ',I4,' in program DATS2',
     1         ' at iline=',I5)

9500  CALL ZDEND
      CALL ZEND
      STOP
      END




      SUBROUTINE DRAWAXES (Xmin,Xmax,Ymin,Ymax,Xtic,Ytic,LABEL1,LABEL2)

C     This subroutine draws a set of axes and labels them using the DGL
C     subroutine set for doing this.  Inputs are minimum, maximum, tic
C     spacing and label for each axis.  It is necessary to define the
C     viewport and window before entering this subroutine.

      INTEGER XL(6),YL(6)           ! integer variables for storage of
                                    ! axis labels

      INTEGER Xlength,Ylength       ! length of axis labels

      INTEGER OPCODE                ! pass parameter for ZIWS call

      INTEGER Isize,Rsize           ! ZIWS return array sizes

      INTEGER Ilist(1)              ! ZIWS return array

      CHARACTER*12 Xlabel,Ylabel    ! axis labels

      CHARACTER*12 LABEL1,LABEL2    ! axis labels from main program

      REAL Rlist(4)                 ! ZIWS return array

      EQUIVALENCE (Xlabel,XL),(Ylabel,YL)

      Xlabel=LABEL1
      Ylabel=LABEL2

C     set window to known size and set character size

      CALL ZWIND (0.0,1.0,0.0,1.0)
      CALL ZCSIZ (.0125,.02)

C     inquire to find character size in world coordinates

      OPCODE=250
      Isize=0
      Rsize=2
      CALL ZIWS (OPCODE,Isize,Rsize,Ilist,Rlist,ierr)

C     find axis label lengths and write them centered on each axis

      Xlength=LENGTH (Xlabel,11)
```

```
      Ylength=LENGTH (Ylabel,11)
      CALL ZMOVE (0.5-Xlength/2.0*Rlist(1),0.0)
      CALL ZTEXT (Xlength,XL)
      CALL ZMOVE (Rlist(2)*0.3,0.5-Ylength/2.0*Rlist(1))
      CALL ZOESC (1057)
      CALL ZTEXT (Ylength,YL)

C     inquire to find out viewport limits, then expand the viewport
C     to give room for tic labels

      OPCODE=451
      Isize=0
      Rsize=4
      CALL ZIWS (OPCODE,Isize,Rsize,Ilist,Rlist,ierr)
      CALL ZVIEW ((Rlist(2)-Rlist(1))*0.05+Rlist(1),
     1            (Rlist(2)-Rlist(1))*0.9+Rlist(1),
     2            (Rlist(4)-Rlist(3))*0.2+Rlist(3),Rlist(4))

C     change window for axes drawing

      Xtotal=Xmax-Xmin
      Ytotal=Ymax-Ymin
      CALL ZWIND (Xmin-0.15*Xtotal,Xmax+0.1*Xtotal,Ymin,Ymax)

C     draw axes

      CALL LAXES (Xtic,Ytic,Xmin,5.0*Ymin,2,2,0.01)
      CALL LAXES (Xtic,Ytic,Xmin-0.5*Xtotal,Ymin,2,2,0.01)

      RETURN
      END

      INTEGER FUNCTION LENGTH (STRING,L)

C     This function checks a string to see how many characters are
C     present in it.  The string is checked for two successive blank
C     characters, when this condition is found, it assumes that the
C     last non-blank character is the last character of the string
C     and the length is given the value of the character position of
C     that last non-blank character.

      CHARACTER*80 STRING
      LENGTH=L
      DO I=2,L-1
          IF (STRING(I:I+1).EQ.'  ') THEN
              LENGTH=I-1
              GOTO 10
          ENDIF
      END DO
      IF (STRING(L:L).EQ.' ') LENGTH=L-1
10    RETURN
      END
```

```
&HRDPL T=00004 IS ON CR  00020 USING 00042 BLKS R=0000
                                           2:40 PM  TUE., 10  JAN., 1984

FTN7X,Q
      PROGRAM HRDPLT(5)

C     This is a segment the plotting program for the Rockfill
C     Research Project DB-31.  This segment is called by program
C     PLOT1 when the operator requests a plot using the HP-9872
C     plotter.

      COMMON /a/ seq
      COMMON logonlu,time1,time2,Minplot,Maxplot,Ticplot,Units
      COMMON ChanNum,ChData,NumPoints,NumChan,DataFile,StartChan
      COMMON ZeroCorrection,CalFactor,Description,TestTitle

      INTEGER dummy                 ! dummy integer variable

      INTEGER ios                   ! I/O status return

      INTEGER logonlu               ! LU of log-on device

      INTEGER NAME(3)               ! array for next segment name

      INTEGER ChanNum(3)            ! array of channels of interest


      INTEGER NumPoints             ! number of points to be plotted for
                                    ! each channel

      INTEGER NumChan               ! total number of channels to be plotted

      INTEGER iline                 ! disc read error locator

      INTEGER Point                 ! point number to be plotted

      INTEGER Channel               ! channel number index

      INTEGER CurveLabel(21)        ! integer variable for curve labeling

      INTEGER TestT(40)             ! test title integer variable

      INTEGER Tlength               ! length of test title

      INTEGER StartChan             ! starting channel number

      REAL Min(3),Max(3)            ! min and max of channel data to be plotted

      REAL Minplot,Maxplot,Ticplot! min, max and tic spacing for plot

      REAL time1,time2              ! time range for plotting

      DIMENSION ChData(3,1500)      ! array to hold converted and scaled data

      DIMENSION lbuf(200)           ! large output print buffer

      DIMENSION ZeroCorrection(48)! channel zero correction factors

      DIMENSION CalFactor(48)       ! channel calibration factors

```

```
      LOGICAL ex                    ! file existence inquiry return
      CHARACTER*1 YORN              ! question response variable
      CHARACTER*11 Units(48)        ! channel units of measurement
      CHARACTER*12 DataFile         ! name of data file
      CHARACTER*32 Description(48)! measurement description for each channel
      CHARACTER*34 Desc             ! channel description
      CHARACTER*80 TestTitle        ! string for test title
      EQUIVALENCE (Desc,CurveLabel),(TestTitle,TestT)
      CALL LGBUF (lbuf,200)
      DATA NAME(1)/'PL'/,NAME(2)/'OT'/,NAME(3)/'1 '/
2     FORMAT(A32)
C     initialization of plotting program
      CALL ZBEGN
      CALL ZDINT (8,0,dummy)
      CALL ZDLIM (0.0,254.0,0.0,190.0)
      CALL ZASPK (254.0,190.0)
C     labeling of graph with test title
      CALL ZWIND (0.0,1.0,0.0,1.0)
      CALL ZCSIZ (0.0125,0.035)
      Tlength=LENGTH(TestTitle,80)
      CALL ZMOVE (0.5-Tlength/2.0*0.01172,0.98)
      CALL ZTEXT (Tlength,TestT)
C     legend for curve 1 ID
      CALL ZCSIZ (0.0125,0.028)
      CALL ZCOLR (2)
      CALL ZMOVE (0.1,0.04)
      CALL ZDRAW (0.15,0.04)
      Desc(1:2)='  '
      Desc(3:34)=Description(ChanNum(1))
      Tlength=LENGTH(Desc,34)
      CALL ZTEXT (Tlength,CurveLabel)
      IF (NumChan.EQ.1) GOTO 100
C     legend for curve 2 ID
      CALL ZCOLR (3)
      CALL ZMOVE (0.1,.02)
      CALL ZDRAW (0.15,0.02)
      Desc(3:34)=Description(ChanNum(2))
      Tlength=LENGTH(Desc,34)
      CALL ZTEXT (Tlength,CurveLabel)
      IF (NumChan.EQ.2) GOTO 100
C     legend for curve 3 ID
```

```
      CALL ZCOLR (4)
      CALL ZMOVE (0.1,0.0)
      CALL ZDRAW (0.15,0.0)
      Desc(3:34)=Description(ChanNum(3))
      Tlength=LENGTH(Desc,34)
      CALL ZTEXT (Tlength,CurveLabel)

C     drawing of axes

100   CALL ZCOLR (1)
C     CALL ZVIEW (0.0,1.0,.10,.95)
      CALL DRAWAXES (time1,time2,Minplot,Maxplot,1.0,Ticplot,
     1              'TIME (SECS)',Units(ChanNum(1)))

C     plotting of first channel of data with color 2

      CALL ZCOLR (2)
      CALL ZMOVE (time1+0.001*(ChanNum(1)-1),ChData(1,1))

C         loop to plot individual points

          DO Point=2,NumPoints
              CALL ZDRAW (time1+0.001*(ChanNum(1)-1)+(Point-1)*0.01,
     1                    ChData(1,Point))
          END DO
      CALL ZMCUR

C     check to see if last channel to plot

      IF (NumChan.EQ.1) GOTO 1000

C     plotting of second channel of data with color 3

      CALL ZCOLR (3)
      CALL ZMOVE (time1+.001*(ChanNum(2)-1),ChData(2,1))
      CALL ZDRAW (time1+.001*(ChanNum(2)-1),ChData(2,1))

          DO Point=2,NumPoints
              CALL ZDRAW (time1+.001*(ChanNum(2)-1)+(Point-1)*.01,
     1                    ChData(2,Point))
          END DO
      CALL ZMCUR

C     check to see if last channel to plot

      IF (NumChan.EQ.2) GOTO 1000

C     plotting of third channel of data with color 4

      CALL ZCOLR (4)
      CALL ZMOVE (time1+.001*(ChanNum(3)-1),ChData(3,1))

          DO Point=2,NumPoints
              CALL ZDRAW (time1+.001*(ChanNum(3)-1)+(Point-1)*.01,
     1                    ChData(3,Point))
          END DO
      CALL ZMCUR

C     ask operator if finished with CRT plots
```

```
 1000  CALL ZCOLR (0)
       CALL ZMCUR
       WRITE (logonlu,1005)
 1005  FORMAT(//'Do you want to plot more (Y or N)?')
       READ (logonlu,1010) YORN
 1010  FORMAT(A1)
       iline=1011
       seg=2
       IF (YORN.EQ.'Y') CALL SEGLD (NAME,ierr)

       IF (ierr.EQ.0) GOTO 9500

C      disc access error handling routine

 9000  WRITE (logonlu,9005) ierr,iline
 9005  FORMAT (//'****Disc access error ',I4,' in program DATS2',
     1          ' at iline=',I5)

 9500  CALL ZDEND
       CALL ZEND
       STOP
       END




       SUBROUTINE DRAWAXES (Xmin,Xmax,Ymin,Ymax,Xtic,Ytic,LABEL1,LABEL2)

C      This subroutine draws a set of axes and labels them using the DGL
C      subroutine set for doing this.  Inputs are minimum, maximum, tic
C      spacing and label for each axis.

       INTEGER XL(6),YL(6)           ! integer variables for storage of
                                     ! axis labels

       INTEGER Xlength,Ylength       ! length of axis labels

       INTEGER OPCODE                ! pass parameter for ZIWS call

       INTEGER Isize,Rsize           ! ZIWS return array sizes

       INTEGER Ilist(1)              ! ZIWS return array

       CHARACTER*12 Xlabel,Ylabel    ! axis labels

       CHARACTER*12 LABEL1,LABEL2    ! axis labels from main program

       REAL Rlist(4)                 ! ZIWS return array

       EQUIVALENCE (Xlabel,XL),(Ylabel,YL)

       Xlabel=LABEL1
       Ylabel=LABEL2

C      find axis label lengths and write them centered on each axis

       Xlength=LENGTH (Xlabel,11)
       Ylength=LENGTH (Ylabel,11)
       CALL ZMOVE (0.56-Xlength/2.0*0.0125,0.08)
```

```
      CALL ZTEXT (Xlength,XL)
      CALL ZMOVE (0.02,0.54-Ylength/2.0*0.02)
      Rlist(1)=0.0
      Rlist(2)=127.999
      CALL ZOESC (250,0,2,Ilist,Rlist)
      CALL ZTEXT (Ylength,YL)
      Rlist(1)=127.999
      Rlist(2)=0.0
      CALL ZOESC (250,0,2,Ilist,Rlist)

C     inquire to find out viewport limits, then expand the viewport
C     to give room for tic labels

      OPCODE=451
      Isize=0
      Rsize=4
      CALL ZIWS (OPCODE,Isize,Rsize,Ilist,Rlist,ierr)
      CALL ZDLIM (0.0,254.0,20.0,180.0)
      CALL ZASPK (254.0,160.0)

C     change window for axes drawing

      Xtotal=Xmax-Xmin
      Ytotal=Ymax-Ymin
      CALL ZWIND (Xmin-0.1*Xtotal,Xmax+0.02*Xtotal,
     1            Ymin-0.05*Ytotal,Ymax+0.02*Ytotal)
      CALL ZCSIZ (0.01*Xtotal,0.04*Ytotal)

C     draw axes

      CALL LAXES (Xtic,Ytic,Xmin,Ymin,2,2,0.01)

      RETURN
      END

      INTEGER FUNCTION LENGTH (STRING,L)

C     This function checks a string to see how many characters are
C     present in it.  The string is checked for two successive blank
C     characters, when this condition is found, it assumes that the
C     last non-blank character is the last character of the string
C     and the length is given the value of the character position of
C     that last non-blank character.

      CHARACTER*80 STRING        ! string to be checked
      LENGTH=L                   ! L is the length of the string variable passed

C     loop to check for a character starting at the end of the string

      DO I=0,L-1
          IF (STRING(L-I:L-I).EQ.' ') THEN
              LENGTH=LENGTH-1
          ELSE
              GOTO 10
          ENDIF
      END DO
10    RETURN
      END
```

APPENDIX D [40]

PERFORMANCE CHARACTERISTICS OF THE HYDRAULIC RAM AND ACCELEROMETER SPECIFICATIONS

Ram No. 1

Vector force lb (kg) 50,000 (22 680)
Stall force lb (kg) 77,000 (34 900)
Maximum stroke inch (mm) 1 (25)
Maximum velocity (in/s) (mm/s) 18 (457)
Maximum frequency (hz) 400

SPECIFICATIONS OF EGA-125 SERIES MINIATURE ACCELEROMETERS*

Model EGA-125 is the uniaxial piezoresistive accelerometer used to measure embankment accelerations.

Model	EGA-125–5	EGA-125–10
Range	± 5 g	± 10 g
Sensitivity	15 mV/g nom.	12 mV/g nom.
Resonant frequency	300 Hz	500 Hz
Nonlinearity	± 1%	± 1%
Transverse sensitivity	3% max.	3% max.
Weight	0.5 g	0.5 g

**Entran Bulletin*, EGAS-582, Entran Devices, Inc., Fairfield, New Jersey, no date.

APPENDIX E

THREE-DIMENSIONAL MODEL PLOTS, MODEL CROSS SECTIONS, AND POINT GAUGE DATA

CONTENTS

FIGURES

TABLES

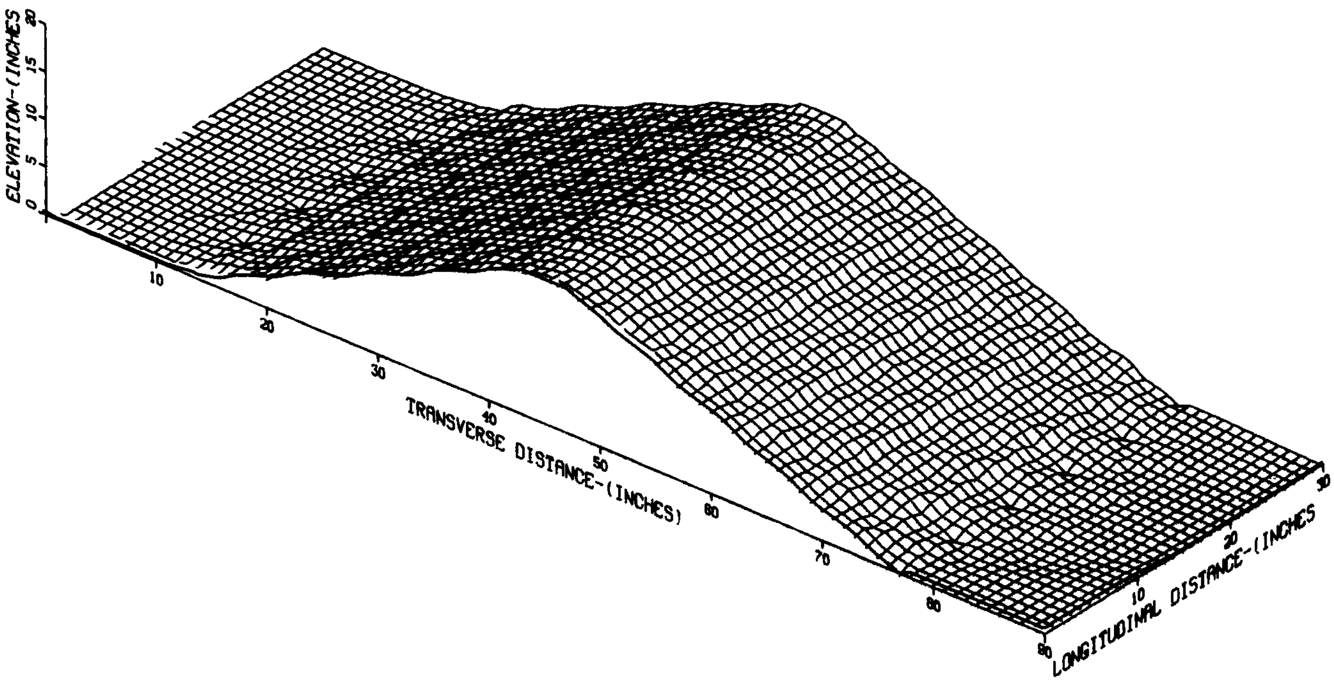


Figure E-1. – Three-dimensional plot for model No. 1 – pretest.

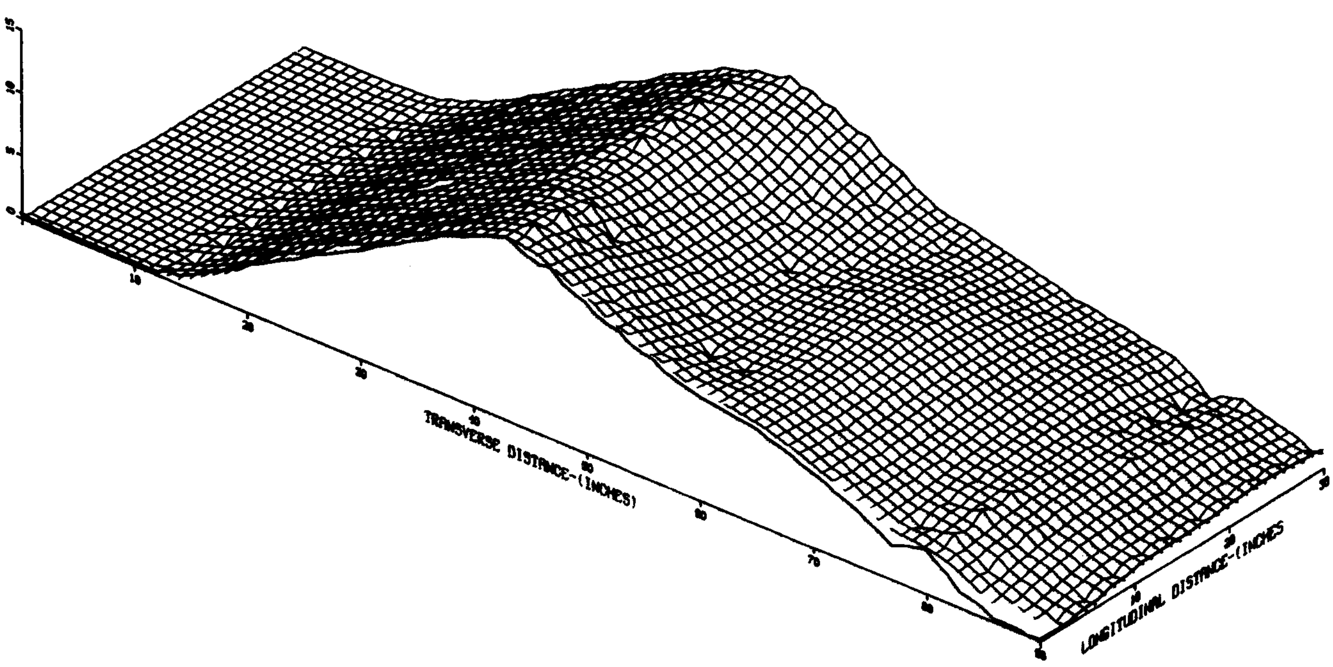


Figure E-2. – Three-dimensional plot for model No. 3 – posttest.

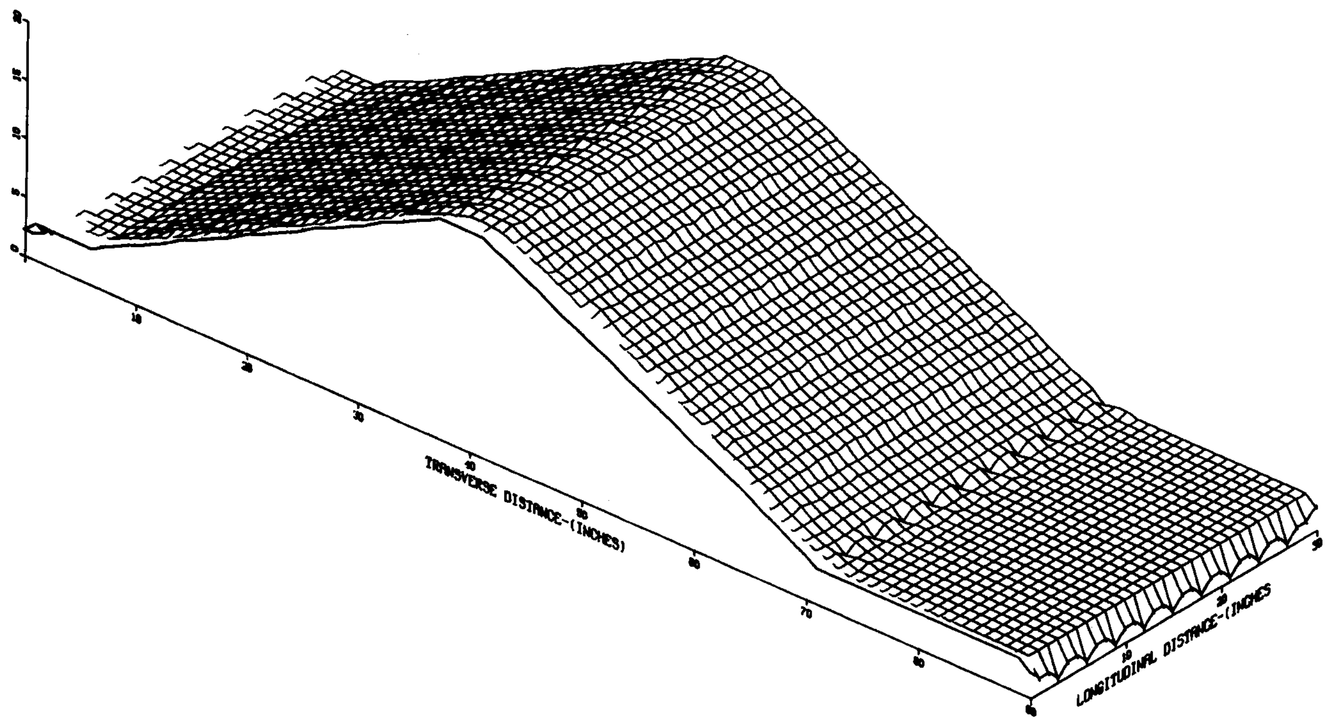


Figure E-3. – Three-dimensional plot for model No. 5 – pretest.

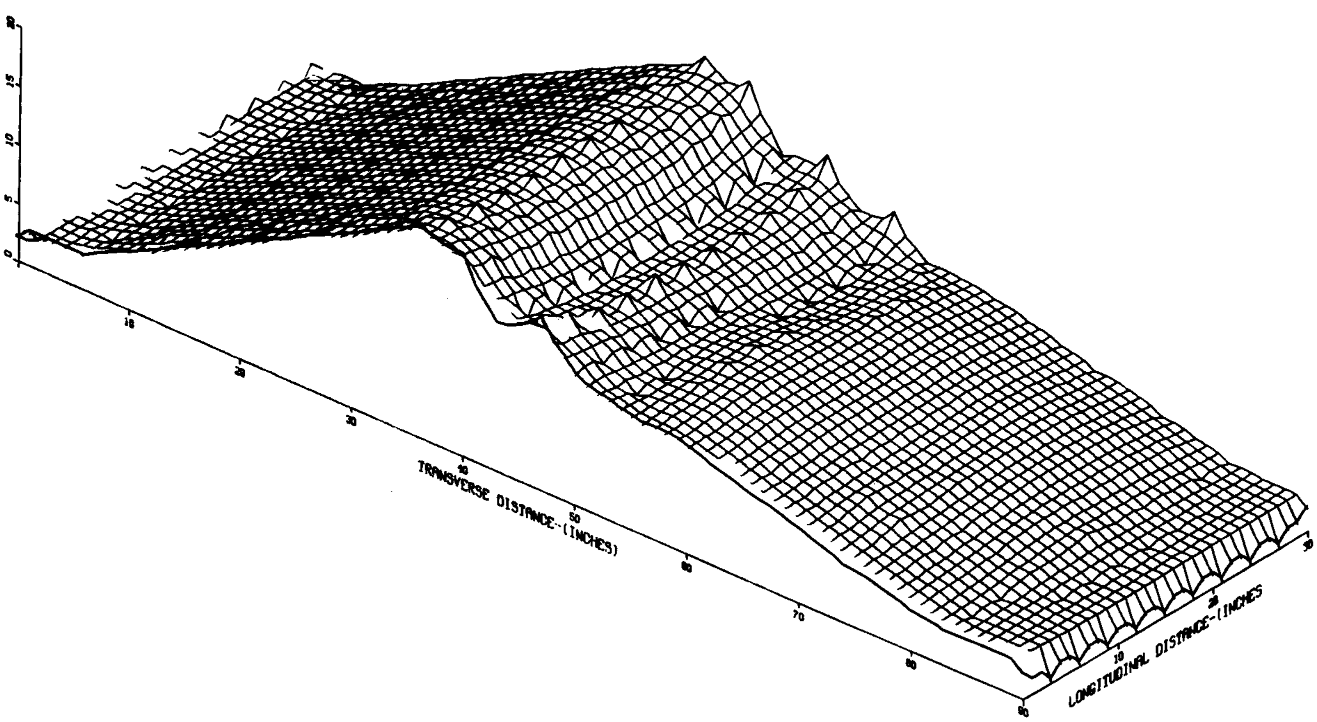


Figure E-4. – Three-dimensional plot for model No. 5 – posttest.

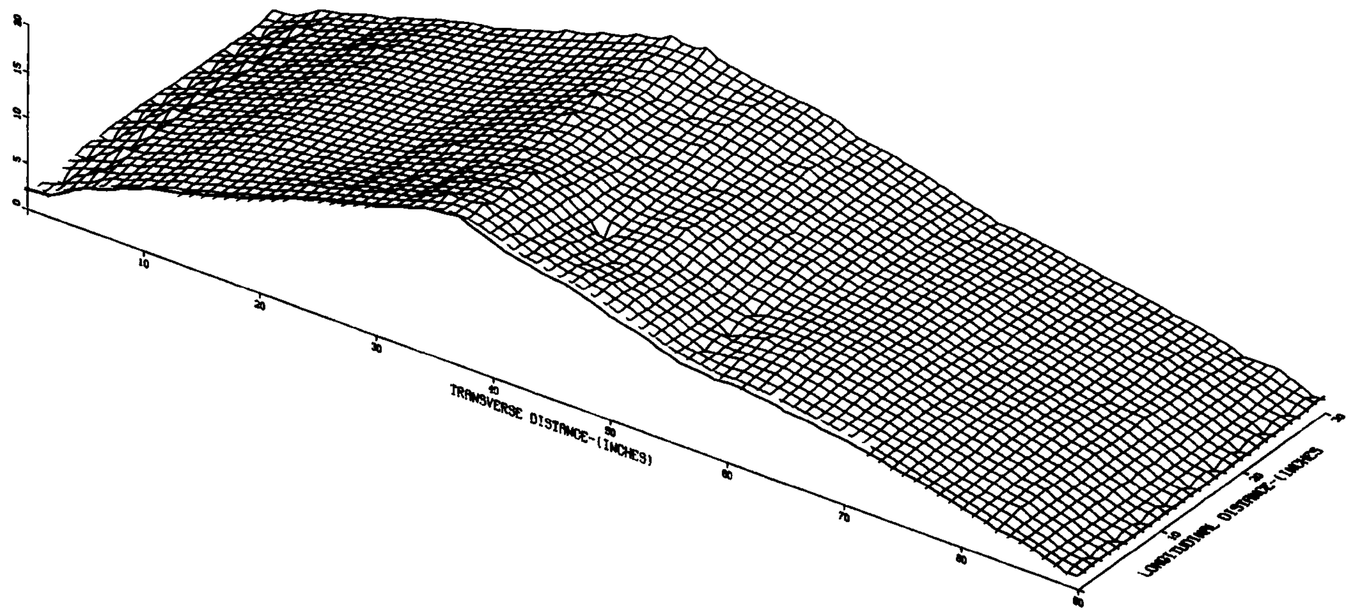


Figure E-5. – Three-dimensional plot for model No. 6 – posttest.

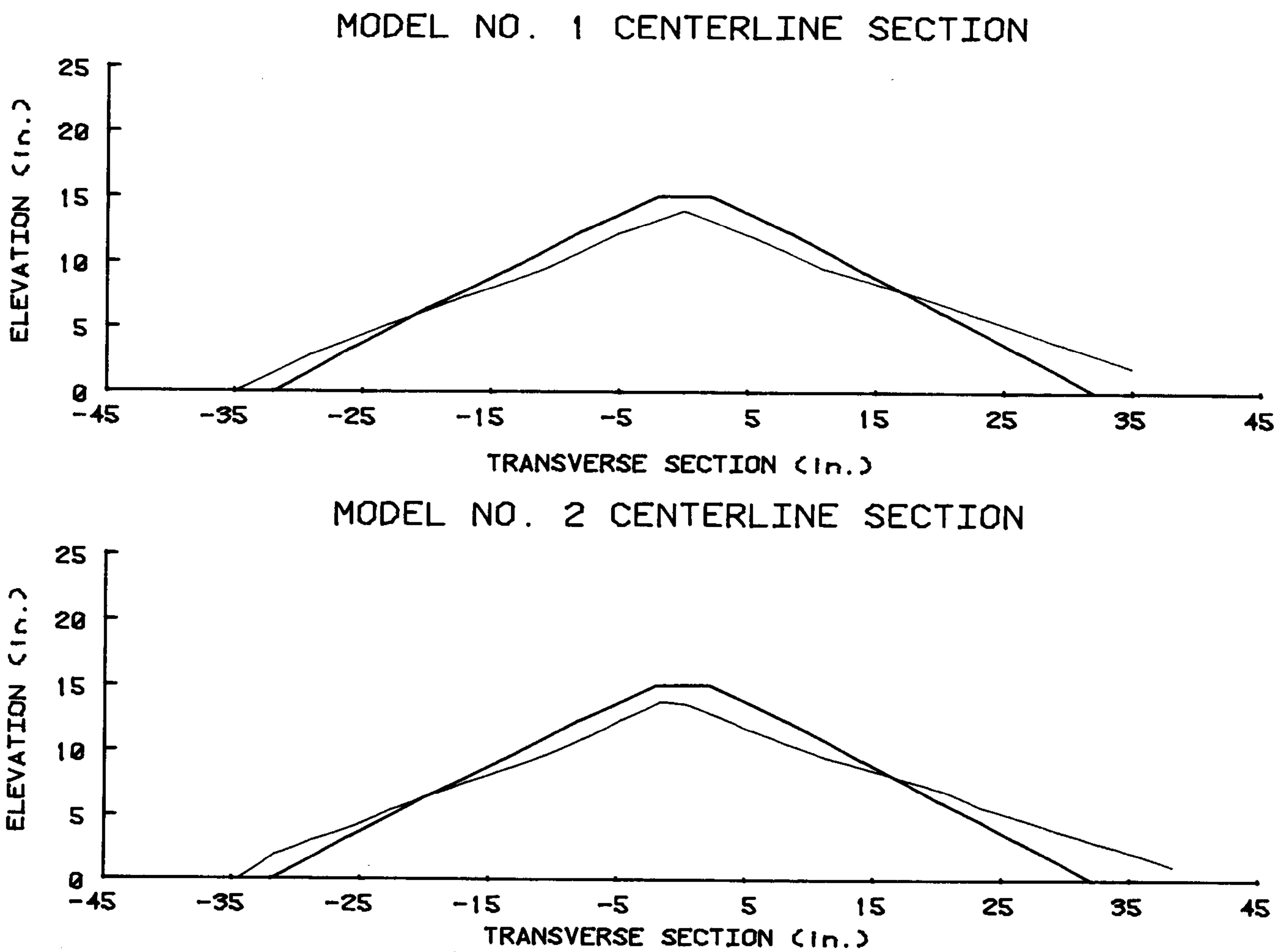
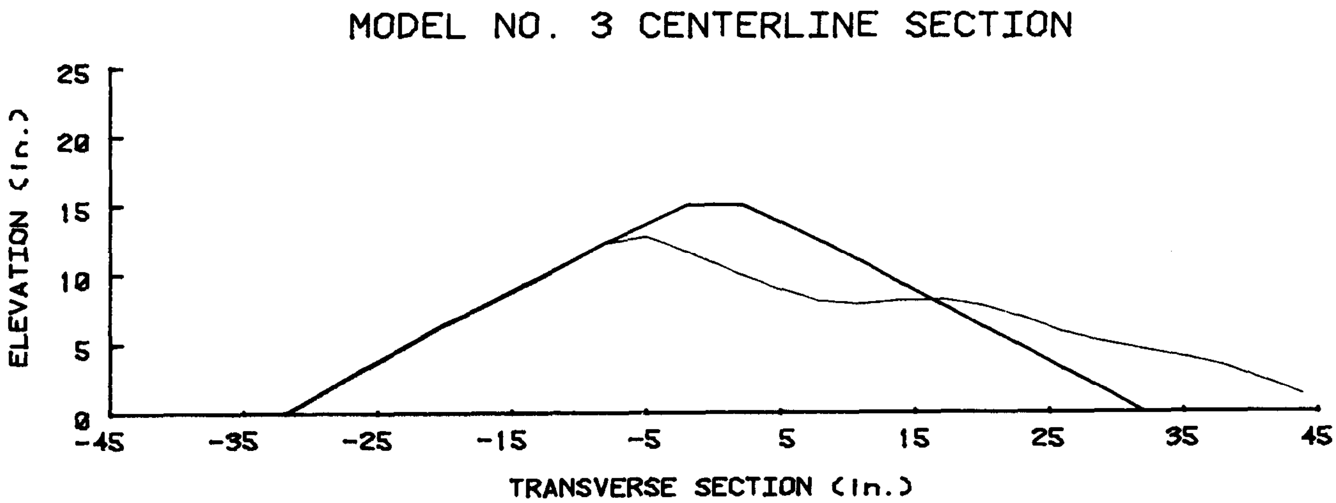


Figure E-6. – Centerline cross sections for models No. 1 and No. 2



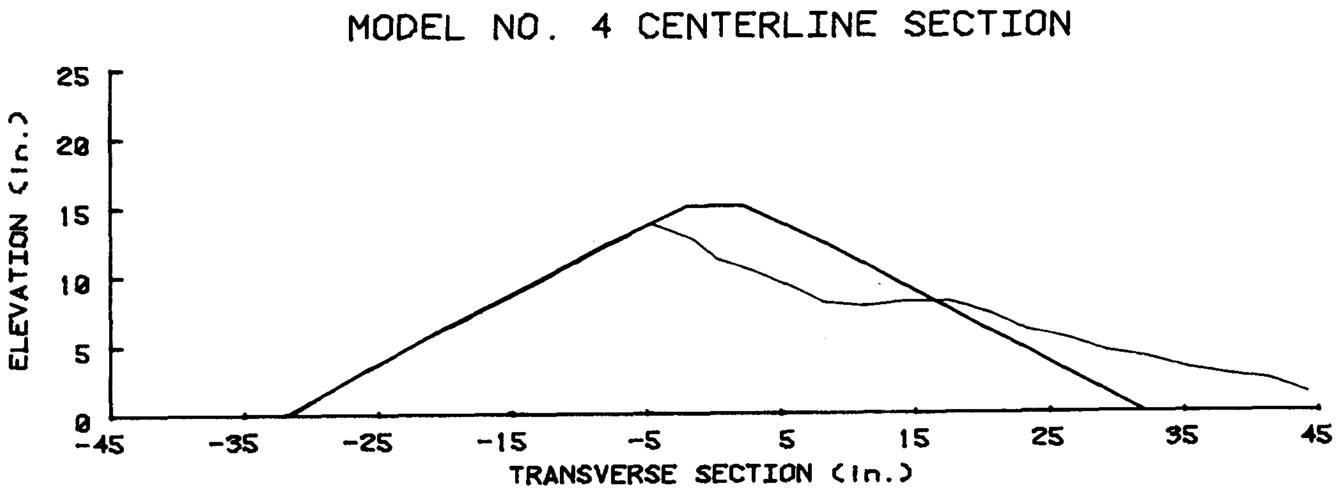


Figure E-7. – Centerline cross sections for models No. 3 and No. 4.

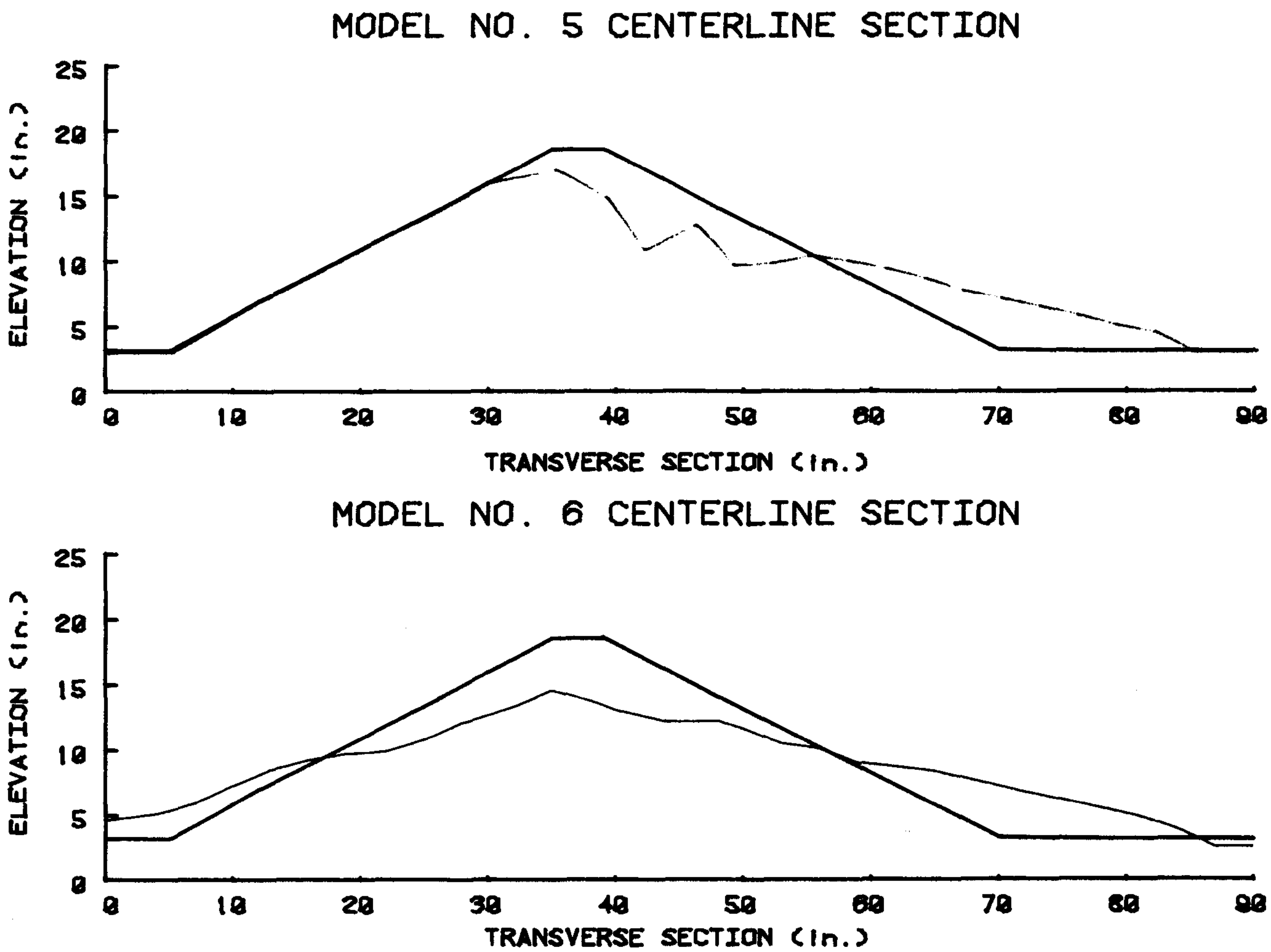
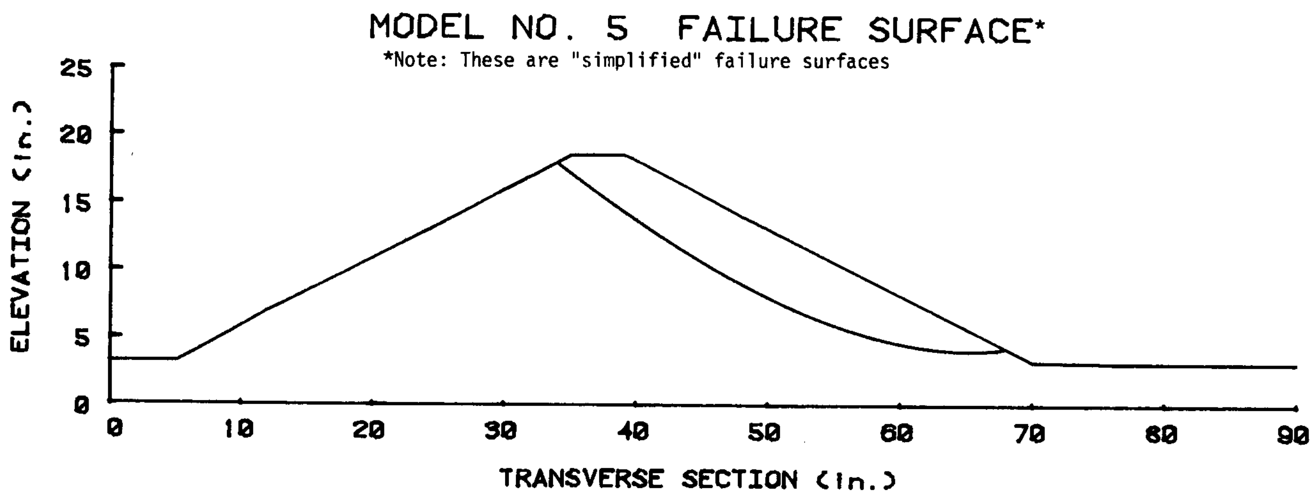


Figure E-8. – Centerline cross sections for models No. 5 and No. 6.



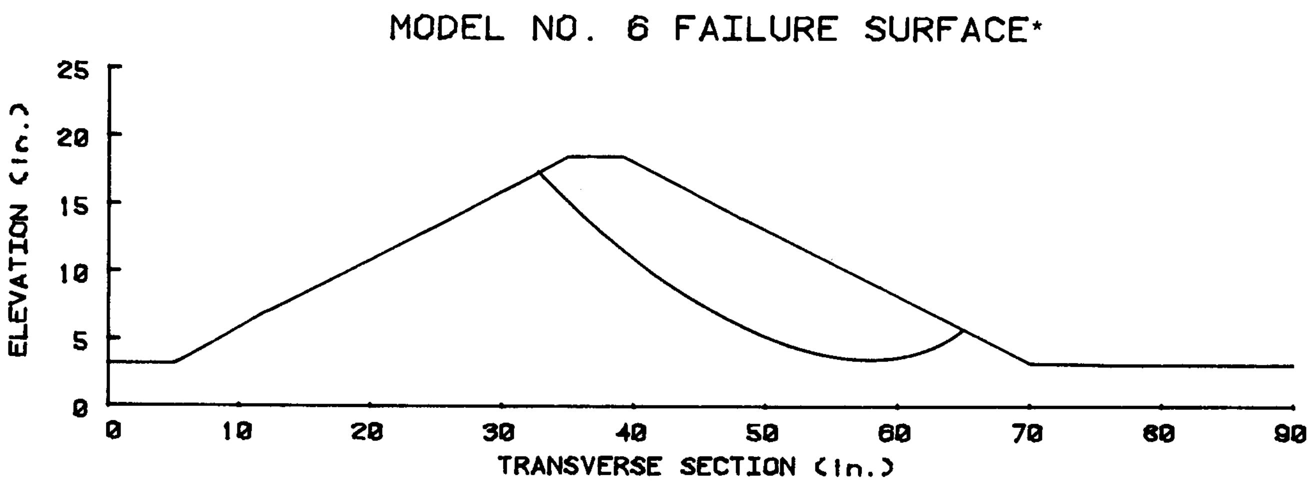
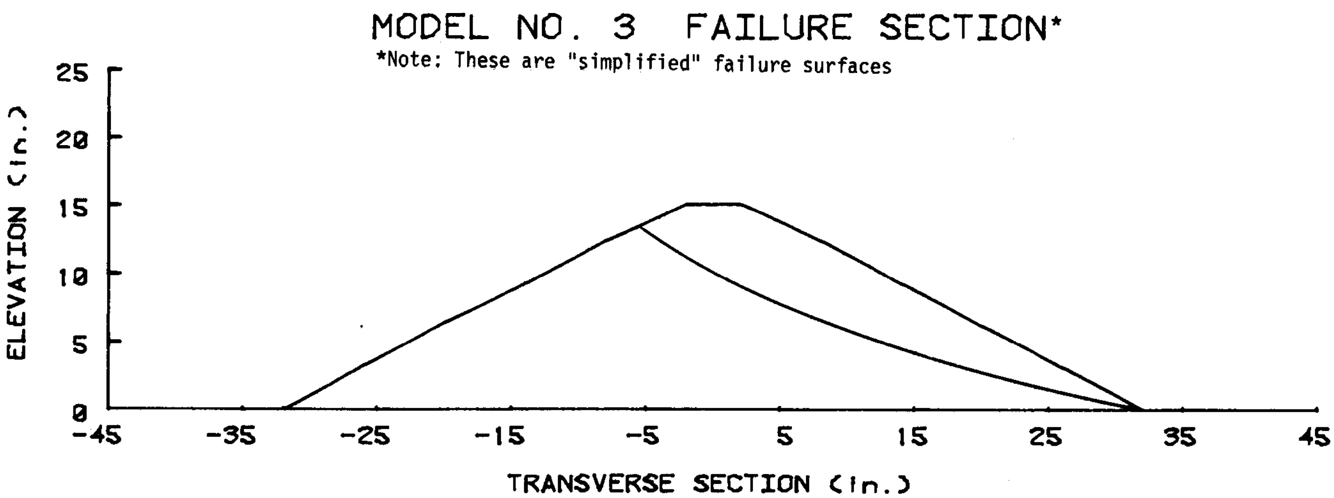


Figure E-9. – Simplified centerline failure surfaces for models No. 3 and No. 4.



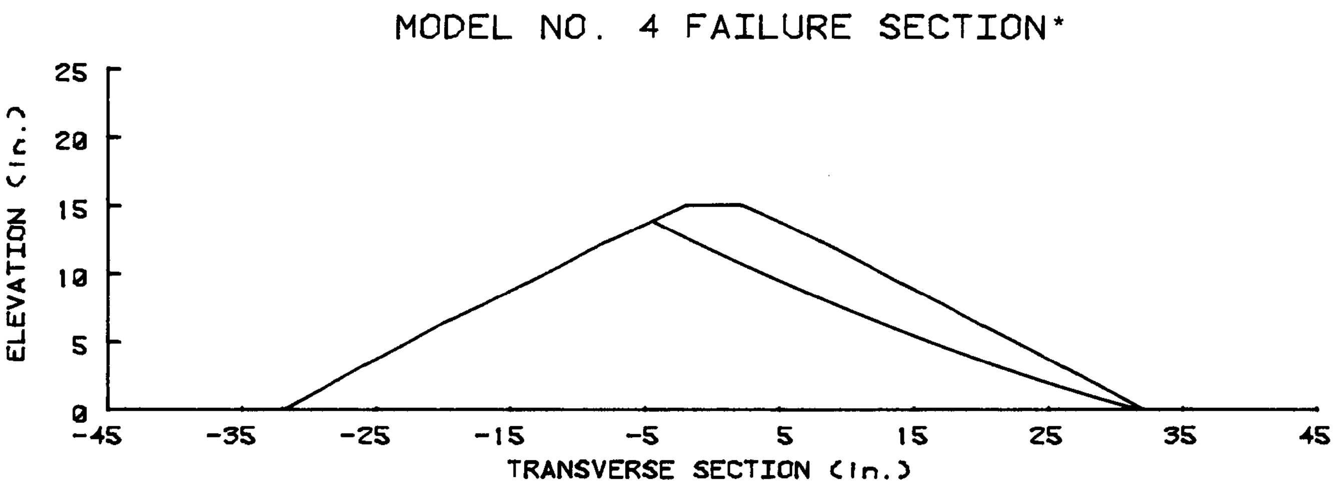


Figure E-10. – Centerline failure surfaces for models No. 5 and No. 6.

Table E-1. – Input data for three-dimensional plots – file DB1.

X	Y	Z
13.	0.	0.
13.	6.	0.
13.	12.	0.
13.	18.	0.
13.	24.	0.
13.	30.	0.
16.	0.	1.1
16.	6.	1.2
16.	12.	1.4
16.	18.	1.3
16.	24.	1.2
16.	30.	1.1
22.	0.	4.4
22.	6.	4.3
22.	12.	4.3
22.	18.	4.3
22.	24.	4.3
22.	30.	4.4
28.	0.	7.3
28.	6.	7.1
28.	12.	7.2
28.	18.	7.3
28.	24.	7.4
28.	30.	7.5
34.	0.	10.4
34.	6.	10.5
34.	12.	10.5
34.	18.	10.5
34.	24.	10.9
34.	30.	10.5
40.	0.	13.7
40.	6.	13.7
40.	12.	13.8
40.	18.	13.7
40.	24.	13.7
40.	30.	13.5
43.	0.	15.1
43.	6.	15.1
43.	12.	15.1
43.	18.	15.1
43.	24.	15.1
43.	30.	15.0
44.	00.0	15.1
44.	06.0	15.1
44.	12.0	15.1
44.	18.0	15.1
44.	24.0	15.1
44.	30.0	15.1
45.	00.0	15.1
45.	06.0	15.1
45.	12.0	15.1
45.	18.0	15.1
45.	24.0	15.1
45.	30.0	15.1
46.	00.0	15.1
46.	06.0	15.1
46.	12.0	15.1
46.	18.0	15.1
46.	24.0	15.1
46.	30.0	15.1
47.	0.	15.1
47.	6.	15.1
47.	12.	15.1
47.	18.	15.1
47.	24.	15.1
47.	30.	15.0
50.	0.	13.7
50.	6.	13.9
50.	12.	13.6
50.	18.	13.6
50.	24.	13.5
50.	30.	13.5
56.	0.	10.6
56.	6.	10.7
56.	12.	10.6
56.	18.	10.8
56.	24.	10.7
56.	30.	10.5
62.	0.	7.6
62.	6.	7.7
62.	12.	7.6
62.	18.	7.6
62.	24.	7.5
62.	30.	7.4
68.	0.	4.4
68.	6.	4.5
68.	12.	4.6
68.	18.	4.6
68.	24.	4.4
68.	30.	4.3
74.	0.	1.5
74.	6.	1.5
74.	12.	1.5
74.	18.	1.4
74.	24.	1.4
74.	30.	1.3
77.	0.	0.
77.	6.0	0.0
77.	12.0	0.0
77.	18.0	0.0
77.	24.0	0.0
77.	30.0	0.0

Table E-2. – Input data for three-dimensional plots – file DB1A.

X	Y	Z	X	Y	Z
10.0	00.0	00.4	45.0	18.0	14.0
10.0	06.0	00.3	45.0	24.0	13.9
10.0	12.0	00.3	45.0	30.0	13.5
10.0	18.0	00.1	50.0	00.0	11.9
10.0	24.0	00.2	50.0	06.0	11.9
10.0	30.0	00.0	50.0	12.0	11.9
16.0	00.0	02.8	50.0	18.0	12.1
16.0	06.0	03.0	50.0	24.0	12.2
16.0	12.0	03.0	50.0	30.0	11.7
16.0	18.0	02.9	56.0	00.0	09.6
16.0	24.0	03.1	56.0	06.0	09.6
16.0	30.0	02.7	56.0	12.0	09.6
22.0	00.0	05.0	56.0	18.0	09.5
22.0	06.0	05.0	56.0	24.0	09.5
22.0	12.0	04.8	56.0	30.0	09.3
22.0	18.0	05.2	62.0	00.0	07.8
22.0	24.0	05.0	62.0	06.0	08.0
22.0	30.0	05.0	62.0	12.0	08.0
28.0	00.0	07.2	62.0	18.0	07.8
28.0	06.0	07.4	62.0	24.0	07.6
28.0	12.0	07.4	62.0	30.0	07.6
28.0	18.0	07.4	68.0	00.0	05.9
28.0	24.0	07.6	68.0	06.0	06.0
28.0	30.0	07.4	68.0	12.0	05.9
34.0	00.0	09.6	68.0	18.0	05.9
34.0	06.0	09.4	68.0	24.0	05.7
34.0	12.0	09.3	68.0	30.0	05.4
34.0	18.0	09.4	74.0	00.0	03.6
34.0	24.0	09.5	74.0	06.0	03.7
34.0	30.0	09.6	74.0	12.0	03.8
40.0	00.0	12.2	74.0	18.0	03.9
40.0	06.0	12.2	74.0	24.0	03.7
40.0	12.0	12.2	74.0	30.0	03.6
40.0	18.0	12.3	80.0	00.0	01.8
40.0	24.0	12.1	80.0	06.0	02.0
40.0	30.0	12.0	80.0	12.0	02.0
45.0	00.0	13.7	80.0	18.0	02.0
45.0	06.0	13.8	80.0	24.0	01.9
45.0	12.0	13.8	80.0	30.0	01.7

Table E-3. – Input data for three-dimensional plots – file DB2A.

X	Y	Z	X	Y	Z	X	Y	Z
10.0	00.0	00.0	28.0	09.0	07.6	45.0	18.0	13.7
10.0	03.0	00.0	28.0	12.0	07.6	45.0	21.0	13.7
10.0	06.0	00.0	28.0	15.0	07.6	45.0	24.0	13.6
10.0	09.0	00.0	28.0	18.0	07.6	45.0	27.0	13.6
10.0	12.0	00.0	28.0	21.0	07.6	45.0	30.0	13.7
10.0	15.0	00.0	28.0	24.0	07.6	47.0	00.0	13.1
10.0	18.0	00.0	28.0	27.0	07.6	47.0	03.0	13.0
10.0	21.0	00.0	28.0	30.0	07.4	47.0	06.0	12.9
10.0	24.0	00.0	31.0	00.0	08.5	47.0	09.0	13.0
10.0	27.0	00.0	31.0	03.0	08.4	47.0	12.0	12.9
10.0	30.0	00.0	31.0	06.0	08.3	47.0	15.0	12.8
13.0	00.0	01.7	31.0	09.0	08.4	47.0	18.0	12.9
13.0	03.0	02.0	31.0	12.0	08.5	47.0	21.0	13.0
13.0	06.0	01.7	31.0	15.0	08.6	47.0	24.0	13.2
13.0	09.0	01.7	31.0	18.0	08.6	47.0	27.0	13.1
13.0	12.0	01.9	31.0	21.0	08.7	47.0	30.0	13.0
13.0	15.0	02.0	31.0	24.0	08.5	50.0	00.0	11.8
13.0	18.0	01.8	31.0	27.0	08.4	50.0	03.0	11.8
13.0	21.0	01.9	31.0	30.0	08.3	50.0	06.0	11.7
13.0	24.0	01.8	34.0	00.0	09.8	50.0	09.0	11.6
13.0	27.0	01.8	34.0	03.0	09.8	50.0	12.0	11.7
13.0	30.0	01.5	34.0	06.0	09.8	50.0	15.0	11.6
16.0	00.0	02.8	34.0	09.0	09.6	50.0	18.0	11.6
16.0	03.0	03.1	34.0	12.0	09.7	50.0	21.0	11.8
16.0	06.0	03.0	34.0	15.0	09.6	50.0	24.0	12.1
16.0	09.0	03.0	34.0	18.0	09.6	50.0	27.0	11.9
16.0	12.0	03.0	34.0	21.0	09.6	50.0	30.0	11.8
16.0	15.0	03.1	34.0	24.0	09.5	53.0	00.0	10.6
16.0	18.0	03.1	34.0	27.0	09.5	53.0	03.0	10.6
16.0	21.0	03.2	34.0	30.0	09.6	53.0	06.0	10.5
16.0	24.0	03.2	37.0	00.0	11.2	53.0	09.0	10.5
16.0	27.0	03.2	37.0	03.0	11.7	53.0	12.0	10.5
16.0	30.0	02.9	37.0	06.0	11.2	53.0	15.0	10.5
19.0	00.0	04.1	37.0	09.0	11.0	53.0	18.0	10.4
19.0	03.0	04.2	37.0	12.0	11.1	53.0	21.0	10.5
19.0	06.0	04.3	37.0	15.0	10.9	53.0	24.0	10.6
19.0	09.0	04.2	37.0	18.0	10.9	53.0	27.0	10.5
19.0	12.0	04.1	37.0	21.0	10.9	53.0	30.0	10.6
19.0	15.0	04.1	37.0	24.0	10.8	56.0	00.0	09.5
19.0	18.0	04.2	37.0	27.0	10.9	56.0	03.0	09.4
19.0	21.0	04.2	37.0	30.0	11.0	56.0	06.0	09.4
19.0	24.0	04.3	40.0	00.0	12.6	56.0	09.0	09.4
19.0	27.0	04.1	40.0	03.0	12.8	56.0	12.0	09.4
19.0	30.0	04.0	40.0	06.0	12.8	56.0	15.0	09.4
22.0	00.0	05.2	40.0	09.0	12.6	56.0	18.0	09.4
22.0	03.0	05.2	40.0	12.0	12.4	56.0	21.0	09.4
22.0	06.0	05.5	40.0	15.0	12.4	56.0	24.0	09.3
22.0	09.0	05.4	40.0	18.0	12.2	56.0	30.0	09.6
22.0	12.0	05.5	40.0	21.0	12.2	59.0	00.0	08.7
22.0	15.0	05.5	40.0	24.0	12.2	59.0	03.0	08.7
22.0	18.0	05.5	40.0	27.0	12.4	59.0	06.0	08.7
22.0	21.0	05.4	40.0	30.0	12.4	59.0	09.0	08.6
22.0	24.0	05.2	43.0	00.0	14.2	59.0	12.0	08.6
22.0	27.0	05.2	43.0	03.0	14.3	59.0	15.0	08.7
22.0	30.0	05.2	43.0	06.0	14.2	59.0	18.0	08.7
25.0	00.0	06.4	43.0	09.0	14.0	59.0	21.0	08.6
25.0	03.0	06.5	43.0	12.0	13.8	59.0	24.0	08.5
25.0	06.0	06.5	43.0	15.0	13.8	59.0	27.0	08.7
25.0	09.0	06.6	43.0	18.0	13.6	59.0	30.0	08.5
25.0	12.0	06.6	43.0	21.0	13.6	62.0	00.0	07.8
25.0	15.0	06.6	43.0	24.0	13.7	62.0	03.0	07.7
25.0	18.0	06.6	43.0	27.0	13.8	62.0	06.0	07.7
25.0	21.0	06.5	43.0	30.0	13.7	62.0	09.0	07.7
25.0	24.0	06.5	45.0	00.0	13.9	62.0	12.0	07.7
25.0	27.0	06.6	45.0	03.0	13.7	62.0	15.0	07.8
25.0	30.0	06.5	45.0	06.0	13.8	62.0	18.0	07.7
28.0	00.0	07.4	45.0	09.0	13.8	62.0	21.0	07.8
28.0	03.0	07.3	45.0	12.0	13.6	62.0	24.0	07.6
28.0	06.0	07.5	45.0	15.0	13.6	62.0	27.0	07.8

Table E-3. – Input data for three-dimensional plots – file DB2A. – continued

X	Y	Z	X	Y	Z
62.0	30.0	07.8	74.0	15.0	03.9
65.0	00.0	06.6	74.0	18.0	03.8
65.0	03.0	06.6	74.0	21.0	03.9
65.0	06.0	06.7	74.0	24.0	03.9
65.0	09.0	06.9	74.0	27.0	03.9
65.0	12.0	06.9	74.0	30.0	03.7
65.0	15.0	07.0	77.0	00.0	02.8
65.0	18.0	07.0	77.0	03.0	02.9
65.0	21.0	06.9	77.0	06.0	03.0
65.0	24.0	06.9	77.0	09.0	03.0
65.0	27.0	06.9	77.0	12.0	02.9
65.0	30.0	06.9	77.0	15.0	03.0
68.0	00.0	05.6	77.0	18.0	03.0
68.0	03.0	05.6	77.0	21.0	03.0
68.0	06.0	05.6	77.0	24.0	03.1
68.0	09.0	05.6	77.0	27.0	03.1
68.0	12.0	05.7	77.0	30.0	03.0
68.0	15.0	05.7	80.0	00.0	01.7
68.0	18.0	05.7	80.0	03.0	02.0
68.0	21.0	05.7	80.0	06.0	02.0
68.0	24.0	05.7	80.0	09.0	02.1
68.0	27.0	05.6	80.0	12.0	02.1
68.0	30.0	05.7	80.0	15.0	02.1
71.0	00.0	04.7	80.0	18.0	02.1
71.0	03.0	04.7	80.0	21.0	02.2
71.0	06.0	04.8	80.0	24.0	02.3
71.0	09.0	04.8	80.0	27.0	02.2
71.0	12.0	04.8	80.0	30.0	02.0
71.0	15.0	04.8	83.0	00.0	00.9
71.0	18.0	04.7	83.0	03.0	01.0
71.0	21.0	04.8	83.0	06.0	01.1
71.0	24.0	04.9	83.0	09.0	01.1
71.0	27.0	05.2	83.0	12.0	01.1
71.0	30.0	04.8	83.0	15.0	01.1
74.0	00.0	03.7	83.0	18.0	01.1
74.0	03.0	03.9	83.0	21.0	01.1
74.0	06.0	03.8	83.0	24.0	01.1
74.0	09.0	03.9	83.0	27.0	01.1
74.0	12.0	03.8	83.0	30.0	01.0

Table E-4. – Input data for three-dimensional plots – file DB3A.

X	Y	Z	X	Y	Z	X	Y	Z
13.0	30.0	00.0	43.0	12.0	12.0	59.0	03.0	08.1
13.0	27.0	00.0	43.0	09.0	13.3	59.0	00.0	08.5
13.0	24.0	00.0	43.0	06.0	14.3	62.0	30.0	06.9
13.0	21.0	00.0	43.0	03.0	14.5	62.0	27.0	07.0
13.0	18.0	00.0	43.0	00.0	14.6	62.0	24.0	07.6
13.0	15.0	00.0	45.0	30.0	13.3	62.0	21.0	08.0
13.0	12.0	00.0	45.0	27.0	12.6	62.0	18.0	08.1
13.0	09.0	00.0	45.0	24.0	11.8	62.0	15.0	08.2
13.0	06.0	00.0	45.0	21.0	11.1	62.0	12.0	08.1
13.0	03.0	00.0	45.0	18.0	10.8	62.0	09.0	07.9
13.0	00.0	00.0	45.0	15.0	10.8	62.0	06.0	07.8
19.0	30.0	03.0	45.0	12.0	10.9	62.0	03.0	08.0
19.0	27.0	03.3	45.0	09.0	11.1	62.0	00.0	08.0
19.0	24.0	03.3	45.0	06.0	12.4	65.0	30.0	06.2
19.0	21.0	03.3	45.0	03.0	13.3	65.0	27.0	06.4
19.0	18.0	03.3	45.0	00.0	13.7	65.0	24.0	07.0
19.0	15.0	03.3	47.0	30.0	12.7	65.0	21.0	07.4
19.0	12.0	03.3	47.0	27.0	12.0	65.0	18.0	07.8
19.0	09.0	03.3	47.0	24.0	10.8	65.0	15.0	07.8
19.0	06.0	03.3	47.0	21.0	10.2	65.0	12.0	07.6
19.0	03.0	03.2	47.0	18.0	10.2	65.0	09.0	07.4
19.0	00.0	03.2	47.0	15.0	10.2	65.0	06.0	07.3
25.0	30.0	06.1	47.0	12.0	10.0	65.0	03.0	07.4
25.0	27.0	06.2	47.0	09.0	11.0	65.0	00.0	07.6
25.0	24.0	06.2	47.0	06.0	12.0	68.0	30.0	05.4
25.0	21.0	06.3	47.0	03.0	12.5	68.0	27.0	05.8
25.0	18.0	06.4	47.0	00.0	13.6	68.0	24.0	06.4
25.0	15.0	06.5	50.0	30.0	11.5	68.0	21.0	06.7
25.0	12.0	06.5	50.0	27.0	10.7	68.0	18.0	06.8
25.0	09.0	06.4	50.0	24.0	09.3	68.0	15.0	06.9
25.0	06.0	06.3	50.0	21.0	08.9	68.0	12.0	06.7
25.0	03.0	06.3	50.0	18.0	08.9	68.0	09.0	06.4
25.0	00.0	06.2	50.0	15.0	08.9	68.0	06.0	06.2
31.0	30.0	09.1	50.0	12.0	09.1	68.0	03.0	06.4
31.0	21.0	09.2	50.0	09.0	09.5	68.0	00.0	06.7
31.0	18.0	09.3	50.0	06.0	10.4	71.0	30.0	05.0
31.0	15.0	09.3	50.0	03.0	10.9	71.0	27.0	05.2
31.0	09.0	09.3	50.0	00.0	12.0	71.0	24.0	05.7
31.0	06.0	09.2	53.0	30.0	09.8	71.0	21.0	05.9
31.0	03.0	09.2	53.0	27.0	08.7	71.0	18.0	05.9
31.0	00.0	09.2	53.0	24.0	08.7	71.0	15.0	05.9
37.0	30.0	12.2	53.0	21.0	08.3	71.0	12.0	05.7
37.0	27.0	12.2	53.0	18.0	07.5	71.0	09.0	05.5
37.0	24.0	12.2	53.0	15.0	08.0	71.0	06.0	05.3
37.0	21.0	12.2	53.0	12.0	08.1	71.0	03.0	05.3
37.0	18.0	12.2	53.0	09.0	08.2	71.0	00.0	05.6
37.0	15.0	12.3	53.0	06.0	08.9	74.0	30.0	04.5
37.0	12.0	12.3	53.0	03.0	09.8	74.0	27.0	04.8
37.0	09.0	12.3	53.0	00.0	10.8	74.0	24.0	05.2
37.0	06.0	12.3	56.0	30.0	08.1	74.0	21.0	05.3
37.0	03.0	12.3	56.0	27.0	08.0	74.0	18.0	05.2
37.0	00.0	12.4	56.0	24.0	07.8	74.0	15.0	05.1
40.0	30.0	13.1	56.0	21.0	07.5	74.0	12.0	04.9
40.0	27.0	13.5	56.0	18.0	07.5	74.0	09.0	04.6
40.0	24.0	13.5	56.0	15.0	07.8	74.0	06.0	04.4
40.0	21.0	13.5	56.0	12.0	07.6	74.0	03.0	04.3
40.0	18.0	13.5	56.0	09.0	07.6	74.0	00.0	04.4
40.0	15.0	12.8	56.0	06.0	07.4	77.0	30.0	04.1
40.0	12.0	12.8	56.0	03.0	08.6	77.0	27.0	04.3
40.0	09.0	13.5	56.0	00.0	09.7	77.0	24.0	04.7
40.0	06.0	13.5	59.0	30.0	07.6	77.0	21.0	04.7
40.0	03.0	13.5	59.0	27.0	07.2	77.0	18.0	04.6
40.0	00.0	13.5	59.0	24.0	07.4	77.0	15.0	04.5
43.0	30.0	13.8	59.0	21.0	07.8	77.0	12.0	04.3
43.0	27.0	13.0	59.0	18.0	08.2	77.0	09.0	04.1
43.0	24.0	12.3	59.0	15.0	08.1	77.0	06.0	03.7
43.0	21.0	12.1	59.0	12.0	07.9	77.0	03.0	03.1
43.0	18.0	11.9	59.0	09.0	07.6	77.0	00.0	02.8
43.0	15.0	11.7	59.0	06.0	07.4	80.0	30.0	01.9

Table E-4. – Input data for three-dimensional plots – file DB3A. – continued

X	Y	Z	X	Y	Z
80.0	27.0	02.4	86.0	27.0	02.4
80.0	24.0	03.1	86.0	24.0	02.8
80.0	21.0	03.6	86.0	21.0	02.8
80.0	18.0	03.9	86.0	18.0	02.4
80.0	15.0	04.0	86.0	15.0	02.4
80.0	12.0	04.1	86.0	12.0	02.4
80.0	09.0	04.3	86.0	09.0	02.3
80.0	06.0	04.3	86.0	06.0	01.7
80.0	03.0	03.8	86.0	03.0	01.0
80.0	00.0	03.6	86.0	00.0	00.2
83.0	30.0	03.1	89.0	30.0	01.2
83.0	27.0	03.2	89.0	27.0	01.6
83.0	24.0	03.7	89.0	24.0	01.8
83.0	21.0	03.7	89.0	21.0	01.7
83.0	18.0	03.9	89.0	18.0	01.3
83.0	15.0	03.4	89.0	15.0	01.3
83.0	12.0	03.3	89.0	12.0	01.3
83.0	09.0	03.0	89.0	09.0	00.9
83.0	06.0	02.6	89.0	06.0	00.3
83.0	03.0	01.8	89.0	03.0	00.0
83.0	00.0	01.2	89.0	00.0	00.0
86.0	30.0	02.2			

Table E-5. – Input data for three-dimensional plots – file DB4A.

X	Y	Z	X	Y	Z	X	Y	Z
13.0	30.0	00.0	47.0	12.0	10.6	65.0	06.0	06.7
13.0	24.0	00.0	47.0	09.0	11.0	65.0	03.0	06.7
13.0	18.0	00.0	47.0	06.0	11.4	65.0	00.0	06.7
13.0	12.0	00.0	47.0	03.0	12.0	68.0	30.0	06.4
13.0	06.0	00.0	47.0	00.0	13.2	68.0	27.0	06.2
13.0	00.0	00.0	50.0	30.0	11.5	68.0	24.0	06.1
19.0	30.0	03.1	50.0	27.0	10.6	68.0	21.0	06.1
19.0	24.0	03.1	50.0	24.0	10.1	68.0	18.0	06.1
19.0	18.0	03.2	50.0	21.0	09.7	68.0	15.0	06.2
19.0	12.0	03.4	50.0	18.0	09.4	68.0	12.0	06.2
19.0	06.0	03.3	50.0	15.0	09.4	68.0	09.0	06.0
19.0	00.0	03.1	50.0	12.0	09.3	68.0	06.0	05.8
25.0	30.0	06.2	50.0	09.0	09.2	68.0	03.0	05.5
25.0	24.0	06.3	50.0	06.0	09.6	68.0	00.0	05.4
25.0	18.0	06.4	50.0	03.0	10.4	71.0	30.0	05.4
25.0	12.0	06.3	50.0	00.0	11.9	71.0	27.0	05.3
25.0	06.0	06.1	53.0	30.0	10.1	71.0	24.0	05.2
25.0	00.0	06.0	53.0	27.0	09.1	71.0	21.0	05.3
31.0	30.0	09.1	53.0	24.0	08.7	71.0	18.0	05.4
31.0	24.0	09.3	53.0	21.0	08.3	71.0	15.0	05.4
31.0	18.0	09.3	53.0	18.0	08.1	71.0	12.0	05.5
31.0	12.0	09.3	53.0	15.0	08.0	71.0	09.0	05.5
31.0	06.0	09.4	53.0	12.0	08.0	71.0	06.0	05.1
31.0	00.0	09.2	53.0	09.0	08.0	71.0	03.0	04.8
37.0	30.0	12.3	53.0	06.0	08.0	71.0	00.0	04.9
37.0	24.0	12.4	53.0	03.0	08.9	74.0	30.0	04.1
37.0	18.0	12.4	53.0	00.0	10.4	74.0	27.0	04.3
37.0	12.0	12.3	56.0	30.0	09.1	74.0	24.0	04.3
37.0	06.0	12.3	56.0	27.0	08.0	74.0	21.0	04.4
37.0	00.0	12.3	56.0	24.0	07.2	74.0	18.0	04.5
40.0	30.0	13.5	56.0	21.0	07.4	74.0	15.0	04.6
40.0	27.0	13.7	56.0	18.0	07.6	74.0	12.0	04.7
40.0	24.0	13.9	56.0	15.0	07.8	74.0	09.0	04.6
40.0	21.0	13.9	56.0	12.0	07.6	74.0	06.0	04.4
40.0	18.0	13.8	56.0	09.0	07.6	74.0	03.0	03.9
40.0	15.0	13.8	56.0	06.0	07.6	74.0	00.0	04.1
40.0	12.0	13.7	56.0	03.0	08.0	77.0	30.0	03.7
40.0	09.0	13.8	56.0	00.0	09.1	77.0	27.0	03.6
40.0	06.0	13.9	59.0	30.0	08.0	77.0	24.0	03.6
40.0	03.0	13.9	59.0	27.0	07.6	77.0	21.0	03.8
40.0	00.0	14.1	59.0	24.0	07.7	77.0	18.0	03.8
43.0	30.0	15.0	59.0	24.0	07.7	77.0	15.0	03.9
43.0	27.0	14.4	59.0	21.0	07.8	77.0	12.0	04.1
43.0	24.0	13.7	59.0	18.0	08.1	77.0	09.0	04.0
43.0	21.0	13.1	59.0	15.0	08.1	77.0	06.0	03.9
43.0	18.0	12.8	59.0	12.0	08.2	77.0	03.0	03.4
43.0	15.0	12.8	59.0	09.0	08.0	77.0	00.0	03.6
43.0	12.0	12.8	59.0	06.0	08.0	80.0	30.0	02.3
43.0	09.0	12.8	59.0	03.0	07.7	80.0	27.0	02.6
43.0	06.0	13.1	59.0	00.0	07.7	80.0	24.0	03.1
43.0	03.0	13.9	62.0	30.0	07.6	80.0	21.0	03.1
43.0	00.0	14.8	62.0	27.0	07.8	80.0	18.0	03.3
45.0	30.0	14.6	62.0	24.0	08.0	80.0	15.0	03.3
45.0	27.0	13.1	62.0	21.0	08.0	80.0	12.0	03.5
45.0	24.0	12.7	62.0	18.0	08.1	80.0	09.0	03.4
45.0	21.0	12.0	62.0	15.0	08.2	80.0	06.0	03.2
45.0	18.0	11.5	62.0	12.0	08.4	80.0	03.0	02.8
45.0	15.0	11.2	62.0	09.0	08.4	80.0	00.0	03.0
45.0	12.0	11.2	62.0	06.0	08.2	83.0	30.0	01.5
45.0	09.0	11.9	62.0	03.0	07.8	83.0	27.0	02.1
45.0	06.0	12.3	62.0	00.0	07.4	83.0	24.0	02.7
45.0	03.0	13.0	65.0	30.0	07.2	83.0	21.0	02.8
45.0	00.0	14.2	65.0	27.0	07.0	83.0	18.0	02.8
47.0	30.0	13.4	65.0	24.0	07.1	83.0	15.0	02.9
47.0	27.0	12.0	65.0	21.0	07.2	83.0	12.0	02.8
47.0	24.0	11.3	65.0	18.0	07.4	83.0	09.0	02.7
47.0	21.0	11.1	65.0	15.0	07.5	83.0	06.0	02.6
47.0	18.0	10.6	65.0	12.0	07.4	83.0	03.0	02.4
47.0	15.0	10.6	65.0	09.0	07.1	83.0	00.0	02.5

Table E-5. – Input data for three-dimensional plots – file DB4A. – continued

X	Y	Z
86.0	30.0	00.5
86.0	27.0	01.3
86.0	24.0	02.6
86.0	21.0	
86.0	18.0	02.4
86.0	15.0	02.4
86.0	12.0	02.1
86.0	09.0	02.0
86.0	06.0	02.0
86.0	03.0	01.7
86.0	00.0	01.7
89.0	30.0	00./
89.0	27.0	00.2
89.0	24.0	01.1
89.0	21.0	01.3
89.0	18.0	01.6
89.0	15.0	01.4
89.0	12.0	01.5
89.0	09.0	01.3
89.0	06.0	01.3
89.0	03.0	01.1
89.0	00.0	00.1
END,,RL		

Table E-6. – Input data for three-dimensional plots – file DB5A.

X	Y	Z	X	Y	Z	X	Y	Z
00.9	30.0	00.0	31.0	21.0	15.5	53.0	12.0	10.0
00.9	27.0	00.0	31.0	18.0	15.5	53.0	09.0	09.9
00.9	24.0	00.0	31.0	15.0	15.5	53.0	06.0	09.4
00.9	21.0	00.0	31.0	12.0	15.5	53.0	03.0	09.5
00.9	18.0	00.0	31.0	09.0	15.5	53.0	00.0	10.4
00.9	15.0	00.0	31.0	06.0	15.5	56.0	30.0	09.0
00.9	12.0	00.0	31.0	03.0	15.5	56.0	27.0	09.2
00.9	09.0	00.0	31.0	00.0	15.5	56.0	24.0	09.6
00.9	06.0	00.0	36.0	30.0	18.3	56.0	21.0	09.5
00.9	03.0	00.0	36.0	27.0	17.4	56.0	18.0	09.9
00.9	00.0	00.0	36.0	24.0	16.9	56.0	15.0	10.3
01.0	30.0	03.0	36.0	21.0	17.2	56.0	12.0	10.6
01.0	27.0	03.0	36.0	18.0	17.2	56.0	09.0	10.6
01.0	24.0	03.0	36.0	15.0	17.2	56.0	06.0	10.3
01.0	21.0	03.0	36.0	12.0	16.9	56.0	03.0	10.0
01.0	18.0	03.0	36.0	09.0	16.9	56.0	00.0	09.6
01.0	15.0	03.0	36.0	06.0	17.4	59.0	30.0	08.9
01.0	12.0	03.0	36.0	03.0	17.6	59.0	27.0	09.5
01.0	09.0	03.0	36.0	00.0	18.3	59.0	24.0	09.9
01.0	06.0	03.0	40.0	30.0	17.9	59.0	21.0	10.0
01.0	03.0	03.0	40.0	27.0	16.4	59.0	18.0	09.9
01.0	00.0	03.0	40.0	24.0	15.4	59.0	15.0	09.9
06.0	30.0	03.0	40.0	21.0	14.8	59.0	12.0	10.0
06.0	27.0	03.0	40.0	18.0	14.8	59.0	09.0	10.2
06.0	24.0	03.0	40.0	15.0	14.7	59.0	06.0	10.0
06.0	21.0	03.0	40.0	12.0	14.8	59.0	03.0	09.7
06.0	18.0	03.0	40.0	09.0	15.1	59.0	00.0	09.6
06.0	15.0	03.0	40.0	06.0	15.2	62.0	30.0	08.4
06.0	12.0	03.0	40.0	03.0	16.0	62.0	27.0	08.8
06.0	09.0	03.0	40.0	00.0	17.4	62.0	24.0	09.1
06.0	06.0	03.0	43.0	30.0	12.8	62.0	21.0	09.3
06.0	03.0	03.0	43.0	27.0	11.7	62.0	18.0	09.2
06.0	00.0	03.0	43.0	24.0	11.1	62.0	15.0	09.3
13.0	30.0	06.5	43.0	21.0	10.8	62.0	12.0	09.3
13.0	27.0	06.5	43.0	18.0	10.9	62.0	09.0	09.4
13.0	24.0	06.5	43.0	15.0	10.7	62.0	06.0	09.3
13.0	21.0	06.5	43.0	12.0	10.4	62.0	03.0	08.9
13.0	18.0	06.5	43.0	09.0	10.5	62.0	00.0	08.6
13.0	15.0	06.5	43.0	06.0	10.9	65.0	30.0	07.7
13.0	12.0	06.5	43.0	03.0	11.2	65.0	27.0	08.1
13.0	09.0	06.5	43.0	00.0	12.7	65.0	24.0	08.5
13.0	06.0	06.5	47.0	30.0	14.6	65.0	21.0	08.7
13.0	03.0	06.5	47.0	27.0	13.1	65.0	18.0	08.5
13.0	00.0	06.5	47.0	24.0	12.5	65.0	15.0	08.5
19.0	30.0	09.5	47.0	21.0	12.2	65.0	12.0	08.5
19.0	27.0	09.5	47.0	18.0	12.3	65.0	09.0	08.5
19.0	24.0	09.5	47.0	15.0	12.6	65.0	06.0	08.2
19.0	21.0	09.5	47.0	12.0	13.0	65.0	03.0	07.8
19.0	18.0	09.5	47.0	09.0	12.8	65.0	00.0	07.7
19.0	15.0	09.5	47.0	06.0	13.0	68.0	30.0	06.9
19.0	12.0	09.5	47.0	03.0	13.9	68.0	27.0	07.3
19.0	09.0	09.5	47.0	00.0	14.9	68.0	24.0	07.6
19.0	06.0	09.5	50.0	30.0	11.0	68.0	21.0	07.8
19.0	03.0	09.5	50.0	27.0	10.6	68.0	18.0	07.7
19.0	00.0	09.5	50.0	24.0	10.4	68.0	15.0	07.6
25.0	30.0	12.5	50.0	21.0	10.5	68.0	12.0	07.5
25.0	27.0	12.5	50.0	18.0	10.4	68.0	09.0	07.3
25.0	24.0	12.5	50.0	15.0	09.6	68.0	06.0	07.3
25.0	21.0	12.5	50.0	12.0	09.8	68.0	03.0	07.0
25.0	18.0	12.5	50.0	09.0	10.0	68.0	00.0	07.0
25.0	15.0	12.5	50.0	06.0	10.6	71.0	30.0	06.0
25.0	12.0	12.5	50.0	03.0	10.9	71.0	27.0	06.5
25.0	09.0	12.5	50.0	00.0	11.3	71.0	24.0	07.0
25.0	06.0	12.5	53.0	30.0	12.2	71.0	21.0	07.2
25.0	03.0	12.5	53.0	27.0	09.1	71.0	18.0	07.1
25.0	00.0	12.5	53.0	24.0	08.5	71.0	15.0	07.0
31.0	30.0	15.5	53.0	21.0	08.7	71.0	12.0	06.7
31.0	27.0	15.5	53.0	18.0	08.9	71.0	09.0	06.7
31.0	24.0	15.5	53.0	15.0	09.7	71.0	06.0	06.9

Table E-6. – Input data for three-dimensional plots – file DB5A. – continued

X	Y	Z	X	Y	Z
71.0	03.0	06.2	83.0	15.0	04.4
71.0	00.0	06.0	83.0	12.0	03.9
74.0	30.0	05.7	83.0	09.0	03.9
74.0	27.0	06.0	83.0	06.0	03.1
74.0	24.0	06.4	83.0	03.0	03.0
74.0	21.0	06.5	83.0	00.0	03.0
74.0	18.0	06.5	86.0	30.0	03.0
74.0	15.0	06.4	86.0	27.0	03.0
74.0	12.0	06.4	86.0	24.0	03.0
74.0	09.0	05.9	86.0	21.0	03.0
74.0	06.0	06.2	86.0	18.0	02.9
74.0	03.0	05.4	86.0	15.0	02.9
74.0	00.0	04.9	86.0	12.0	02.9
77.0	30.0	04.6	86.0	09.0	02.9
77.0	27.0	05.3	86.0	06.0	03.1
77.0	24.0	05.6	86.0	03.0	03.1
77.0	21.0	05.7	86.0	00.0	03.1
77.0	18.0	05.8	89.0	30.0	03.0
77.0	15.0	05.7	89.0	27.0	03.0
77.0	12.0	05.4	89.0	24.0	03.0
77.0	09.0	05.0	89.0	21.0	03.0
77.0	06.0	04.8	89.0	18.0	03.0
77.0	03.0	04.6	89.0	15.0	02.9
77.0	00.0	04.4	89.0	12.0	02.9
80.0	30.0	04.0	89.0	09.0	02.9
80.0	27.0	04.6	89.0	06.0	02.9
80.0	24.0	04.8	89.0	03.0	03.1
80.0	21.0	04.8	89.0	00.0	03.1
80.0	18.0	04.8	89.1	30.0	00.0
80.0	15.0	04.9	89.1	27.0	00.0
80.0	12.0	04.8	89.1	24.0	00.0
80.0	09.0	04.3	89.1	21.0	00.0
80.0	06.0	04.1	89.1	18.0	00.0
80.0	03.0	04.0	89.1	15.0	00.0
80.0	00.0	03.5	89.1	12.0	00.0
83.0	30.0	03.1	89.1	09.0	00.0
83.0	27.0	03.2	89.1	06.0	00.0
83.0	24.0	03.7	89.1	03.0	00.0
83.0	21.0	03.7	89.1	00.0	00.0
83.0	18.0	03.8			

Table E-7. – Input data for three-dimensional plots – file DB5B.

X	Y	Z	X	Y	Z	X	Y	Z
00.9	30.0	00.0	06.0	21.0	03.0	38.0	12.0	18.5
00.9	27.0	00.0	06.0	18.0	03.0	38.0	09.0	18.5
00.9	24.0	00.0	06.0	15.0	03.0	38.0	06.0	18.5
00.9	21.0	00.0	06.0	12.0	03.0	38.0	03.0	18.5
00.9	18.0	00.0	06.0	09.0	03.0	38.0	00.0	18.5
00.9	15.0	00.0	06.0	06.0	03.0	39.0	30.0	18.5
00.9	12.0	00.0	06.0	03.0	03.0	39.0	27.0	18.5
00.9	09.0	00.0	06.0	00.0	03.0	39.0	24.0	18.5
00.9	06.0	00.0	13.0	30.0	06.5	39.0	21.0	18.5
00.9	03.0	00.0	13.0	27.0	06.5	39.0	18.0	18.5
00.9	00.0	00.0	13.0	24.0	06.5	39.0	15.0	18.5
01.0	30.0	03.0	13.0	21.0	06.5	39.0	12.0	18.5
01.0	27.0	03.0	13.0	18.0	06.5	39.0	09.0	18.5
01.0	24.0	03.0	13.0	15.0	06.5	39.0	06.0	18.5
01.0	21.0	03.0	13.0	12.0	06.5	39.0	03.0	18.5
01.0	18.0	03.0	13.0	09.0	06.5	39.0	00.0	18.5
01.0	15.0	03.0	13.0	06.0	06.5	40.0	30.0	18.5
01.0	12.0	03.0	13.0	03.0	06.5	40.0	27.0	18.5
01.0	09.0	03.0	13.0	00.0	06.5	40.0	24.0	18.5
01.0	06.0	03.0	19.0	30.0	09.5	40.0	21.0	18.5
01.0	03.0	03.0	19.0	27.0	09.5	40.0	18.0	18.5
01.0	00.0	03.0	19.0	24.0	09.5	40.0	15.0	18.5
03.0	30.0	03.0	19.0	21.0	09.5	40.0	12.0	18.5
03.0	27.0	03.0	19.0	18.0	09.5	40.0	09.0	18.5
03.0	24.0	03.0	19.0	15.0	09.5	40.0	06.0	18.5
03.0	21.0	03.0	19.0	12.0	09.5	40.0	03.0	18.5
03.0	18.0	03.0	19.0	09.0	09.5	40.0	00.0	18.5
03.0	15.0	03.0	19.0	06.0	09.5	40.5	30.0	18.5
03.0	12.0	03.0	19.0	03.0	09.5	40.5	27.0	18.5
03.0	09.0	03.0	19.0	00.0	09.5	40.5	24.0	18.5
03.0	06.0	03.0	25.0	30.0	12.5	40.5	21.0	18.5
03.0	03.0	03.0	25.0	27.0	12.5	40.5	18.0	18.5
03.0	00.0	03.0	25.0	24.0	12.5	40.5	15.0	18.5
04.0	30.0	03.0	25.0	21.0	12.5	40.5	12.0	18.5
04.0	27.0	03.0	25.0	18.0	12.5	40.5	09.0	18.5
04.0	24.0	03.0	25.0	15.0	12.5	40.5	06.0	18.5
04.0	21.0	03.0	25.0	12.0	12.5	40.5	03.0	18.5
04.0	18.0	03.0	25.0	09.0	12.5	40.5	00.0	18.5
04.0	15.0	03.0	25.0	06.0	12.5	41.0	30.0	18.5
04.0	12.0	03.0	25.0	03.0	12.5	41.0	27.0	18.5
04.0	09.0	03.0	25.0	00.0	12.5	41.0	24.0	18.5
04.0	06.0	03.0	31.0	30.0	15.5	41.0	21.0	18.5
04.0	03.0	03.0	31.0	27.0	15.5	41.0	18.0	18.5
04.0	00.0	03.0	31.0	24.0	15.5	41.0	15.0	18.5
05.0	30.0	03.0	31.0	21.0	15.5	41.0	12.0	18.5
05.0	27.0	03.0	31.0	18.0	15.5	41.0	09.0	18.5
05.0	24.0	03.0	31.0	15.0	15.5	41.0	06.0	18.5
05.0	21.0	03.0	31.0	12.0	15.5	41.0	03.0	18.5
05.0	18.0	03.0	31.0	09.0	15.5	41.0	00.0	18.5
05.0	15.0	03.0	31.0	06.0	15.5	47.0	30.0	15.5
05.0	12.0	03.0	31.0	03.0	15.5	47.0	27.0	15.5
05.0	09.0	03.0	31.0	00.0	15.5	47.0	24.0	15.5
05.0	06.0	03.0	37.0	30.0	18.5	47.0	21.0	15.5
05.0	03.0	03.0	37.0	27.0	18.5	47.0	18.0	15.5
05.0	00.0	03.0	37.0	24.0	18.5	47.0	15.0	15.5
05.5	30.0	03.0	37.0	21.0	18.5	47.0	12.0	15.5
05.5	27.0	03.0	37.0	18.0	18.5	47.0	09.0	15.5
05.5	24.0	03.0	37.0	15.0	18.5	47.0	06.0	15.5
05.5	21.0	03.0	37.0	12.0	18.5	47.0	03.0	15.5
05.5	18.0	03.0	37.0	09.0	18.5	47.0	00.0	15.5
05.5	15.0	03.0	37.0	06.0	18.5	53.0	30.0	12.5
05.5	12.0	03.0	37.0	03.0	18.5	53.0	27.0	12.5
05.5	09.0	03.0	37.0	00.0	18.5	53.0	24.0	12.5
05.5	06.0	03.0	38.0	30.0	18.5	53.0	21.0	12.5
05.5	03.0	03.0	38.0	27.0	18.5	53.0	18.0	12.5
05.5	00.0	03.0	38.0	24.0	18.5	53.0	15.0	12.5
06.0	30.0	03.0	38.0	21.0	18.5	53.0	12.0	12.5
06.0	27.0	03.0	38.0	18.0	18.5	53.0	09.0	12.5
06.0	24.0	03.0	38.0	15.0	18.5	53.0	06.0	12.5

Table E-7. – Input data for three-dimensional plots – file DB5B. – continued

X	Y	Z
53.0	03.0	12.5
53.0	00.0	12.5
59.0	30.0	09.5
59.0	27.0	09.5
59.0	24.0	09.5
59.0	21.0	09.5
59.0	18.0	09.5
59.0	15.0	09.5
59.0	12.0	09.5
59.0	09.0	09.5
59.0	06.0	09.5
59.0	03.0	09.5
59.0	00.0	09.5
65.0	30.0	06.5
65.0	27.0	06.5
65.0	24.0	06.5
65.0	21.0	06.5
65.0	18.0	06.5
65.0	15.0	06.5
65.0	12.0	06.5
65.0	09.0	06.5
65.0	06.0	06.5
65.0	03.0	06.5
65.0	00.0	06.5
71.0	30.0	03.0
71.0	27.0	03.0
71.0	24.0	03.0
71.0	21.0	03.0
71.0	18.0	03.0
71.0	15.0	03.0
71.0	12.0	03.0
71.0	09.0	03.0
71.0	06.0	03.0
71.0	03.0	03.0
71.0	00.0	03.0
77.0	30.0	03.0
77.0	27.0	03.0
77.0	24.0	03.0
77.0	21.0	03.0
77.0	18.0	03.0
77.0	15.0	03.0
77.0	12.0	03.0
77.0	09.0	03.0
77.0	06.0	03.0
77.0	03.0	03.0
77.0	00.0	03.0
83.0	30.0	03.0
83.0	27.0	03.0
83.0	24.0	03.0
83.0	21.0	03.0
83.0	18.0	03.0
83.0	15.0	03.0
83.0	12.0	03.0
83.0	09.0	03.0
83.0	06.0	03.0
83.0	03.0	03.0
83.0	00.0	03.0
89.0	30.0	03.0
89.0	27.0	03.0
89.0	24.0	03.0
89.0	21.0	03.0
89.0	18.0	03.0
89.0	15.0	03.0
89.0	12.0	03.0
89.0	09.0	03.0
89.0	06.0	03.0
89.0	03.0	03.0
89.0	00.0	03.0
89.1	30.0	00.0
89.1	27.0	00.0
89.1	24.0	00.0
89.1	21.0	00.0
89.1	18.0	00.0
89.1	15.0	00.0
89.1	12.0	00.0
89.1	09.0	00.0
89.1	06.0	00.0
89.1	03.0	00.0
89.1	00.0	00.0

Table E-8. – Input data for three-dimensional plots – file DB6A.

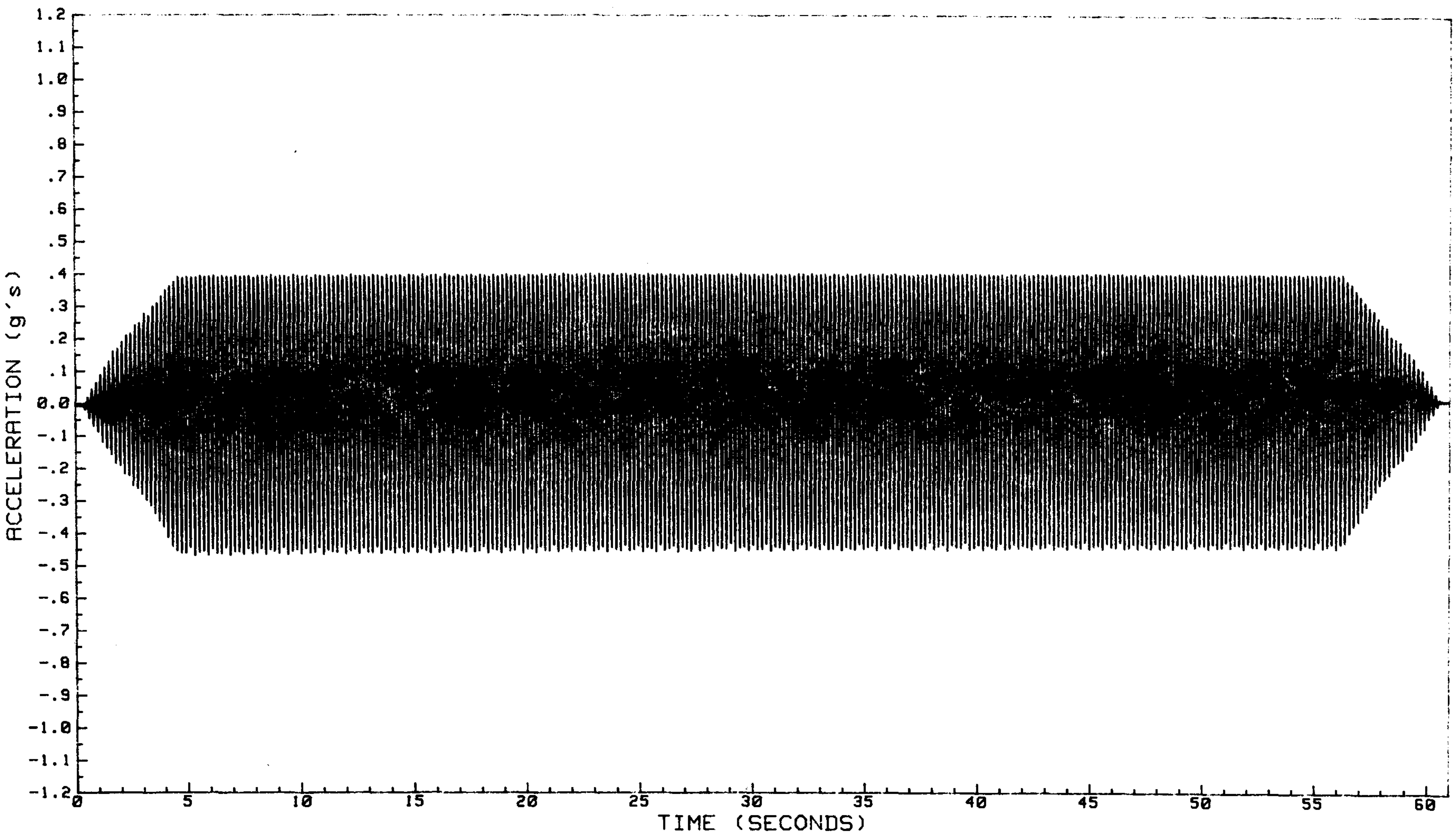
X	Y	Z	X	Y	Z	X	Y	Z
01.0	30.0	00.0	16.0	21.0	09.4	34.0	12.0	14.6
01.0	27.0	00.0	16.0	18.0	09.4	34.0	09.0	14.6
01.0	24.0	00.0	16.0	15.0	09.4	34.0	06.0	15.0
01.0	21.0	00.0	16.0	12.0	09.2	34.0	03.0	15.6
01.0	18.0	00.0	16.0	09.0	09.4	34.0	00.0	15.6
01.0	15.0	00.0	16.0	06.0	09.1	37.0	30.0	15.6
01.0	12.0	00.0	16.0	03.0	08.7	37.0	27.0	14.8
01.0	09.0	00.0	16.0	00.0	08.7	37.0	24.0	13.8
01.0	06.0	00.0	19.0	30.0	09.4	37.0	21.0	14.2
01.0	03.0	00.0	19.0	27.0	09.6	37.0	18.0	14.3
01.0	00.0	00.0	19.0	24.0	09.8	37.0	15.0	13.9
01.0	30.0	03.5	19.0	21.0	09.6	37.0	12.0	14.3
01.0	27.0	04.0	19.0	18.0	09.8	37.0	09.0	15.4
01.0	24.0	04.2	19.0	15.0	09.8	37.0	06.0	15.6
01.0	21.0	04.4	19.0	12.0	09.8	37.0	03.0	15.9
01.0	18.0	04.4	19.0	09.0	09.6	37.0	00.0	16.1
01.0	15.0	04.7	19.0	06.0	09.7	40.0	30.0	14.4
01.0	12.0	04.7	19.0	03.0	09.8	40.0	27.0	13.6
01.0	09.0	04.4	19.0	00.0	09.8	40.0	24.0	13.2
01.0	06.0	04.7	22.0	30.0	10.6	40.0	21.0	13.3
01.0	03.0	03.1	22.0	27.0	10.4	40.0	18.0	12.8
01.0	00.0	02.4	22.0	24.0	10.3	40.0	15.0	13.1
04.0	30.0	04.6	22.0	21.0	10.1	40.0	12.0	13.7
04.0	27.0	04.8	22.0	18.0	10.2	40.0	09.0	14.0
04.0	24.0	04.6	22.0	15.0	10.0	40.0	06.0	14.1
04.0	21.0	04.9	22.0	12.0	10.1	40.0	03.0	14.0
04.0	18.0	05.0	22.0	09.0	10.2	40.0	00.0	14.5
04.0	15.0	05.2	22.0	06.0	10.4	43.0	30.0	13.7
04.0	12.0	05.5	22.0	03.0	11.1	43.0	27.0	13.9
04.0	09.0	05.5	22.0	00.0	11.2	43.0	24.0	13.7
04.0	06.0	05.0	25.0	30.0	12.0	43.0	21.0	13.4
04.0	03.0	04.3	25.0	27.0	11.3	43.0	18.0	12.6
04.0	00.0	03.8	25.0	24.0	11.1	43.0	15.0	12.5
07.0	30.0	05.5	25.0	21.0	11.0	43.0	12.0	12.4
07.0	27.0	05.7	25.0	18.0	10.8	43.0	09.0	10.4
07.0	24.0	05.8	25.0	15.0	10.9	43.0	06.0	12.6
07.0	21.0	05.7	25.0	12.0	10.8	43.0	03.0	13.0
07.0	18.0	05.8	25.0	09.0	10.9	43.0	00.0	13.5
07.0	15.0	06.0	25.0	06.0	11.2	46.0	30.0	13.3
07.0	12.0	05.9	25.0	03.0	12.0	46.0	27.0	13.1
07.0	09.0	05.9	25.0	00.0	12.4	46.0	24.0	12.9
07.0	06.0	05.8	28.0	30.0	13.1	46.0	21.0	12.4
07.0	03.0	05.4	28.0	27.0	12.5	46.0	18.0	12.0
07.0	00.0	05.2	28.0	24.0	12.2	46.0	15.0	12.2
10.0	30.0	06.4	28.0	21.0	12.2	46.0	12.0	12.4
10.0	27.0	06.5	28.0	18.0	12.2	46.0	09.0	12.1
10.0	24.0	07.0	28.0	15.0	12.2	46.0	06.0	11.7
10.0	21.0	07.2	28.0	12.0	12.5	46.0	03.0	12.7
10.0	18.0	07.2	28.0	09.0	12.2	46.0	00.0	12.9
10.0	15.0	07.4	28.0	06.0	12.3	48.0	30.0	12.8
10.0	12.0	07.3	28.0	03.0	13.1	48.0	27.0	12.6
10.0	09.0	07.4	28.0	00.0	13.4	48.0	24.0	12.0
10.0	06.0	07.2	31.0	30.0	14.3	48.0	21.0	12.1
10.0	03.0	06.7	31.0	27.0	13.7	48.0	18.0	12.1
10.0	00.0	06.7	31.0	24.0	13.4	48.0	15.0	12.3
13.0	30.0	07.7	31.0	21.0	13.3	48.0	12.0	12.4
13.0	27.0	08.1	31.0	18.0	13.5	48.0	09.0	12.0
13.0	24.0	08.6	31.0	15.0	13.4	48.0	06.0	11.5
13.0	21.0	08.5	31.0	12.0	13.4	48.0	03.0	12.3
13.0	18.0	08.5	31.0	09.0	13.1	48.0	00.0	12.1
13.0	15.0	08.6	31.0	06.0	13.3	50.0	30.0	11.7
13.0	12.0	08.6	31.0	03.0	13.9	50.0	27.0	11.6
13.0	09.0	08.5	31.0	00.0	14.4	50.0	24.0	12.2
13.0	06.0	08.4	34.0	30.0	15.5	50.0	21.0	11.8
13.0	03.0	07.8	34.0	27.0	15.0	50.0	18.0	11.5
13.0	00.0	07.6	34.0	24.0	14.6	50.0	15.0	11.7
16.0	30.0	08.8	34.0	21.0	15.0	50.0	12.0	11.7
16.0	27.0	09.1	34.0	18.0	14.8	50.0	09.0	10.9
16.0	24.0	09.4	34.0	15.0	14.6	50.0	06.0	11.1

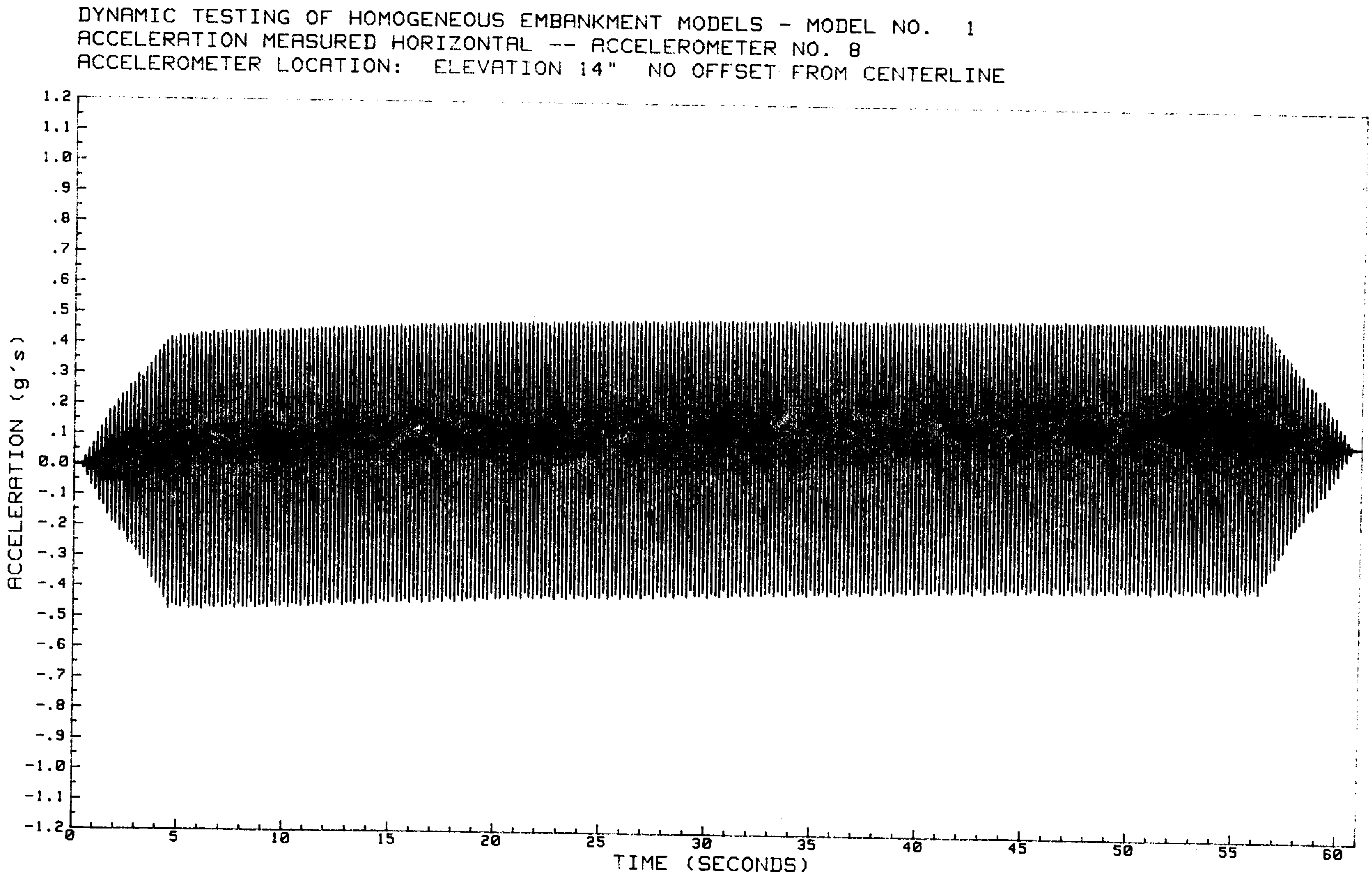
Table E-8. – Input data for three-dimensional plots – file DB6A. – continued

X	Y	Z	X	Y	Z	X	Y	Z
50.0	03.0	11.4	65.0	12.0	08.5	80.0	21.0	04.6
50.0	00.0	11.1	65.0	09.0	08.8	80.0	18.0	04.5
53.0	30.0	11.2	65.0	06.0	08.7	80.0	15.0	04.9
53.0	27.0	11.1	65.0	03.0	08.2	80.0	12.0	04.9
53.0	24.0	11.1	65.0	00.0	07.7	80.0	09.0	04.7
53.0	21.0	10.9	68.0	30.0	07.0	80.0	06.0	04.8
53.0	18.0	10.5	68.0	27.0	07.0	80.0	03.0	04.4
53.0	15.0	10.6	68.0	24.0	07.5	80.0	00.0	04.3
53.0	12.0	10.3	68.0	21.0	07.6	83.0	30.0	03.1
53.0	09.0	09.8	68.0	18.0	07.8	83.0	27.0	03.6
53.0	06.0	10.0	68.0	15.0	07.7	83.0	24.0	03.9
53.0	03.0	10.1	68.0	12.0	07.9	83.0	21.0	03.9
53.0	00.0	10.2	68.0	09.0	08.0	83.0	18.0	04.2
56.0	30.0	10.2	68.0	06.0	08.0	83.0	15.0	03.9
56.0	27.0	10.2	68.0	03.0	07.5	83.0	12.0	04.1
56.0	24.0	10.0	68.0	00.0	07.2	83.0	09.0	04.0
56.0	21.0	10.2	71.0	30.0	06.3	83.0	06.0	04.1
56.0	18.0	10.0	71.0	27.0	06.5	83.0	03.0	04.1
56.0	15.0	10.2	71.0	24.0	06.6	83.0	00.0	03.3
56.0	12.0	09.6	71.0	21.0	06.7	86.0	30.0	03.2
56.0	09.0	08.6	71.0	18.0	06.7	86.0	27.0	03.1
56.0	06.0	08.3	71.0	15.0	06.8	86.0	24.0	02.8
56.0	03.0	08.6	71.0	12.0	06.9	86.0	21.0	02.8
56.0	00.0	08.8	71.0	09.0	07.0	86.0	18.0	02.6
59.0	30.0	09.0	71.0	06.0	07.2	86.0	15.0	02.6
59.0	27.0	09.2	71.0	03.0	06.9	86.0	12.0	02.6
59.0	24.0	09.2	71.0	00.0	06.7	86.0	09.0	02.6
59.0	21.0	09.3	74.0	30.0	05.7	86.0	06.0	02.6
59.0	18.0	09.2	74.0	27.0	05.8	86.0	03.0	02.8
59.0	15.0	09.1	74.0	24.0	06.1	86.0	00.0	02.7
59.0	12.0	09.1	74.0	21.0	06.1	89.0	30.0	03.2
59.0	09.0	09.3	74.0	18.0	06.2	89.0	27.0	03.1
59.0	06.0	09.4	74.0	15.0	06.2	89.0	24.0	02.8
59.0	03.0	09.1	74.0	12.0	06.1	89.0	21.0	02.8
59.0	00.0	08.4	74.0	09.0	06.3	89.0	18.0	02.6
62.0	30.0	07.9	74.0	06.0	06.3	89.0	15.0	02.6
62.0	27.0	08.1	74.0	03.0	06.3	89.0	12.0	02.6
62.0	24.0	08.5	74.0	00.0	05.9	89.0	09.0	02.6
62.0	21.0	08.7	77.0	30.0	04.9	89.0	06.0	02.6
62.0	18.0	08.9	77.0	27.0	05.3	89.0	03.0	02.8
62.0	15.0	08.7	77.0	24.0	05.4	89.0	00.0	02.7
62.0	12.0	08.9	77.0	21.0	05.4	89.0	30.0	00.0
62.0	09.0	09.2	77.0	18.0	05.4	89.0	27.0	00.0
62.0	06.0	09.3	77.0	15.0	05.6	89.0	24.0	00.0
62.0	03.0	08.9	77.0	12.0	05.7	89.0	21.0	00.0
62.0	00.0	08.3	77.0	09.0	05.8	89.0	18.0	00.0
65.0	30.0	07.6	77.0	06.0	05.4	89.0	15.0	00.0
65.0	27.0	07.8	77.0	03.0	05.2	89.0	12.0	00.0
65.0	24.0	08.1	77.0	00.0	05.3	89.0	09.0	00.0
65.0	21.0	08.3	80.0	30.0	04.3	89.0	06.0	00.0
65.0	18.0	08.5	80.0	27.0	04.5	89.0	03.0	00.0
65.0	15.0	08.4	80.0	24.0	04.9	89.0	00.0	00.0

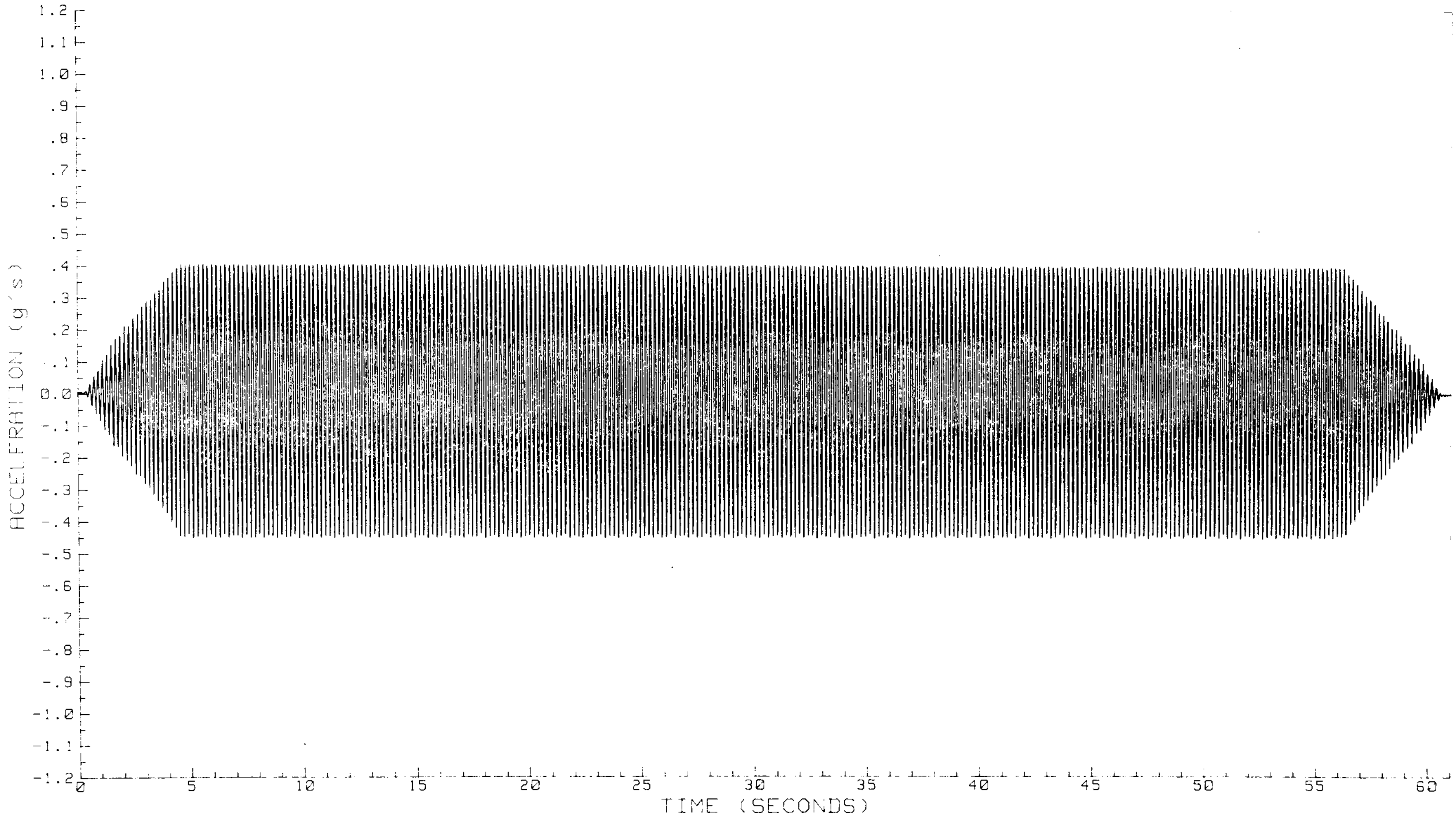
APPENDIX F
DATA ACQUISITION – SUMMARY OF PLOTS

DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 1
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 7
ACCELEROMETER LOCATION: ELEVATION 10" OFFSET FROM CENTERLINE 10" RT

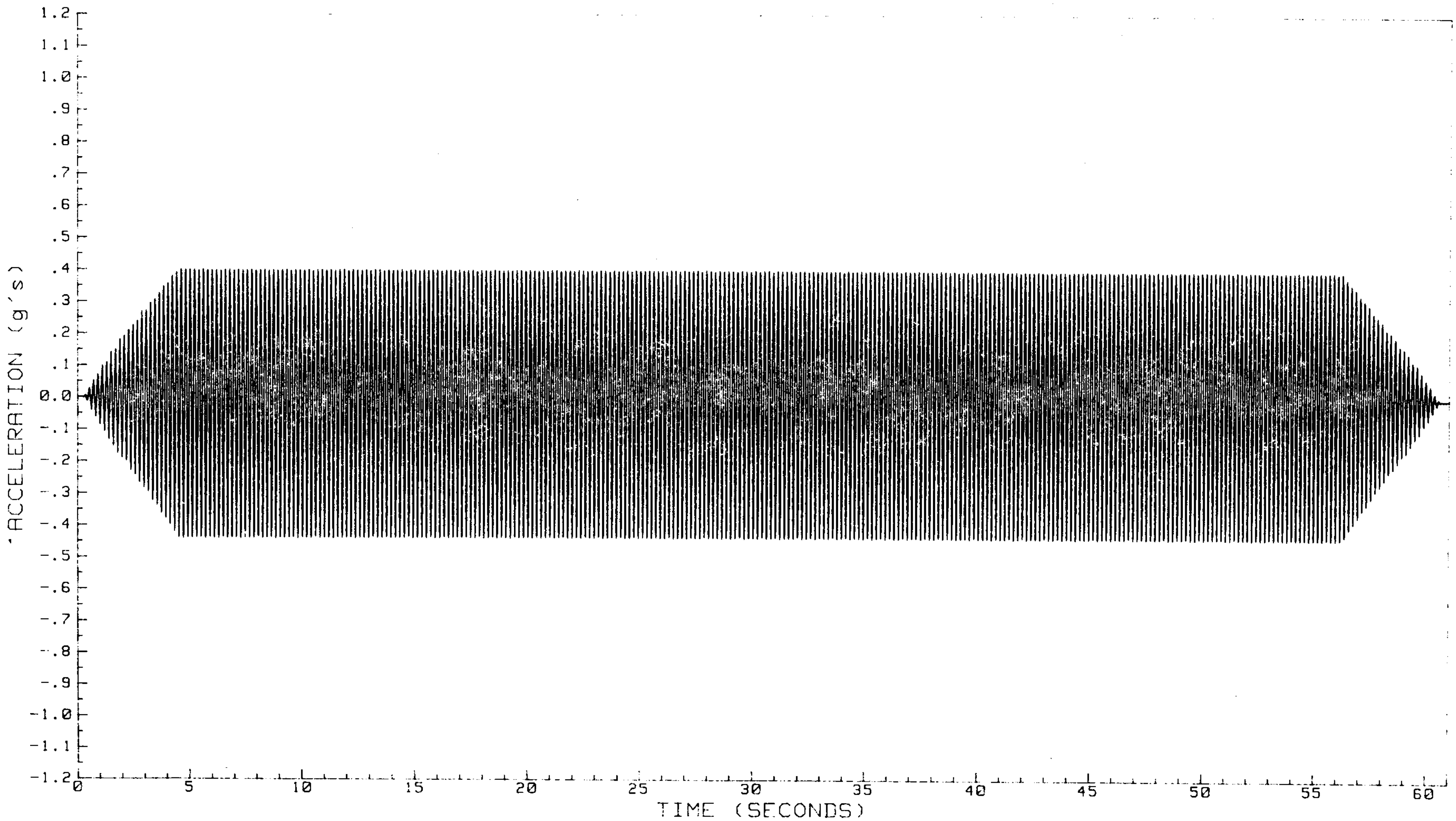



DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 1
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 8
ACCELEROMETER LOCATION: ELEVATION 14" NO OFFSET FROM CENTERLINE
ACCELERATION (g's)
TIME (SECONDS)
1.2
1.1
1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
0.0
-.1
-.2
-.3
-.4
-.5
-.6
-.7
-.8
-.9
-1.0
-1.1
-1.2
0
5
10
15
20
25
30
35
40
45
50
55
60

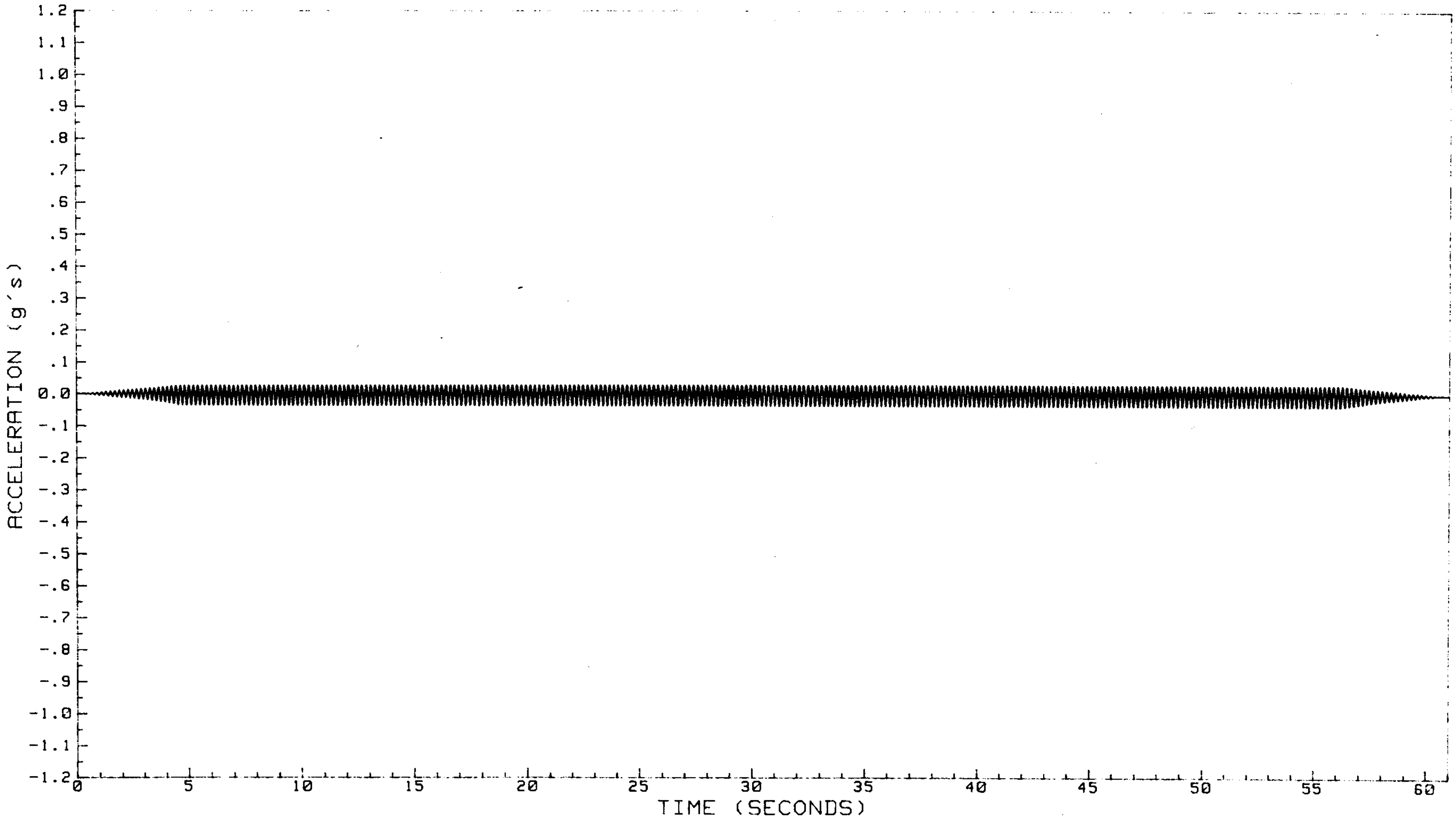
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 1
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 18
ACCELEROMETER LOCATION: ELEVATION 10" NO OFFSET FROM CENTERLINE



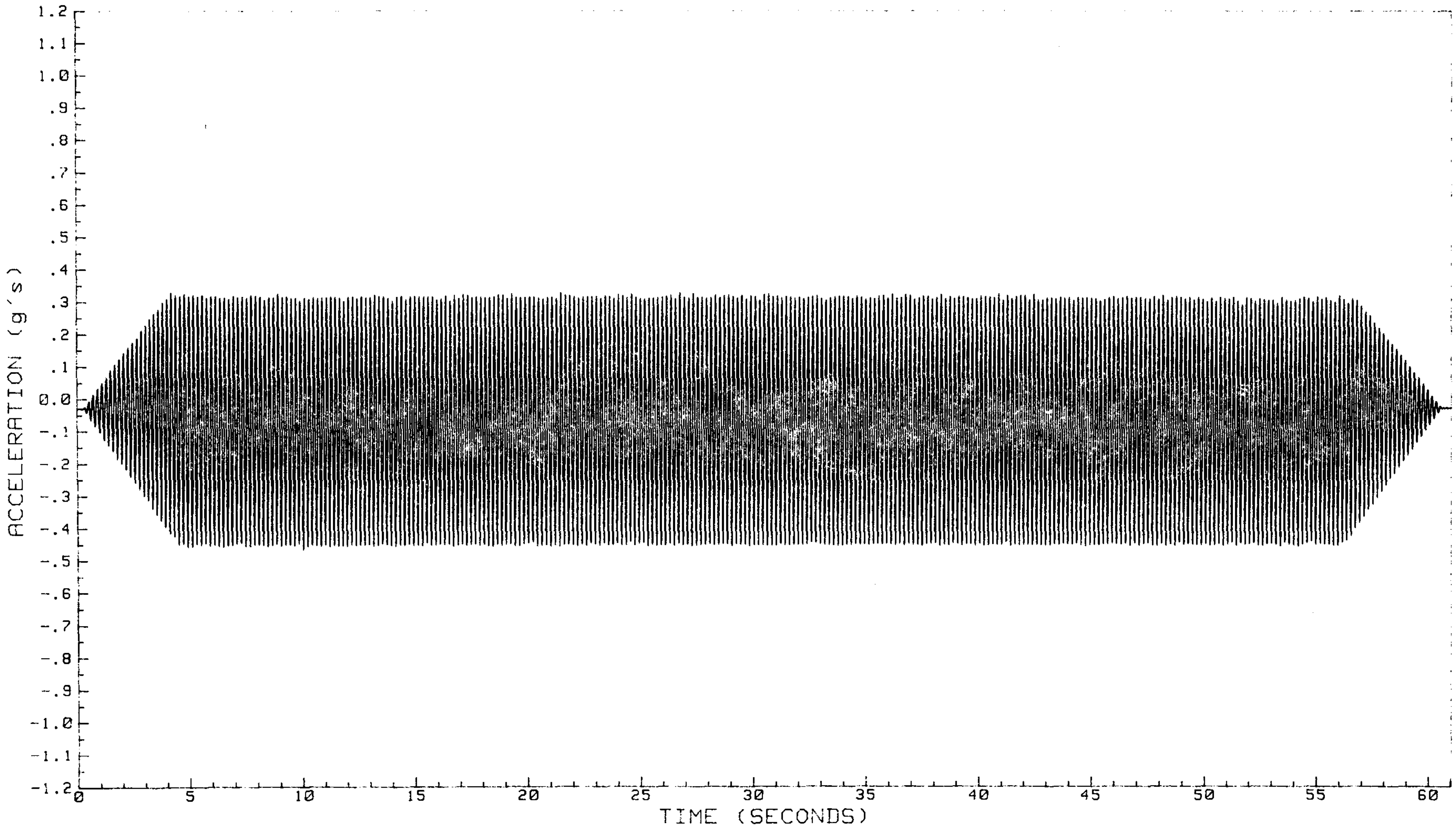
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 1
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 13
ACCELEROMETER LOCATION: ELEVATION 5" NO OFFSET FROM CENTERLINE



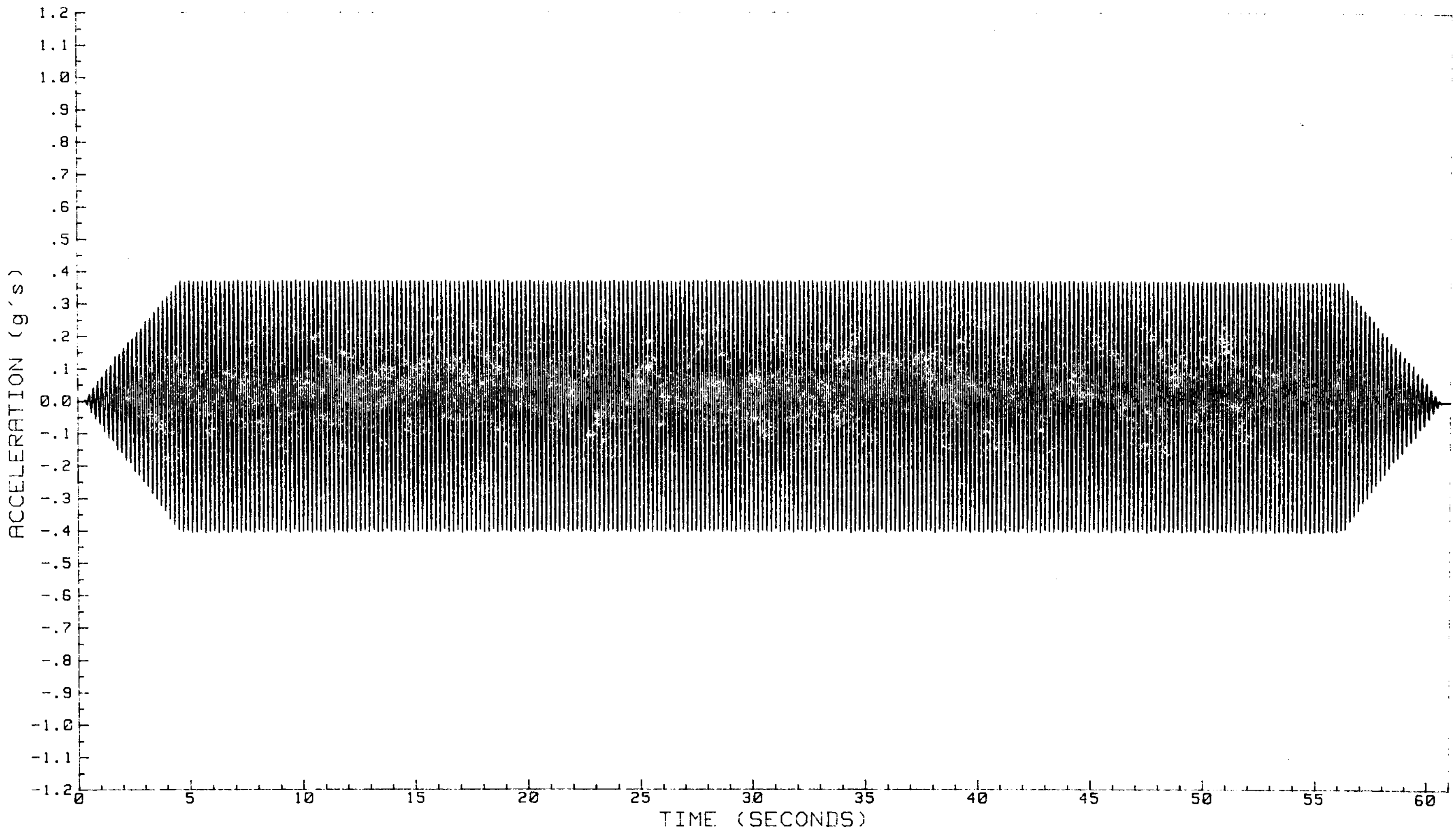
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 1
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 17
ACCELEROMETER LOCATION: ELEVATION 1.0" NO OFFSET FROM CENTERLINE



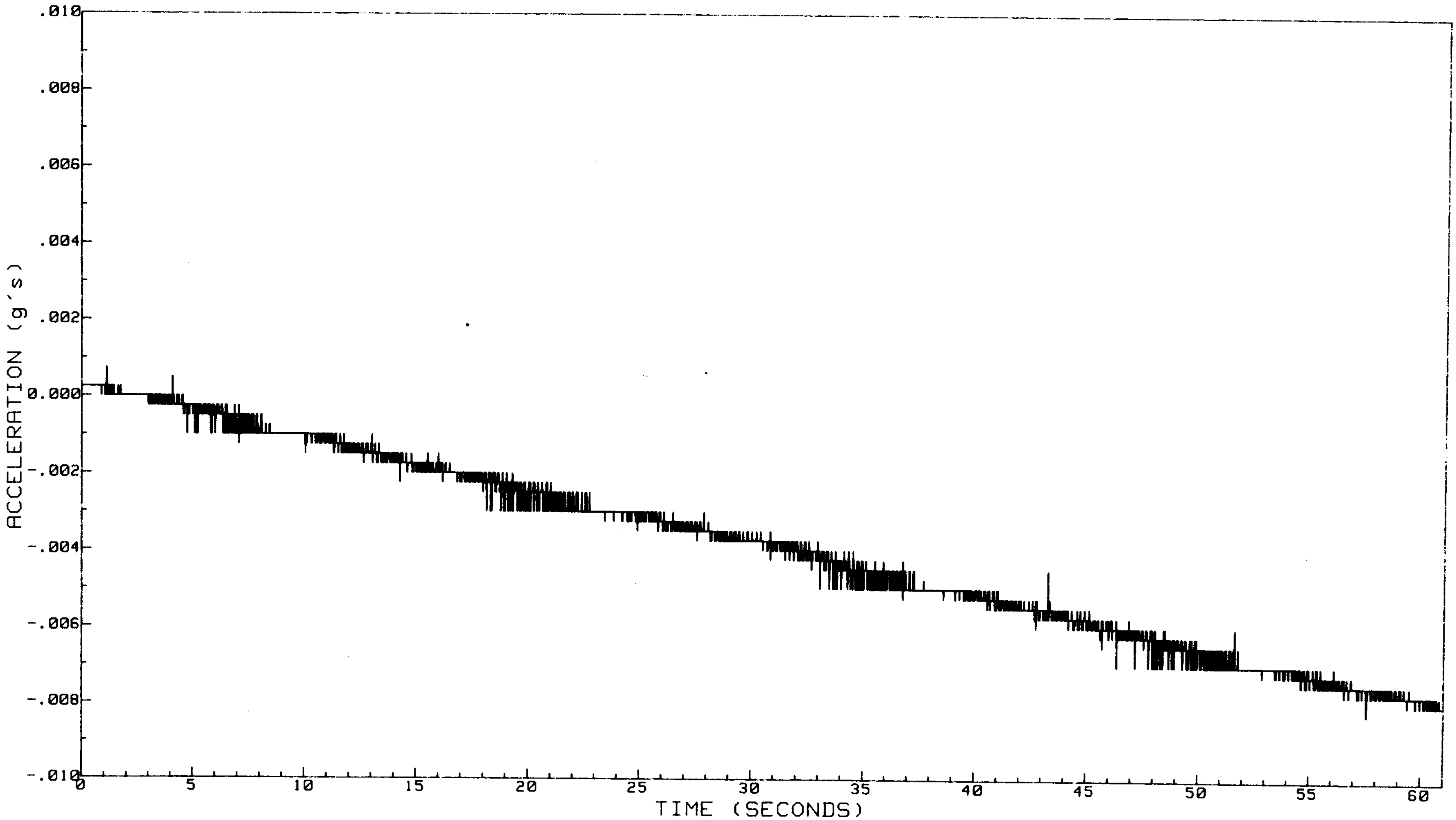
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 1
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 74
ACCELEROMETER LOCATION: ACTUATOR

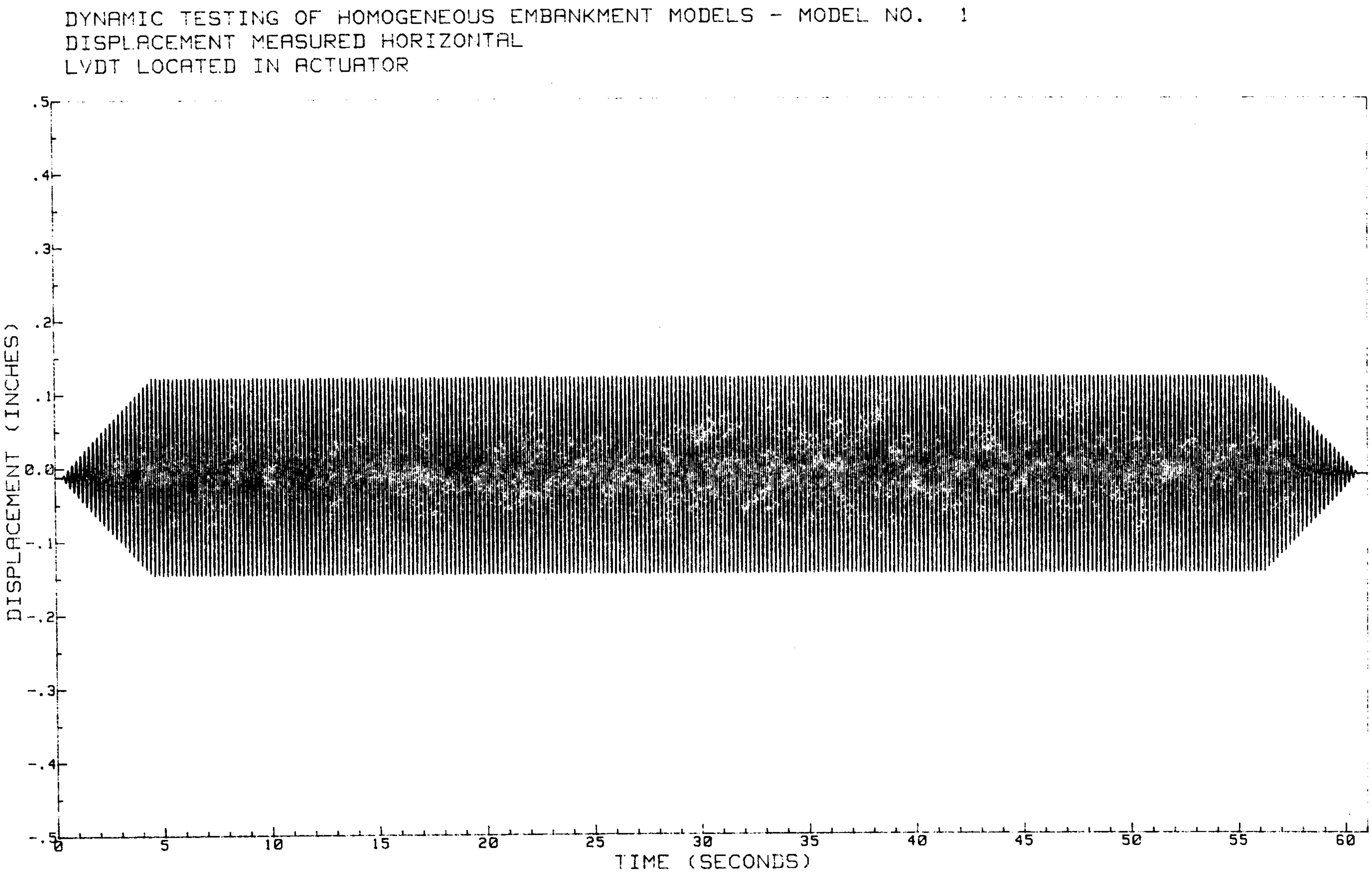


DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 1
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 72
ACCELEROMETER LOCATION: BOX FRAME

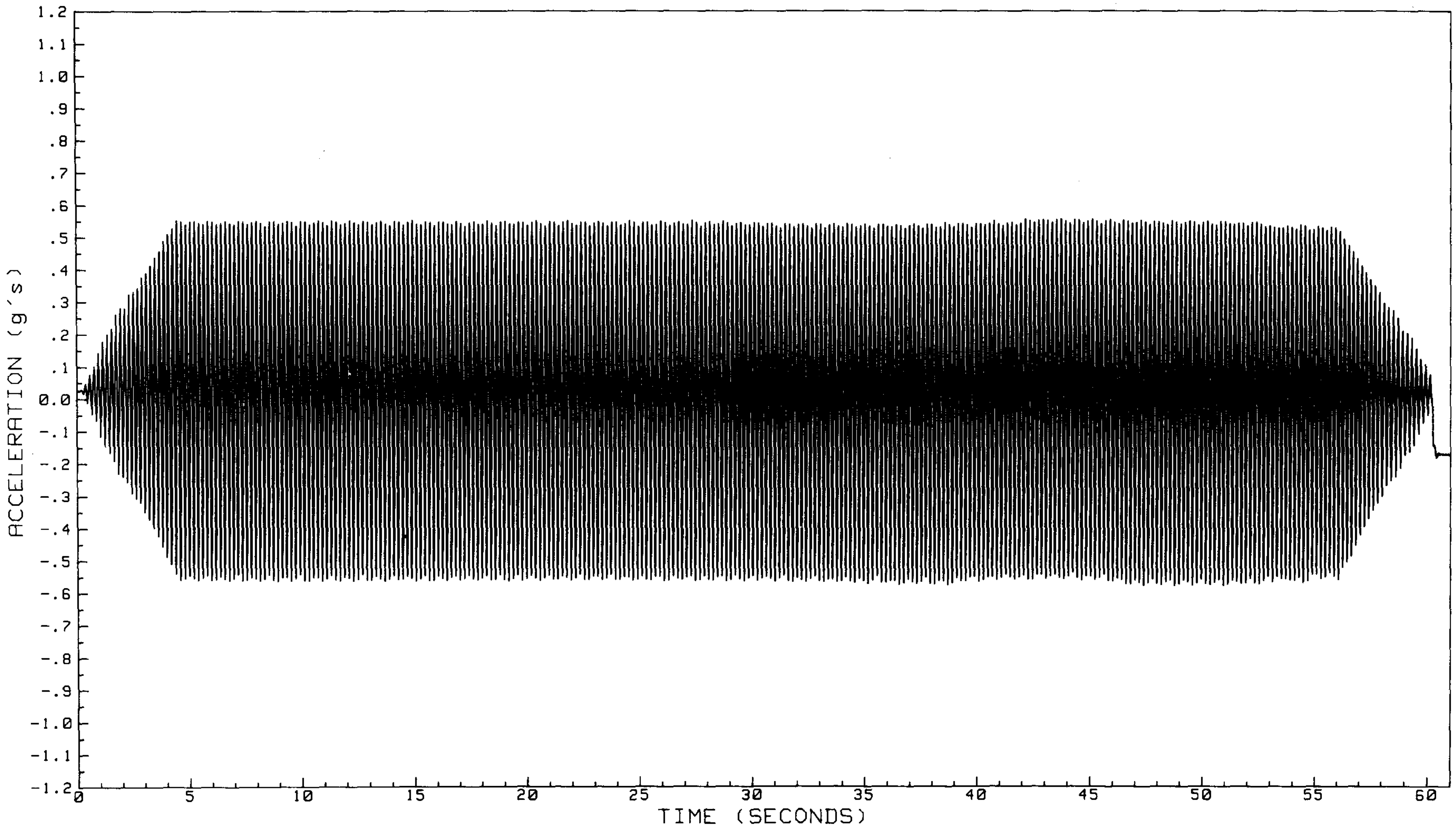


DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 1
ACCELERATION MEASURED VERTICAL -- ACCELEROMETER NO. 77
ACCELEROMETER LOCATION: BOX FRAME

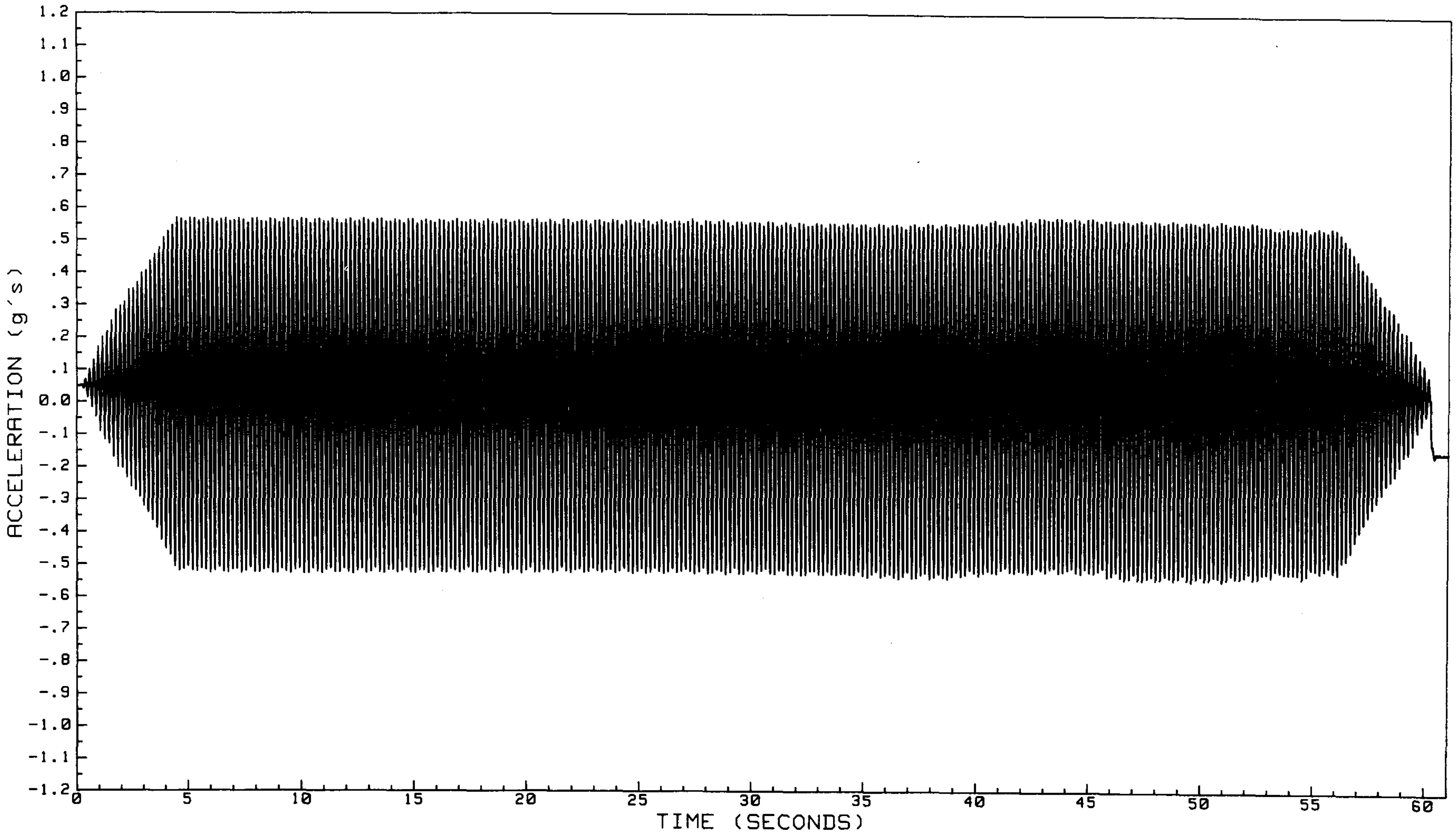



DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 1
DISPLACEMENT MEASURED HORIZONTAL
LVDT LOCATED IN ACTUATOR
DISPLACEMENT (INCHES)
.5
.4
.3
.2
.1
0.0
-.1
-.2
-.3
-.4
-.5
0
5
10
15
20
25
30
35
40
45
50
55
60
TIME (SECONDS)

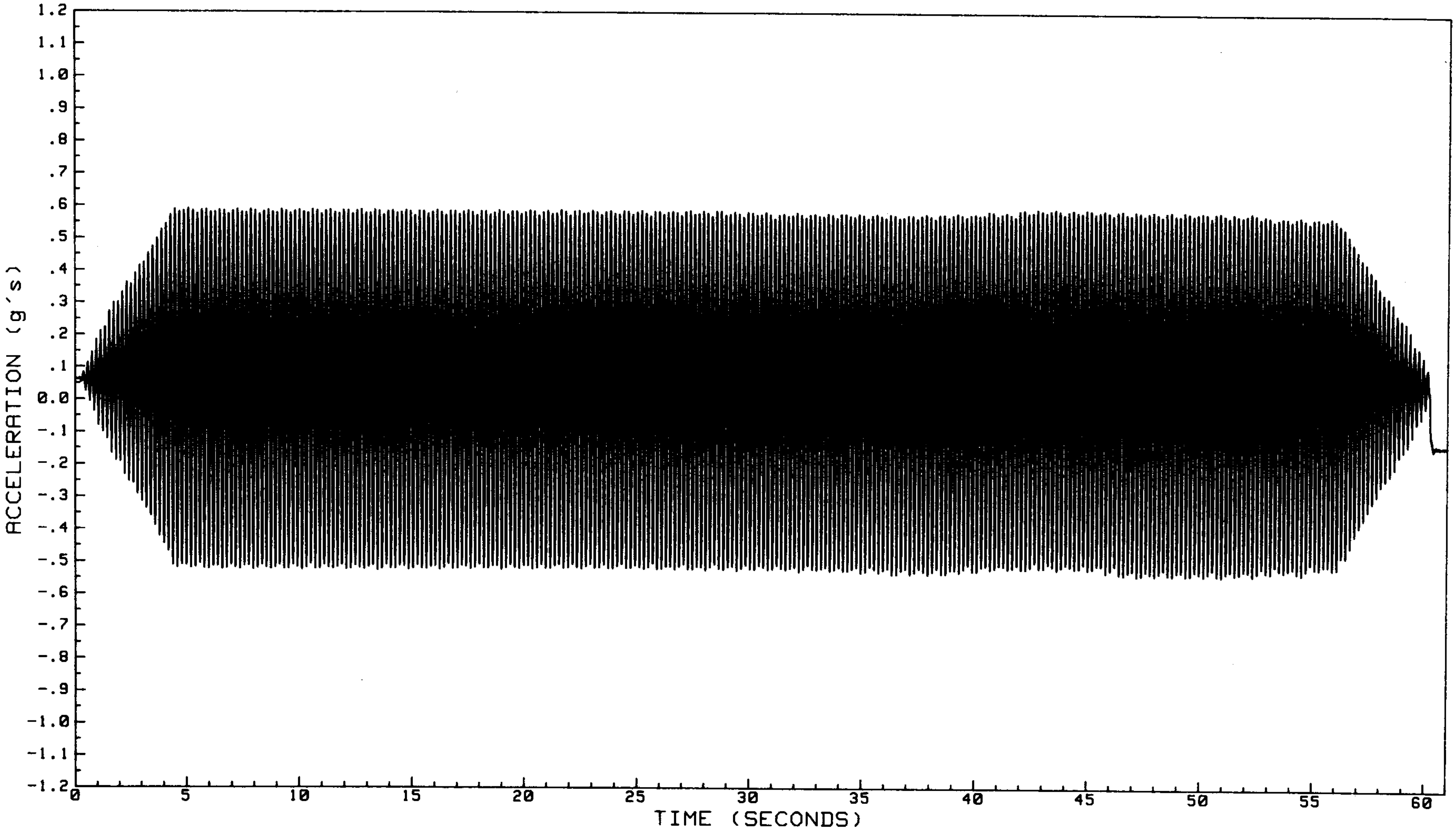
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 2
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 7
ACCELEROMETER LOCATION: ELEVATION 12.0" OFFSET FROM CENTERLINE 0.5" RT



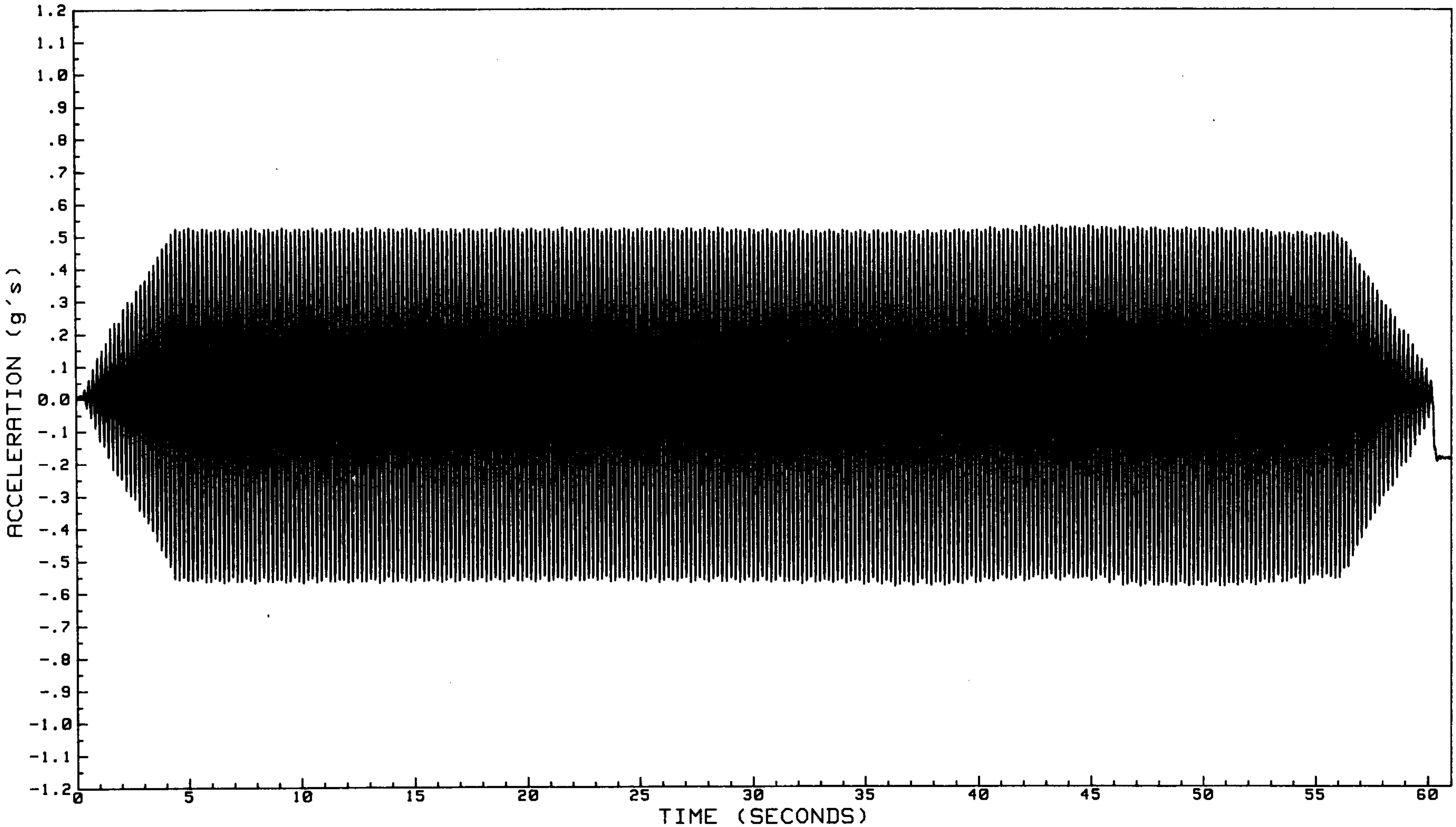
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 2
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 8
ACCELEROMETER LOCATION: ELEVATION 14.5" OFFSET FROM CENTERLINE 1.5" LT



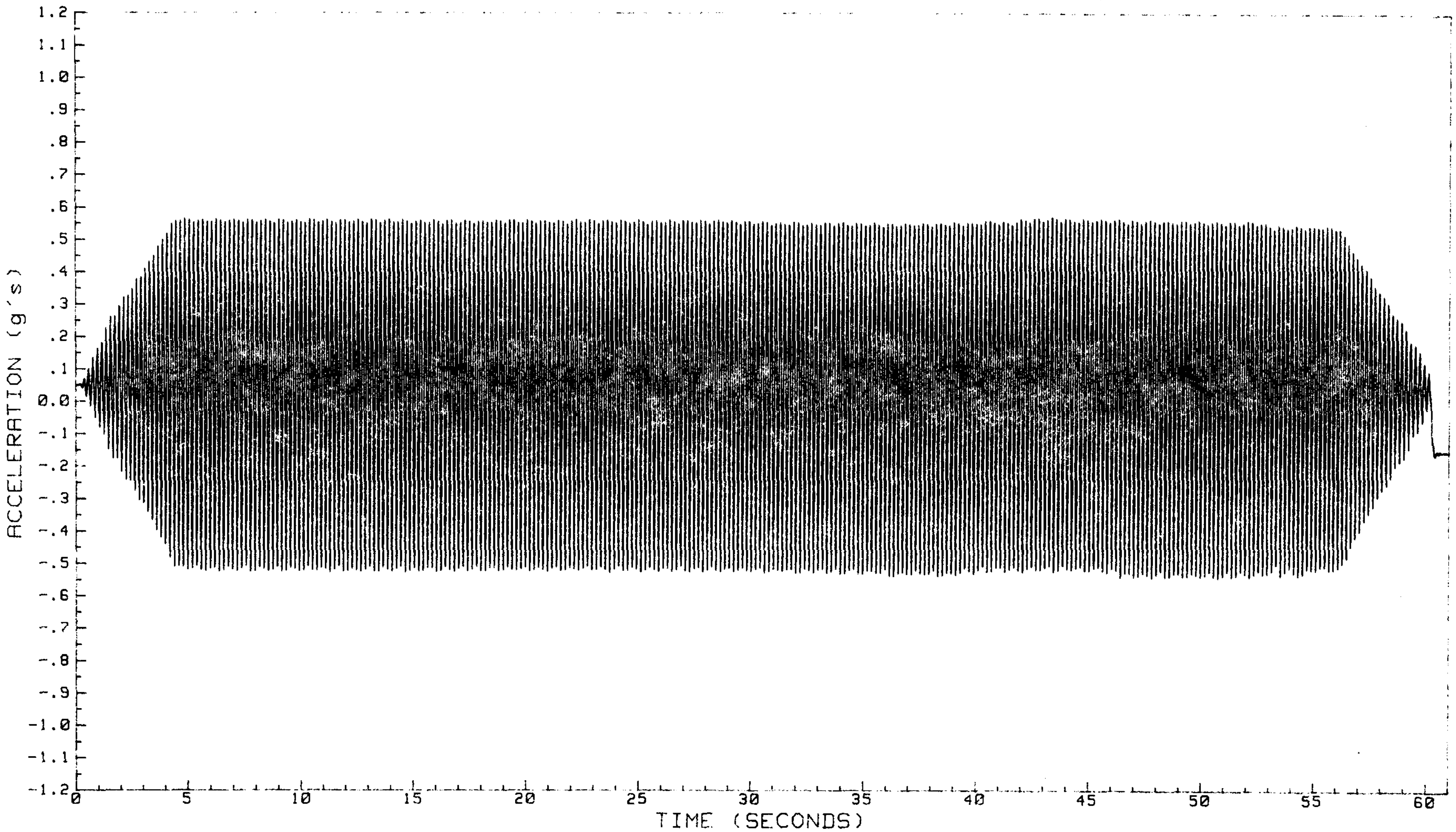
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 2
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 18
ACCELEROMETER LOCATION: ELEVATION 10.0" OFFSET FROM CENTERLINE 1.5" RT



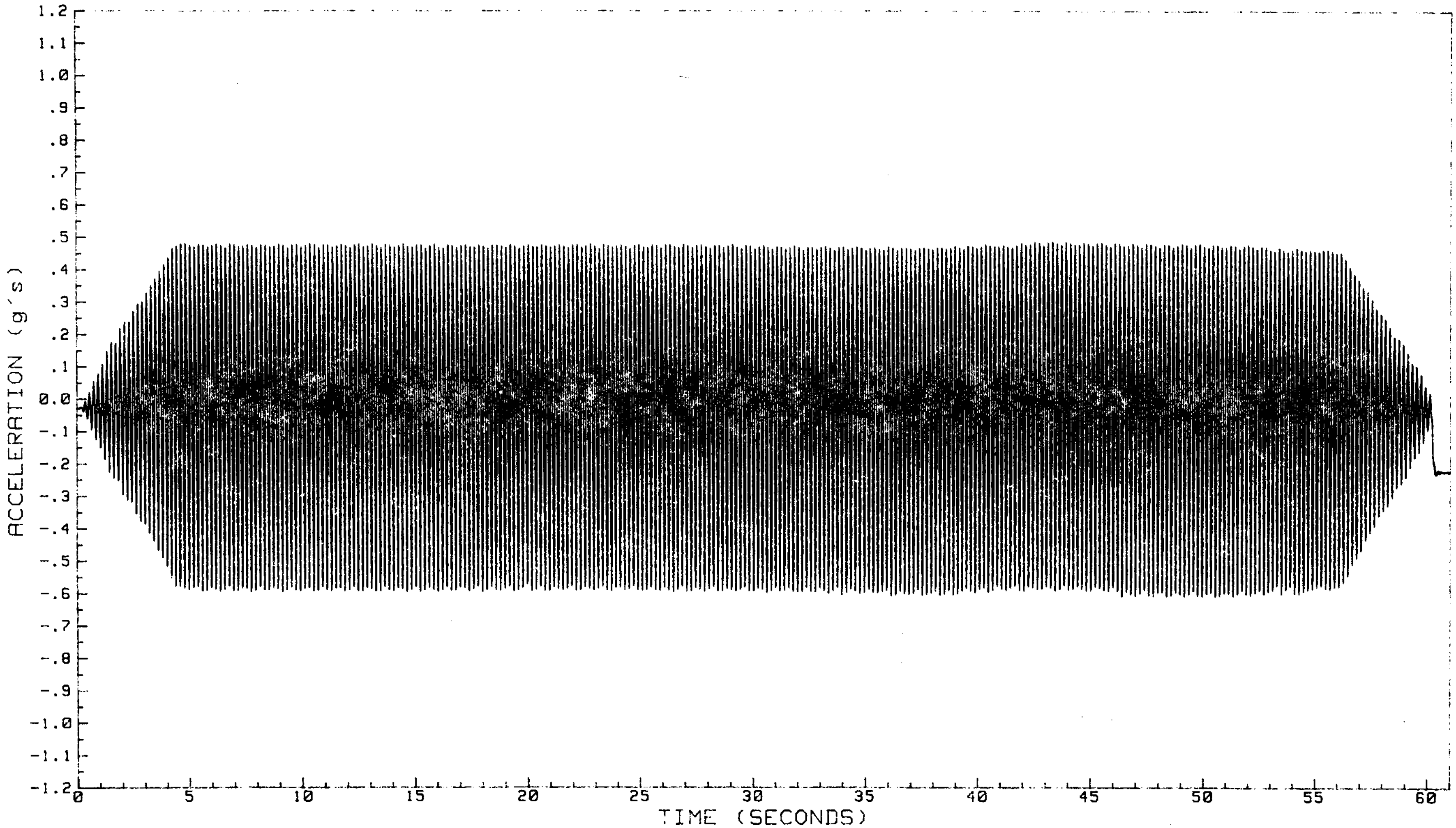
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 2
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 13
ACCELEROMETER LOCATION: ELEVATION 5.0" OFFSET FROM CENTERLINE 2.0" RT



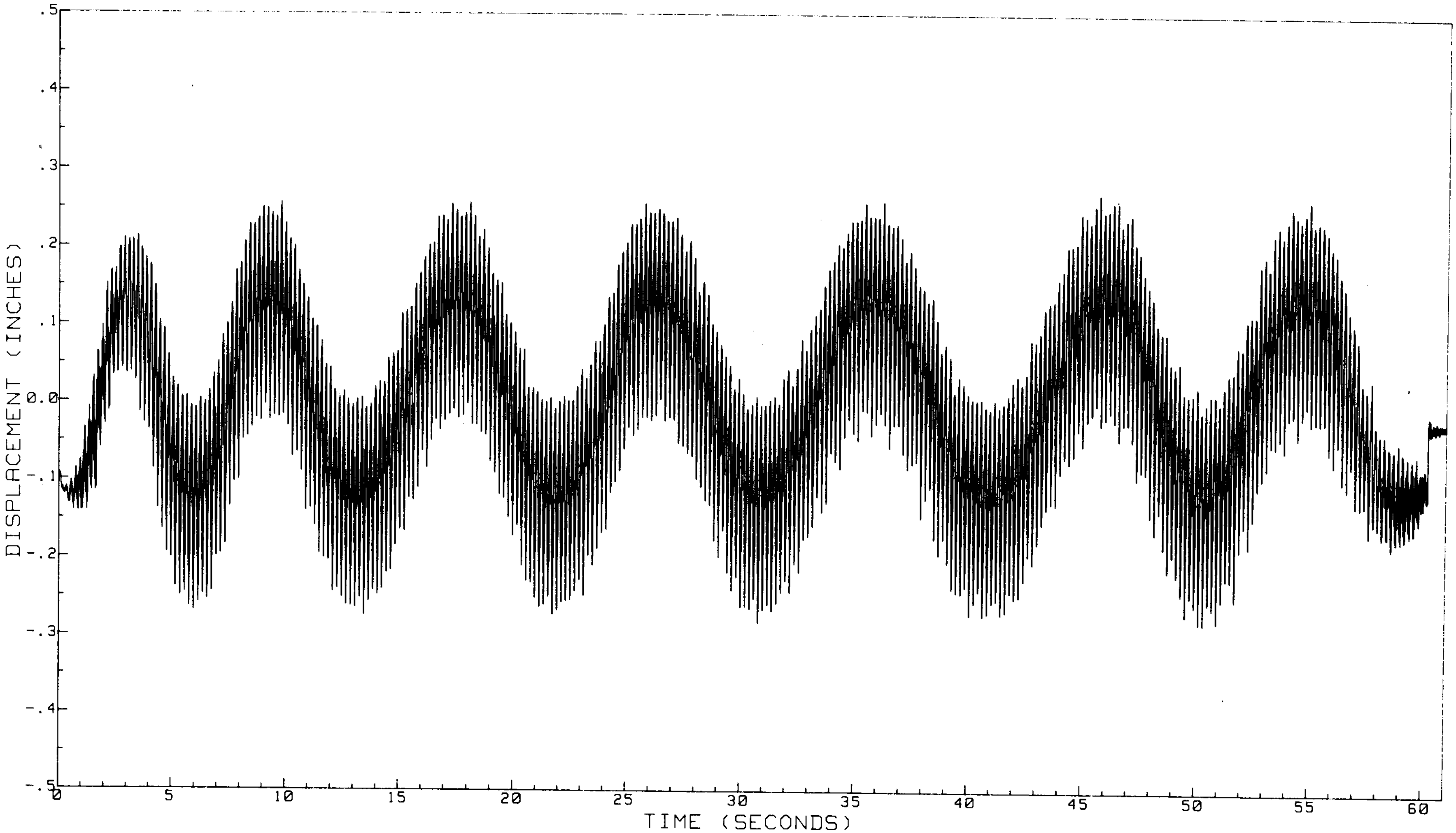
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 2
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 74
ACCELEROMETER LOCATION: ACTUATOR



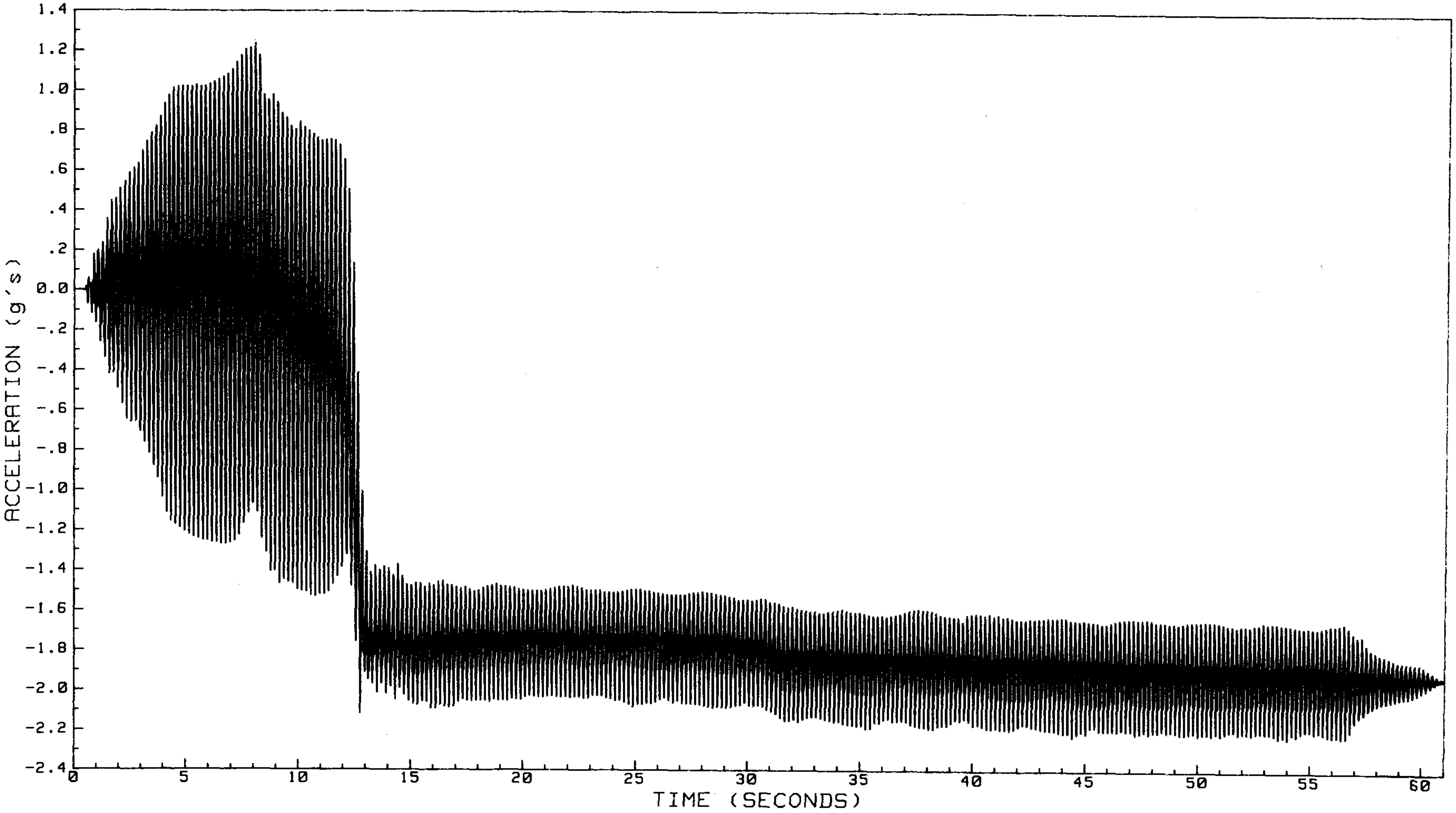
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 2
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 72
ACCELEROMETER LOCATION: BOX FRAME



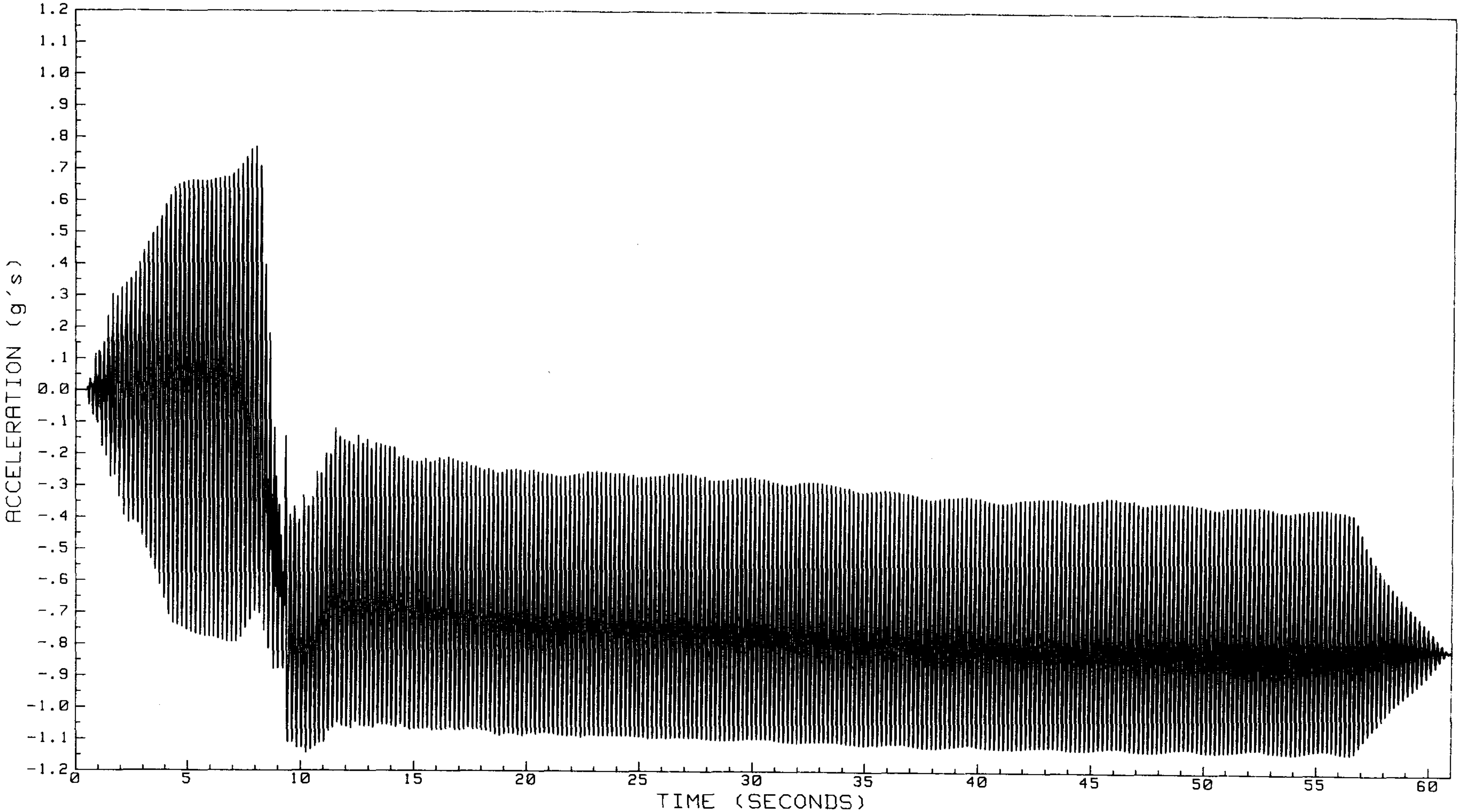
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 2
DISPLACEMENT MEASURED HORIZONTAL
LVDT LOCATED IN ACTUATOR



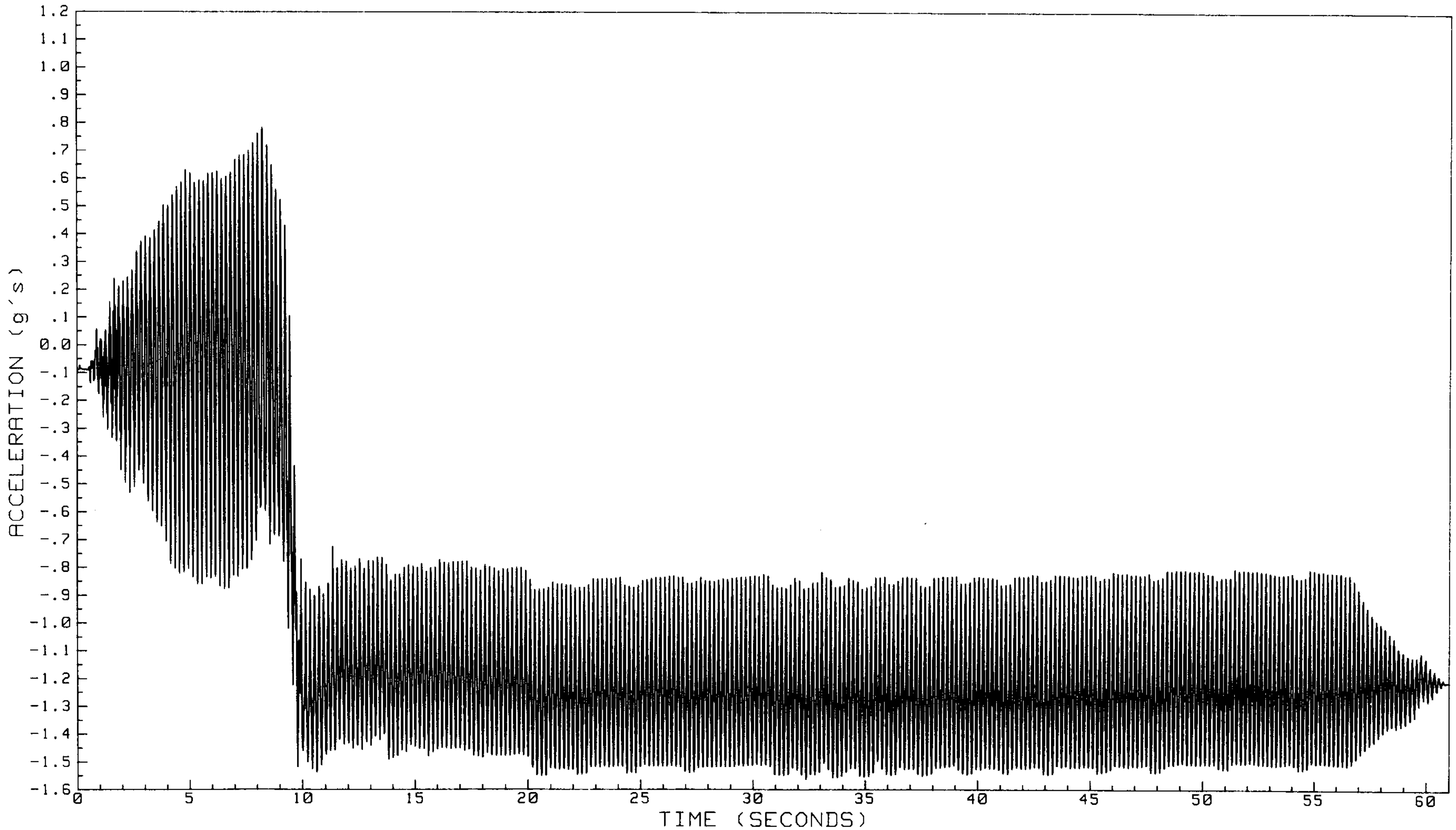
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 3
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 7
ACCELEROMETER LOCATION: ELEVATION 12.0" OFFSET FROM CENTERLINE 7.0" LT



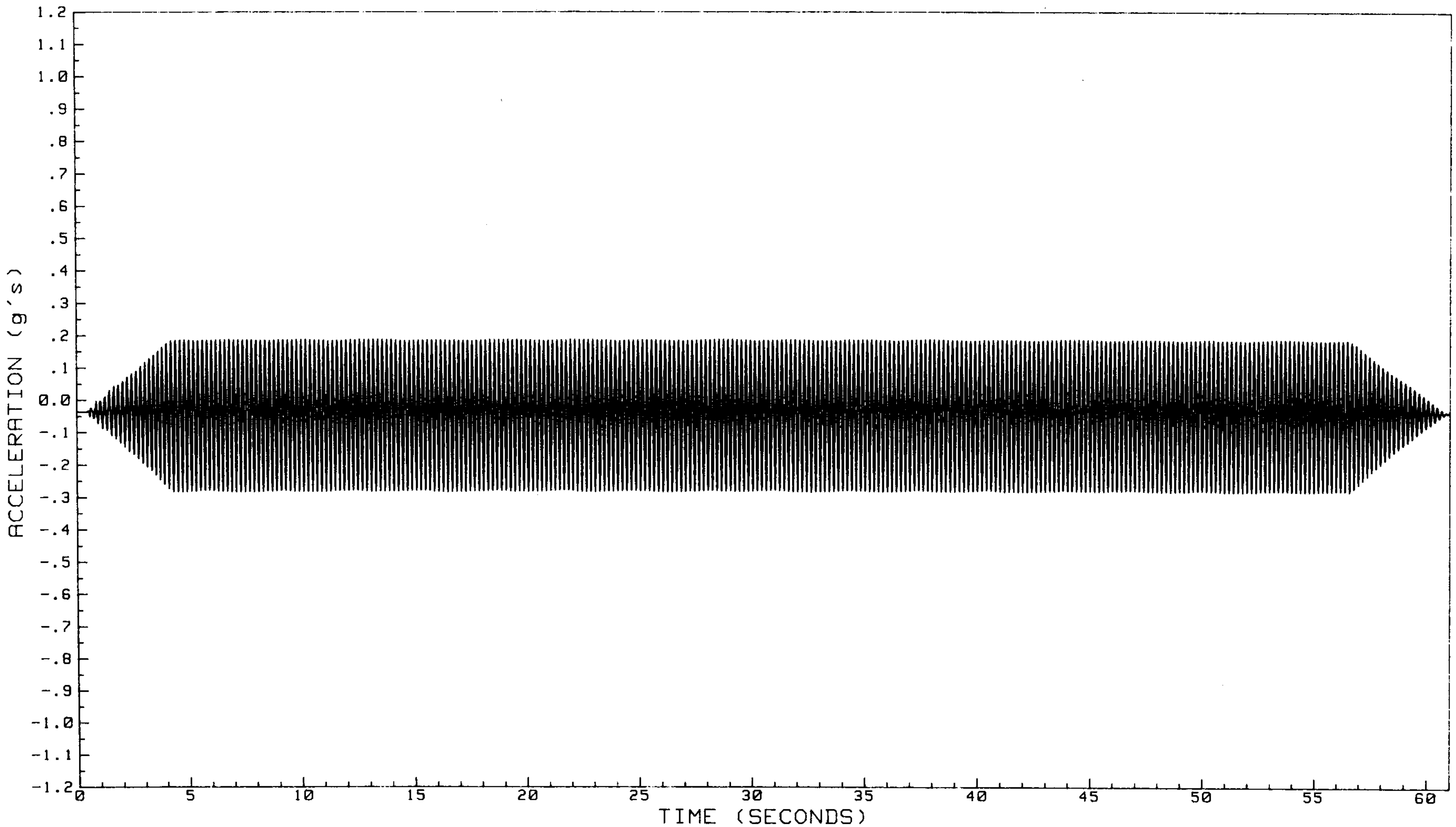
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 3
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 8
ACCELEROMETER LOCATION: ELEVATION 14.0" OFFSET FROM CENTERLINE 3.0" RT



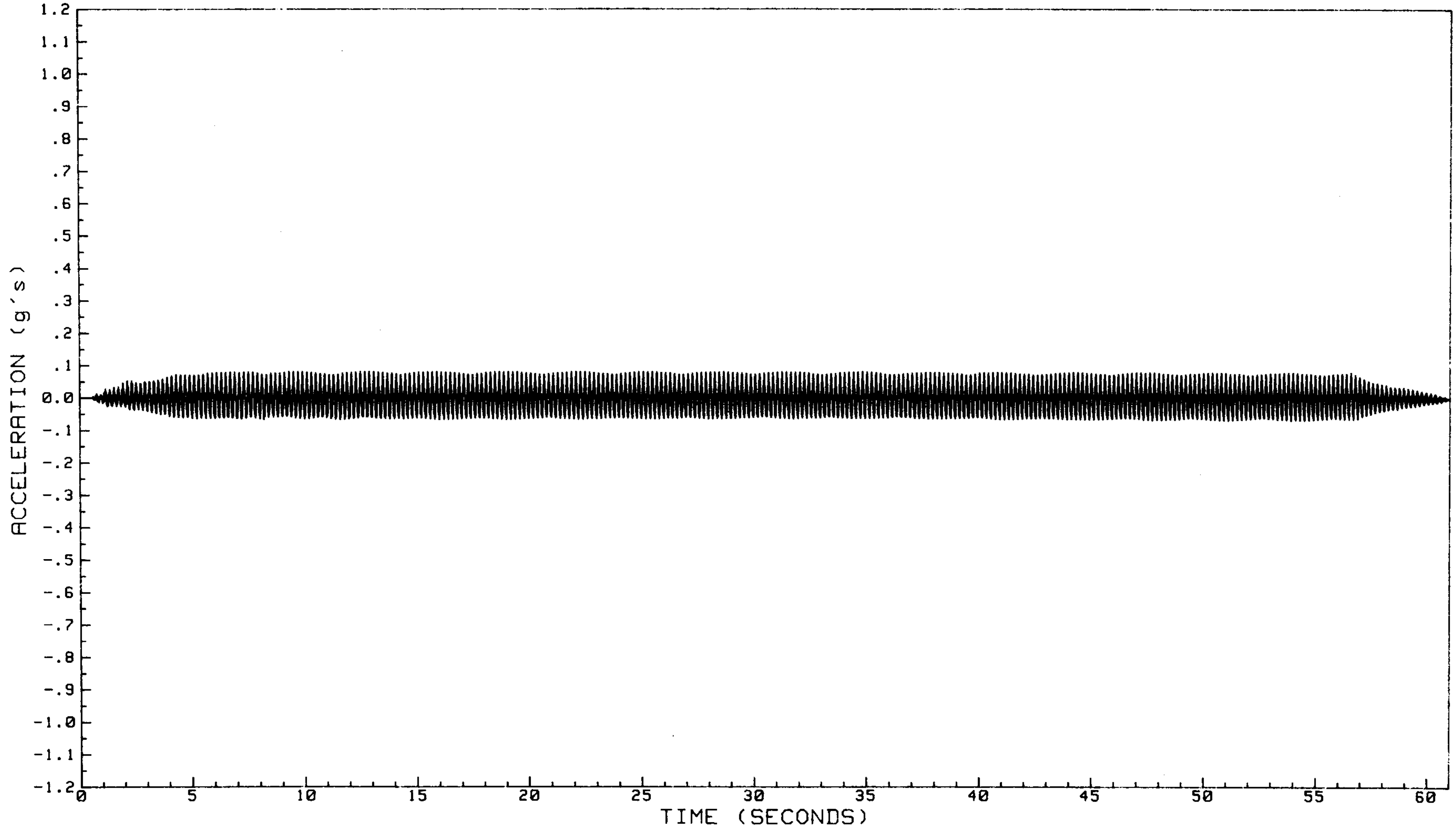
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 3
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 18
ACCELEROMETER LOCATION: ELEVATION 14.0" NO OFFSET FROM CENTERLINE



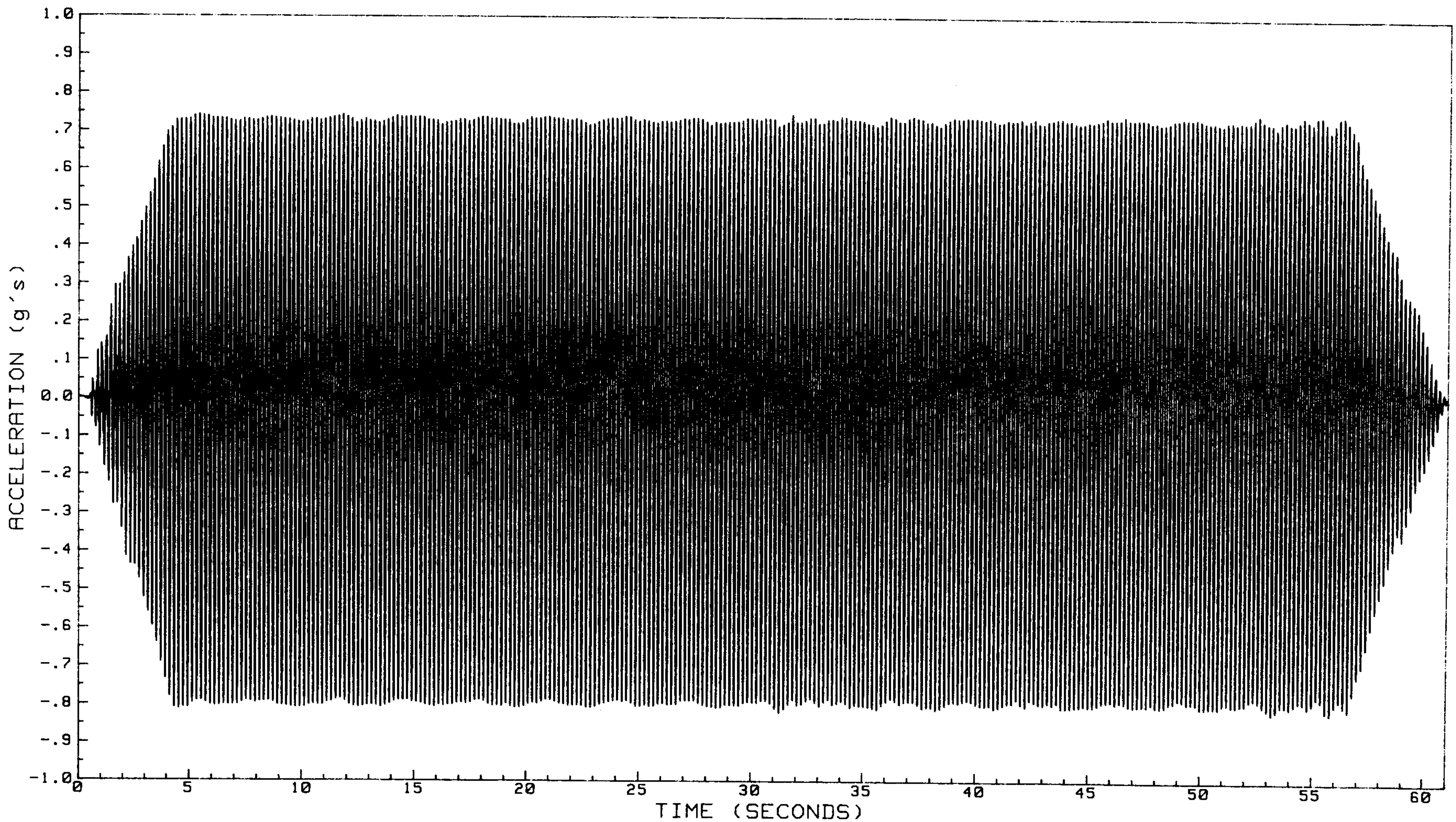
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 3
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 74
ACCELEROMETER LOCATION: ACTUATOR



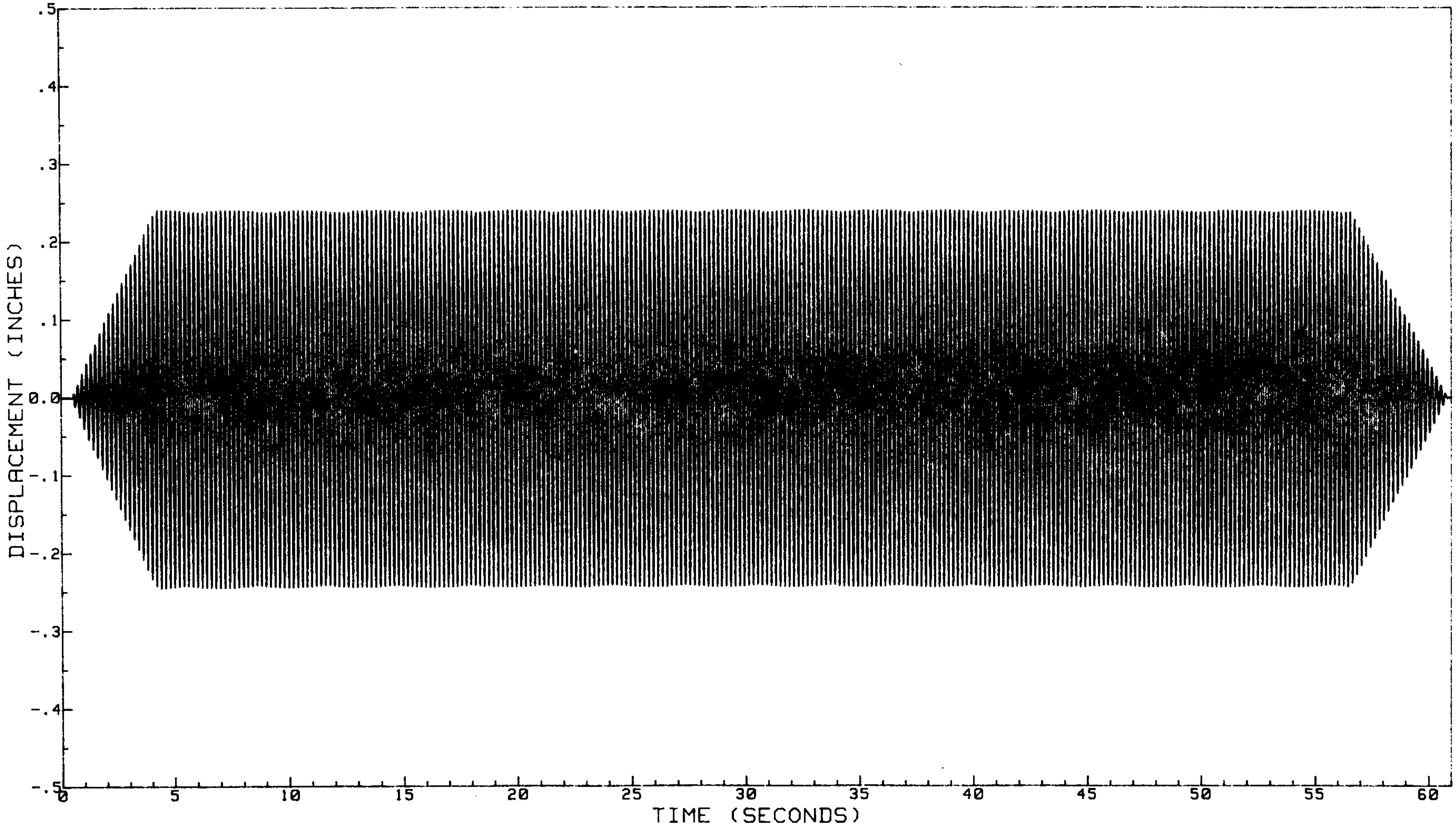
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 3
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 72
ACCELEROMETER LOCATION: BOX FRAME



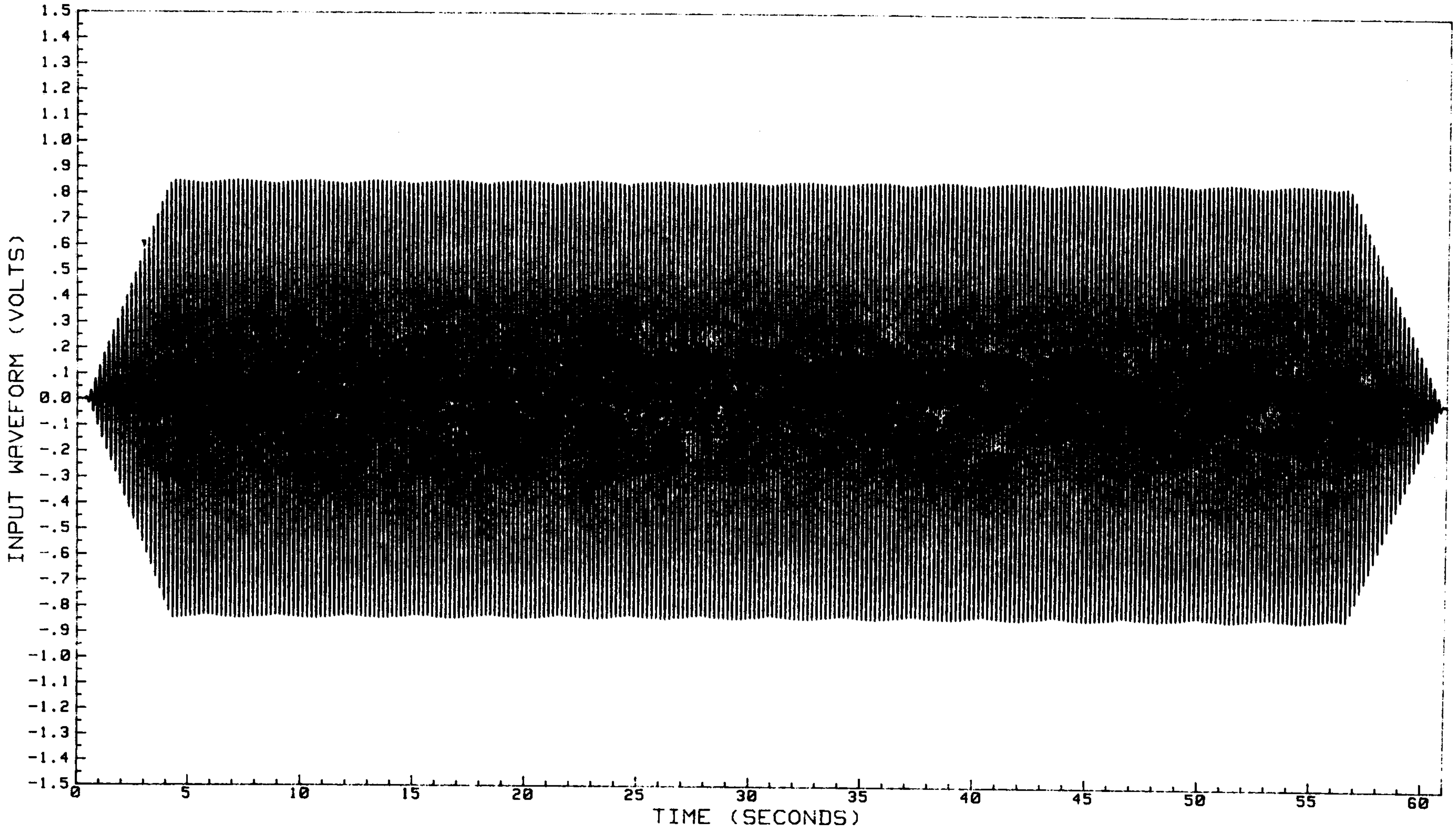
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 3
ACCELERATION MEASURED VERTICAL -- ACCELEROMETER NO. 77
ACCELEROMETER LOCATION: BOX FRAME



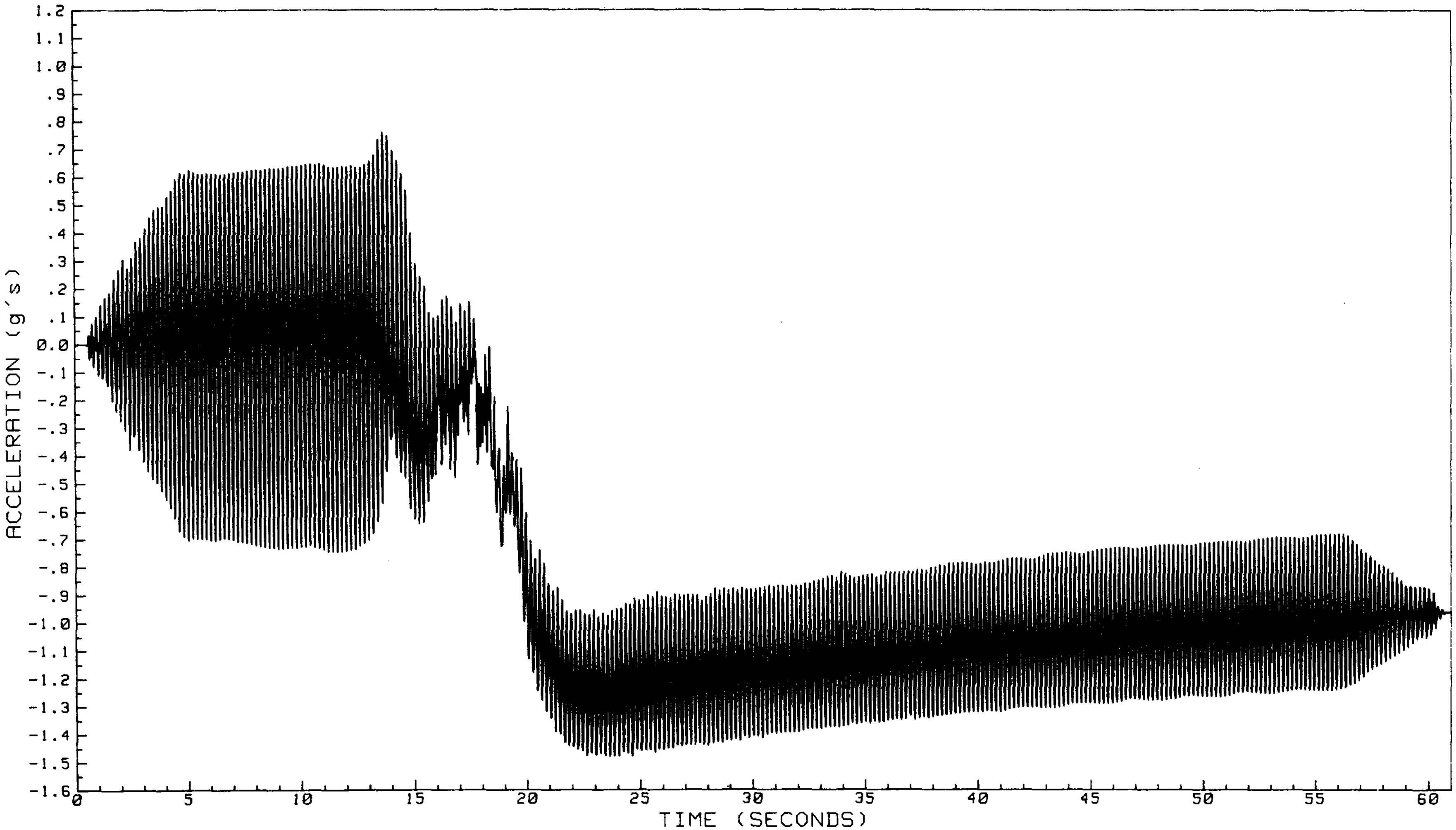
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 3
DISPLACEMENT MEASURED HORIZONTAL
LVDT LOCATED IN ACTUATOR



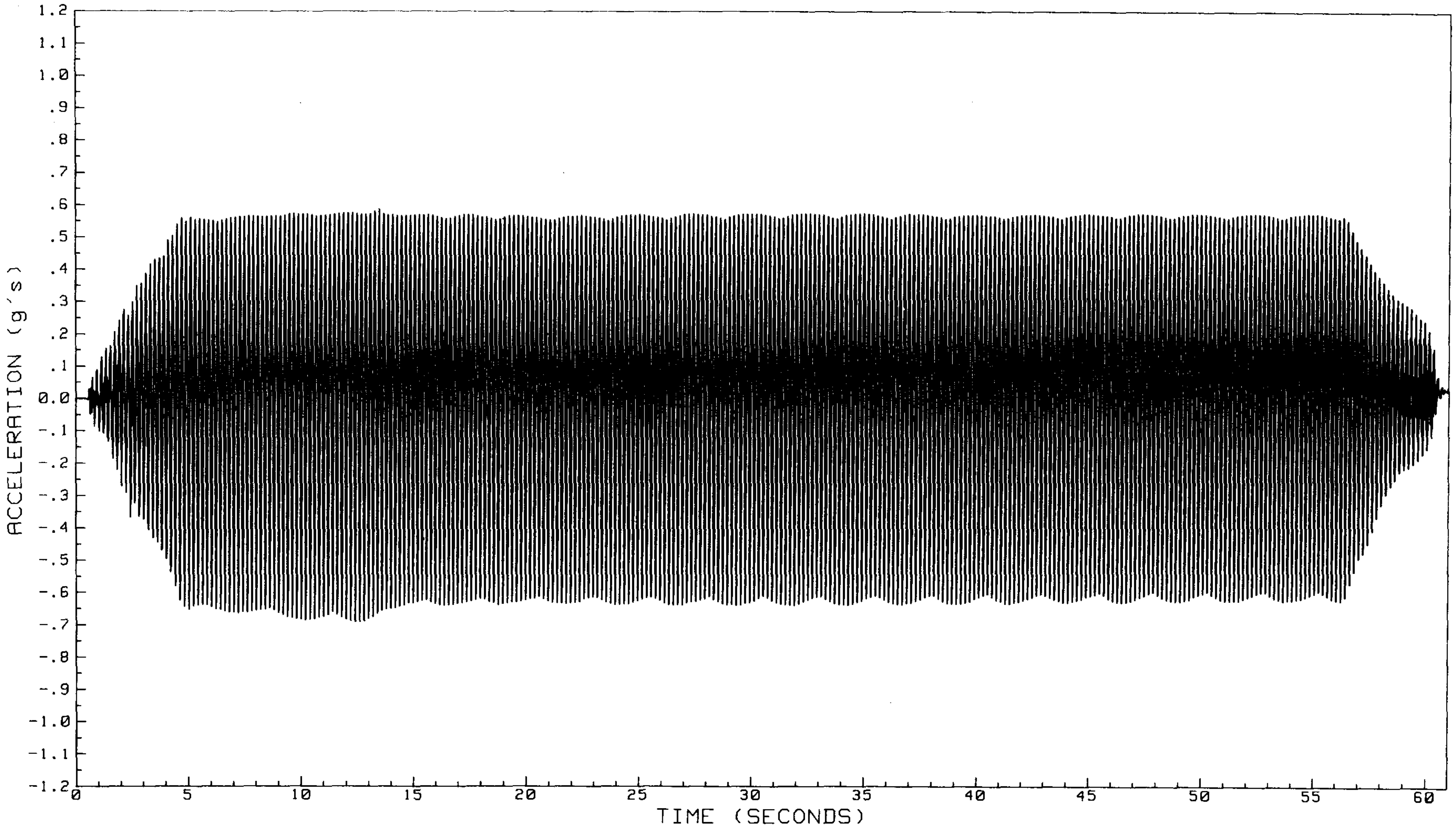
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 3
INPUT RAMP WAVEFORM
MEASURED AT RAMP GENERATOR OUTPUT



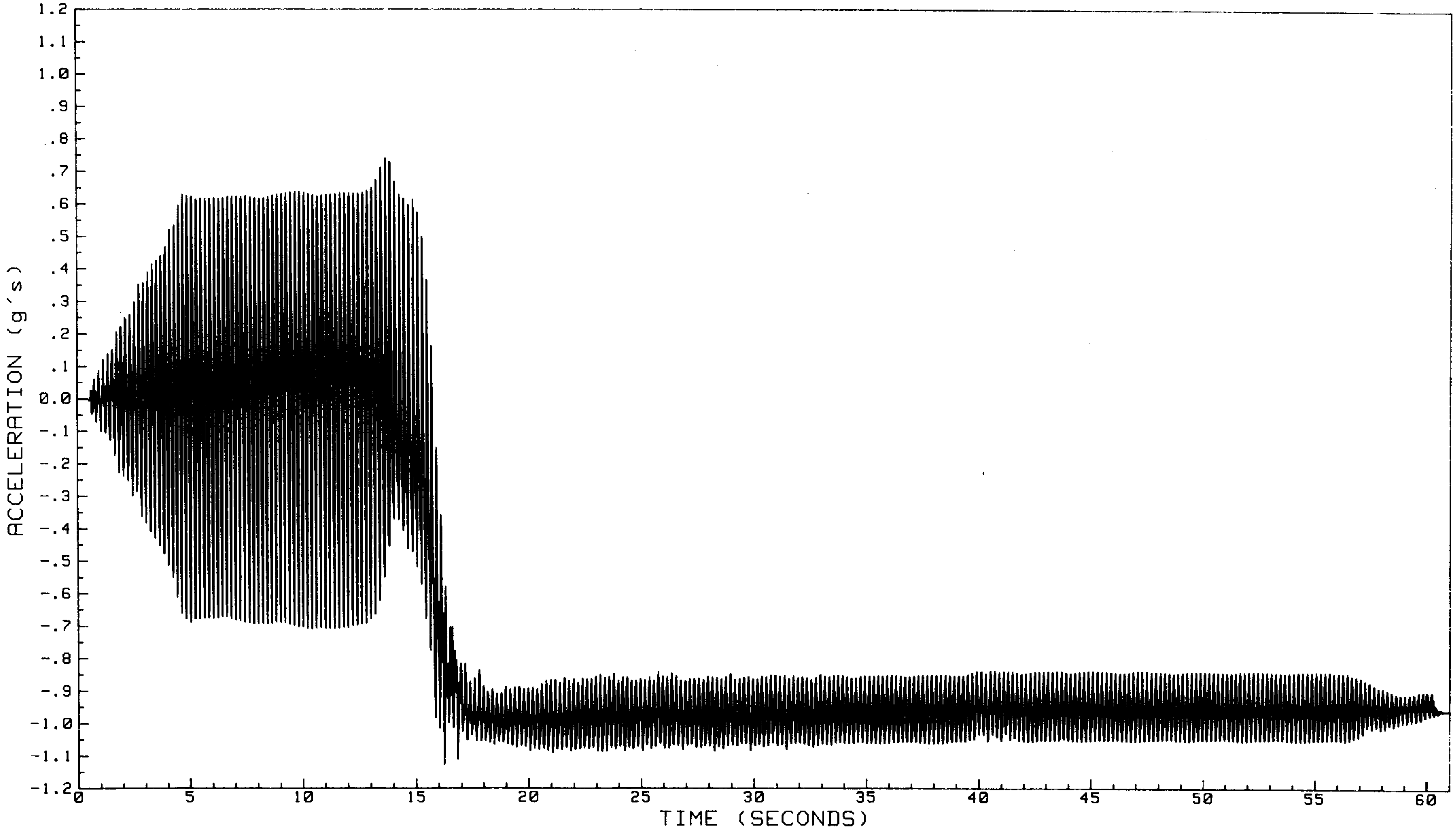
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 4
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 10
ACCELEROMETER LOCATION: ELEVATION 13.0" OFFSET FROM CENTERLINE 5.0" LT



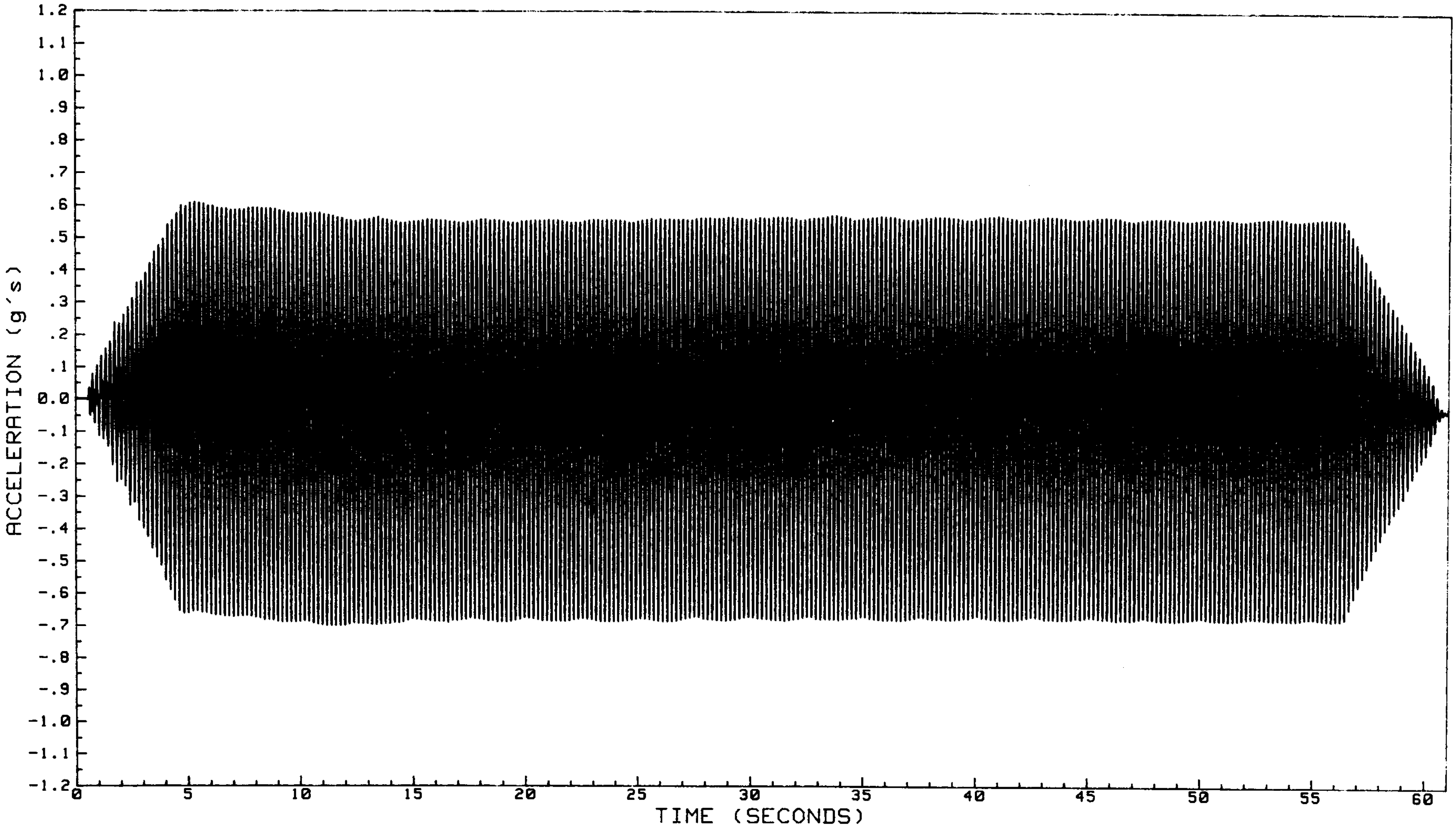
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 4
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 8
ACCELEROMETER LOCATION: ELEVATION 6.0" OFFSET FROM CENTERLINE 14.0" LT



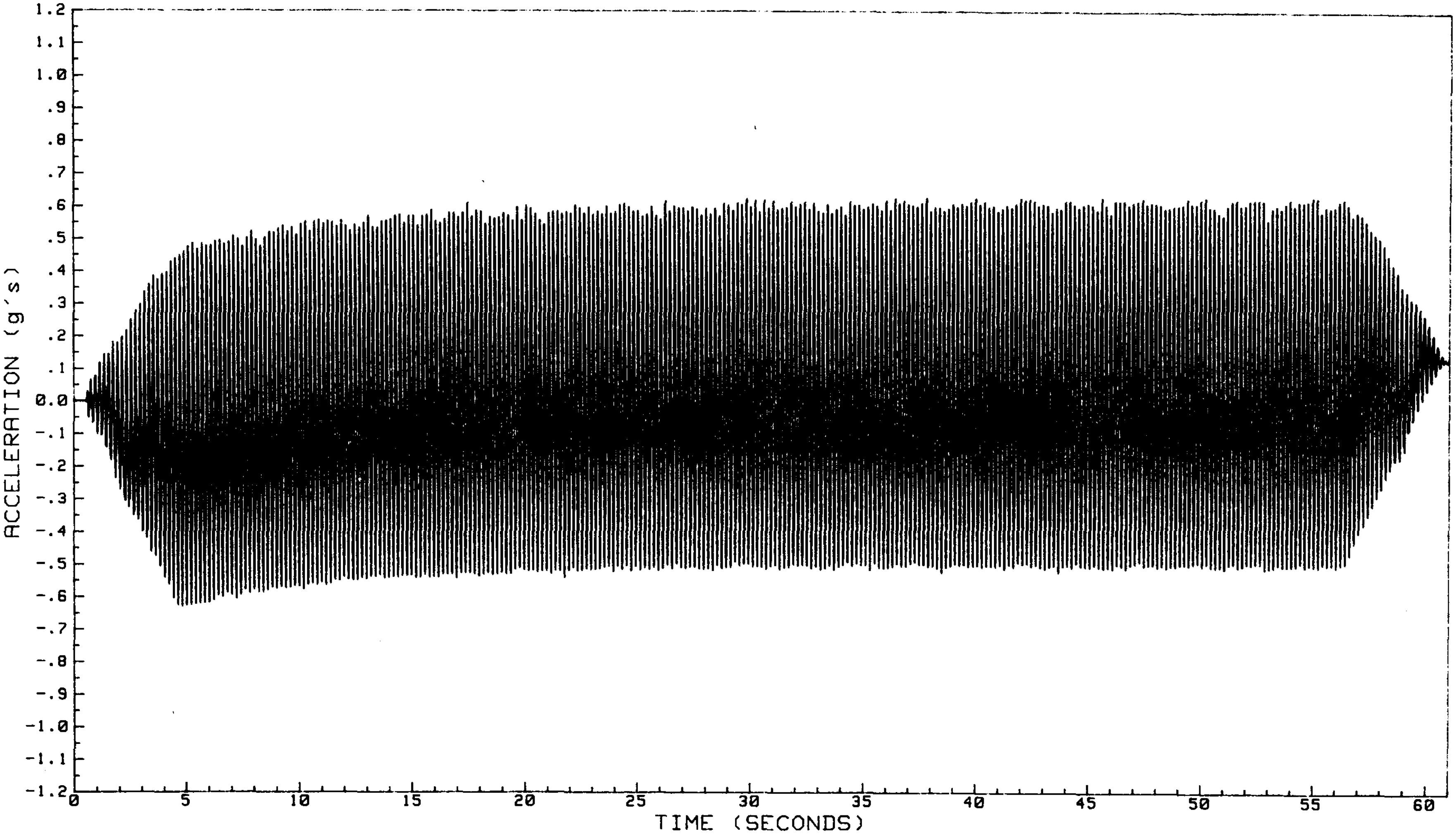
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 4
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 13
ACCELEROMETER LOCATION: ELEVATION 7.0" OFFSET FROM CENTERLINE 18.0" LT



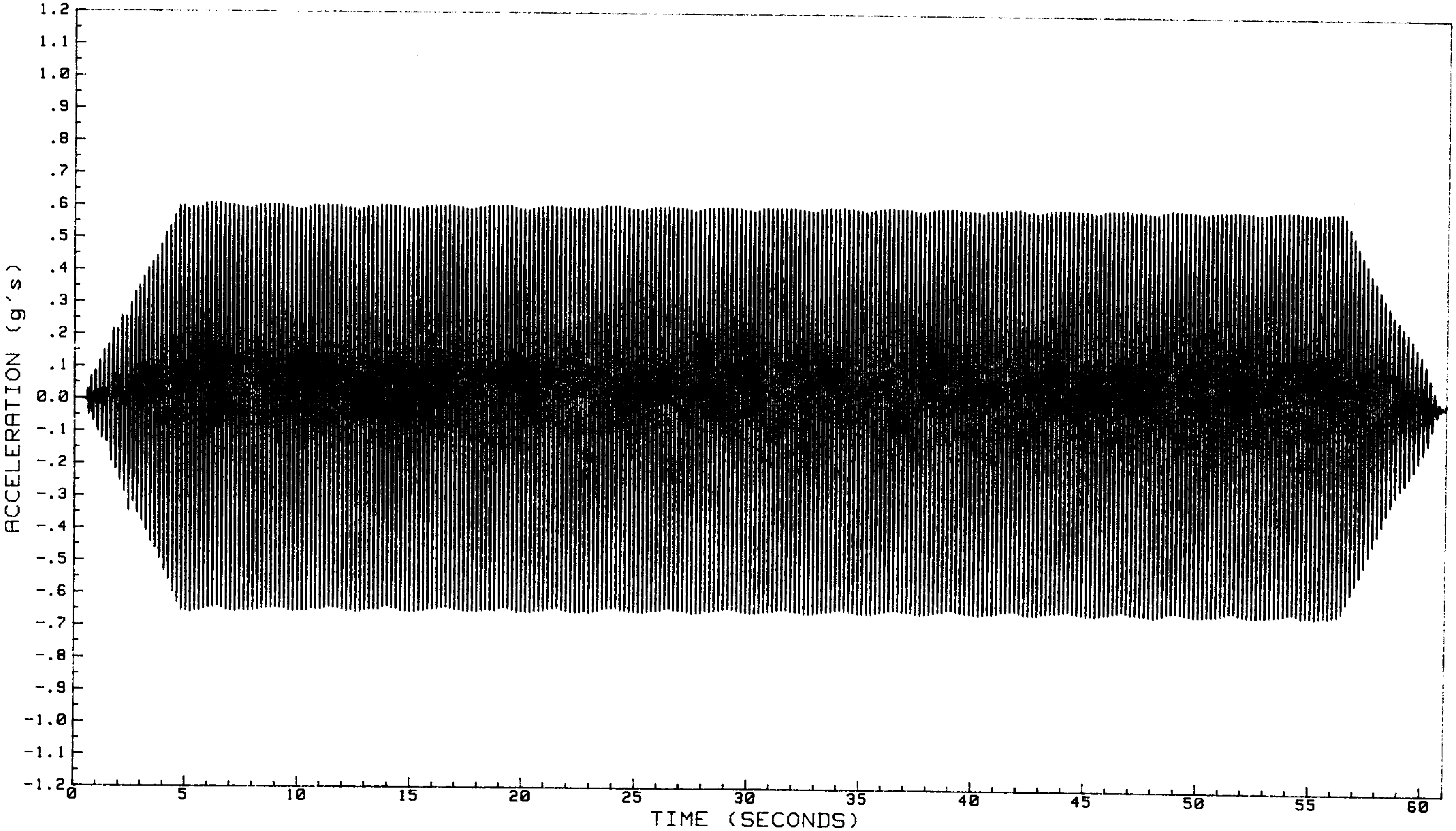
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 4
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 9
ACCELEROMETER LOCATION: ELEVATION 11.4" OFFSET FROM CENTERLINE 7.5" RT



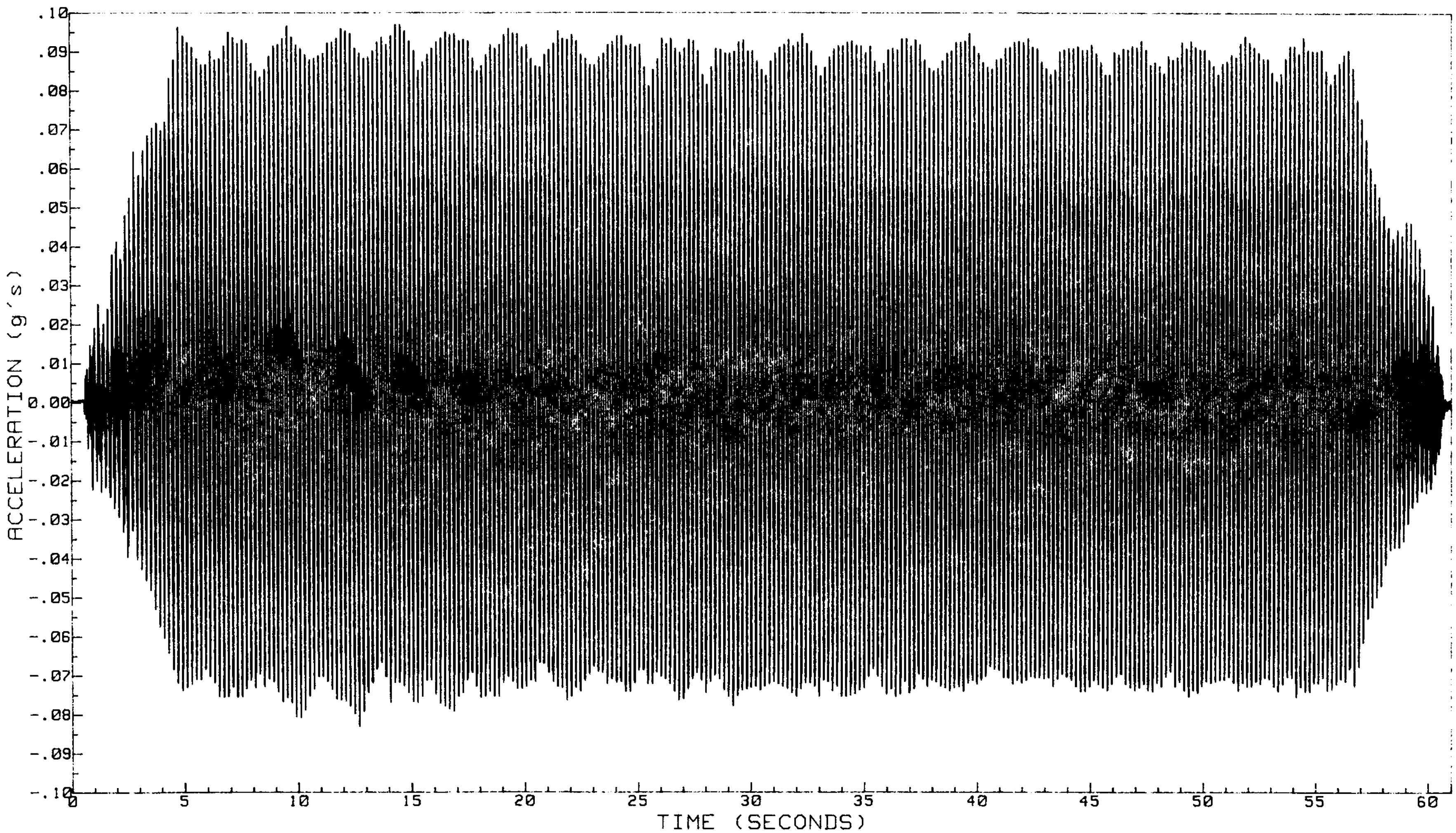
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 4
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 74
ACCELEROMETER LOCATION: ACTUATOR



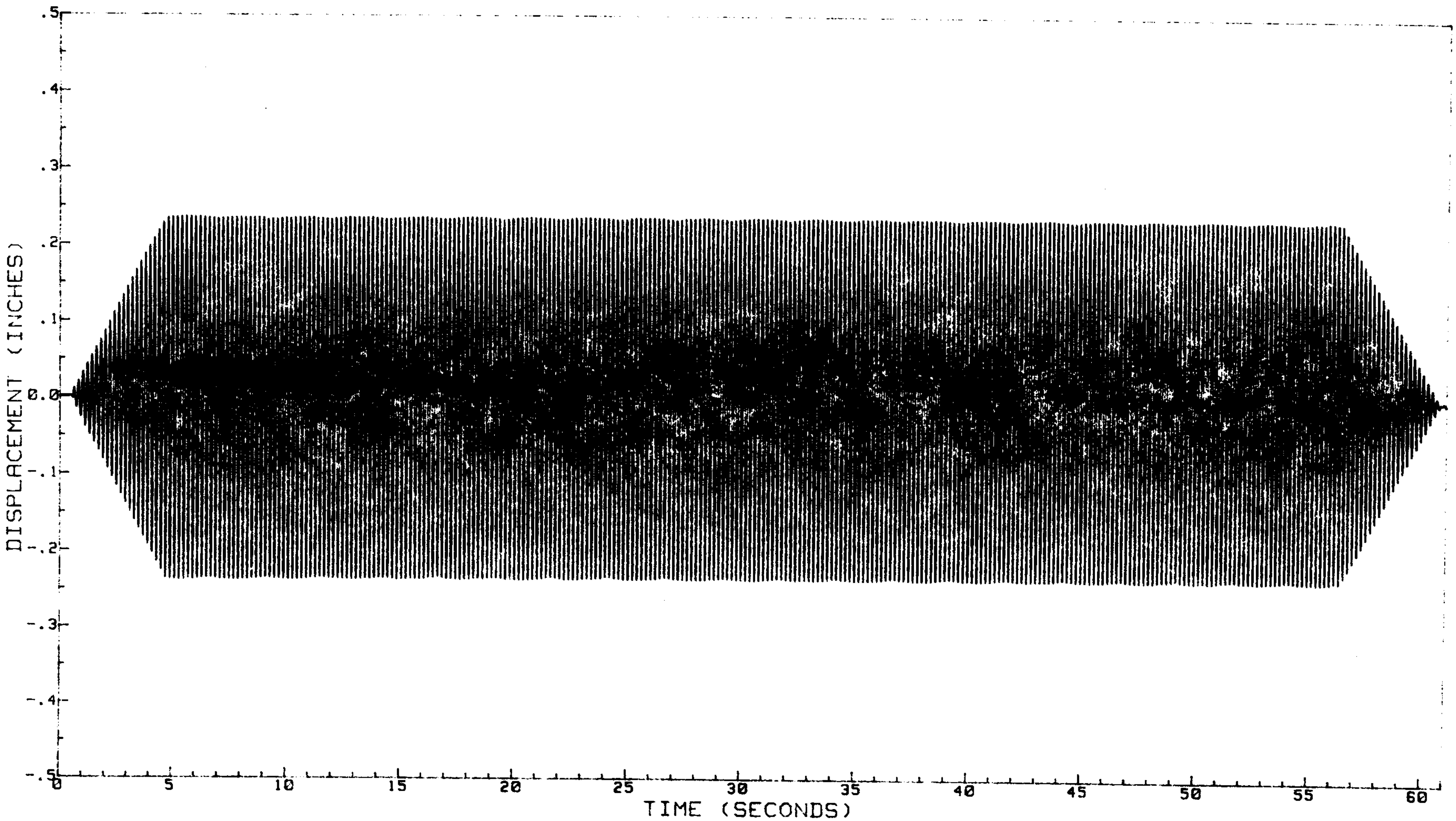
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 4
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 72
ACCELEROMETER LOCATION: BOX FRAME



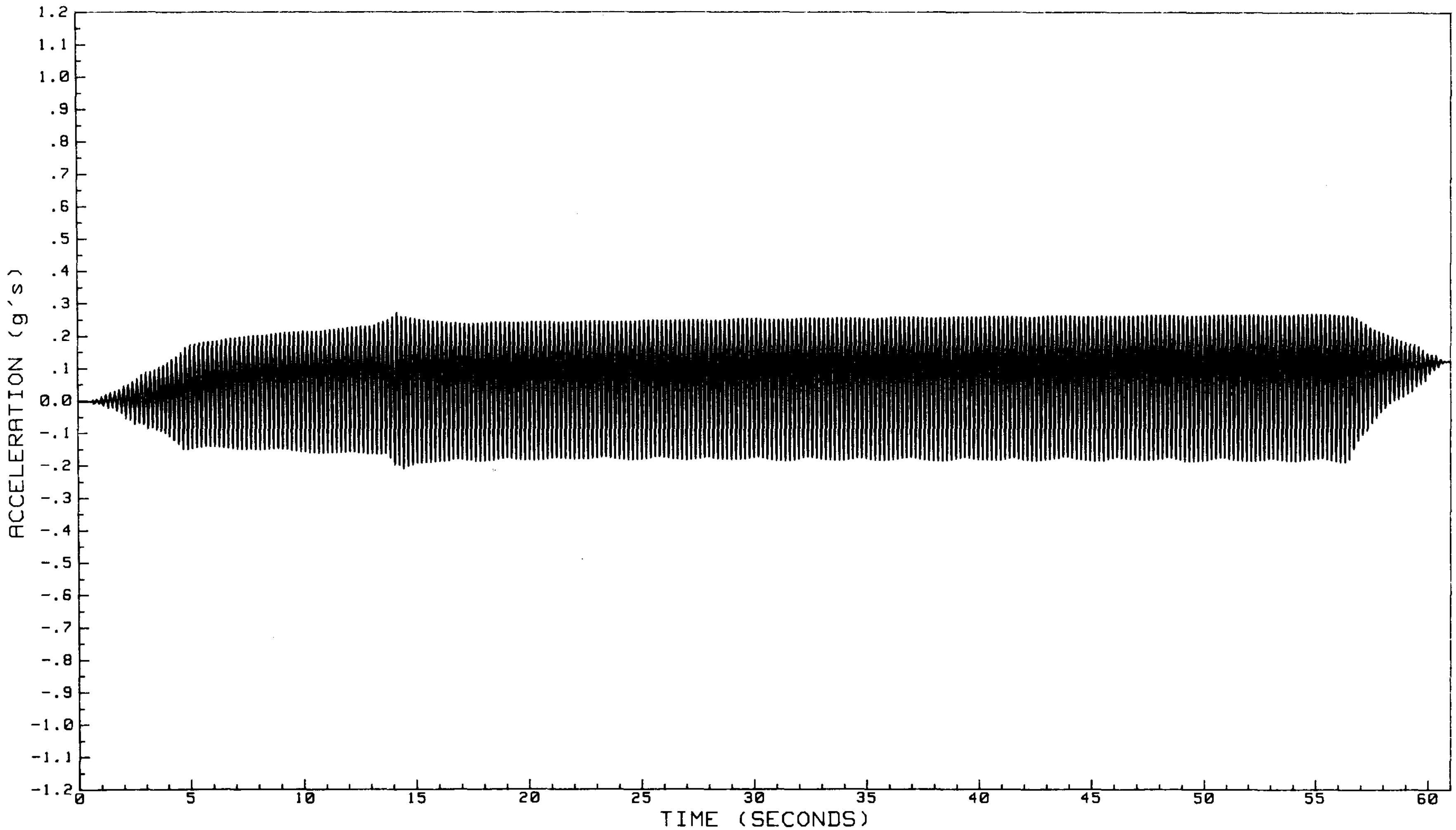
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 4
ACCELERATION MEASURED VERTICAL -- ACCELEROMETER NO. 77
ACCELEROMETER LOCATION: BOX FRAME



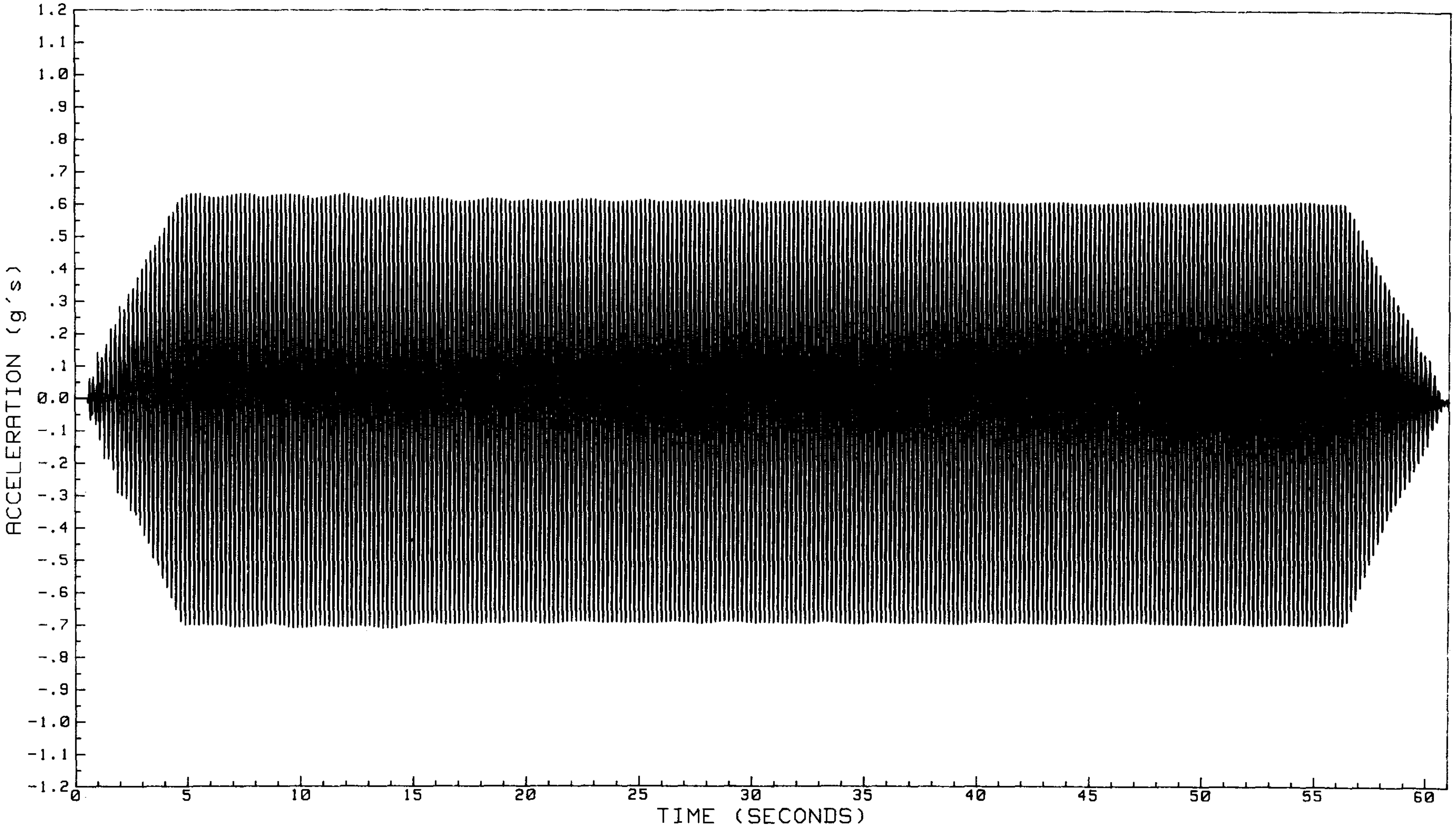
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 4
DISPLACEMENT MEASURED HORIZONTAL
LVDT LOCATED IN ACTUATOR



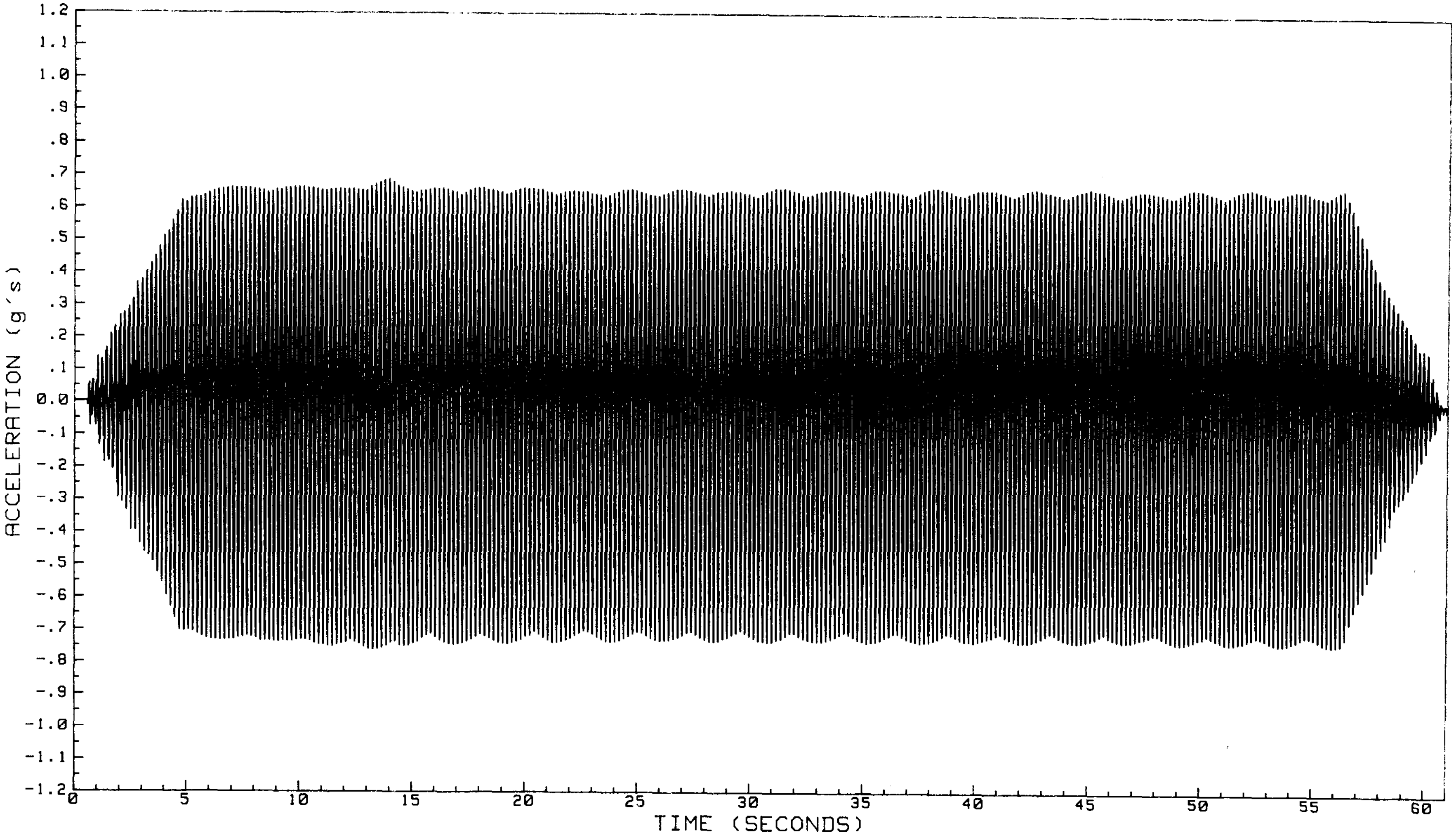
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 5
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 10
ACCELEROMETER LOCATION: ELEVATION 17.0" OFFSET FROM CENTERLINE 3.5" RT



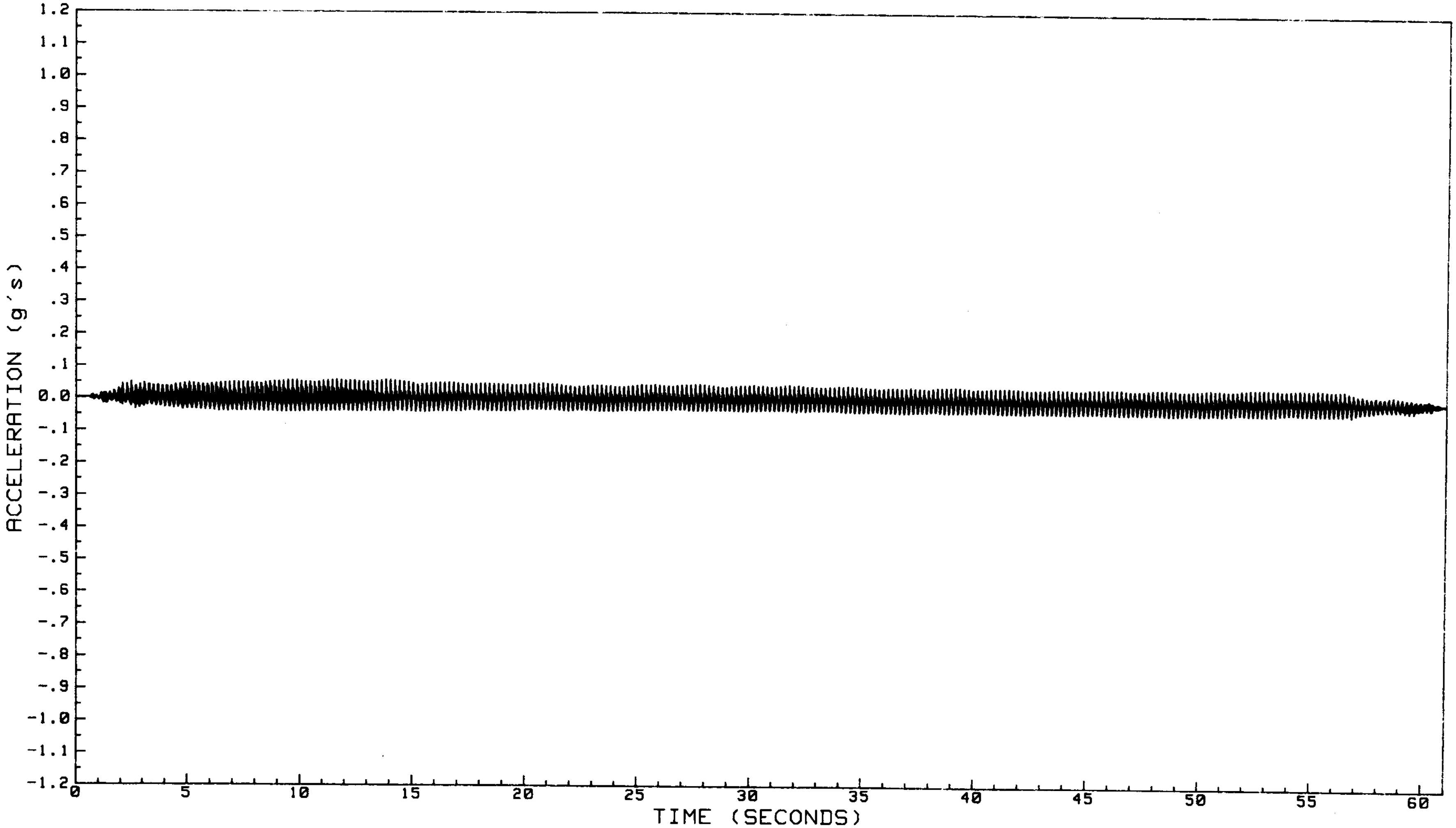
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 5
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 8
ACCELEROMETER LOCATION: ELEVATION 7.0" OFFSET FROM CENTERLINE 18.0" LT



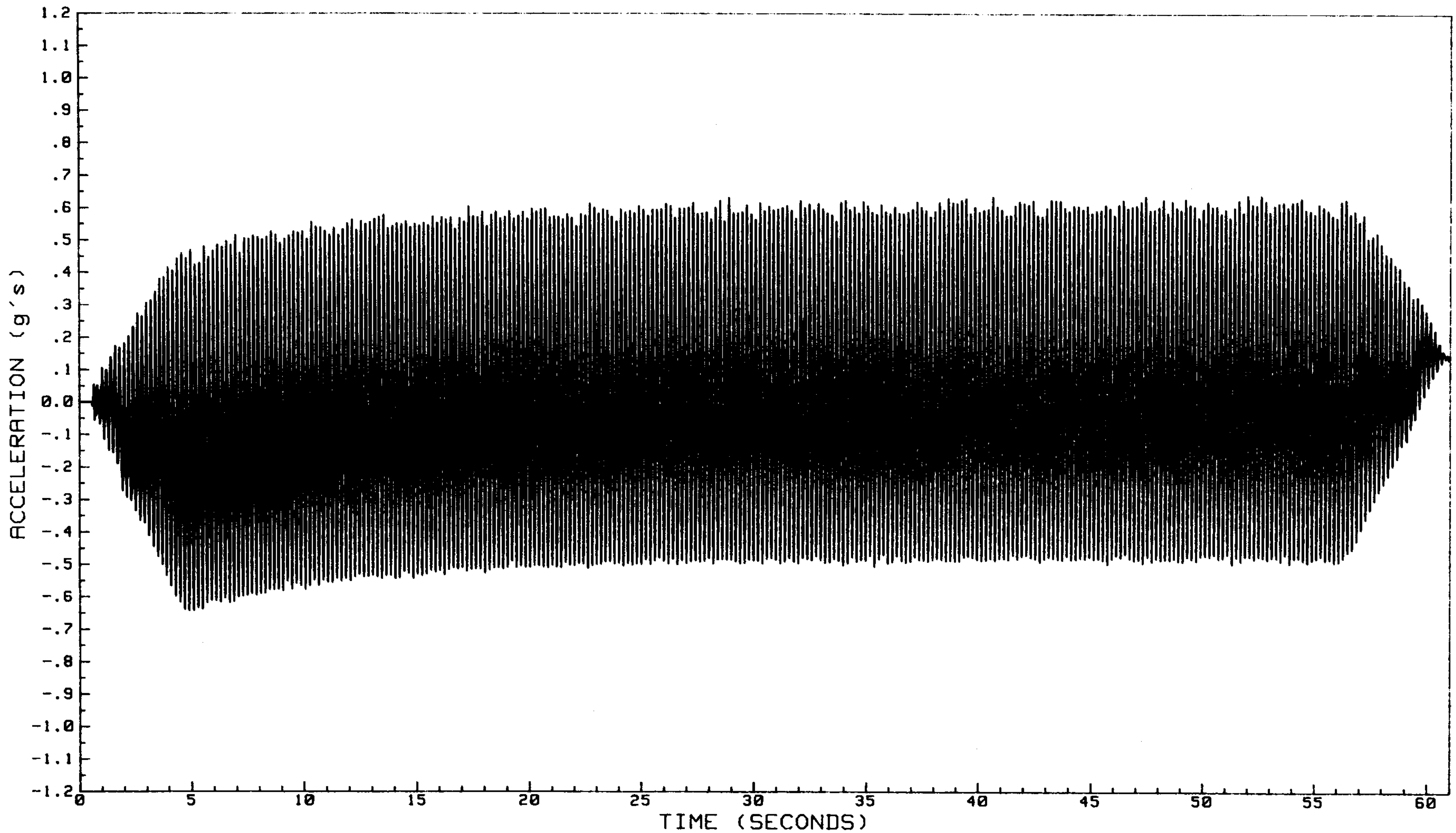
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 5
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 13
ACCELEROMETER LOCATION: ELEVATION 14.0" OFFSET FROM CENTERLINE 10.0" RT



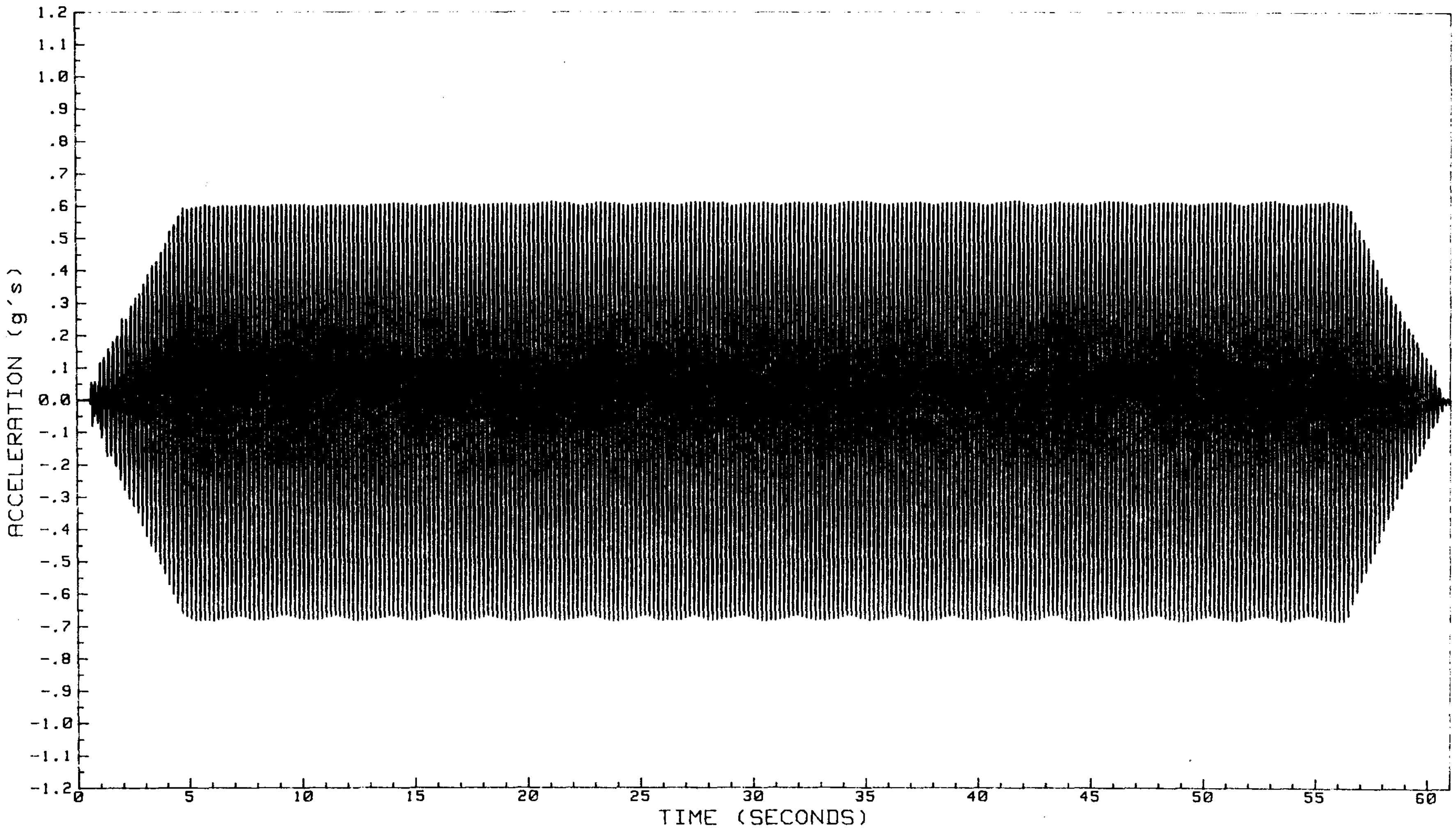
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 5
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 9
ACCELEROMETER LOCATION: ELEVATION 12.0" OFFSET FROM CENTERLINE 4.0" RT



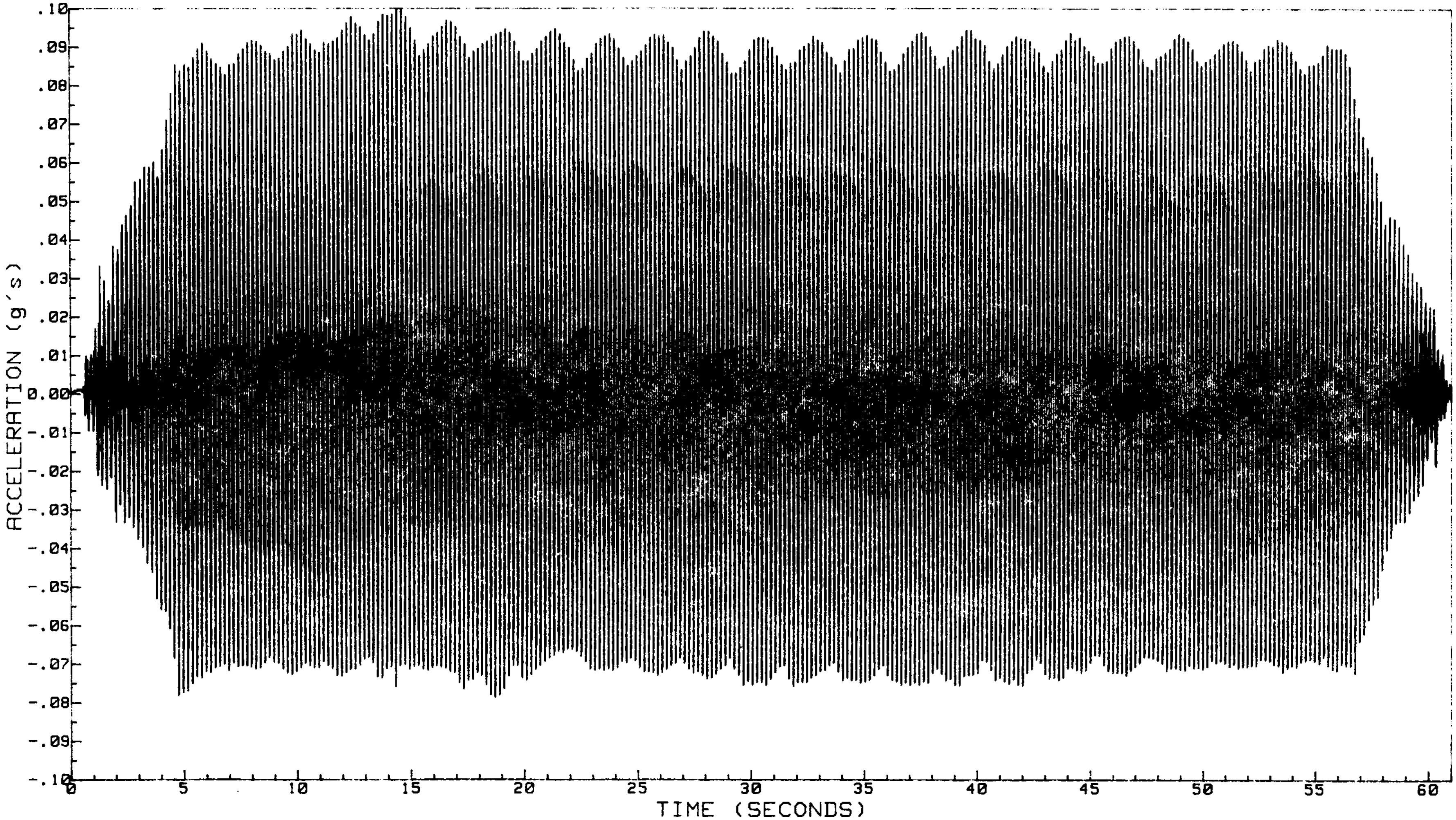
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 5
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 74
ACCELEROMETER LOCATION: ACTUATOR

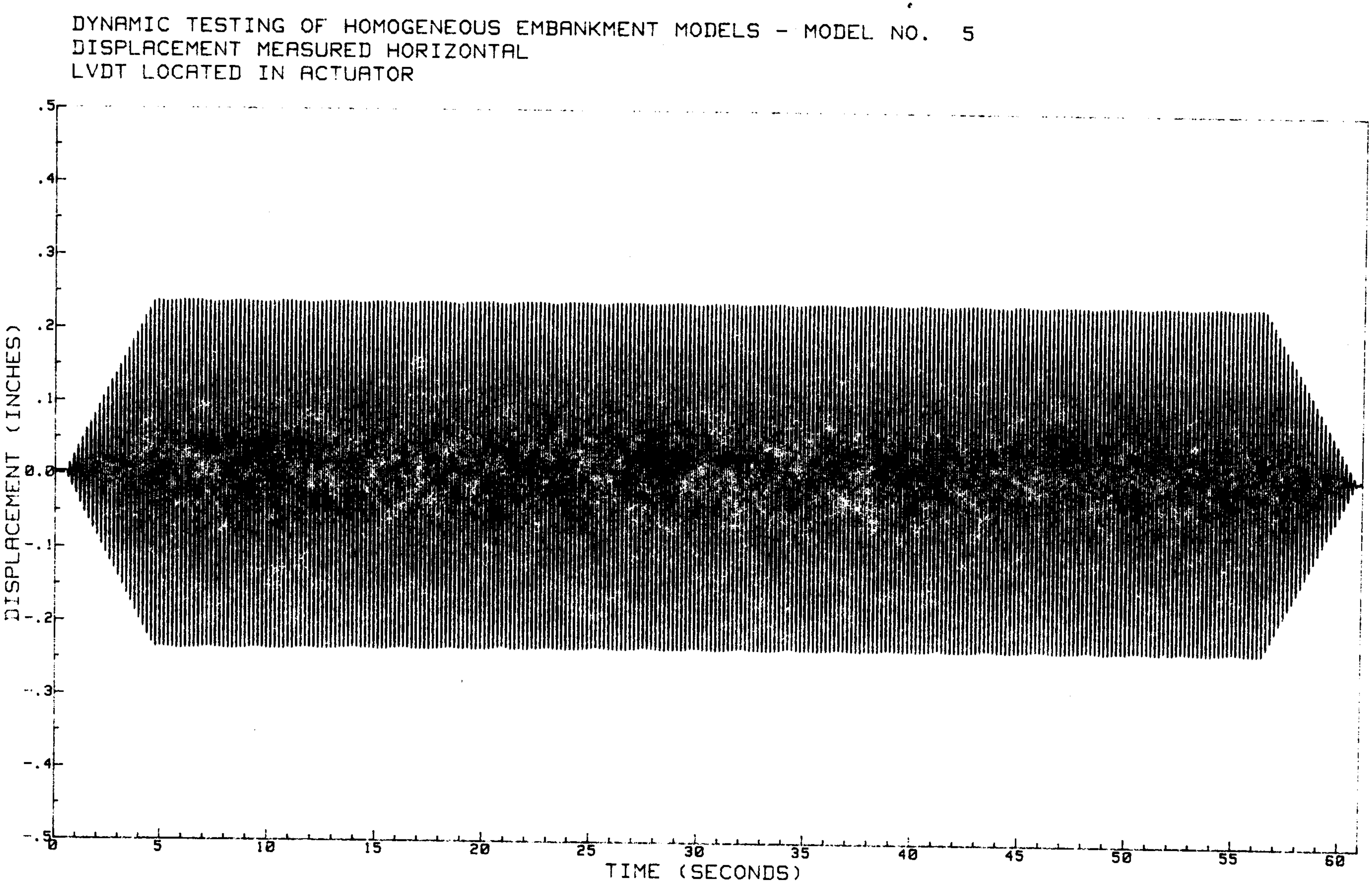


DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 5
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 72
ACCELEROMETER LOCATION: BOX FRAME

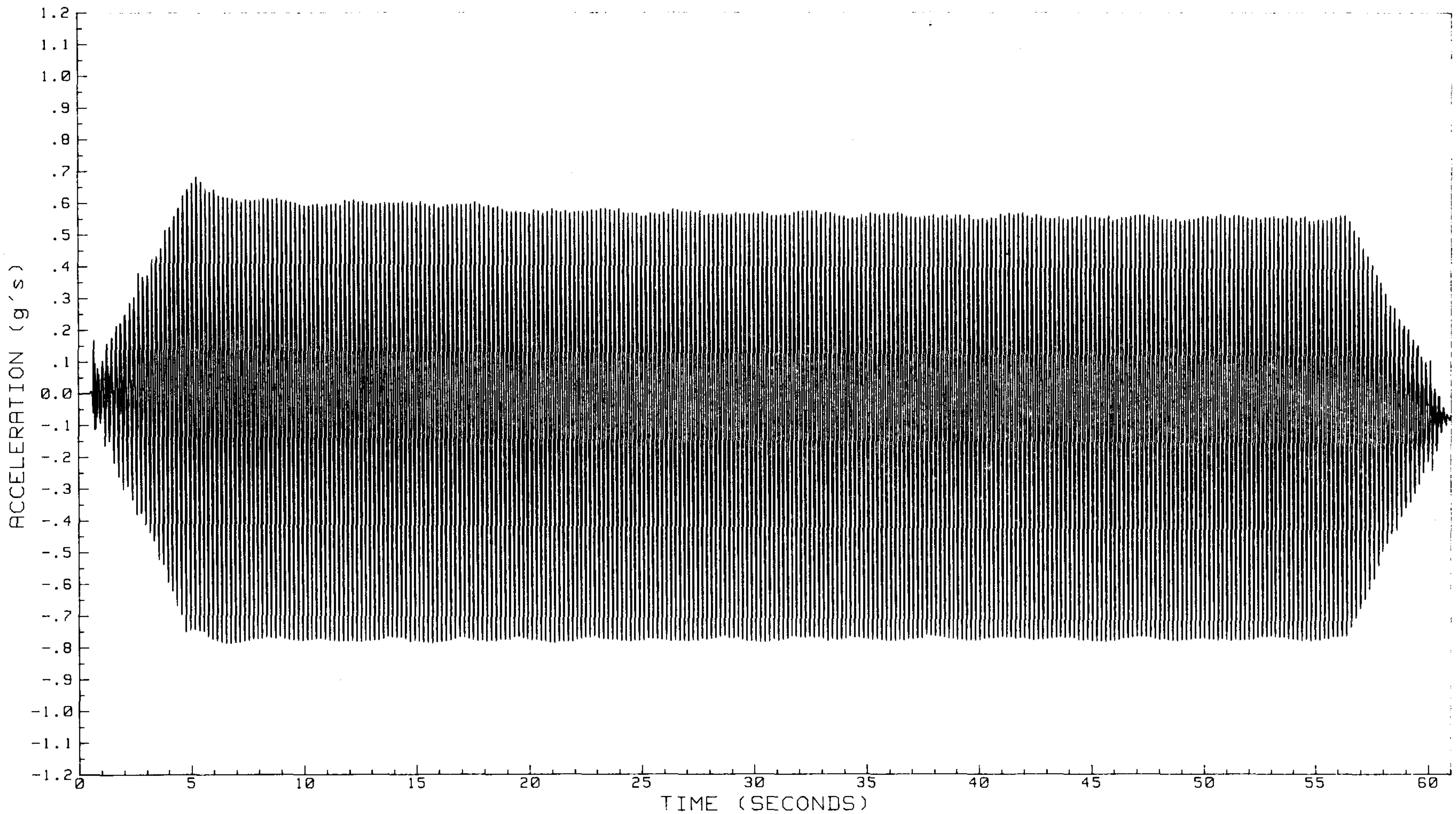


DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 5
ACCELERATION MEASURED VERTICAL -- ACCELEROMETER NO. 77
ACCELEROMETER LOCATION: BOX FRAME

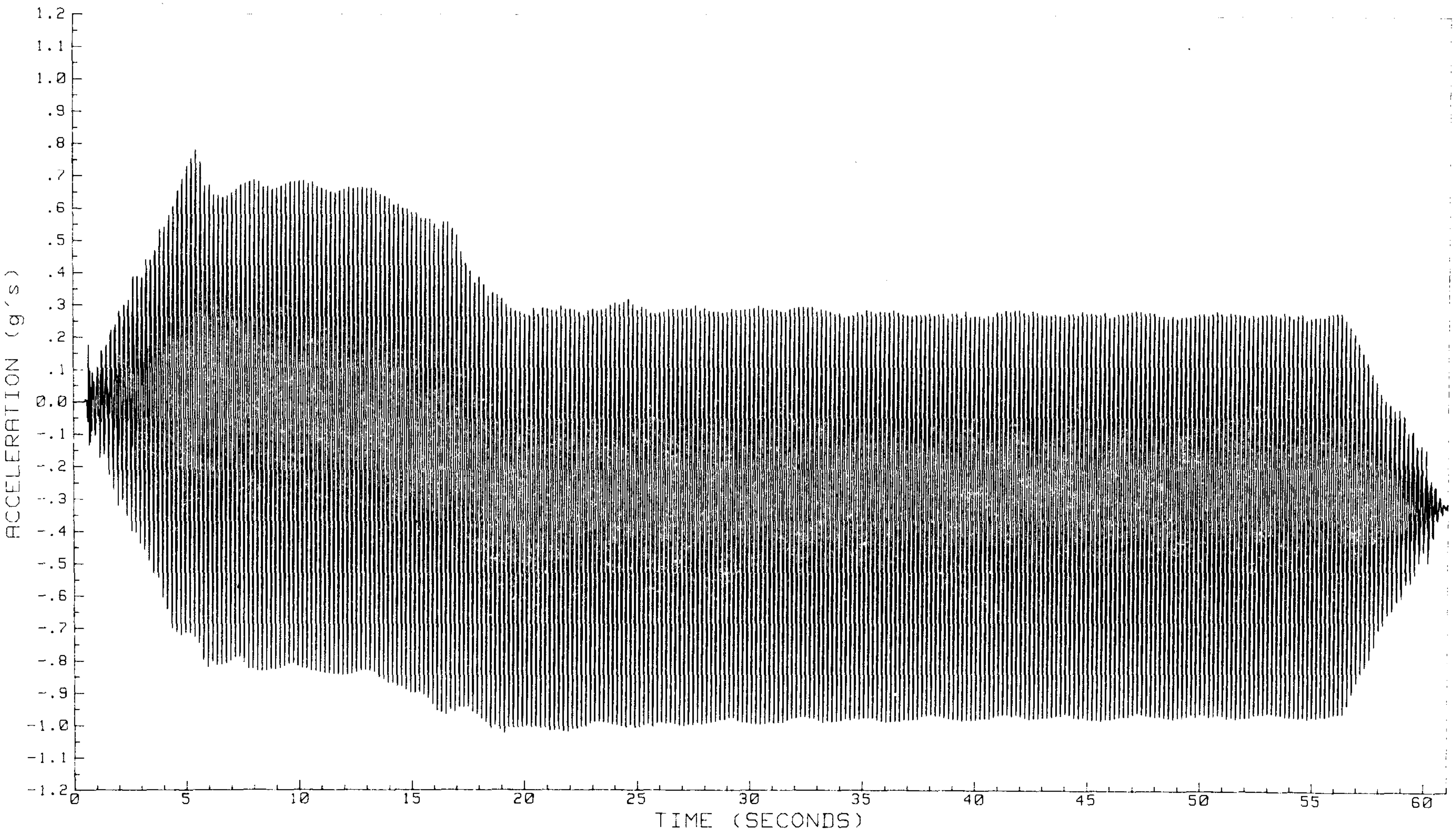



DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 5
DISPLACEMENT MEASURED HORIZONTAL
LVDT LOCATED IN ACTUATOR
DISPLACEMENT (INCHES)
.5
.4
.3
.2
.1
0.0
-.1
-.2
-.3
-.4
-.5
0
5
10
15
20
25
30
35
40
45
50
55
60
TIME (SECONDS)

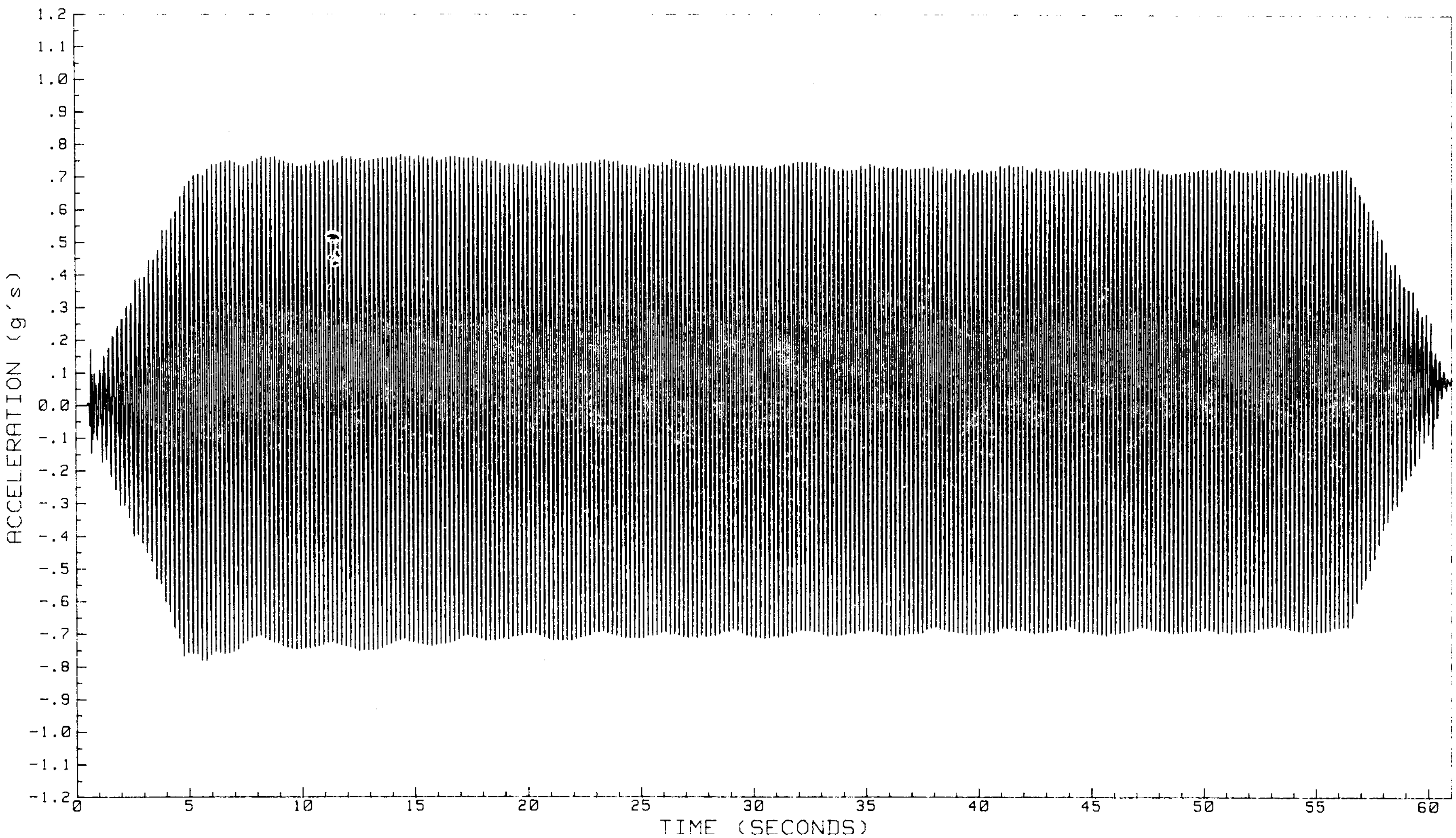
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 6
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 17
ACCELEROMETER LOCATION: ELEVATION 7.2" OFFSET FROM CENTERLINE 11.5 " LT



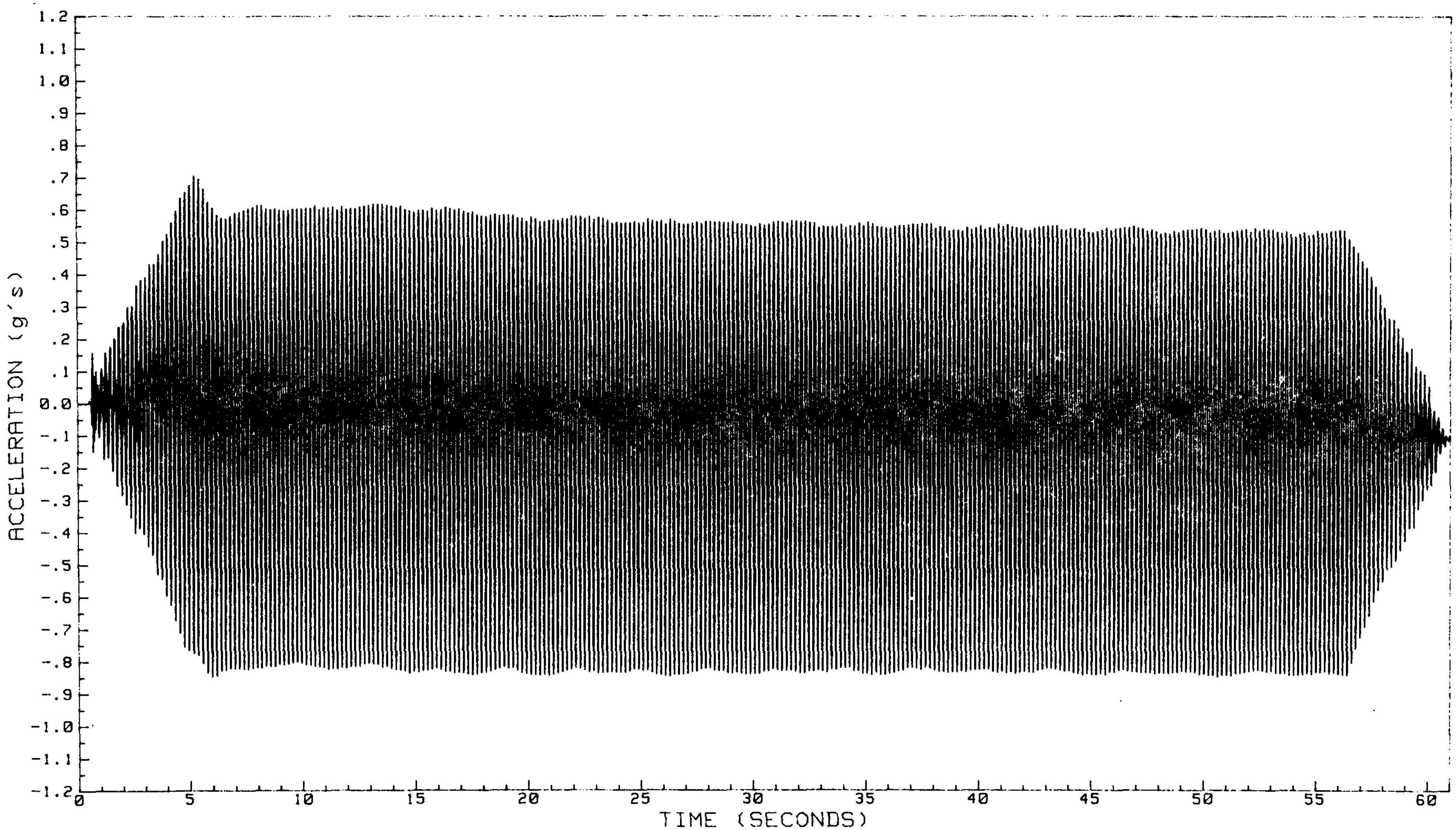
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 6
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 8
ACCELEROMETER LOCATION: ELEVATION 12.1" OFFSET FROM CENTERLINE 3" LT



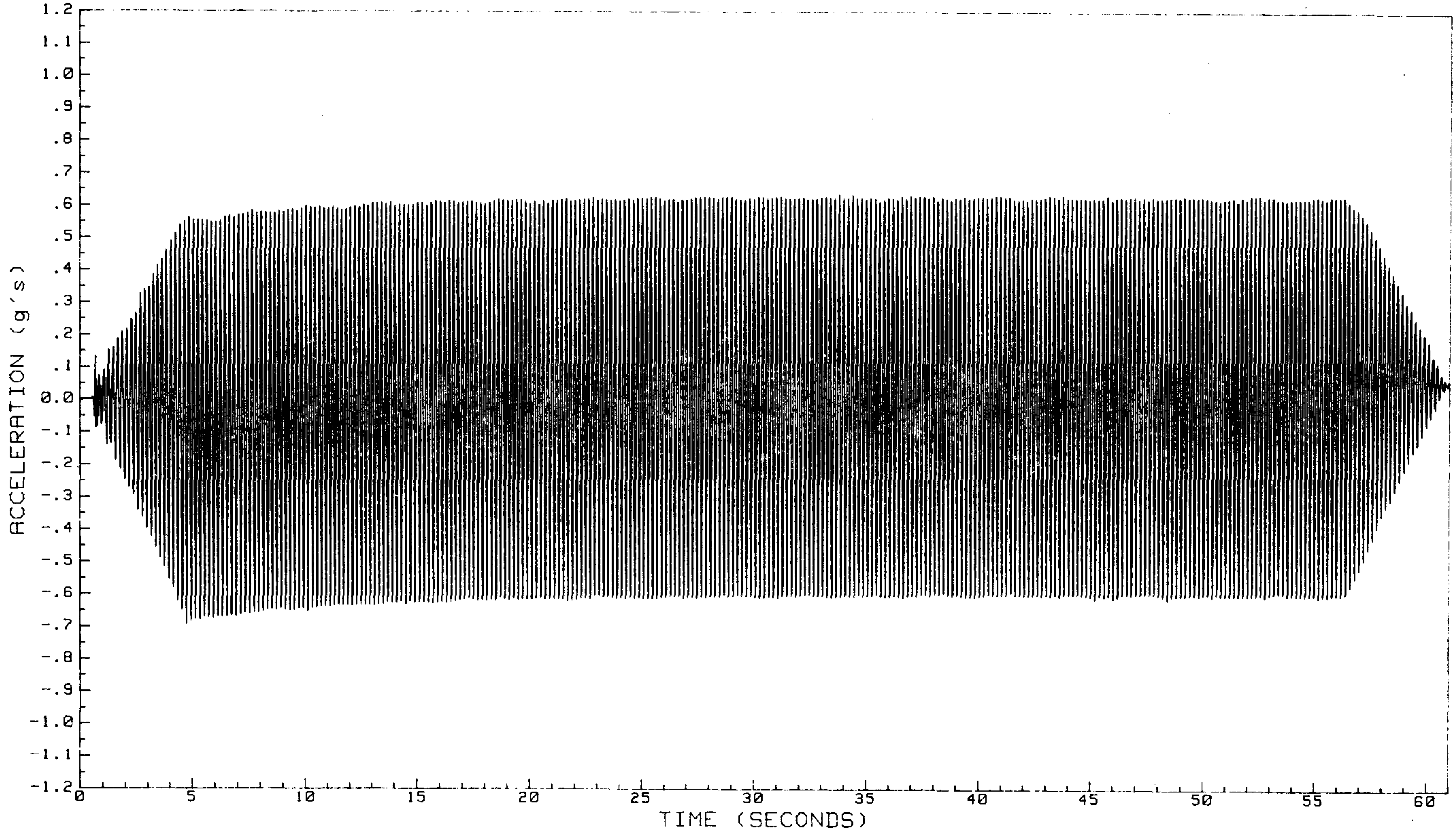
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 6
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 1
ACCELEROMETER LOCATION: ELEVATION 9.6" OFFSET FROM CENTERLINE 5" RT



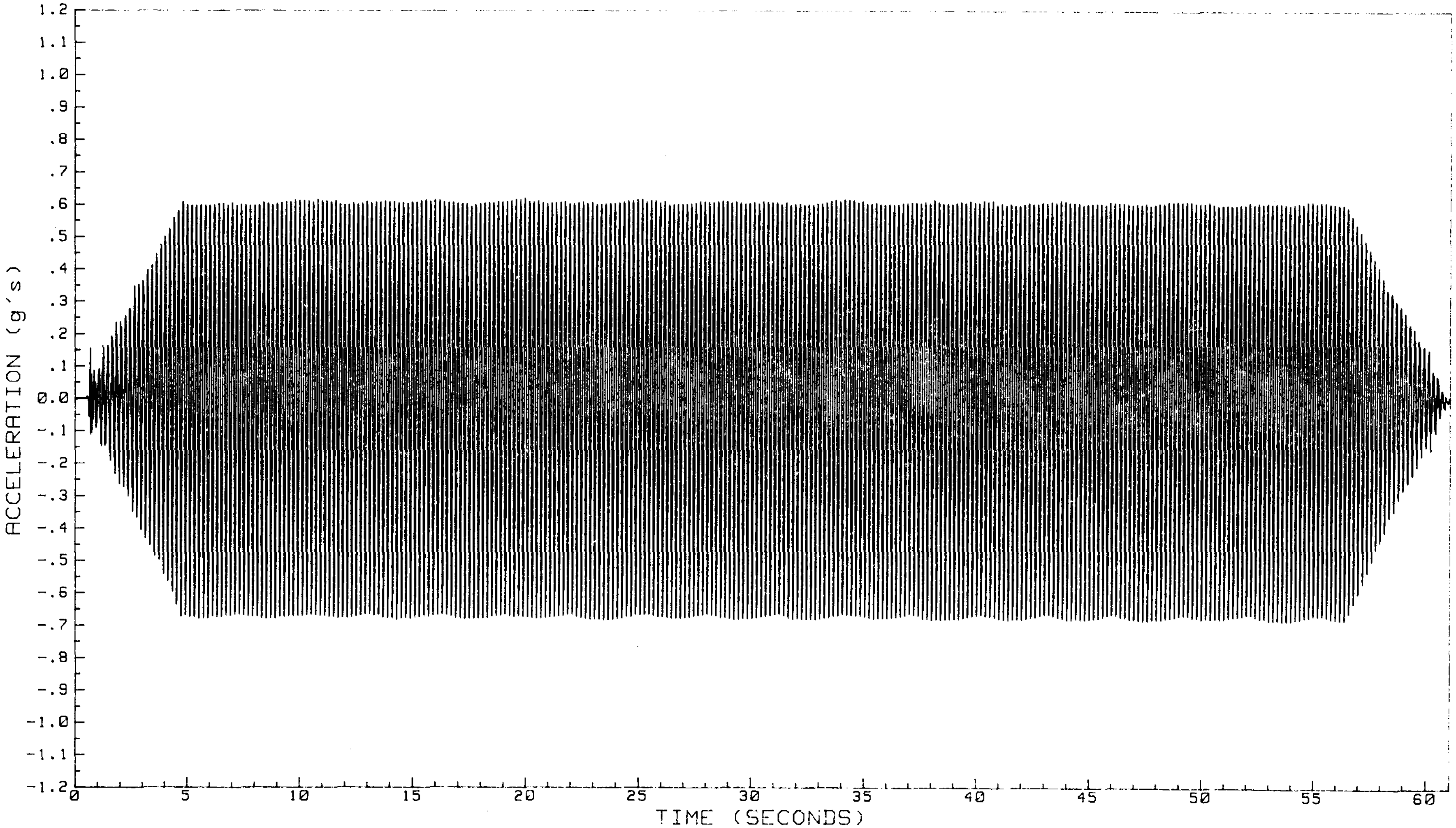
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 6
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 9
ACCELEROMETER LOCATION: ELEVATION 8.7" OFFSET FROM CENTERLINE 11.5" LT



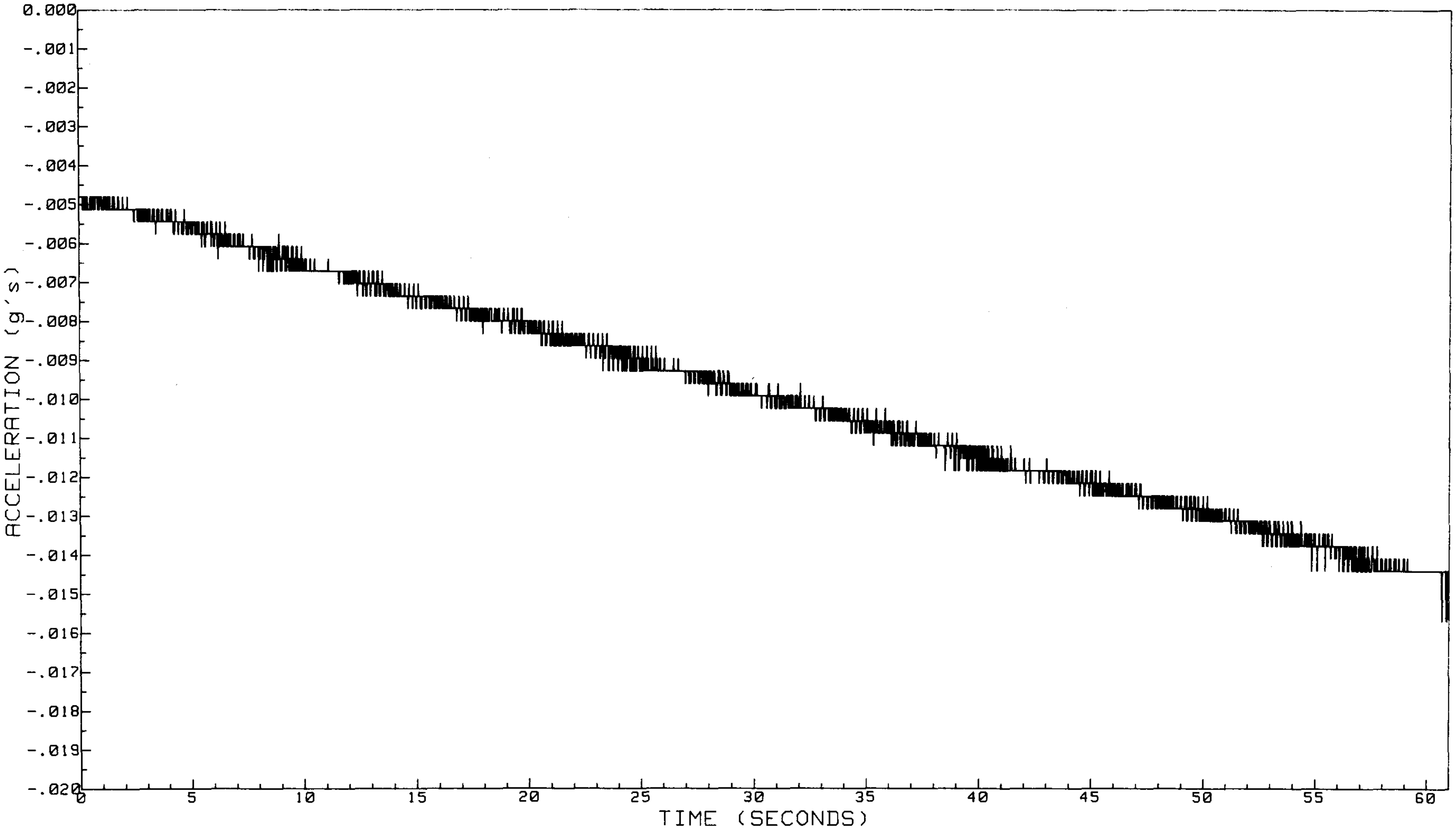
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 6
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 74
ACCELEROMETER LOCATION: ACTUATOR



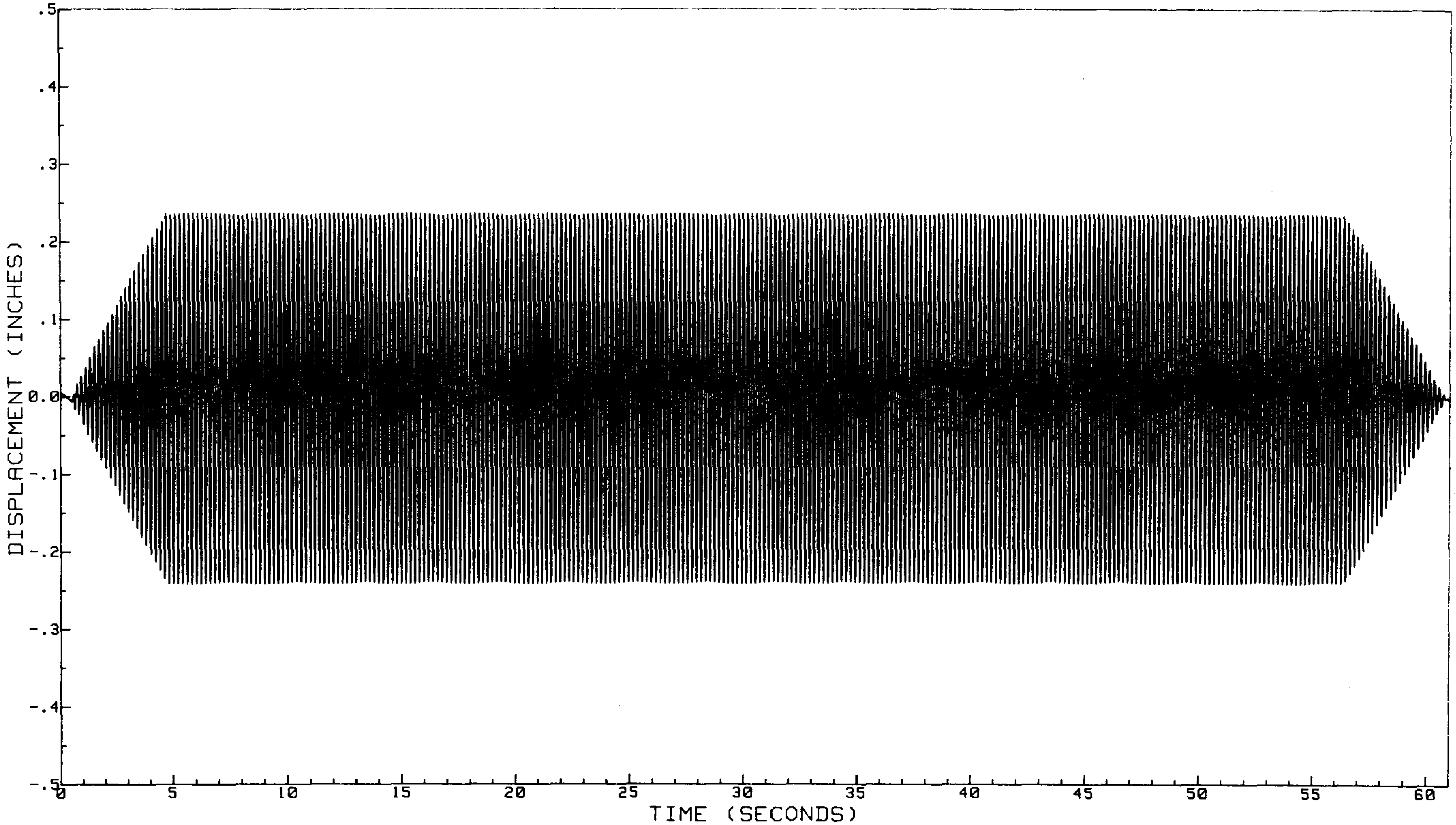
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 6
ACCELERATION MEASURED HORIZONTAL -- ACCELEROMETER NO. 72
ACCELEROMETER LOCATION: BOX FRAME



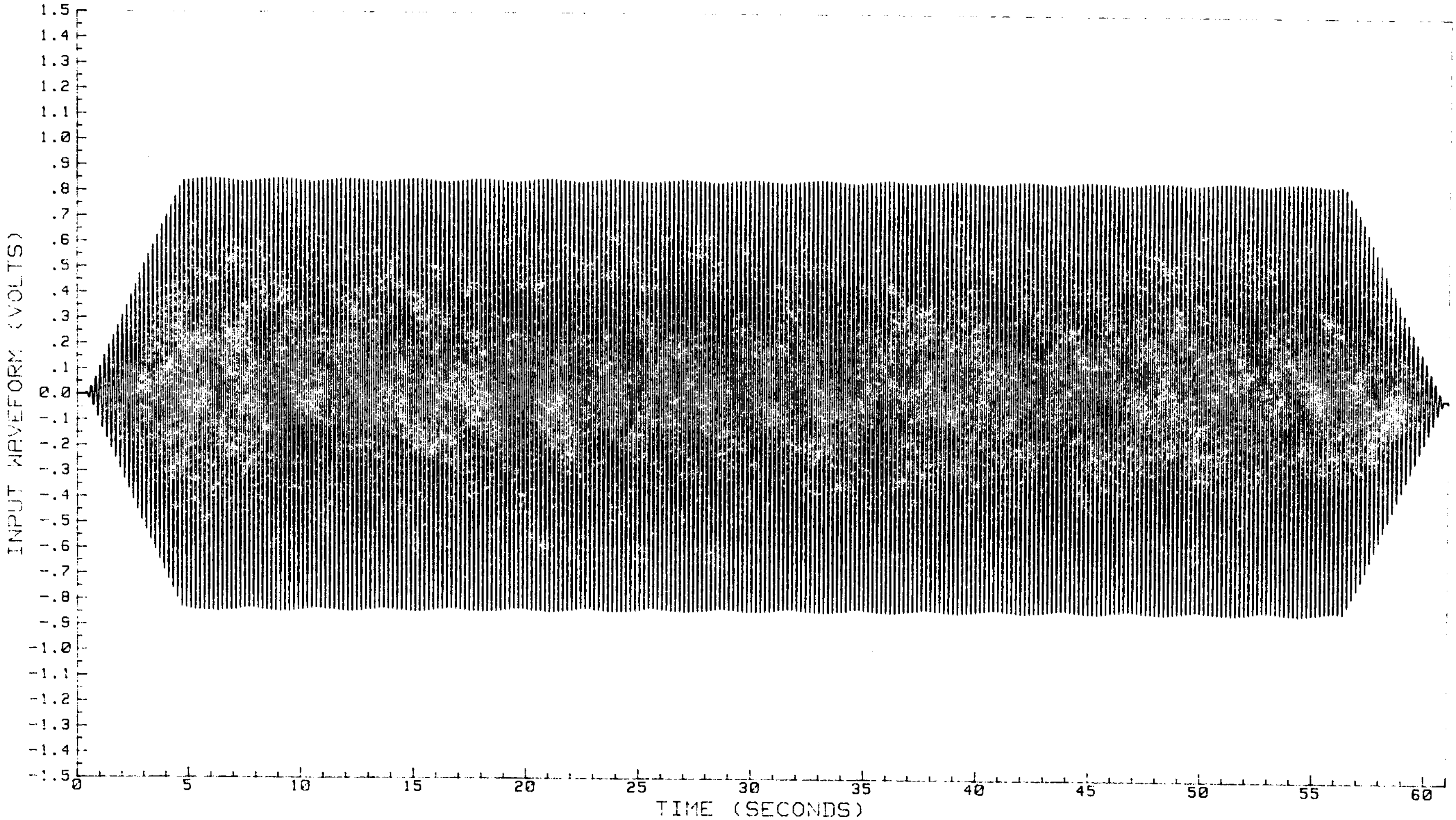
DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 6
ACCELERATION MEASURED VERTICAL -- ACCELEROMETER NO. 77
ACCELEROMETER LOCATION: BOX FRAME



DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 6
DISPLACEMENT MEASURED HORIZONTAL
LVDT LOCATED IN ACTUATOR



DYNAMIC TESTING OF HOMOGENEOUS EMBANKMENT MODELS - MODEL NO. 6
INPUT RAMP WAVEFORM
MEASURED AT RAMP GENERATOR OUTPUT



GPO 854-197

Mission of the Bureau of Reclamation

The Bureau of Reclamation of the U.S. Department of the Interior is responsible for the development and conservation of the Nation's water resources in the Western United States.

The Bureau's original purpose "to provide for the reclamation of arid and semiarid lands in the West" today covers a wide range of interrelated functions. These include providing municipal and industrial water supplies; hydroelectric power generation; irrigation water for agriculture; water quality improvement; flood control; river navigation; river regulation and control; fish and wildlife enhancement; outdoor recreation; and research on water-related design, construction, materials, atmospheric management, and wind and solar power.

Bureau programs most frequently are the result of close cooperation with the U.S. Congress, other Federal agencies, States, local governments, academic institutions, water-user organizations, and other concerned groups.

A free pamphlet is available from the Bureau entitled "Publications for Sale." It describes some of the technical publications currently available, their cost, and how to order them. The pamphlet can be obtained upon request from the Bureau of Reclamation, Attn D-822A, P O Box 25007, Denver Federal Center, Denver CO 80225-0007.